Politecnico di Torino

Politecnico di Torino

Physics of Complex Systems

A.a. 2024/2025

Graduation Session 12/ 2025

Variational approach to imaginary time propagation towards the solution of hard combinatorial problems

Supervisors:

Fabrizio Dolcini

Jan Carl Budich

Candidate:

Giovanni Ferrannini

Acknowledgements

Un grazie ai miei genitori. Grazie papà per essere stato il mio primo consigliere, il mio primo confidente e per aver sempre creduto in me. Grazie mamma per essere stata la mia ancora, il mio porto sicuro e per avermi sempre sostenuto... E sì, mamma, grazie anche per le stesse cose per cui ho ringraziato papà.

Un grazie a mia sorella. Ma che te devo dì, Giubba... già sai tutto.

Un grazie a tutta la mia famiglia. Mi siete mancati in questi anni, ma tornare a casa dei nonni mi ha sempre riempito di gioia e felicità, soprattutto quando eravamo tutti insieme.

Un grazie al professor Jan Budich e al professor Fabrizio Dolcini per la pazienza, il supporto e la guida che mi hanno fornito in questo percorso.

Un grazie al professor Bartolomeo Montrucchio, a Deborah Volpe, a Giovanni Cirillo e ad Andrea Marchesin per essere stati i miei punti di riferimento nel mio percorso all'interno del team studentesco.

Un grazie ai miei fratelli di dado — Alex, Emiliano, Rocco e Daniele. Che dire: grazie per esserci stati, grazie per esserci. Sono fiero di avervi nella mia vita.

Un grazie alla mia girl— Alice. Grazie per essermi stata vicino in questo periodo con amore e affetto.

Un grazie ai miei fratelli di progetti e ricerche — Dario e Federico. Con voi ho scoperto e riscoperto la passione per la nostra materia e ho trovato non solo dei colleghi, ma delle persone che ammiro.

Un grazie ai miei colleghi e amici. Un grazie a Guglielmo (il primo e il più alto dei miei broh), Sara F., Edoardo, Carmen, Carlo e Gabbo. La mia vita sarebbe stata più triste e meno zingara senza di voi.

Un grazie a Silvia e Francesco per aver condiviso questo percorso di magistrale con me, tra messaggi, audio infiniti e appunti scribacchiati.

Un grazie ai miei coinquilini – Daniel e Jacopo. Grazie per avermi accolto con calore e gentilezza a casetta.

Un grazie a tutti i ragazzi del gruppo di ricerca a Dresda: Joachim, Tim, Lukas, Benjamin, Carl, Florian e Yumin.

Un ultimo grazie a chiunque non ho potuto ringraziare per tempo e a chi ho ringraziato ma mai abbastanza.

Abstract

This thesis presents and evaluates a variational approach designed to address 3-SAT problems at critical clause density, examining how parameter-space trajectories generated by deterministic time-dependent variational principle (TDVP) dynamics, seeded by an initial imaginary-time evolution, can steer the ansatz toward low-energy configurations and, ultimately, the ground states of the encoded instances.

Each Boolean formula is mapped to a spin Hamiltonian, whose low-energy spectrum encodes the satisfying assignments. Within this framework, the algorithm employs an ansatz constructed as a superposition of uncorrelated Bloch-disk product states. The initialization is carried out via a short imaginary-time evolution, starting from a transverse-field-like state with equal overlap with all classical configurations. During this preliminary phase, no parameter optimization is performed; instead, the procedure systematically generates and expands a set of product states, steering their linear combination toward energetically favorable regions. After a predefined number of iterations, the process transitions to a fully variational phase, during which both the amplitudes and the local spin orientations are refined by applying the TDVP in imaginary time.

To address computational costs and memory usage, several mitigation strategies have been implemented. During the expansion phase, the parameter space is meticulously compressed, retaining only the most relevant contributions. In the subsequent phase, variational updates are carried out within an effective subspace of the manifold, selected at each step from the full set of directions. In this phase, the algorithm also employs a local metric rescaling (LMR) scheme, together with the regularization of the local metric tensor and the resolution of the corresponding linear system via MINRES-QLP, thereby circumventing the explicit inversion of large, dense, and ill-conditioned matrices. In this context, the combined imaginary-time evolution and TDVP scheme systematically drives the ansatz toward low-energy configurations, even for challenging 3-SAT instances within the range of system sizes we analyzed.

At the same time, the analysis reveals structural bottlenecks whose impact becomes increasingly severe as the spin count grows. The rapid escalation of both memory consumption and wall-clock time is directly tied to the dimension of the manifold generated by the imaginary-time expansion, which grows drastically with both system size and expansion depth.

Overall, this work shows that imaginary-time dynamics and variational geometry can be repurposed as a deterministic heuristic for an NP-complete problem such as 3-SAT, while also clarifying where and why intrinsic limitations on scalability arise.

Table of Contents

Chapter 1

Introduction

The resolution of problems belonging to the NP-complete class represents one of the central challenges in theoretical computer science and computational physics [1, 2]. Among these, the 3-SAT problem plays a particularly significant role, both for its paradigmatic value and for the practical implications arising from its solution or approximation [3]. The study of such problems is not only of purely theoretical interest: the methods developed to tackle them find applications in areas ranging from combinatorial optimization to cryptography [4], and they are also tightly connected to the statistical physics of disordered systems, where ideas such as frustration, metastability, and glassy dynamics naturally emerge [3, 5].

In particular, random instances of 3-SAT (and K-SAT in general) exhibit the well-known "easy–hard–easy" pattern as the clause density is varied [1, 6]: formulas with low clause density are typically satisfiable and easy to solve, while highly overconstrained formulas are typically unsatisfiable and again easy to classify. In contrast, in a narrow critical region around the satisfiability threshold at clause density $\alpha \simeq 4.3$ the problem becomes algorithmically hard [1, 6]. In that regime, the space of satisfying assignments (when it exists) breaks up into many well-separated clusters (or "valleys") at large Hamming distances, and simple local-search heuristics tend to get trapped in metastable configurations [2, 7]. This behavior is commonly interpreted as the algorithmic signature of a glass transition: under the standard spin-glass mapping, the associated energy landscape becomes rough and highly clustered, dominated by barriers and long-lived metastable states [8, 5]. For this reason, instances close to the threshold are routinely used as demanding benchmarks for both classical and quantum-inspired optimization methods [9, 10].

In recent years, the rapid development of quantum technologies has renewed interest in tackling such hard combinatorial problems with tools inspired by quantum mechanics [11, 12]. Several approaches aim to prepare low-energy (ideally ground) states of an Ising-like cost Hamiltonian that encodes the problem. Quantum annealing, in its original formulation, attempts to reach the ground state of a problem Hamiltonian by slowly interpolating from a simple transverse-field Hamiltonian, hoping to remain in the instantaneous ground state through an adiabatic-like path [9].Variational quantum algorithms, such as the Quantum Approximate Optimization Algorithm (QAOA) [13] and VQE-type methods [14, 10], instead construct a parameterized trial state and iteratively optimize its parameters to minimize the expectation value of the cost Hamiltonian. These hybrid strategies are attractive because they combine a quantum state ansatz with classical optimization, and they can, in principle, adapt their expressive power to the structure of a specific instance [10].

Even when large-scale fault-tolerant quantum hardware is not available, these ideas have motivated a broad class of hybrid variational schemes, where a parameterized quantum state is trained by a classical optimizer to approximate the ground state of a problem Hamiltonian [14, 10]. The underlying philosophy is to borrow physical mechanisms that are known to drive a many-body system toward its ground state and reinterpret them as optimization procedures over a variational manifold. This is especially appealing for NP-hard optimization problems (like 3-SAT near the threshold) that resemble frustrated spin glasses: if we can navigate the energy landscape using tools originally developed to relax quantum systems toward their ground state, we may gain an advantage over naive local descent.

This work fits into that line. We study a variational, quantum-inspired algorithm designed

to search for low-energy (ideally satisfying) assignments of 3-SAT instances at clause densities close to $\alpha \approx 4.3$, i.e. in the "hard SAT" regime. The key idea is to map the logical clauses of 3-SAT to an Ising-type Hamiltonian $\hat{H}$ acting on spin variables. In this mapping, each Boolean assignment corresponds to a computational-basis spin configuration, and $\hat{H}$ counts the number of violated clauses. Satisfying assignments, therefore, correspond to ground states with zero energy (or a degenerate zero energy manifold if multiple satisfying assignments exist). The original SAT problem is thus reformulated as a ground-state search in a frustrated spin system.

Rather than representing and evolving the full state in the 2^L-dimensional Hilbert space (where L is the number of spins/variables), we approximate imaginary-time evolution within a variational subspace.

Our ansatz is a quantum superposition of uncorrelated product states of L spins. Each product state is specified by a set of Bloch angles, one per spin, and the superposition is weighted by real amplitudes. The procedure is initialized by performing a short imaginary-time propagation (ITP) starting from a transverse-field type state that has a nonzero overlap with every classical assignment. Each ITP-step inflates our variational parameter vector, generating new product states with the corresponding amplitudes and updating the old amplitudes while keeping the old states untouched.

After this initialization stage, the evolution is performed by integrating the imaginary time-dependent variational principle (TDVP) in imaginary time *within* the ansatz manifold. In imaginary time, TDVP can be viewed as a projection of

$$\partial_\tau |\Psi\rangle = -\hat{H} |\Psi\rangle$$

onto the tangent space of the variational state. This projection produces a deterministic linear system

$$S_{\bar{w}} \, \dot{\bar{w}} = -F_{\bar{w}},$$

or, in discrete-update form,

$$S_{\bar{w}} \, \delta\bar{w} = -\gamma \, F_{\bar{w}}, \tag{1.1}$$

where $\bar{w}$ collects all variational parameters, $S_{\bar{w}}$ is a metric (or covariance) tensor built from overlaps of tangent vectors, and $F_{\bar{w}}$ is the projected "force" generated by $\hat{H}$. In practice, this is a principled, geometry-aware parameter update that solves for the best local imaginary-time descent direction *within the chosen ansatz*.

In an idealized scenario, one could continue applying ITP until the number of product states saturates the intended expressive budget of the ansatz For sufficiently large imaginary time, this would drive the state close to the true ground state. From that point on, both the amplitudes and the local spin orientations could be treated as variational parameters and further refined through TDVP to fully reach the solution of our problem. In practice, however, the number of parameters grows drastically with the number of ITP steps, and this scaling becomes prohibitive even for moderately large system sizes. As a consequence, we cannot afford to let ITP reach saturation; instead after a small number of initialization steps we switch to the TDVP dynamic.

Within this framework, we raise the following questions:

- How does the number of parameters — and thus the computational cost — scale with system size, given that our ansatz is a superposition of product states whose amplitudes and Bloch angles are all variational?
- How effectively can a TDVP-driven imaginary-time variational update converge toward satisfying (or nearly satisfying) assignments in the hardest region of 3-SAT, i.e., around the satisfiability threshold?

The structure of this work is organized as follows:

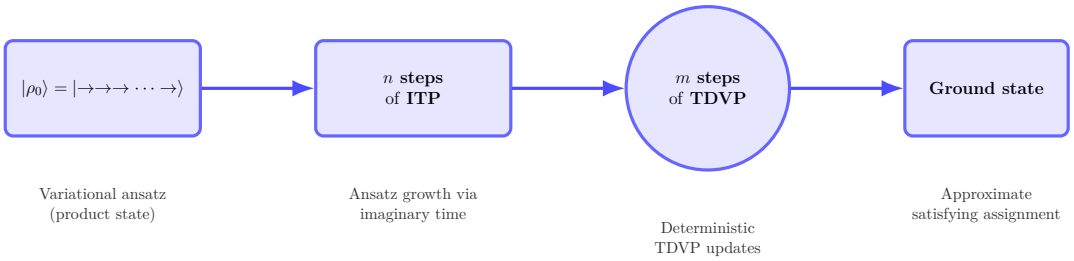


Figure 1.1: Overall structure of the algorithm. We (i) initialize a simple product-state ansatz with broad support over all assignments; (ii) apply a short exact imaginary-time propagation (ITP), which inflates the variational manifold by generating a superposition of product states; (iii) switch to TDVP to perform deterministic, geometry-aware updates of all variational parameters. The goal is to approach the ground state of the 3-SAT clause Hamiltonian, i.e. a satisfying assignment.

- **Chapter 2** introduces the essential aspects of the 3-SAT problem, the mapping that allows us to study it as a spin system, and its connection to disordered systems and spin glasses. It also presents the notion of the "critical clause density threshold" (and its analogy with a glass transition), explaining why this regime produces particularly hard instances and how we generated them.
- **Chapter 3** develops imaginary-time propagation (ITP) based on a clause-structured frontier. We introduce the Bloch-disk variational ansatz, specify the frontier growth and pruning rules, and analyze how the size of this manifold scales with the number of variables and clauses.
- **Chapter 4** presents the time-dependent variational principle (TDVP) in imaginary time, showing how the Schrödinger-like equation projects to the linear system in Eq (1.1) on the variational parameters. We derive explicit expressions for the metric tensor $S_{\bar{w}}$ and the force term $F_{\bar{w}}$ specialized to our ansatz.
- **Chapter 5** benchmarks the variational imaginary-time dynamics on small 3-SAT instances. We compare frontier-based ITP with exact imaginary-time evolution and assess how subsequent TDVP refinement updates the variational parameters and improves ground-state probabilities.
- **Chapter 6** introduces sensitivity-driven compression strategies for both TDVP and ITP. For TDVP, we restrict updates to an effective low-dimensional subspace without reducing the underlying manifold, while for ITP we explicitly compress the frontier manifold. We compare the resulting accuracy, stability, and runtime on single instances and finite-size statistics at critical clause density.

Chapter 2

The 3-SAT Problem

Chapter 3

The 3-SAT Problem

In this chapter, we introduce the canonical combinatorial problem that underlies much of our analysis: the Boolean satisfiability problem, and in particular, its 3-SAT restriction. From the complexity-theoretic point of view, 3-SAT is a prototypical NP-complete problem and serves as a universal source of hardness through polynomial-time reductions. From the statistical-physics point of view, random 3-SAT provides one of the simplest ensembles where a sharp SAT/UNSAT phase transition, an easy–hard–easy pattern of typical algorithmic complexity, and a glassy, spin-glass-like structure of the solution space can all be studied within a unified framework. In what follows, we start from the formal definitions of decision problems, NP, NP-completeness, and SAT, then specialize to 3-SAT and its random ensembles, with particular emphasis on the hard-SAT regime near the satisfiability threshold. We then review the SAT/UNSAT transition and its interpretation in terms of spin-glass models, introduce the planted-solution instances that define our benchmark ensemble, and finally present the explicit Ising mapping that allows these formulas to be treated as spin systems in the variational imaginary-time framework developed in the subsequent chapters.

3.1 NP-Completeness and Boolean Satisfiability

Problems in the class NP-complete play a central role in computational complexity theory. Informally, these are the "hardest" problems in NP: if we could solve any one of them efficiently, then *every* problem in NP could be solved efficiently. We begin by recalling what NP-completeness means and then explaining how SAT, and in particular 3-SAT, fits into this picture.

Definition 1 (Class NP and NP-completeness)**.** *The class* NP *(nondeterministic polynomial time) is the set of all decision problems for which, whenever the correct answer is* Yes*, there exists a* certificate *(also called a* witness*) that can be verified in polynomial time by a deterministic Turing machine. Intuitively: finding a solution may be computationally difficult, but checking that a proposed solution is correct is efficient.*

Definition 2. *A decision problem A is* NP-hard *if every problem in NP can be transformed (reduced) to A by a polynomial-time reduction that preserves* Yes/No *answers. In other words, solving A efficiently would allow us to solve* all *problems in NP efficiently.*

A decision problem A is NP-complete *if:*

1. *$A \in NP$, and*
2. *A is NP-hard.*

NP-complete problems are therefore the most difficult problems in NP: if any NP-complete problem can be solved in polynomial time, then all problems in NP can be solved in polynomial time, implying $P = NP$.

The Boolean satisfiability problem (SAT) was historically the first problem proven to be NP-complete. This result, due to Cook and Levin, established SAT as a kind of "universal" NP problem: any other NP problem can be efficiently encoded as an instance of SAT. Because

of this, the SAT problem is often used as a starting point to prove that other problems are NP-complete via polynomial-time reductions [15].

In what follows, we introduce SAT and, more specifically, its restricted form, 3-SAT, which is also NP-complete.

Definition 3 (Boolean Formula)**.** *A* Boolean formula ϕ *is an expression built from:*

- *Boolean variables* $x_1, x_2, \ldots, x_N$ *taking values in* $\{True, False\}$ *(or equivalently* $\{0, 1\}$*),*
- *logical connectives such as* $\wedge$ *(AND),* $\vee$ *(OR), and* $\neg$ *(NOT),*
- *and parentheses for grouping.*

Given an assignment for the N *variables, a Boolean formula* ϕ *can be either* True *or* False*.*

Definition 4 (Satisfiability Problem (SAT))**.** *Given a Boolean formula* ϕ*, the* satisfiability problem *(SAT) asks whether there exists an assignment of Boolean values to the variables of* ϕ *such that the entire formula evaluates to* True*. If such an assignment exists, the formula is said to be* satisfiable*, and any assignment that makes* ϕ *evaluate to* True *is called a* satisfying assignment *(or* certificate*). If no such assignment exists, the formula is* unsatisfiable*.*

SAT is in NP because, given a candidate assignment of the variables, we can efficiently evaluate ϕ and check whether it is *True*. Moreover, SAT is NP-hard, since any other NP problem can be reduced to SAT. Therefore, SAT is NP-complete. For this reason, SAT is a canonical reference problem for NP-completeness.

In this work, we focus on a particular structured version of SAT known as *3-SAT* [15]. To define 3-SAT, we first describe a standard normal form for Boolean formulas.

Definition 5 (Conjunctive Normal Form (CNF))**.** *A* Conjunctive Normal Form *(CNF) is a Boolean formula that can be written as:*

$$\phi = C_1 \wedge C_2 \wedge \cdots \wedge C_M,$$

where each C_c *is a* clause *of the form*

$$C_c = l_{c1} \vee l_{c2} \vee \cdots \vee l_{ck_c}.$$

The terms l_{cq} *are* literals*, each being either a variable* x_k *or its negation* $\neg x_k$*. A* k-CNF *formula is a CNF in which all clauses contain at most* k *literals.*

Definition 6 (3-SAT Satisfiability Problem)**.** *A* 3-SAT *instance is a Boolean formula in 3-CNF, i.e., a conjunction of* M *clauses, each containing exactly three literals drawn from* N *Boolean variables. The decision problem of whether a given 3-SAT instance is satisfiable is the* 3-SAT problem*.*

$$\phi = \bigwedge_{c=1}^{M} (l_{i_c} \vee l_{j_c} \vee l_{h_c}). \tag{3.1}$$

Here l_{i_c} *denotes the literal in position* i_c*, which can be either* x_{i_c} *or* $\neg x_{i_c}$*. The parameter* M *is the number of clauses,* N *is the number of variables, and for each clause* C_c *the triple* (i_c, j_c, h_c) *indicates the indices of its three literals, with* $i_c \neq j_c \neq h_c$*.*

3-SAT is a restriction of SAT (because it only allows formulas in 3-CNF, with exactly three literals per clause); despite this restriction, it remains NP-complete. In practice, this makes 3-SAT extremely important: many NP problems are shown to be NP-complete by reduction *from* 3-SAT rather than from general SAT, since 3-SAT already has a very uniform and combinatorial structure that is convenient to encode in other settings.

3.2 The SAT transition

A striking empirical and theoretical property of random k-SAT (and, in particular, random 3-SAT) is the emergence of a sharp *phase transition* as we vary the *clause density*

$$\alpha = \frac{M}{N},$$

i.e., the ratio between the number of clauses M and the number of Boolean variables N. For small α the instance is typically underconstrained and therefore satisfiable; for large α it is typically overconstrained and therefore unsatisfiable. Between these two extremes, there is a very narrow "critical" window where the probability of satisfiability drops abruptly from almost 1 to almost 0 as $N \to \infty$ [1, 2]. In complexity-theoretic language, this window is where typical instances of NP-complete SAT become algorithmically hardest, giving rise to the famous *easy–hard–easy* pattern [1].

SAT/UNSAT threshold

Fix k and generate a random k-SAT instance by sampling each clause uniformly at random from all k-clauses over $\{x_1, \ldots, x_N\}$ (each literal negated with probability 1/2). In the limit of large N, it is now widely believed (and can be rigorously established for large k) that such random formulas exhibit a *sharp threshold*: there exists a critical density $\alpha_c(k)$ such that for $\alpha < \alpha_c(k)$, a random instance is satisfiable with probability tending to 1, whereas for $\alpha > \alpha_c(k)$, it is unsatisfiable with probability tending to 1 [1]. For 3-SAT, refined statistical-physics calculations based on cavity methods and survey propagation predict this threshold at

$$\alpha_c(3) \simeq 4.267,$$

a non-rigorous value with subleading finite-size effects that is nevertheless in very good agreement with numerical experiments [2, 16]. Below α_c, satisfying assignments still exist (the formula is *SAT*); above α_c, no assignment satisfies all clauses (*UNSAT*).

Although α_c is defined in purely combinatorial terms ("does a satisfying assignment exist?"), it also marks a sharp change in *typical* algorithmic difficulty. Empirically, general-purpose search procedures (e.g., DPLL-style backtracking with heuristics) solve most random instances very quickly when $\alpha \ll \alpha_c$ (underconstrained) and also when $\alpha \gg \alpha_c$ (so overconstrained that contradictions surface early). Runtime blows up dramatically in a narrow band around α_c [1]. This gives a characteristic easy–hard–easy curve: as α increases from small to critical values, instances suddenly become hard; as α increases past the critical region, they become easy again.

This spike in hardness is not accidental. Near α_c, the solver is effectively forced to explore a huge combinatorial search tree because many variables are "almost" forced but not quite, and incorrect early guesses lead to long backtracking paths. In this regime, the search cost grows exponentially with N in practice [1].

Statistical-physics viewpoint

From the statistical-physics point of view, a random 3-SAT formula with N variables and M clauses is naturally mapped to a diluted Ising p-spin glass (with $p = 3$) on a random hypergraph or factor graph: each Boolean variable becomes an Ising spin $\sigma_i \in \{-1, +1\}$, each clause contributes a local three-spin energy term that penalizes exactly the unique assignment that violates the clause, and the total "energy" $E(\boldsymbol{\sigma})$ is the number of unsatisfied clauses [3, 2, 16]. One then considers the Boltzmann–Gibbs measure

$$\mu_\beta(\boldsymbol{\sigma}) \propto \exp(-\beta E(\boldsymbol{\sigma})),$$

on this random Hamiltonian, and studies the zero-temperature limit $\beta \to \infty$ as a function of the control parameter $\alpha = M/N$. In this language, SAT corresponds to a phase in which the ground-state energy density

$$e_0(\alpha) \;=\; \frac{1}{N} \min_{\boldsymbol{\sigma}} E(\boldsymbol{\sigma})$$

is zero (there exist configurations that satisfy all clauses), while UNSAT corresponds to a phase in which $e_0(\alpha) > 0$. The SAT/UNSAT transition is therefore a $T = 0$ phase transition in which $e_0(\alpha)$ first becomes strictly positive, and at the same time, the *structure* of low-energy states changes qualitatively.

The cavity/replica analysis shows that for random 3-SAT, this transition is described by a *discontinuous one-step replica-symmetry-breaking* (1-RSB) solution: above a dynamical (clustering) threshold α_d, the set of satisfying assignments (when it still exists) shatters into an exponential number of clusters with positive complexity $\Sigma(e)$, and at the critical density $\alpha_c \simeq 4.27$ the complexity at zero energy $\Sigma(0)$ vanishes; beyond this point the ground-state energy density $e_0(\alpha)$ becomes strictly positive [2, 16]. In parallel, order parameters that probe long-range structure—such as the fraction of frozen (backbone) variables shared by all solutions in a cluster, or more refined frustration parameters in the sense of Zhou [1, 17]—develop a discontinuity at the transition, very much as in mean-field p-spin glasses [8]. In this precise sense, random 3-SAT realizes a *glass-like* phase transition on a sparse random graph: the hardest window near α_c is the regime where the underlying spin-glass system enters a discontinuous 1-RSB phase, with a clustered, rugged solution space that directly manifests as typical-case computational hardness.

In the modern picture, the glassy regime of random k-SAT may resemble an *RFOT* (Random First-Order Transition): at the level of the energy landscape, it shares the same mathematical structure (a discontinuous 1-RSB solution with an exponential number of metastable states) as the Random First-Order Transition scenario of mean-field structural glasses [8]. The crucial difference is that here the control parameter is the clause density α at zero temperature, and the RFOT-like 1-RSB structure already develops within the SAT phase (at the clustering/condensation thresholds) rather than exactly at the satisfiability threshold. More generally, Monasson *et al.* showed on the 2+p-SAT ensemble that random 3-SAT sits on the "random first-order" side of a broader phase diagram: for small p, the SAT/UNSAT transition is continuous, with a backbone that turns on smoothly, while beyond a critical p_0 (in particular, in the pure 3-SAT case), the transition becomes random first-order, with a discontinuous backbone and exponentially large typical running times for DPLL-type (backtracking) algorithms near the threshold [1]. At the same time, recent rigorous work reframes this glassy picture in terms of the Overlap Gap Property (OGP): for large k and related p-spin glasses, Kızıldağ proves a sharp threshold for the emergence of a multi-overlap gap, i.e. a "forbidden" band of overlaps between near-optimal configurations [18]. This provides a mathematically precise version of the shattered, clustered geometry suggested by the cavity/1-RSB analysis and indicates that the *glassy phase* of random k-SAT is best understood as governed by a glass-like discontinuous 1-RSB landscape rather than by a standard equilibrium first-order transition in the thermodynamic sense.

Structure of the solution space

The studies discussed above provide a precise explanation of why and how the hard regime emerges precisely at α_c it is a consequence of the geometry of the set of satisfying assignments, which undergoes its own sequence of transitions before the instance actually becomes UNSAT.[2, 16, 7].

Let us track α for random 3-SAT:

- **Easy SAT phase ($\alpha < \alpha_d$).** For small densities, all satisfying assignments belong to one "liquid" region of configuration space: variables are only weakly correlated, and local search heuristics quickly find a satisfying assignment. Survey propagation (SP) converges

to a trivial, essentially uninformative fixed point in which no variable is strongly biased [2]. For 3-SAT this regime persists up to a *dynamic/clustering threshold* $\alpha_d \approx 3.92$ [2, 16].

- **Hard SAT (glassy) phase ($\alpha_d < \alpha < \alpha_c$).** Beyond α_d, the solution set *shatters* into an exponential number of "clusters" or pure states. Each cluster is internally well connected (you can move between solutions by flipping a bounded number of variables at a time), but different clusters are separated by extensive Hamming distance and energetic or entropic barriers [16, 7]. In the spin-glass language this is a one-step replica-symmetry-breaking (1-RSB) phase: the Gibbs measure (or, at zero temperature, the set of ground states) decomposes into exponentially many states with positive *complexity* $\Sigma(e)$, i.e. an exponential density of metastable valleys at a given energy density e.

  Survey propagation now converges to a nontrivial fixed point encoding long-range biases and "frozen" variables within each cluster [2, 16]. In this window, formulas are still satisfiable with high probability, but finding *one* satisfying assignment becomes algorithmically difficult, because local moves get trapped inside suboptimal clusters and complete search must effectively navigate an exponentially branching space of states. This regime is sometimes called the *hard-SAT phase* [2], and is the algorithmic manifestation of the RFOT-like glassy structure.

- **UNSAT phase ($\alpha > \alpha_c$).** Past $\alpha_c \simeq 4.267$, no satisfying assignments remain: every assignment violates some clauses. In the cavity/1-RSB solution this is the point where the complexity of zero-energy states $\Sigma(0)$ drops to zero and the ground-state energy density $e_0(\alpha)$ becomes strictly positive [16]. At this point, many solvers can quickly detect contradictions and prove UNSAT, and the empirical runtime often drops again — hence the "easy" tail on the high-α side of the curve [1, 2].

More generally, for random k-SAT with $k \geq 4$, one finds that beyond the clustering transition at α_d an additional *condensation* transition can occur before α_c: although there are still exponentially many clusters, only a few of them carry almost all the weight (i.e. almost all satisfying assignments live in a tiny number of dominant clusters). This condensation is accompanied by a nonanalytic change in the entropy density of solutions and marks the limit of effectiveness for message-passing heuristics such as belief propagation [7]. In this condensed regime, the 1-RSB cavity formalism and SP are the natural tools to describe the landscape. In the specific case of random 3-SAT, the clustering and condensation transitions are believed to lie very close to each other in α, so the genuinely condensed regime does not extend far below the SAT/UNSAT threshold.

Algorithm-dependent easy/hard boundaries

Finally, "hardness" is not an intrinsic property of the formula alone. For a simple branch-and-propagate decision procedure that recursively branches on variables and rewrites the residual subproblems back into CNF, an analytic treatment of the associated recursion shows that the solver itself exhibits a sharp algorithmic phase transition: in one region of the (M, N, k) parameter space the expected running time remains bounded ("easy"), whereas beyond a critical surface the branching process leads to an uncontrolled combinatorial blow-up ("hard") [6]. Remarkably, this solver-induced easy/hard boundary does *not* coincide exactly with the satisfiable/unsatisfiable threshold of random k-SAT. This reinforces a central message of SAT phase-transition studies: the easy–hard–easy pattern is, to a significant extent, a property of the search dynamics of a given algorithm, rather than solely a consequence of the underlying logical satisfiability structure.

3.3 Planted satisfiable instances and search objective

In classical complexity terms, SAT is formulated as a *decision problem*: given a Boolean formula ϕ, decide whether there exists at least one assignment of its variables that satisfies all clauses (i.e. makes $\phi = \text{True}$). In that formulation, the algorithm must answer *Yes* (SAT) or *No* (UNSAT), and this is exactly the NP-complete problem discussed in Section 3.1.

In this thesis, our setup is intentionally different and directly follows the "hidden (planted) solution" ensemble introduced by Barthel, Hartmann, Leone, Ricci-Tersenghi, Weigt, and Zecchina [19]. The central idea in [19] is to generate SAT instances that are *guaranteed satisfiable by construction*, but which are nevertheless algorithmically hard near the clause-density threshold $\alpha_c \simeq 4.27$ where random 3-SAT exhibits its easy–hard–easy transition [2, 16]. This lets us probe true search hardness without the confound "maybe the instance was UNSAT".

From the spin-glass viewpoint, the planted ensemble can be interpreted as a glassy spin system with a hidden ferromagnetic bias along the planted configuration: the Hamiltonian still has a rugged landscape with many metastable states, but one or more particular configurations are guaranteed to be a ground state.

From decision to search We do *not* ask "Is the instance satisfiable?". By construction, our instances *are* satisfiable. Instead, our task is of a different flavour:

- Given a formula that we already know to be satisfiable, we aim to *recover one or more* satisfying assignments.
- In stronger form, we are interested in exploring (part of) the space of satisfying assignments.

In other words, our problem is no longer a pure *decision* problem (*SAT or UNSAT?*) but a *search / reconstruction* problem: *find* satisfying configurations and characterize how they are organized. This shift from decision to search mirrors the philosophy in [19], where the instance is designed so that at least one ground-truth solution exists, and the interesting question becomes how difficult it is for an algorithm to actually reconstruct it.

Planted (a.k.a. hidden) solutions The formulas we study are not arbitrary random 3-SAT instances. They are generated via a *planted-solution* procedure that guaranties satisfiability while trying to "hide" the planted assignment [19]. The procedure, as implemented in code developed within the Cluster of Excellence "Complexity and Topology in Quantum Matter" (Prof. Jan Budich, TU Dresden), is:

1. Choose a "planted" assignment $\sigma^\star \in \{0,1\}^N$ that we want to satisfy for all clauses. In practice, we start from $\sigma^\star = (1,1,\ldots,1)$ and optionally apply a global random sign flip (see below) so that $\sigma^\star$ is effectively random.
2. For each clause, decide how many of the three distinct variables will appear negated.
3. Accept only clauses that are *satisfied* by $\sigma^\star$; reject clauses that $\sigma^\star$ would violate.
4. Repeat until we reach the target number of clauses M, hence a desired clause density

$$\alpha = \frac{M}{N},$$

   which we tune close to the hard-SAT regime around $\alpha_c \simeq 4.27$ [2, 16].

More explicitly, each 3-clause is drawn from three "types":

- **type 0:** all three literals are positive $(x \vee y \vee z)$,

- **type 1:** exactly one negated literal ($\neg x \vee y \vee z$),
- **type 2:** exactly two negated literals ($\neg x \vee \neg y \vee z$).

A clause with three negations ($\neg x \vee \neg y \vee \neg z$) is *forbidden*, because it would be *false* under $\sigma^\star = (1,1,\ldots,1)$. By restricting ourselves to types 0, 1, and 2, we guaranty that $\sigma^\star$ satisfies every generated clause.

The generator then chooses the clause type probabilistically. We denote by p_0 the probability of drawing a type-0 clause (all positive literals). Following [19], the probabilities for the other types are set as

$$p_1 = \frac{1 - 4p_0}{6}, \qquad p_2 = \frac{1 + 2p_0}{6}, \tag{3.2}$$

so that the normalization condition

$$p_0 \; + \; 3p_1 \; + \; 3p_2 \; = \; 1 \tag{3.3}$$

holds, where $3p_1$ is the total weight of all clauses with exactly one negation (since there are three possible positions for that negation), and $3p_2$ is the total weight of all clauses with exactly two negations.

The parameter p_0 is chosen in the range

$$0.077 \lesssim p_0 \lesssim 0.25$$

has two important effects [19]:

- it prevents trivial "majority-vote" recovery of $\sigma^\star$ (so the planted solution is genuinely hidden),
- it produces formulas that are empirically hard for both local-search heuristics and complete solvers, especially when α is tuned near the satisfiability threshold.

In code, this is exactly how we generate each clause: with probability p_0 we emit a type-0 clause (no negations); with probability $3p_1$ we emit a clause with a *single* negated literal; with probability $3p_2$ we emit a clause with *two* negated literals. Clauses with three negations are never produced.

Finally, to avoid the planted assignment being trivially the "all-True" vector, we optionally perform a global "gauge" transformation: for each variable x_i we randomly decide whether to flip its sign everywhere in the formula. This corresponds to the `shuffle` option in our implementation and is the practical version of the "hidden"/"concealed" planted assignment discussed in [19].

3.4 Our framework

In this thesis, we focus on clause densities α in the hard-SAT region, i.e., close to the SAT/UNSAT threshold α_c. In this near-critical regime, typical instances (i) remain satisfiable with non-negligible probability, (ii) display a clustered, glassy solution space with long-range constraints, and (iii) are empirically among the hardest for modern solvers due to backbone freezing, deep backtracking, and, conjecturally, geometric barriers of OGP type [2, 16, 7]. Working here, therefore, provides a controlled stress test on maximally challenging yet meaningful inputs. Our aim is not only to assess whether the proposed procedure works, but also to understand how and where it fails, using its breakdowns as a probe of the underlying hardness landscape.

To this end, we consider planted-SAT instances in the hard-SAT regime described in Section 3.2. This choice has several advantages for our goals:

- **A known ground truth.** By construction, we know at least one satisfying assignment $\sigma^\star$. Whenever our procedure outputs a candidate configuration, we can verify it clause by clause (using a dedicated `check_assignment` routine) and evaluate performance in terms of correctness, not only runtime.

- **Direct access to the glassy regime.** By tuning $\alpha = M/N$ close to $\alpha_c \simeq 4.27$, where random 3-SAT exhibits clustering, frozen backbones, and exponential search cost [2, 16, 7], we force our method to operate in a phase where the solution space is rugged and metastable. Barthel *et al.* show that, for suitable choices of (p_0, p_1, p_2), planted instances in this regime remain satisfiable but are algorithmically difficult [19], making them ideal testbeds.

  [2, 16, 7] and are directly tied to the trapping behavior observed in [19].

- **Clean separation of modeling vs satisfiability issues.** If our method fails to reconstruct a satisfying assignment in this planted setting, we know that the failure is algorithmic (search, heuristics, sampling), rather than a trivial consequence of UNSAT. This was precisely the motivation in [19]: by eliminating the "maybe the instance is unsatisfiable" explanation, hardness can be attributed directly to the search dynamics and the geometry of the solution space.

Conceptual perspective In summary, the computational problem we investigate can be phrased as:

> *Given a 3-SAT instance ϕ whose satisfiability is guaranteed by construction but is unknown by the procedure, can we efficiently recover one (or more) satisfying assignments?*

3.5 Ising mapping

As already discussed in computationally hard problems — whose worst-case behavior exhibits exponential time or memory, such as NP-complete problems — they admit one-to-one correspondences with the ground-state structure of disordered spin systems [3]. A highly structured family of satisfiable 3-SAT formulas can be mapped to an ordered spin-glass model with distinctly *glassy* behavior: exponentially many metastable minima separated by barriers that trap local search and hinder the discovery of the true ground state [20]. The SAT transition has been argued to play a role analogous to a random first-order phase transition, in the sense of structural glasses, where the system freezes into a mosaic of aperiodic, statistically distributed states [5, 2, 16]. This provides a natural framework for theoretical interpretation. Several mappings from SAT to Ising / QUBO models exist [15]; in this work we adopt the following Hamiltonian for 3-SAT:

$$H = \sum_c H_c = \sum_{c:(i,j,h)} O_{(i,j,h)} \tag{3.4}$$

which is the sum of local clause contributions H_c, each associated with a clause c (M in total) of the SAT problem.

For each Boolean literal l_i, we introduce a spin variable s_i defined as

$$s_i = \begin{cases} -1 & \text{if } l_i \text{ is the positive literal } x_i, \\ 1 & \text{if } l_i \text{ is the negated literal } \neg x_i, \end{cases} \tag{3.5}$$

where the spin is related to the Pauli operator σ_i^z. Each clause involving three literals on variables (i, j, h) is mapped to a projector

$$O_{(i,j,h)} = \bigotimes_{k \neq i,j,h} \mathbb{I} \bigotimes_{q=i,j,h} \frac{(1 - s_q \sigma_q^z)}{2} \tag{3.6}$$

which acts as the identity on all qubits not in the clause and as a penalty projector on the clause qubits. In this way, $O_{(i,j,h)}$ contributes a positive energy only if the clause is *violated*, and contributes zero otherwise. Therefore, the spin configuration that minimizes H corresponds to an assignment satisfying the largest possible number of clauses; if the minimum energy is $H_{\min} = 0$, the corresponding assignment is a satisfying solution of the SAT problem.

Sign convention In the computational basis

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}, \qquad |1\rangle = \begin{pmatrix}0\\1\end{pmatrix}, \qquad \sigma^z = \begin{pmatrix}1 & 0\\0 & -1\end{pmatrix}.$$

and identify Boolean values by $x_i = 0 \leftrightarrow |0\rangle$ (FALSE) and $x_i = 1 \leftrightarrow |1\rangle$ (TRUE). For a clause literal l_q referring to variable x_q, define

$$s_q = \begin{cases} -1, & \text{if } l_q = x_q \text{ (positive literal)},\\ +1, & \text{if } l_q = \neg x_q \text{ (negated literal)}. \end{cases}$$

Then

$$P_{l_q} := \frac{1 - s_q\,\sigma_q^z}{2}$$

is exactly the projector onto the *falsifying* value of that literal:

literal l_q	false when	desired projector	$\frac{1-s_q\,\sigma_q^z}{2}$
x_q	$x_q = 0$ ($\|0\rangle$)	$\frac{1+\sigma_q^z}{2}$	since $s_q = -1$
$\neg x_q$	$x_q = 1$ ($\|1\rangle$)	$\frac{1-\sigma_q^z}{2}$	since $s_q = +1$

Therefore, for a 3-literal clause (i, j, h) the operator

$$O_{(i,j,h)} = \bigotimes_{k\neq i,j,h} \mathbb{I} \bigotimes_{q=i,j,h} \frac{1 - s_q\,\sigma_q^z}{2}$$

projects onto the *unique* assignment where *all three* literals in that clause are false, i.e. the clause is violated; it yields 0 otherwise. Consequently, the Hamiltonian

$$H = \sum_c O_c$$

is diagonal in the computational basis and returns, on any assignment $|\mathbf{x}\rangle$, exactly the number of unsatisfied clauses. Minimizing H is therefore equivalent to maximizing the number of satisfied clauses, with $H_{\min} = 0$ if and only if the formula is satisfiable.
Hence

$$\sigma^z\,|0\rangle = +\,|0\rangle \equiv |\uparrow\rangle \text{ (spin up)}, \qquad \sigma^z\,|1\rangle = -\,|1\rangle \equiv -\,|\downarrow\rangle \text{ (spin down)}.$$

Therefore, we can express the projectors ont the eigenstates as

$$P_\uparrow = \tfrac{1+\sigma^z}{2} \text{ (selects } |0\rangle), \qquad P_\downarrow = \tfrac{1-\sigma^z}{2} \text{ (selects } |1\rangle).$$

Example: Consider the clause

$$C = (\,x_1 \;\vee\; \neg x_2 \;\vee\; x_3\,),$$

so that $s_1 = -1$ (positive literal x_1), $s_2 = +1$ (negated literal $\neg x_2$), and $s_3 = -1$ (positive literal x_3). Suppressing identities on qubits $\neq$ 1,2,3, the clause operator is

$$O_{(1,2,3)} = P_{x_1} \otimes P_{\neg x_2} \otimes P_{x_3} \quad \text{with} \quad P_x = \tfrac{1+\sigma^z}{2}, \;\; P_{\neg x} = \tfrac{1-\sigma^z}{2}.$$

Take the (satisfying) assignment $(x_1, x_2, x_3) = (1,1,0)$, i.e.

$$|\mathbf{x}\rangle = |1\rangle_1 \otimes |1\rangle_2 \otimes |0\rangle_3\,,$$

for which the clause C is True because $x_1 = 1$. Then

$$P_{x_1}\,|1\rangle_1 = \tfrac{1+\sigma_1^z}{2}\,|1\rangle_1 = \tfrac{1-1}{2}\,|1\rangle_1 = \mathbf{0},$$

and therefore

$$O_{(1,2,3)}\,|\mathbf{x}\rangle = (P_{x_1}\,|1\rangle_1) \otimes (P_{\neg x_2}\,|1\rangle_2) \otimes (P_{x_3}\,|0\rangle_3) = \mathbf{0} \otimes (\cdot) \otimes (\cdot) = \mathbf{0}.$$

Hence a satisfying assignment yields $O_{(1,2,3)}\,|\mathbf{x}\rangle = \mathbf{0}$ and contributes zero energy, as claimed.

Correctness of the clause-projector Hamiltonian. The construction above matches the standard statistical-mechanics encoding of K-SAT: one introduces one spin degree of freedom per Boolean variable and assigns an energy penalty to each violated constraint [3, 16]. In that language, for any classical assignment $\mathbf{x}$, the cost Hamiltonian is

$$H_{\text{cost}}(\mathbf{x}) \;=\; (\#\text{ of violated clauses under } \mathbf{x}).$$

Our Hamiltonian $H = \sum_c O_c$ *exactly* reproduces this. By construction:

- O_c is a projector that evaluates to 1 if and only if clause c is violated by $\mathbf{x}$, and 0 otherwise.
- Therefore $H(\mathbf{x}) = \sum_c O_c(\mathbf{x})$ is an integer equal to the number of violated clauses.

It immediately follows that any assignment $\mathbf{x}$ minimizing H is an assignment that violates the fewest possible clauses. In particular, $H_{\min} = 0$ if and only if there exists an assignment that violates zero clauses, i.e. a satisfying assignment. Thus, finding the ground state of H is equivalent to solving the SAT instance. This is precisely the viewpoint under which random K-SAT is treated as a spin-glass energy landscape, and SAT corresponds to the existence of zero-energy ground states [3, 20, 2, 16].

From the conceptual standpoint, this is exactly the Hamiltonian analyzed by Mézard, Parisi, and Zecchina using the cavity method at zero temperature: each clause contributes a multi-spin interaction that penalizes one local pattern, and the competition between these random constraints generates the glassy phase diagram described in Section 3.2 [2, 16]. The same Hamiltonian (or closely related diluted p-spin models) is also the playground for Zhou's long-range frustration analysis [17] and for rigorous OGP results at large k [18].

Chapter 4

Imaginary Time Propagation

A straightforward ("naive") strategy for solving a 3-SAT instance mapped to an Ising Hamiltonian would be to fully diagonalize the Hamiltonian and read off its lowest-energy eigenstate(s). However, exact diagonalization scales exponentially with the number of spins and becomes intractable even for modest system sizes. A more practical alternative is *imaginary-time propagation* (ITP), which allows one to project onto the ground state without computing the full spectrum. [21]

Imaginary time is an unphysical but extremely powerful mathematical device used throughout quantum mechanics, statistical mechanics, and quantum field theory. Formally, one performs a *Wick rotation* $t \mapsto -i\tau$, which maps real (Minkowski) time to imaginary (Euclidean) time and connects unitary quantum evolution to Euclidean/statistical evolution. In this picture, thermal weights of the form $e^{-\beta H}$ and ground-state projectors arise from imaginary-time evolution operators $e^{-\tau H/\hbar}$, and ground-state preparation appears as the limit $\tau \to \infty$ of $e^{-\tau H/\hbar}$ acting on a suitable trial state. [21, 22]

On classical hardware, this principle underlies projector and diffusion Monte Carlo methods, which implement imaginary-time evolution as a stochastic diffusion-and-branching process. [21] On near-term quantum hardware, the non-unitarity of $e^{-\tau H/\hbar}$ prevents its direct realization; instead, one approximates imaginary-time evolution variationally. Variational Quantum Imaginary Time Evolution (VarQITE) and related schemes update the parameters of an ansatz state so that its evolution best approximates the action of $e^{-\tau H/\hbar}$, enabling approximate ground-state preparation for Ising-encoded SAT instances on noisy intermediate-scale quantum (NISQ) devices. [23, 24]

In the remainder of this chapter, we first introduce imaginary time more formally via Wick rotation and make explicit the connection between Euclidean evolution and thermal weights (Sec. **??**). We then highlight the ground-state projection property of imaginary-time dynamics and discuss its interpretation as a formal cooling process (Secs. **??**–**??**). Building on this foundation, we present the Bloch-disk variational ansatz that underpins our algorithm (Sec. 4.3) and derive a frontier-based ITP scheme tailored to the clause structure of the 3-SAT Ising Hamiltonian (Sec. **??**). Finally, we analyze the scaling of the resulting variational representation, emphasizing the proliferation of configurations and its algorithmic implications (Sec. **??**). Technical details, including analytical estimates of frontier proliferation and related scaling arguments, are deferred to Appendix A.2, while explicit pseudo-code and implementation details for the frontier-based ITP routine are collected in Appendix A.1.

4.1 Imaginary time and Wick rotation

A Wick rotation replaces real time by an imaginary variable,

$$t \;\mapsto\; -\,i\,\tau, \qquad \tau \in \mathbb{R}, \tag{4.1}$$

which can be related to inverse temperature by introducing

$$\tau \;=\; \hbar\beta, \qquad \beta \;=\; \frac{1}{k_B T}. \tag{4.2}$$

Geometrically, this is a rotation by $\pi/2$ in the complex time plane, mapping Minkowski to Euclidean signature. [22]

In quantum mechanics, the real-time propagator generated by a time-independent Hamiltonian H is

$$U(t) \;=\; e^{-\frac{i}{\hbar}Ht}. \tag{4.3}$$

Under the Wick rotation $t = -i\tau$ one obtains the Euclidean (imaginary-time) evolution operator

$$U_E(\tau) \;=\; e^{-\tau H/\hbar}. \tag{4.4}$$

Setting $\tau = \hbar\beta$ and working in units with $\hbar = 1$, one writes

$$U_E(\beta) \;=\; e^{-\beta H}, \tag{4.5}$$

which is precisely the exponential appearing in the Gibbs state

$$\rho_\beta \;=\; \frac{e^{-\beta H}}{Z}, \qquad Z \;=\; \mathrm{Tr}\Big(e^{-\beta H}\Big), \qquad \beta \;=\; \frac{1}{k_B T}. \tag{4.6}$$

Thus the correspondence

$$\frac{it}{\hbar} \;\longrightarrow\; \beta \qquad (\text{or } t \to -i\hbar\beta)$$

links unitary quantum time evolution to thermal physics: imaginary time in quantum mechanics plays the role of inverse temperature $1/T$ in statistical mechanics. This same correspondence underlies Euclidean path integrals, Matsubara (imaginary-time) Green's functions, and the standard ground-state projection $\lim_{\beta\to\infty} e^{-\beta H} \propto |E_0\rangle$. [21, 22]

Given an initial trial state $|\psi(0)\rangle$, imaginary-time propagation is defined by

$$|\psi(\tau)\rangle \;=\; \frac{e^{-\tau H/\hbar}\,|\psi(0)\rangle}{\big\|e^{-\tau H/\hbar}\,|\psi(0)\rangle\big\|}, \tag{4.7}$$

with $\tau \in \mathbb{R}_{\geq 0}$. The propagator $e^{-\tau H/\hbar}$ is non-unitary and exponentially suppresses excited-state components: high-energy contributions decay as $e^{-\tau E_n/\hbar}$ faster than the ground-state component $e^{-\tau E_0/\hbar}$, so the large-τ limit projects onto the ground state without explicit diagonalization. [21]

4.2 Ground-state projection

The ground-state projection property of imaginary-time evolution can be made explicit by working in the energy eigenbasis. For simplicity, we set $\hbar = 1$ henceforth. Let $H\,|E_n\rangle = E_n\,|E_n\rangle$ with $E_0 \leq E_1 \leq E_2 \leq \cdots$ and expand an arbitrary initial state as

$$|\psi\rangle = \sum_n c_n\,|E_n\rangle\,, \qquad c_n \in \mathbb{C}. \tag{4.8}$$

Then

$$e^{-\beta H}\,|\psi\rangle = \sum_n c_n e^{-\beta E_n}\,|E_n\rangle \tag{4.9}$$

$$= e^{-\beta E_0}\left(c_0\,|E_0\rangle + \sum_{n\neq 0} c_n e^{-\beta(E_n - E_0)}\,|E_n\rangle\right). \tag{4.10}$$

Assuming E_0 is non-degenerate (or that $|\psi\rangle$ has support only in one ground-state vector), one obtains

$$\boxed{|E_0\rangle \;=\; \lim_{\beta\to\infty} \frac{e^{-\beta H}\,|\psi\rangle}{\big\|e^{-\beta H}\,|\psi\rangle\big\|}} \qquad \text{for any } |\psi\rangle \text{ with } \langle E_0|\psi\rangle \neq 0. \tag{4.11}$$

Indeed,

$$\lim_{\beta\to\infty} \frac{e^{-\beta H}\,|\psi\rangle}{\|e^{-\beta H}\,|\psi\rangle\|} = \lim_{\beta\to\infty} \frac{e^{-\beta E_0}\Big(c_0\,|E_0\rangle + \sum_{n\neq 0} c_n e^{-\beta(E_n-E_0)}\,|E_n\rangle\Big)}{e^{-\beta E_0}\sqrt{|c_0|^2 + \sum_{n\neq 0}|c_n|^2 e^{-2\beta(E_n-E_0)}}} \tag{4.12}$$

$$= \frac{c_0}{|c_0|}\,|E_0\rangle\,, \tag{4.13}$$

i.e. convergence to the ground state up to an irrelevant global phase. This is the fundamental reason why imaginary-time propagation is a ground-state preparation primitive. [21, 23]

Simple imaginary-time propagation algorithm

The operator $e^{-\beta H}$ can be represented as the limit of a product of short imaginary-time steps. Starting from the identity

$$e^{A} \;=\; \lim_{n\to\infty}\Big(\mathbf{1} + \tfrac{A}{n}\Big)^n,$$

set $A = -\beta H$ and write

$$e^{-\beta H} = \lim_{n\to\infty}\Big(\mathbf{1} - \tfrac{\beta}{n}H\Big)^n. \tag{4.14}$$

Equivalently, one may start from the truncated Taylor series

$$e^{-\beta H} = \sum_{l=0}^{\infty} \frac{(-\beta H)^l}{l!}, \tag{4.15}$$

and introduce a short imaginary-time step $\delta\tau := \beta/n$. For large n one has $\delta\tau \ll 1$, and each factor

$$e^{-\delta\tau H} = \mathbf{1} - \delta\tau\, H + \mathcal{O}(\delta\tau^2)$$

is well approximated by a first-order Euler step for the imaginary-time Schrödinger equation

$$\partial_\tau\,|\phi(\tau)\rangle = -H\,|\phi(\tau)\rangle\,, \tag{4.16}$$

where $|\phi(\tau)\rangle = e^{-\tau H}\,|\psi(0)\rangle$ denotes the *unnormalized* state. The corresponding normalized state is

$$|\psi(\tau)\rangle = \frac{|\phi(\tau)\rangle}{\|\,|\phi(\tau)\rangle\,\|},$$

which satisfies

$$\partial_\tau\,|\psi(\tau)\rangle = -\big(H - \langle H\rangle_\tau\big)\,|\psi(\tau)\rangle\,, \qquad \langle H\rangle_\tau := \langle\psi(\tau)|\,H\,|\psi(\tau)\rangle\,.$$

One usually applies finite-difference updates to the non-normalized state, then explicitly renormalizes.

Thus, for a small time step $\delta\tau$,

$$e^{-\beta H} \approx \big(\mathbf{1} - \delta\tau\, H\big)^n, \qquad n = \beta/\delta\tau.$$

This suggests the following simple iterative ITP scheme:

$$|\Psi\rangle \;\leftarrow\; \big(\mathbf{1} - \delta\tau\, H\big)\,|\Psi\rangle\,, \qquad \text{followed by } |\Psi\rangle \mapsto |\Psi\rangle\,/\|\,|\Psi\rangle\,\| \tag{4.17}$$

at each step. In the limit $\delta\tau \to 0$ and $n \to \infty$ with $n\delta\tau = \beta$ fixed, the iteration converges to $e^{-\beta H}\,|\psi(0)\rangle$, i.e. to the imaginary-time evolved state approaching the ground state. In numerical implementations, one often shifts the Hamiltonian by a constant, $H \to H - E_{\text{ref}}\mathbf{1}$, to keep the overall scale of the propagator under control.

In practice, numerical stability requires $\delta\tau$ to be sufficiently small compared to the energy scales of H, and explicit renormalization after each step prevents the norm from diverging or vanishing. This "power-method-like" projector iteration is the basis of projector and diffusion Monte Carlo methods on classical hardware. [21] On quantum hardware, variational imaginary-time evolution algorithms aim to reproduce the same flow in parameter space: instead of directly applying the non-unitary operator $(\mathbf{1} - \delta\tau H)$ to a state vector, one updates the parameters of a variational ansatz so that its infinitesimal change best approximates $-H\,|\psi\rangle$. [23, 24]

This subsection clarifies the meaning and limitations of ITP as a sort of formal (yet unphysical) "cooling" process.

Using the definition of imaginary-time evolution in Eq. (4.7) and the spectral decomposition employed in Sec. 4.2 [cf. Eq. (4.11)], the imaginary-time evolution essentially re-weights each energy eigencomponent of the initial state by a factor $e^{-\tau E_n}$(or $e^{-\beta E_n}$ in thermal notation). Relative to the ground-state contribution, the weight of an excited state with energy E_n is suppressed by a factor $e^{-\tau(E_n-E_0)}$. For imaginary times $\tau \gg 1/\Delta$, where Δ denotes the gap between the ground state and the first excited state, the amplitudes of excited eigenstates are exponentially suppressed, and the state becomes exponentially close to the ground state subspace.

Formally, the operator $e^{-\tau H}$ has the same algebraic structure as the Boltzmann weight $e^{-\beta H}$, with the identification $\beta = \tau$ (in units where $\hbar = 1$). Large imaginary time, therefore, corresponds to a large inverse temperature and, hence, to a low effective temperature. This reweighting is precisely where the analogy with a cooling process lies: imaginary-time evolution suppresses high-energy components in the same way that lowering the temperature suppresses high-energy contributions in a Gibbs state.

It is crucial, however, to emphasize that imaginary time is not a physical time coordinate. It is an auxiliary Euclidean parameter obtained by analytic continuation $t \mapsto -i\tau$ of the real-time Schrödinger equation, and the non-unitary map $e^{-\tau H}$ does not describe the dynamical evolution of an isolated quantum system in real laboratory time. From the operator-theoretic point of view, two features distinguish imaginary-time propagation from genuine dynamics:

- **Non-unitarity.** The map $|\psi\rangle \mapsto e^{-\tau H}|\psi\rangle$ is non-unitary and generically changes the norm of the state. Relative to the ground-state component, all excited-state components are exponentially suppressed as $e^{-\tau(E_n-E_0)}$. In numerical implementations one therefore renormalizes the state after each step, or equivalently shifts the Hamiltonian by a reference energy to control the overall norm. This non-unitarity is precisely what makes imaginary-time evolution an efficient projector onto low-energy states, but also makes it a purely mathematical construction rather than a literal cooling process for an isolated quantum system.

- **Many-body non-locality.** Even if H is a sum of local terms, the finite-time propagator $e^{-\tau H}$ is, in general, a generically nonlocal and highly entangling operator on the many-body Hilbert space. Quantum algorithms for imaginary-time evolution therefore approximate $e^{-\tau H}$ using sequences of local unitaries and measurements, or by reconstructing its action within a variational manifold, because there is no simple local dissipative channel whose exact action coincides with $e^{-\tau H}$ at finite τ.

Within this perspective, the "cooling" picture should be interpreted strictly as a formal analogy: imaginary time plays the role of an effective inverse temperature in the weight factors, but there is no underlying thermal bath, no real-time stochastic thermalization, and no causal local dynamics in physical spacetime that literally follows τ.

Loss of excited-state information. The same formal cooling picture explains why imaginary-time evolution deliberately discards information about excited states. Since excited-state contributions are exponentially suppressed relative to the ground state, beyond some imaginary time their amplitudes fall below numerical precision, and the resulting state is indistinguishable from a pure ground state. At that stage the excitation spectrum cannot be reconstructed from $|\psi(\tau)\rangle$ alone.

State preparation and practical advantages/limitations. Because of its projector character, imaginary-time propagation is best viewed as a state-preparation primitive rather than as a model of real-time dynamics. On classical hardware it underlies projector and diffusion Monte Carlo methods for ground-state properties of many-body systems, while on

quantum hardware it motivates quantum imaginary-time evolution (QITE) and variational imaginary-time algorithms, in which unitary circuits and measurements are engineered to approximate the non-unitary map $e^{-\tau H}$.

In this context, ITP exhibits several characteristic advantages and limitations:

- *Advantages:* (i) imaginary-time dynamics is governed by a diffusion-like equation rather than by the oscillatory unitary Schrödinger equation, which typically improves convergence towards low-energy states and enhances numerical stability; (ii) the conceptual structure is simple — formally, one "applies $e^{-\tau H}$ and renormalizes" — and admits both classical Monte Carlo realizations and hybrid quantum–classical variational implementations.
- *Limitations:* (i) many imaginary-time iterations are often required, especially in the presence of small spectral gaps; (ii) information about excited states is intentionally suppressed, so accessing excited spectra requires additional projections or modified schemes; (iii) simulating large systems remains computationally demanding, either due to sampling cost in classical projector Monte Carlo or due to repeated measurements and parameter updates in variational quantum implementations.

These features motivate the use of ITP as a ground-state preparation tool for Ising-encoded SAT instances, while also delineating the practical reach of both classical and quantum realizations of imaginary-time evolution.

4.3 Variational Ansatz

In the previous section, imaginary-time propagation was introduced as an ideal projector onto the ground state of the 3-SAT cost Hamiltonian, assuming access to the full 2^L-dimensional Hilbert space. In practice, such an exact evolution is infeasible for large L, and one must work with a restricted, efficiently representable family of states and an approximate imaginary-time update rule acting within that family.

In what follows, we instantiate this idea for Ising-encoded 3-SAT by specifying (i) a concrete variational ansatz based on superpositions of simple product states restricted to a Bloch disk, and (ii) a "frontier-based" imaginary-time propagation scheme that selectively grows this ansatz under the action of the clause projectors. The resulting state at the end of the frontier-based ITP stage will then serve as the initial procedure for a subsequent TDVP refinement.

Definition We work with a family of fully separable L-qubit product states indexed by $\lambda = 0, \ldots, D$,

$$|\rho_\lambda\rangle = \bigotimes_{k=0}^{L-1} |B_{k\lambda}\rangle\,, \qquad |B_{k\lambda}\rangle := \cos\theta_k^\lambda\ |0_k\rangle + \sin\theta_k^\lambda\ |1_k\rangle\,. \tag{4.18}$$

Each single-qubit factor $|B_{k\lambda}\rangle$ is parameterized by a single real angle θ_k^λ. We restrict every qubit to the x–z plane of the Bloch sphere (a "Bloch disk"), i.e. we fix the azimuthal phase to $\phi_k^\lambda = 0$ and only vary the polar angle θ_k^λ. By construction,

$$\cos^2\theta_k^\lambda + \sin^2\theta_k^\lambda = 1 \qquad (\forall\, k, \lambda), \tag{4.19}$$

so each $|B_{k\lambda}\rangle$ and hence each $|\rho_\lambda\rangle$ is normalized.

The full variational state at imaginary time τ is a linear superposition of $D{+}1$ such product configurations,

$$|\Psi(\tau)\rangle = \sum_{\lambda=0}^{D} a_\lambda(\tau)\ |\rho_\lambda\rangle\,, \qquad \sum_{\lambda=0}^{D} |a_\lambda(\tau)|^2 = 1, \tag{4.20}$$

where $a_\lambda(\tau)$ are real amplitudes in our implementation. Although each $|\rho_\lambda\rangle$ is not entangled, the superposition (4.20) can represent nontrivial correlations through interference between different product configurations.

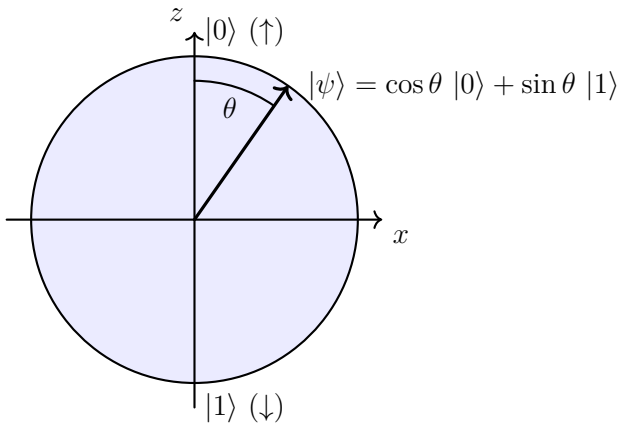


Figure 4.1: *Bloch disk*: single-qubit states restricted to the x–z plane. The polar angle θ rotates the state from $|0\rangle$ (up, $+z$) toward $|1\rangle$ (down, $-z$).

All variational degrees of freedom are collected in the parameter vector

$$\bar{w} = (a_0, \ldots, a_D;\, \theta_0^0, \ldots, \theta_{L-1}^0;\, \theta_0^1, \ldots, \theta_{L-1}^1;\, \ldots;\, \theta_0^D, \ldots, \theta_{L-1}^D). \tag{4.21}$$

Here λ labels distinct product configurations, and k runs over qubits.

Initialization We initialize along $+x$ on every qubit,

$$|\Psi_0\rangle = \bigotimes_{k=0}^{L-1} |{\rightarrow_k}\rangle\,, \qquad |{\rightarrow_k}\rangle := \frac{1}{\sqrt{2}}(|0_k\rangle + |1_k\rangle), \tag{4.22}$$

which corresponds to the seed configuration $\lambda = 0$ with

$$\theta_k^0 = \frac{\pi}{4}, \qquad \forall\, k. \tag{4.23}$$

This choice gives nonzero overlap with every computational-basis assignment (including satisfying assignments of the 3-SAT instance) and avoids starting in already "pinned" configurations such as $\theta = 0$ or $\theta = \pi/2$.

Clause projectors The cost Hamiltonian H is a sum of clause projectors, each diagonal in the computational σ^z basis (see Chapter 3.5). A clause acting on qubits (i, j, h) can be written as

$$H = \sum_c H_c = \sum_{c:(i,j,h)} O_{(i,j,h)}, \tag{4.24}$$

$$O_{(i,j,h)} = \bigotimes_{k \notin \{i,j,h\}} \mathbb{I}_k \;\otimes \bigotimes_{q \in \{i,j,h\}} \frac{1 - s_q\, \sigma_q^z}{2}. \tag{4.25}$$

where $s_q = \pm 1$ fixes which σ^z eigenstate is selected on qubit q. On a single-qubit factor

$$|B_{q\lambda}\rangle = \cos\theta_q^\lambda\, |0_q\rangle + \sin\theta_q^\lambda\, |1_q\rangle\,,$$

we use $\sigma^z\,|0\rangle = +\,|0\rangle$ and $\sigma^z\,|1\rangle = -\,|1\rangle$ to obtain

$$\frac{1 - s_q\, \sigma_q^z}{2}\, |B_{q\lambda}\rangle = \begin{cases} \cos\theta_q^\lambda\, |0_q\rangle\,, & s_q = -1, \\ \sin\theta_q^\lambda\, |1_q\rangle\,, & s_q = +1. \end{cases} \tag{4.26}$$

Thus, on each qubit in the clause, the projector keeps only the computational-basis component it demands ($|0\rangle$ if $s_q = -1$, $|1\rangle$ if $s_q = +1$) and discards the other one. In the special cases $\theta_q^\lambda = 0$ (already $|0_q\rangle$) or $\theta_q^\lambda = \pi/2$ (already $|1_q\rangle$), Eq. (4.26) can also return the zero vector when the clause demands the opposite value; such branches are interpreted as logical contradictions and are discarded.

Applying $O_{(i,j,h)}$ to a parent product state $|\rho_\lambda\rangle$ either produces:

- a *child product state* $|\rho'_\lambda\rangle_{(i,j,h)}$, in which qubits (i, j, h) are now pinned to definite computational values, and all other qubits remain unchanged, with total weight

$$A_{(i,j,h),\lambda} = A_{i\lambda} A_{j\lambda} A_{h\lambda}, \qquad A_{q\lambda} = \begin{cases} \cos\theta_q^\lambda, & s_q = -1, \\ \sin\theta_q^\lambda, & s_q = +1, \end{cases} \tag{4.27}$$

  or

- the zero vector, which we drop.

In this preliminary framework, each clause either fixes a subset of spins in a product state or excludes the state as inconsistent. Within this context, we will henceforth refer to $|\rho_\lambda\rangle$ as "replicas," since in this initialization procedure they effectively reproduce partial (or potentially complete) invalid assignments. This terminology is adopted solely for convenience and does not intend to establish any conceptual connection with the replica method discussed in 3.

4.4 Frontier-based Imaginary-Time Propagation

We now describe how imaginary-time propagation is implemented *within* the variational family introduced in Sec. 4.3. Starting Eq. (4.20)), we approximate one step of imaginary-time evolution by the first-order Euler rule

$$|\Psi(\tau + d\tau)\rangle \propto \left(\mathbb{I} - d\tau\, \hat{H}\right) |\Psi(\tau)\rangle\,, \tag{4.28}$$

where $\hat{H}$ is the 3-SAT cost Hamiltonian written as a sum of clause projectors [cf. Eq. (4.25)]. The role of the frontier-based ITP scheme is to realize Eq. (4.28) while (i) never leaving the span of product states of the form (4.18), and (ii) enlarging this span in a controlled, sparse way.

Naive clause expansion and combinatorial issues

Plugging the variational expansion (4.20) into (4.28), and using the clause structure of $\hat{H} = \sum_c O_c$ together with the single-qubit action in Eq. (4.26), one obtains the Euler-expanded update

$$|\Psi(\tau + d\tau)\rangle \approx \sum_\lambda a_\lambda\, |\rho_\lambda\rangle \;-\; d\tau \sum_\lambda \sum_c a_\lambda\, A_{c\lambda}\, |\rho'_\lambda\rangle_c\,, \tag{4.29}$$

which is the variational version of Eq. (4.28). Here $A_{c\lambda}$ is the clause weight defined in Eq. (4.27), and $|\rho'_\lambda\rangle_c$ denotes the (possibly pinned) child product state produced by clause c acting on the parent $|\rho_\lambda\rangle$.

If we were to take Eq. (4.29) at face value and, at *every* step, include in the ansatz all children $|\rho'_\lambda\rangle_c$ generated by all pairs (λ, c), the number D of distinct product states would grow combinatorially. In fact , each pair (λ, c) produces a candidate child state $|\rho'_\lambda\rangle_c$ that, in principle, should be added to the pool.

Two key mechanisms help reduce the *true* growth of the basis:

1. **Clones.** Different (λ, c) pairs can generate the *same* product configuration, i.e. the same pattern of pinned spins (same angles $\theta_k^\lambda \in \{0, \pi/2, \pi/4\}$). This happens when distinct sequences of clause projectors pin qubits in a logically equivalent way. All such children are *clones* and must be merged into a single basis state with the sum of their amplitude contributions.

2. **Incompatible branches.** If a clause demands a value for some qubit that contradicts an already pinned value, Eq. (4.26) yields the zero vector on that branch. These logically inconsistent children are projected to $|0\rangle$ and are discarded from the variational manifold.

Clone merging and annihilation of incompatible branches imply that many of the formal children in (4.29) either collapse onto existing states or vanish. After a few layers of clause applications, most new branches are either clones or contradictions, and the manifold spanned by the $|\rho_\lambda\rangle$ tends to *saturate* (see Appendix A.2).

However, the algorithmic cost of the naive update remains high: even if D eventually saturates , at each imaginary-time step one would still

- apply all clauses to all D parents,
- and, for each child, perform an expensive search over the current pool to decide whether it is a clone or a genuinely new state.

The frontier-based scheme introduced below is designed precisely to avoid this cost: instead of treating all parents symmetrically, we identify the subset of states that are actually responsible for discovering *new* product configurations at step k, and we restrict the full "create-or-merge" logic to this subset.

Toy example with two clause operators

To illustrate this more concretely, consider a simple Hamiltonian

$$\hat{H} = O_{c_1} + O_{c_2}, \tag{4.30}$$

with two clause projectors O_{c_1} and O_{c_2}. We start from a single product state $|\rho_0\rangle$ with amplitude $a_0^{(0)}$ at step $k = 0$. For clarity we focus only on the Hamiltonian part $-d\tau\,\hat{H}$ and omit the identity channel.

Step 1. Acting with the two clauses on $|\rho_0\rangle$ gives

$$-d\tau\, a_0^{(0)}\, O_{c_1}\,|\rho_0\rangle = -d\tau\, a_0^{(0)}\, A_{c_1 0}\,|\rho_1\rangle\,, \tag{4.31}$$

$$-d\tau\, a_0^{(0)}\, O_{c_2}\,|\rho_0\rangle = -d\tau\, a_0^{(0)}\, A_{c_2 0}\,|\rho_2\rangle\,, \tag{4.32}$$

where $A_{c_1 0}, A_{c_2 0}$ are the clause weights defined as in Eq. (4.27), and $|\rho_1\rangle$, $|\rho_2\rangle$ are the (possibly distinct) child product states obtained by pinning the qubits involved in c_1 and c_2. After merging and normalization, we can view the pools as

$$\mathcal{S}_0 = \{|\rho_0\rangle\}, \qquad \mathcal{S}_1 = \{|\rho_1\rangle\,, |\rho_2\rangle\}, \tag{4.33}$$

with amplitudes $(a_0^{(1)}, a_1^{(1)}, a_2^{(1)})$. Denoting the norm at this step by

$$\mathcal{N}_1 := \sqrt{\left(a_0^{(0)}\right)^2 + \left(d\tau\, A_{c_1 0} a_0^{(0)}\right)^2 + \left(d\tau\, A_{c_2 0} a_0^{(0)}\right)^2},$$

the amplitudes can be written explicitly as

$$a_0^{(1)} = \frac{a_0^{(0)}}{\mathcal{N}_1}, \tag{4.34}$$

$$a_1^{(1)} = \frac{-d\tau\, a_0^{(0)} A_{c_1 0}}{\mathcal{N}_1}, \tag{4.35}$$

$$a_2^{(1)} = \frac{-d\tau\, a_0^{(0)} A_{c_2 0}}{\mathcal{N}_1}. \tag{4.36}$$

Here $\mathcal{S}_0$ denotes the set of states first created at step 0, and $\mathcal{S}_1$ those first created at step 1. The frontier at step 1 is $\mathcal{F}^{(1)} = \mathcal{S}_1$.

Step 2. At the next step both the old state $|\rho_0\rangle$ and the new states $|\rho_1\rangle, |\rho_2\rangle$ contribute to $-d\tau\,\hat{H}\,|\Psi^{(1)}\rangle$.

From the old state $|\rho_0\rangle$ we obtain

$$-d\tau\, a_0^{(1)}\, O_{c_1}\,|\rho_0\rangle = -d\tau\, a_0^{(1)}\, A_{c_1 0}\,|\rho_1\rangle\,, \tag{4.37}$$

$$-d\tau\, a_0^{(1)}\, O_{c_2}\,|\rho_0\rangle = -d\tau\, a_0^{(1)}\, A_{c_2 0}\,|\rho_2\rangle\,. \tag{4.38}$$

Crucially, $|\rho_0\rangle$ has *regenerated* exactly the same children as at the previous step: $|\rho_1\rangle$ and $|\rho_2\rangle$. These are already in $\mathcal{S}_1$, so they contribute only to updating the amplitudes $a_1^{(2)}, a_2^{(2)}$; they do not create new basis states.

By contrast, acting with O_{c_1}, O_{c_2} on the frontier state $|\rho_1\rangle$ can generate genuinely new configurations:

$$-d\tau\, a_1^{(1)}\, O_{c_1}\,|\rho_1\rangle = -d\tau\, a_1^{(1)}\, A_{c_1 1}\,|\rho_3\rangle\,, \tag{4.39}$$

$$-d\tau\, a_1^{(1)}\, O_{c_2}\,|\rho_1\rangle = -d\tau\, a_1^{(1)}\, A_{c_2 1}\,|\rho_4\rangle\,, \tag{4.40}$$

and similarly for the other frontier state $|\rho_2\rangle$. Here $|\rho_3\rangle, |\rho_4\rangle$ denote new product states (unless they happen to coincide with existing ones), and $A_{c_1 1}, A_{c_2 1}$ are the corresponding clause weights. These new states populate the next layer $\mathcal{S}_2$ and become the new frontier $\mathcal{F}^{(2)} = \mathcal{S}_2$.

After collecting all contributions and renormalizing, the new amplitude of $|\rho_1\rangle$ at step 2 has the schematic form

$$a_1^{(2)} = \frac{\underbrace{a_1^{(1)}}_{\text{identity}} - d\tau\, a_0^{(1)} A_{c_1 0} + \dots}{\mathcal{N}_2}, \tag{4.41}$$

where $\mathcal{N}_2$ is the normalization factor at step 2, the first term in the numerator comes from the identity channel, and the ellipsis includes additional contributions from other parents (such as $|\rho_2\rangle$) that could generate a clone of $|\rho_1\rangle$.

This example highlights three key facts:

- Old states (here $|\rho_0\rangle$) still contribute to the imaginary-time update by modifying the amplitudes of already-known states ($|\rho_1\rangle, |\rho_2\rangle$), but they do not discover new configurations.
- Only the *current frontier* (here $|\rho_1\rangle, |\rho_2\rangle$ at step 1) can generate new states ($|\rho_3\rangle, |\rho_4\rangle, \dots$).
- The adjacency pattern (which parent can generate which children) is fixed once the states are created; what changes during the evolution are the amplitudes $a_\lambda^{(k)}$ flowing along this graph.

The frontier can therefore be seen as the outermost "active layer" in a genealogical graph of states.

Layered decomposition and active frontier

The toy example suggests a more general organization of the dynamics. Let $\mathcal{S}_i$ denote the set of product states that are created for the first time at ITP step i. By construction,

$$\mathcal{S}^{(k)} = \bigcup_{i=0}^{k} \mathcal{S}_i, \qquad \mathcal{F}^{(k)} = \mathcal{S}_k, \tag{4.42}$$

and the variational state after k steps can be written as

$$|\Psi^{(k)}\rangle = \sum_{i=0}^{k} \sum_{\lambda \in \mathcal{S}_i} a_\lambda^{(k)}\; |\rho_\lambda\rangle\,. \tag{4.43}$$

The product states $|\rho_\lambda\rangle$ themselves do *not* depend on k: once a state is created and assigned to some layer $\mathcal{S}_i$, its bitstring of pinned spins is fixed. The imaginary-time evolution modifies only the amplitudes $a_\lambda^{(k)}$.

Applying one Euler step,

$$|\Psi^{(k+1)}\rangle \propto (\mathbb{I} - d\tau\, \hat{H})\, |\Psi^{(k)}\rangle\,, \tag{4.44}$$

we can separate the Hamiltonian contribution according to the layers:

$$-d\tau\, \hat{H}\, |\Psi^{(k)}\rangle = -d\tau \sum_{i=0}^{k} \sum_{\lambda\in\mathcal{S}_i} a_\lambda^{(k)}\, \hat{H}\, |\rho_\lambda\rangle \tag{4.45}$$

$$= -d\tau \sum_{i=0}^{k-1} \sum_{\lambda\in\mathcal{S}_i} a_\lambda^{(k)}\, \hat{H}\, |\rho_\lambda\rangle \;-\; d\tau \sum_{\mu\in\mathcal{S}_k} a_\mu^{(k)}\, \hat{H}\, |\rho_\mu\rangle\,. \tag{4.46}$$

By the construction of the algorithm, the action of $\hat{H}$ on a given parent $|\rho_\lambda\rangle$ is completely determined by:

- the pattern of pinned spins in $|\rho_\lambda\rangle$,
- the fixed list of clauses $\{O_c\}$.

This data does not depend on the step index k or on the amplitudes. Therefore, the set of children that can appear in $\hat{H}\,|\rho_\lambda\rangle$ is fixed once and for all.

When λ first appears in layer $\mathcal{S}_i$, it lies in the frontier $\mathcal{F}^{(i)}$ and, at the next update $i \to i{+}1$:

- applies *all* clauses to $|\rho_\lambda\rangle$,
- for each compatible child, check whether it already exists in $\bigcup_{j=0}^{i} \mathcal{S}_j$.
- if it is new, insert it into $\mathcal{S}_{i+1}$.

At that point, we have *exhausted* all children of λ that can ever appear in a single application of $\hat{H}$: any such child is either already in $\bigcup_{j=0}^{i} \mathcal{S}_j$ or has just been put into $\mathcal{S}_{i+1}$. Hence, for all later steps $k \geq i+1$,

$$\hat{H}\, |\rho_\lambda\rangle \in \operatorname{span}\left(\bigcup_{j=0}^{i+1} \mathcal{S}_j\right) \subseteq \operatorname{span}\Big(\mathcal{S}^{(k)}\Big)\,, \qquad \lambda \in \mathcal{S}_i,\ i < k. \tag{4.47}$$

In other words, the first term on the right-hand side of Eq. (4.46) *cannot* generate new product configurations at step $k{+}1$: it only redistributes amplitude among states already in $\mathcal{S}^{(k)}$. The *only* part of $-d\tau\, \hat{H}\, |\Psi^{(k)}\rangle$ that can create genuinely new states is the second term in Eq. (4.46), where $\hat{H}$ acts on the current frontier $\mathcal{S}_k$. The compatible children of these frontier parents that do not yet exist in $\mathcal{S}^{(k)}$ are precisely the states assigned to $\mathcal{S}_{k+1}$ and thus to the new frontier $\mathcal{F}^{(k+1)}$.

We can conclude that:

- the *frontier* $\mathcal{S}_k$ is the unique *generative layer*: only its states can enlarge the variational manifold at step $k{+}1$;
- all older layers $\mathcal{S}_i$ with $i < k$ are still dynamically important: they provide amplitude flow into already-known states, but they never introduce new product configurations again.

This layered viewpoint makes the hierarchical, graph-like structure explicit: each $|\rho_\lambda\rangle$ is a node, edges connect parents to children under clause applications, and imaginary-time evolution amounts to repeatedly pushing amplitude along this fixed graph. The frontier is the moving outer shell of this graph where new nodes may still appear.

Amplitude update rules

It is important to stress that a single Euler step at most produces new product states while no acting on the old ones, but that is not the only effect. Each step updates the $a_\lambda^{(k)}$ of all the states. For each ordered pair of product states $|\rho_\lambda\rangle, |\rho_\mu\rangle$ we define a transition coefficient

$$\kappa_{\mu\lambda} := \sum_{c:\ \mathrm{child}(\lambda,c)=\mu} A_{c\lambda}, \tag{4.48}$$

where the sum runs over all clauses c such that applying O_c to $|\rho_\lambda\rangle$ produces the child $|\rho_\mu\rangle$ with weight $A_{c\lambda}$ (as in Eq. (4.27)). If no clause connects λ to μ we have $\kappa_{\mu\lambda} = 0$.

Collecting all contributions and using

$$|\Psi^{(k)}\rangle = \sum_{\lambda\in\mathcal{S}^{(k)}} a_\lambda^{(k)}\ |\rho_\lambda\rangle\,,$$

the unnormalized amplitude of $|\rho_\mu\rangle$ after one step is

$$\tilde{a}_\mu^{(k+1)} = a_\mu^{(k)} - d\tau \sum_{\lambda\in\mathcal{S}^{(k)}} \kappa_{\mu\lambda}\, a_\lambda^{(k)}, \tag{4.49}$$

where we understand $a_\mu^{(k)} = 0$ if $|\rho_\mu\rangle \notin \mathcal{S}^{(k)}$. The first term in Eq. (4.49) comes from the identity channel, while the second term collects the flow of amplitude from all parents $|\rho_\lambda\rangle$ that can generate $|\rho_\mu\rangle$ in one clause application. After computing all $\tilde{a}_\mu^{(k+1)}$ for $|\rho_\mu\rangle \in \mathcal{S}^{(k+1)}$, we renormalize:

$$\mathcal{N}_{k+1} = \sqrt{\sum_{\nu\in\mathcal{S}^{(k+1)}} \big(\tilde{a}_\nu^{(k+1)}\big)^2}, \qquad a_\mu^{(k+1)} = \frac{\tilde{a}_\mu^{(k+1)}}{\mathcal{N}_{k+1}}. \tag{4.50}$$

The frontier rule does not change Eq. (4.49) itself, but constrains *where* new indices μ are allowed to appear. At step $k \to k{+}1$:

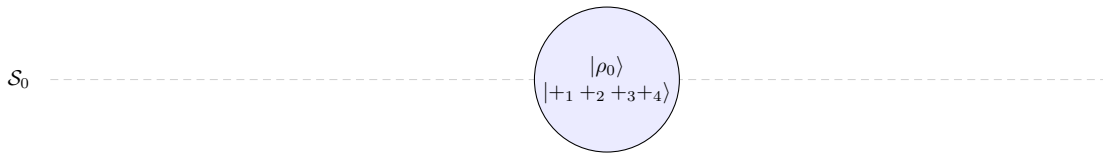
- if $|\rho_\mu\rangle \in \mathcal{S}^{(k)}$ is an old state, we include in the sum in Eq. (4.49) all parents $\lambda \in \mathcal{S}^{(k)}$ such that $\kappa_{\mu\lambda} \neq 0$, regardless of whether they are in the frontier or not;
- If $|\rho_\mu\rangle$ is a genuinely new state created at step $k{+}1$, it must first be produced by at least one parent $\lambda \in \mathcal{F}^{(k)}$. Once created, it can then receive additional contributions from non-frontier parents $\lambda \in \mathcal{S}^{(k)} \setminus \mathcal{F}^{(k)}$ that also regenerate $|\rho_\mu\rangle$.

In this way the set of states $\{|\rho_\lambda\rangle\}$, and therefore the index set $\mathcal{S}^{(k)}$, can grow only through the frontier, while the amplitude updates remain faithful to the full imaginary-time action $(\mathbb{I} - d\tau\,\hat{H})$ on the current variational state.

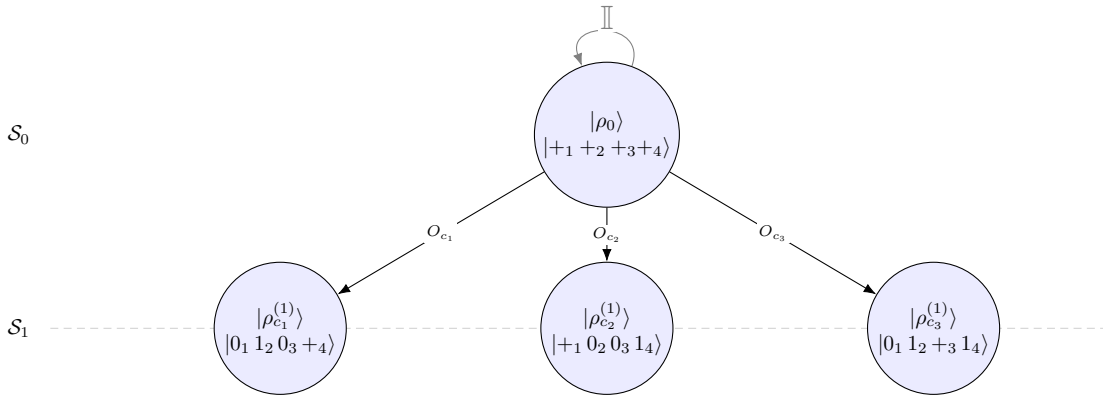
Graphical interpretation

Figures 4.2 and 4.3 provide a graphical interpretation of the frontier-based ITP update. The three panels show three snapshots of the same clause structure taken at successive ITP steps $k = 0,1,2$. The left-hand labels $\mathcal{S}_\prime, \mathcal{S}_\infty, \mathcal{S}_\in$ indicate the layer to which each node belongs, i.e. the step at which that product state first appeared (cf. the layered decomposition of Sec. 4.4).

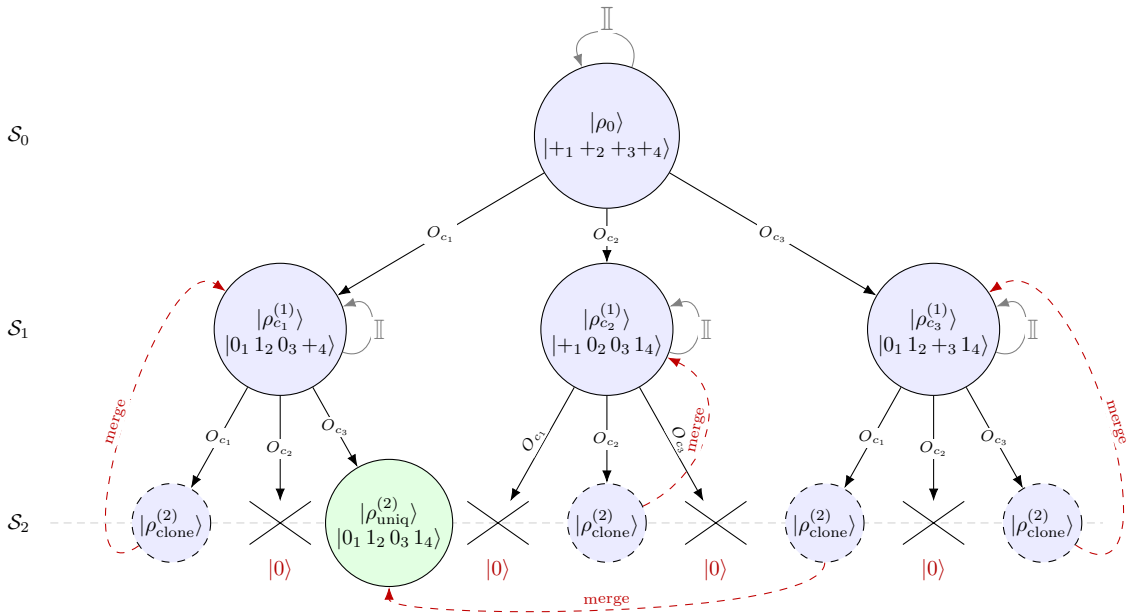
The graphical legend in Fig. 4.3 explains the symbols. Filled circles represent valid product states $|\rho_\lambda\rangle$ (consisitent partial assignments); a colored halo marks states that are in the frontier at the corresponding step. A grey self-loop represents the identity contribution in $(\mathbb{I} - d\tau\,\hat{H})$. Edges labelled by O_{c_i} show the action of clause projectors. Red dashed arrows labelled "merge" indicate clones whose amplitude is merged into an existing node, while nodes ending in a cross and labelled $|0\rangle$ correspond to incompatible branches where the clause demands a value that contradicts an already pinned spin and the projector returns the zero vector. The bottom panel



(a) Step 0



(b) Step 1



(c) Step 2

Figure 4.2: Frontier-based imaginary-time update with pruning and saturation

of Fig. 4.3 specifies the particular 3-SAT instance used (three clauses on four variables) and the literal convention, so each node can be read as a partial assignment of the SAT variables.

Step 0 (panel a). The first panel shows the situation before any ITP update has been applied. We start from a single delocalized product state $|\rho_0\rangle$ in which no spin is pinned (all qubits at $\theta = \pi/4$, i.e. $|+\rangle$). This state is the only element of the pool, $\mathcal{S}^{(0)} = \{\rho_0\}$, and at the same time the only element of the frontier, $\mathcal{F}^{(0)} = \{\rho_0\}$. No self-loop is drawn here, because we

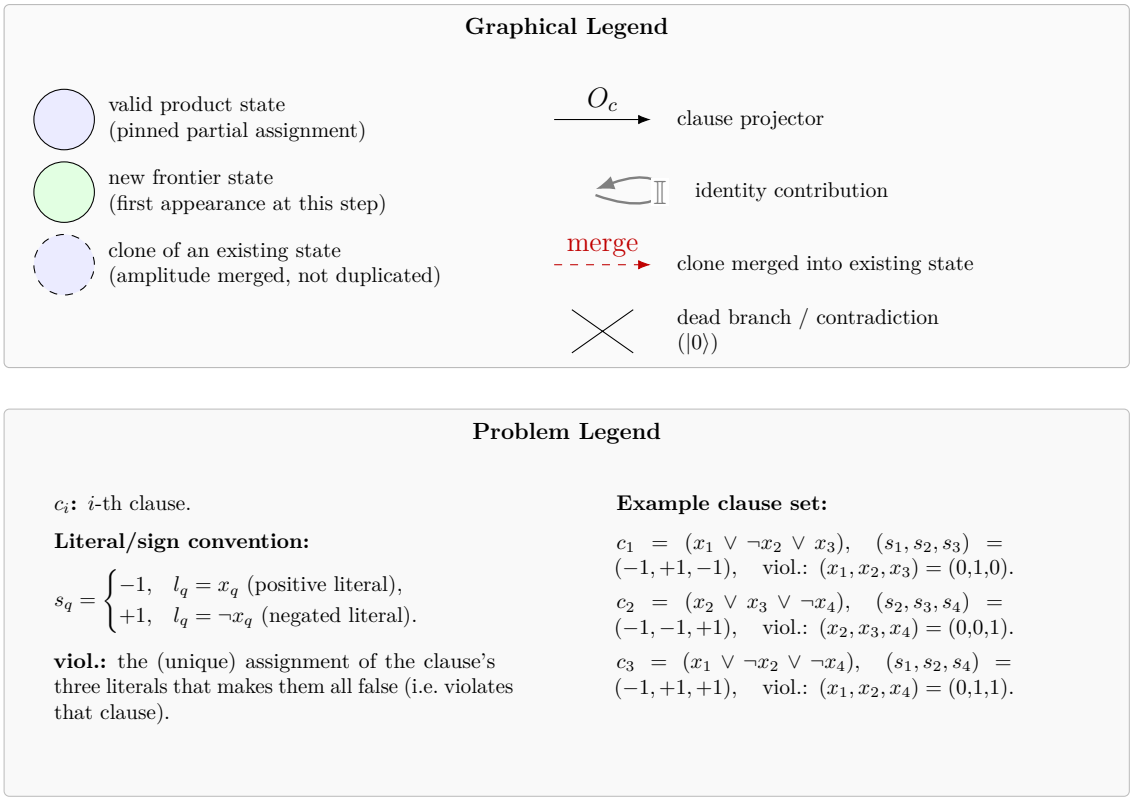


Figure 4.3: Graphical legend (top) and problem legend (bottom).

have not yet applied $(\mathbb{I} - d\tau\,\hat{H})$.

Step 1 (panel b). The second panel represents the graph immediately after the first ITP step $k = 0 \to 1$. The root $|\rho_0\rangle$ is now equipped with a grey self-loop, encoding the identity contribution $\mathbb{I}\,|\rho_0\rangle$. Each clause projector O_{c_i} acts once on the frontier state $|\rho_0\rangle$, producing children along the three outgoing edges. For this particular problem instance, each clause yields a distinct, compatible child, so three new product states are created in the layer $\mathcal{S}_\infty$; these are drawn in the row $\mathcal{S}_\infty$ and form the new frontier $\mathcal{F}^{(1)}$. At this stage no clones or contradictions appear, so there are no red "merge" arrows and no crossed $|0\rangle$ nodes. The pool is now $\mathcal{S}^{(\infty)} = \mathcal{S}_\prime \cup \mathcal{S}_\infty$.

Step 2 (panel c). The third panel depicts the graph after the second ITP step $k = 1 \to 2$. The identity channel now acts on *all* states in $\mathcal{S}^{(1)}$, so both $|\rho_0\rangle$ (in S_0) and its children in S_1 carry grey self-loops. Clause projectors are applied to every state, but with different status depending on whether the parent is in the frontier or not:

- The old state $|\rho_0\rangle$ is no longer in the frontier. When O_{c_i} acts on $|\rho_0\rangle$ at Step 2, it regenerates exactly the same children as at Step 1. These contributions are represented by red dashed arrows that merge back into the existing nodes in S_1: $|\rho_0\rangle$ continues to pump amplitude into those states, but cannot create any new configuration.

- The states in S_1, highlighted as the frontier $\mathcal{F}^{(1)}$ in panel (b), are now the generative layer. Each clause O_{c_i} is applied to each of these frontier parents. Some branches are incompatible with previously pinned spins and terminate in crossed $|0\rangle$ nodes; others lead to children that coincide with states already present in S_0 or S_1, and are therefore shown as red "merge" arrows into those existing nodes (clone merging). Only those branches that reach genuinely new, consistent product states appear as new nodes in the row S_2 and form the new frontier $\mathcal{F}^{(2)}$ (in the figure one of them is additionally shaded, as it will later be used as TDVP initial state).

A striking feature of panel (c) is that almost all branches from $\mathcal{F}^{(1)}$ either (i) end in contradictions or (ii) fall back onto states that were already in $\mathcal{S}^{(1)}$. In this sense the graph has reached *saturation*: no additional product states are discovered. The imaginary-time evolution continues to update the amplitudes $a_\lambda^{(k)}$ through $(\mathbb{I} - d\tau\,\hat{H})$, but the underlying manifold of product states $\{|\rho_\lambda\rangle\}$ is now fixed. The three panels together make visually explicit how identity contributions, clause projections, clone merging, and annihilation of incompatible branches work in concert with the frontier restriction to yield a controlled, hierarchical exploration of the space of partial assignments.

4.5 From ITP to TDVP

Each ITP step $k \to k{+}1$ is implemented in three phases: (i) an identity pass that copies all states in $\mathcal{S}^{(k)}$ to a working pool and seeds their amplitudes, (ii) a frontier expansion in which only $\mathcal{F}^{(k)}$ is allowed to create new product configurations, and (iii) a non-frontier expansion in which older states merely redistribute amplitude into already known configurations. The resulting amplitudes are then globally normalized.

The detailed algorithm, together with the bitmask representation of product states and the corresponding pseudocode, is given in Appendix A.1

After a fixed number of ITP steps we obtain

$$|\Psi^{(n)}\rangle = \sum_{\lambda \in \mathcal{S}^{(n)}} a_\lambda^{(n)}\,|\rho_\lambda\rangle\,,$$

a structured, low-rank superposition of product states. Each $|\rho_\lambda\rangle$ corresponds to a consistent partial assignment of SAT variables, with some spins already pinned in the computational basis according to the clause structure.

$|\Psi^{(n)}\rangle$ is then used as the initial state for the time-dependent variational principle (TDVP)5. In TDVP we freeze the set $\{|\rho_\lambda\rangle\}$ and treat all angles $\{\theta_k^\lambda\}$ and amplitudes $\{a_\lambda\}$ as continuous variational parameters. The evolution follows the projected imaginary-time equation

$$S_{\bar{w}}\,\delta\bar{w} = -\gamma\,F_{\bar{w}},$$

where $S_{\bar{w}}$ is the TDVP metric tensor, and $F_{\bar{w}}$ is the corresponding force vector. Operationally: frontier-based ITP constructs and populates an expressive but still tractable ansatz manifold; TDVP then performs a geometry-aware optimization.

4.6 Scaling of the Variational Representation

At every step of the frontier-based imaginary-time propagation (ITP) discussed in Sec. 4.4, the wavefunction is represented as a superposition of product states $\{|\rho_\lambda\rangle\}_{\lambda=0}^{D(k)}$ with real amplitudes $a_\lambda^{(k)}$. Once a product configuration has been created, its internal structure (pinned spins and Bloch-disk angles) never changes again: the only way to enlarge the variational manifold is to *discover new product states* through the action of $-d\tau\,\hat{H}$ on the current frontier. The total number of real variational parameters at step k scales as

$$N_{\text{params}}(k) \;\sim\; \big(D(k)+1\big)(L+1),$$

So, understanding the scaling of $D(k)$ is equivalent to understanding the scaling of the ansatz dimension.

Layer–resolved mean-field model

In Appendix A.2 we make this picture quantitative by constructing a *layer–resolved mean-field* model for the growth of the frontier manifold. Product configurations are grouped into layers labeled by the number k of pinned spins, and for each layer and ITP step, we track the expected frontier population $F_k(t)$ (states first created at step t) and the cumulative number of distinct states $D_k(t)$. The total frontier size and manifold size are $N(t) = \sum_k F_k(t)$ and $D_{\mathrm{tot}}(t) = \sum_k D_k(t)$, respectively.

When a clause acts on a parent in layer k, it can touch three unfrozen spins, two unfrozen spins, and one frozen spin, and so on. The corresponding combinatorial factors determine the probabilities that the clause produces a locally consistent child in layers $k+1, k+2, k+3$. All local compatibility effects are summarized by a single parameter $\rho \in [0,1]$, which is the probability that a literal on a pinned spin is satisfied by the current assignment. Explicit expressions for the local success probabilities $p_0(k; L, \rho)$, $p_1(k; L, \rho)$, $p_2(k; L, \rho)$ and for the resulting branching $Y^{(t)}_{k \to k'}$ are given in Appendix A.2.

Local branching alone would vastly overcount configurations because different parents (or different clauses on the same parent) frequently produce the *same* child. To incorporate these global clone collisions, the mean-field model applies a simple occupancy argument within each layer: the $Y^{(t)}_k$ candidate children in layer k are treated as independent draws from the S_k possible configurations in that layer, so that only a fraction of them are genuinely new. This yields explicit recursions for $F_k(t)$ and $D_k(t)$ (see Eq. (A.19) in the appendix).

Crucially, the compatibility parameter ρ entering the recursions must be interpreted as an *effective phenomenological* quantity. It absorbs correlations due to repeatedly applying the same clause set, the heterogeneity of frontier states, and the enhanced production of clones once the frontier concentrates in a small region of configuration space. By fitting the mean-field predictions for $D_{\mathrm{tot}}(t)$ to frontier-based ITP data across system sizes and step numbers, we obtain a system-size-dependent estimate of ρ_{eff}. We then use this estimate to predict manifold growth over many time steps for much larger L, where direct ITP simulation would be computationally prohibitive. After this calibration, the mean-field model has no free parameters and can predict the growth and saturation of the frontier manifold for larger instances, where full ITP simulations are too costly (see Chapter 6).

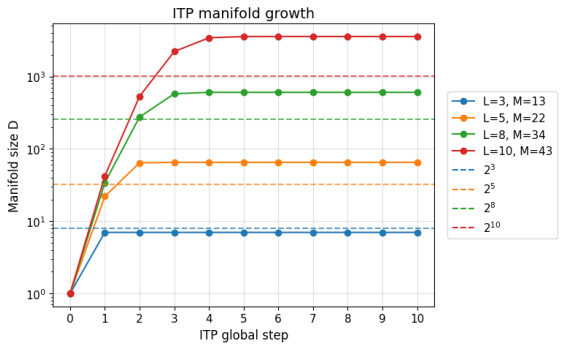
Comparison with frontier-based ITP

The quality of the mean-field description is illustrated in Fig. 4.4, which summarizes the frontier dynamics for $L \in \{3,5,8,10\}$ at critical clause density $\alpha \simeq 4.3$.

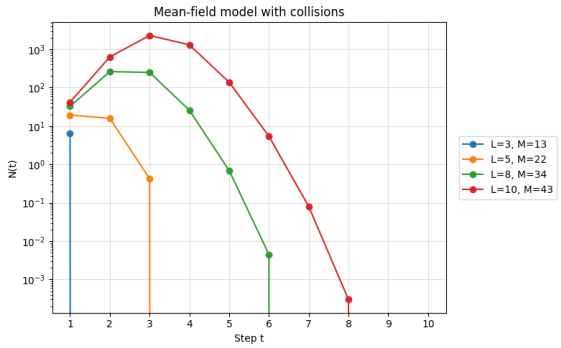
Panel (a) shows the manifold size $D^{\mathrm{ITP}}(t)$ measured directly in the frontier-based ITP. For all system sizes we observe a common qualitative pattern: a very rapid initial proliferation of product states, followed by an equally rapid saturation. Already after $t \simeq 3$–4 global steps the number of distinct configurations approaches a plateau value $D_\star(L)$ and the frontier effectively "freezes". The saturation time is $O(1)$ and only weakly dependent on L, but the plateau height grows quickly with the system size: for $L = 10$ the saturated manifold is already of order 10^3, larger than the Hilbert-space dimension $2^{10} = 1024$ indicated by the dashed line.

Panel (b) displays the mean-field prediction for the number of *new* frontier states $N(t)$ per step. With the calibrated value $\rho_{\mathrm{eff}} = 0.1$ the model reproduces the sharp initial burst (effective branching factor $\beta_t \gtrsim M$ at early times) and the subsequent rapid collapse of $N(t)$ to values compatible with zero. This behaviour is controlled by two mechanisms that are explicit in the recursions of Appendix A.2: on the one hand, clauses are increasingly likely to hit frozen spins with incompatible literals, which suppresses the local success probability $p_{\mathrm{loc}}(k; L, \rho)$; on the other hand, even when many candidates are generated, the occupancy factor in Eq. (A.19) forces the number of *truly new* configurations to saturate once a layer has been largely explored.

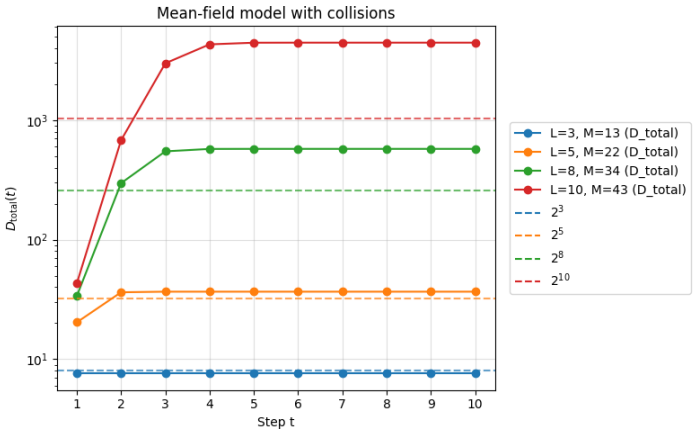
The cumulative mean-field curves in panel (c) show the resulting $D_{\mathrm{tot}}(t)$ and can be directly compared with the ITP data of panel (a). On the logarithmic scale used here the agreement is



(a) ITP manifold growth $D^{\mathrm{ITP}}(t)$ for $L = 3,5,8,10$. Dashed lines indicate the Hilbert-space sizes 2^L.



(b) Mean-field number of new frontier states $N(t) = \sum_k F_k(t)$ for the same system sizes, using $\rho_{\mathrm{eff}} = 0.1$.



(c) Mean-field cumulative number of distinct product states $D_{\mathrm{tot}}(t) = \sum_k D_k(t)$. Dashed lines again show the corresponding 2^L.

Figure 4.4: Frontier manifold growth at critical clause density $\alpha \simeq 4.3$. Panel (a) shows the manifold size measured in the full frontier-based ITP, while panels (b) and (c) display the corresponding predictions of the layer–resolved mean-field model with a single phenomenological parameter $\rho_{\mathrm{eff}} \approx 0.1$.

essentially quantitative: both the initial exponential burst and the saturation plateaux for each L are captured by the mean-field dynamics once the effective compatibility parameter is fixed. This supports the interpretation of ρ as a compact phenomenological summary of the complex correlations created by the frontier-based ITP.

Scaling of the saturated manifold

Beyond a certain depth $t_\star(L)$ the frontier becomes extinct, $N(t) \to 0$, and $D_{\mathrm{tot}}(t)$ saturates to a finite limit $D_\star(L) = \lim_{t\to\infty} D_{\mathrm{tot}}(t)$, which we call the *saturated basis size*. The mean-field analysis of Appendix A.2, performed with the calibrated value $\rho_{\mathrm{eff}} \approx 0.1$, shows that

- for fixed L, the early-time growth of $D_{\mathrm{tot}}(t)$ is approximately exponential in t until the occupancy effects become dominant and the frontier is quenched;
- for fixed depth in the pre-saturation regime, the manifold size grows very rapidly with L, and the saturated value obeys an essentially exponential scaling

$$D_\star(L) \;\sim\; \exp\!\big[\gamma(\alpha)\, L\big], \tag{4.51}$$

with an effective rate $\gamma(\alpha)$ that is smaller than, but of the same order as, $\log 2$ for the critical density $\alpha \simeq 4.3$.

In other words, frontier pruning and clone merging substantially reduce the effective exponent and prefactor compared to the full Hilbert-space dimension 2^L, but the size of the variational manifold remains *essentially exponential* in L (at least as long as one has not yet fully saturated the layers) and it saturates at a value that exceed the Hilbert space dimension . This explains why, in practice, we restrict ourselves to shallow ITP schedules where $D_{\mathrm{tot}}(t) \ll 2^L$, and why the TDVP refinement stage introduced later in the work is crucial: frontier-based ITP alone still explores an exponentially large set of product configurations, and only through a subsequent variational compression can this manifold be projected onto a much smaller, dynamically relevant subspace (see Chapters 5 and 6).

Chapter 5

Time-Dependent Variational Principle (TDVP)

This chapter derives and motivates the imaginary-time TDVP dynamics for our variational ansatz, and explains how we solve the resulting equations in practice. Throughout, we work with the 3-SAT Ising Hamiltonian encoding the cost function of a clause instance (Chapter 3), so that imaginary-time flow corresponds to energy minimization, i.e. clause satisfaction, within a restricted variational family.

The time-dependent variational principle provides a systematic way of restricting the exact time evolution generated by a Hamiltonian $\hat{H}$ to a lower-dimensional manifold of variational states $\{|\Psi_{\bar{w}}\rangle\}$. Instead of evolving a full state in Hilbert space, one tracks the evolution of a finite set of parameters $\bar{w}(\tau)$ that define the variational ansatz and are chosen such that $|\Psi_{\bar{w}(\tau)}\rangle$ approximates the exact trajectory as closely as possible. In the imaginary-time setting considered here, this induces effective descent dynamics for the energy functional restricted to the variational manifold.

We begin by reviewing the conceptual framework that underlies the imaginary-time time-dependent variational principle (TDVP) and its associated geometric interpretation. Subsequently, we formulate the imaginary-time TDVP as a projection of the exact imaginary-time evolution onto the chosen variational manifold. We then proceed to compute the metric tensor S and the force vector F that enter the TDVP equations, and we apply this formalism to our explicit multi-product-state ansatz. We discuss the numerical stabilization of the update via regularization of S and the solution of the resulting linear system with MINRES-QLP [25], emphasizing the close analogy to preconditioned SR updates in VMC. This includes a detailed comparison between an explicit and an operator-based representation of the TDVP metric, together with practical pseudo-code for the full TDVP+MINRES-QLP update (App. A.4). We further relate these implementation choices to the frontier dynamics by analyzing how the proliferation of product replicas controls the scaling of the metric S and force F, and how this, in turn, constrains both the memory footprint and the convergence behavior of MINRES-QLP (Sec. 5.5 and App. A.2).

5.1 Conceptual framework

At a conceptual level, the time-dependent variational principle provides a systematic way of restricting the exact time evolution generated by a Hamiltonian $\hat{H}$ to a lower-dimensional manifold of variational states. Historically, this idea goes back to the Dirac–Frenkel and McLachlan formulations of the time-dependent variational principle, where one requires that the time derivative of a variational state minimizes the norm of the residual of the time-dependent Schrödinger equation within the variational manifold.[1]

In imaginary time, the same construction leads to an effective gradient flow for the energy

[1] See, e.g., McLachlan's original paper on variational solutions of the time-dependent Schrödinger equation [26] and modern expositions such as [27].

functional restricted to the variational manifold. In the full Hilbert space, normalized imaginary-time evolution implements a projector onto the ground state (or, more generally, onto the lowest-energy state with nonzero overlap with the initial state). When one restricts the dynamics to a variational family $\{|\Psi_{\bar{w}}\rangle\}$, imaginary-time TDVP selects the trajectory that best approximates this projector in the sense of McLachlan's minimal-residual principle. Geometrically, this can be viewed as a steepest-descent flow of the energy with respect to the Fubini–Study metric on projective Hilbert space, pulled back to the parameter space $\bar{w}$. The corresponding metric tensor is precisely the matrix $S_{\bar{w}}$ introduced below and is closely related to the quantum geometric tensor and quantum Fisher information.

The logic followed here is parallel to the geometric derivation underlying stochastic reconfiguration (SR) in variational Monte Carlo (VMC), originally proposed by Sorella [28, 29] and later reformulated in geometric terms for neural-network quantum states by Kaubruegger, Pastori and Budich [30]. In SR, one interprets the variational update of the parameters as an approximate imaginary-time step projected onto the variational manifold, with the SR matrix playing the role of a quantum Fisher metric. In our setting we adopt the same geometric viewpoint, but we implement it *deterministically* for our specific multi-product-state ansatz, without Monte Carlo sampling: all matrix elements of $S_{\bar{w}}$ and of the force vector $F_{\bar{w}}$ are evaluated analytically.

Imaginary-time TDVP and its real-time counterpart have been successfully applied to a wide range of variational families. A notable example is the TDVP for matrix product states (MPS), where projecting the Schrödinger equation onto the MPS manifold leads to efficient algorithms for simulating both real- and imaginary-time dynamics of quantum lattice systems [31, 32, 33]. More recently, the same formalism has been adapted to neural-network quantum states [34, 35, 36], where SR is interpreted explicitly as an imaginary-time TDVP in parameter space. In parallel, variational quantum algorithms such as the variational quantum eigensolver (VQE) and its imaginary-time variants perform a similar projection of the dynamics onto the manifold of parametrized quantum circuits [14, 10]. Our setting can be viewed as a fully classical, deterministic analogue of these approaches, tailored to the multi-product-state ansatz introduced in the previous chapter.

In the remainder of this chapter we make this conceptual framework explicit for our ansatz. We first write down the projected imaginary-time evolution on the variational manifold and identify the associated metric $S_{\bar{w}}$ and force vector $F_{\bar{w}}$. We then specialize these objects to the multi-product-state structure and finally discuss how to solve the resulting linear systems efficiently and in a numerically stable way.

5.2 Imaginary-time TDVP as a projected evolution

We start from the exact imaginary-time evolution generated by a Hamiltonian $\hat{H}$,

$$|\Psi(\tau)\rangle \;=\; \frac{e^{-\tau\hat{H}}\,|\Psi(0)\rangle}{\|e^{-\tau\hat{H}}\,|\Psi(0)\rangle\,\|}, \tag{5.1}$$

which solves the imaginary-time Schrödinger equation

$$\frac{\partial}{\partial\tau}\,|\Psi(\tau)\rangle = -\big(\hat{H} - E(\tau)\big)\,|\Psi(\tau)\rangle\,, \qquad E(\tau) = \langle\Psi(\tau)|\hat{H}|\Psi(\tau)\rangle\,, \tag{5.2}$$

with a time-dependent energy shift $E(\tau)$ enforcing the normalization $\langle\Psi(\tau)|\Psi(\tau)\rangle = 1$ at all times. The exponential $e^{-\tau\hat{H}}$ suppresses high-energy components of the initial state exponentially in τ, so that $|\Psi(\tau)\rangle$ converges towards the ground state (or any lowest-energy state overlapping with $|\Psi(0)\rangle$). In our setting $\hat{H}$ is the 3-SAT Ising Hamiltonian encoding the cost function of a clause instance (Chapter 3.5); thus imaginary-time flow corresponds to energy minimization, i.e. clause satisfaction.

In principle, evolving according to this equation would require manipulating a full vector in Hilbert space, whose dimension grows exponentially with the number of spins. To circumvent

this, we constrain the wavefunction to lie in a variational manifold $\{|\Psi_{\bar{w}}\rangle\}$ parametrized by a set of variables $\bar{w} = (w_1, w_2, \dots)$. We assume that the (normalized) state depends on τ only through these parameters:

$$|\hat{\Psi}(\tau)\rangle \equiv |\hat{\Psi}_{\bar{w}(\tau)}\rangle \;=\; \frac{|\Psi_{\bar{w}(\tau)}\rangle}{\sqrt{\langle\Psi_{\bar{w}(\tau)}|\Psi_{\bar{w}(\tau)}\rangle}}. \tag{5.3}$$

The TDVP then prescribes how the parameters $\bar{w}(\tau)$ should evolve so that $|\hat{\Psi}_{\bar{w}(\tau)}\rangle$ approximates the exact imaginary-time trajectory as closely as possible, in the sense of a local variational principle.

A short exact imaginary-time step $d\tau$ acts as

$$|\hat{\Psi}(\tau + d\tau)\rangle = e^{-d\tau \hat{H}}\,|\hat{\Psi}(\tau)\rangle \simeq |\hat{\Psi}(\tau)\rangle - d\tau\,\hat{H}\,|\hat{\Psi}(\tau)\rangle\,, \tag{5.4}$$

where in the last expression we have expanded the exponential to first order in $d\tau$ and implicitly assumed a subsequent renormalization of the state.[2] On the other hand, within the variational family, varying the parameters by $w_k \mapsto w_k + d\tau\,\dot{w}_k$ changes the state as

$$|\hat{\Psi}(\tau + d\tau)\rangle \simeq |\hat{\Psi}(\tau)\rangle + d\tau \sum_k \dot{w}_k \Big[\, |\partial_{w_k}\hat{\Psi}\rangle - |\hat{\Psi}\rangle\,\langle\hat{\Psi}|\partial_{w_k}\hat{\Psi}\rangle\,\Big], \tag{5.5}$$

where

$$|\partial_{w_k}\hat{\Psi}\rangle \equiv \frac{\partial}{\partial w_k}\,|\hat{\Psi}_{\bar{w}}\rangle\,, \qquad \dot{w}_k \equiv \frac{dw_k}{d\tau}. \tag{5.6}$$

The subtraction of the component parallel to $|\hat{\Psi}\rangle$ enforces norm conservation to first order and removes the gauge freedom associated with an overall phase and amplitude of the variational state. In geometric terms, the vectors in square brackets are tangent vectors to the projective manifold of normalized states.

Equating (5.4) and (5.5) to leading order in $d\tau$, and projecting onto the tangent space of the variational manifold, gives the TDVP equations. To do this, it is convenient to define the orthogonalized tangent vectors

$$|j_{\bar{w}}\rangle \;=\; |\partial_{w_j}\hat{\Psi}_{\bar{w}}\rangle - |\hat{\Psi}_{\bar{w}}\rangle\,\langle\hat{\Psi}_{\bar{w}}|\partial_{w_j}\hat{\Psi}_{\bar{w}}\rangle\,. \tag{5.7}$$

By construction $\langle\hat{\Psi}_{\bar{w}}|j_{\bar{w}}\rangle = 0$ for every j, so these $|j_{\bar{w}}\rangle$ span the tangent plane at $\bar{w}$ in the manifold of normalized variational states.

Projecting the evolution equation onto $\langle j_{\bar{w}}|$ yields

$$\sum_k \dot{w}_k \Big[\langle\partial_{w_j}\hat{\Psi}|\partial_{w_k}\hat{\Psi}\rangle - \langle\partial_{w_j}\hat{\Psi}|\hat{\Psi}\rangle\,\langle\hat{\Psi}|\partial_{w_k}\hat{\Psi}\rangle\Big] = -\Big[\langle\partial_{w_j}\hat{\Psi}|\hat{H}|\hat{\Psi}\rangle - \langle\partial_{w_j}\hat{\Psi}|\hat{\Psi}\rangle\,\langle\hat{\Psi}|\hat{H}|\hat{\Psi}\rangle\Big]\,. \tag{5.8}$$

The left-hand side encodes how fast the state moves within the variational manifold when the parameters change at rates $\dot{w}_k$, while the right-hand side is the projection of $-\hat{H}\,|\hat{\Psi}\rangle$ onto the same tangent directions.

It is convenient to introduce two central objects:

$$(S_{\bar{w}})_{jk} \equiv \langle\partial_{w_j}\hat{\Psi}|\partial_{w_k}\hat{\Psi}\rangle - \langle\partial_{w_j}\hat{\Psi}|\hat{\Psi}\rangle\,\langle\hat{\Psi}|\partial_{w_k}\hat{\Psi}\rangle\,, \tag{5.9}$$

$$(F_{\bar{w}})_j \equiv \langle\partial_{w_j}\hat{\Psi}|\hat{H}|\hat{\Psi}\rangle - \langle\partial_{w_j}\hat{\Psi}|\hat{\Psi}\rangle\,\langle\hat{\Psi}|\hat{H}|\hat{\Psi}\rangle\,. \tag{5.10}$$

The matrix $S_{\bar{w}}$ is the pullback of the Fubini–Study metric to the variational parameter space. In the context of VMC it coincides (up to complex conjugation and a real part) with the SR

[2] More precisely, one should replace $\hat{H}$ by $\hat{H} - E(\tau)$ in (5.4). This change only affects the component of the evolution parallel to $|\hat{\Psi}\rangle$ and is therefore irrelevant once we project onto the tangent space orthogonal to $|\hat{\Psi}\rangle$.

matrix and with the quantum Fisher information associated with the variational family. As such, $S_{\bar{w}}$ is positive semi-definite and becomes singular whenever two directions in parameter space generate the same physical variation of the state. The vector $F_{\bar{w}}$ is the effective "force" driving imaginary-time descent of the energy: for normalized states one can show that, up to an overall factor and taking the real part, $(F_{\bar{w}})_j$ is precisely the derivative of the energy expectation value with respect to w_j.

With these definitions, (5.8) becomes

$$\sum_k (S_{\bar{w}})_{jk}\, \dot{w}_k = -(F_{\bar{w}})_j, \qquad \forall j, \tag{5.11}$$

or, in vector form,

$$S_{\bar{w}}\, \dot{\bar{w}} = -F_{\bar{w}}. \tag{5.12}$$

Equation (5.12) is the imaginary-time TDVP equation of motion on the variational manifold. It can be interpreted as a natural-gradient flow in parameter space: the velocity $\dot{\bar{w}}$ is obtained by raising the energy gradient $F_{\bar{w}}$ with the metric $S_{\bar{w}}^{-1}$. This equation is identical in structure to the SR equation of motion used in VMC and neural-network quantum states [28, 29, 34, 30], except that here we regard it as *deterministic*: we compute $S_{\bar{w}}$ and $F_{\bar{w}}$ analytically for our ansatz, rather than sampling them stochastically.

For numerical updates it is convenient to recast (5.12) in discrete form:

$$S_{\bar{w}}\, \delta\bar{w} = -\gamma\, F_{\bar{w}}, \tag{5.13}$$

where $\gamma > 0$ plays the role of an imaginary-time step. Solving this linear system provides the parameter displacement $\delta\bar{w}$ that best approximates an infinitesimal imaginary-time step within the variational manifold. In practice, γ controls the trade-off between faithfully following the continuous TDVP flow (small γ) and making rapid progress in energy minimization (larger γ). The potentially ill-conditioned nature of $S_{\bar{w}}$ and the large number of variational parameters motivate the use of regularization and Krylov-subspace solvers such as MINRES-QLP, as discussed in Sec. 5.4.2.

5.3 Our variational ansatz and its tangent space

5.3.1 Parametrization of the state

We consider a variational state given by a coherent superposition of $D{+}1$ product states:

$$|\Psi\rangle = \sum_{\lambda=0}^{D} a_\lambda\, |\rho_\lambda\rangle\,, \qquad |\rho_\lambda\rangle = \bigotimes_{k=0}^{L-1} |B_{k\lambda}\rangle\,, \tag{5.14}$$

where each local spin state is parameterized as

$$|B_{k\lambda}\rangle = \cos\theta_k^\lambda\ |{\uparrow_k}\rangle + \sin\theta_k^\lambda\ |{\downarrow_k}\rangle\,. \tag{5.15}$$

The complex amplitudes a_λ and the angles θ_i^λ are our variational parameters.

We collect them into a single parameter vector

$$\bar{w} = \Big(a_0, \ldots, a_D;\, \theta_0^0, \ldots, \theta_{L-1}^0;\, \theta_0^1, \ldots, \theta_{L-1}^1;\, \ldots;\, \theta_0^D, \ldots, \theta_{L-1}^D\Big), \tag{5.16}$$

and we will write $w_j \in \{a_\lambda\} \cup \{\theta_i^\lambda\}$ with the understanding that ∂_{w_j} means partial differentiation with respect to one of these coordinates.

The cost Hamiltonian we minimize is the 3-SAT Ising Hamiltonian

$$\hat{H} = \sum_{c:(y,z,h)} \hat{O}_{(y,z,h)}, \tag{5.17}$$

$$\hat{O}_{(y,z,h)} = \bigotimes_{k\neq y,z,h} \mathbb{I}_k \bigotimes_{q=y,z,h} \frac{\mathbb{I}_q - s_q \sigma_z^q}{2}, \tag{5.18}$$

where the clause-dependent sign s_q is

$$s_q = \begin{cases} -1, & \text{literal } l_q \text{ appears un-negated (variable } x_q), \\ +1, & \text{literal } l_q \text{ appears negated } (\neg x_q). \end{cases} \tag{5.19}$$

$\hat{H}$ counts violated clauses. Ground states of $\hat{H}$ therefore correspond to satisfying assignments (energy 0).

5.3.2 Tangent vectors: derivatives of the ansatz

To construct $S_{\bar{w}}$ and $F_{\bar{w}}$, we need the parameter derivatives of $|\Psi\rangle$.

Derivative with respect to a_λ. Varying a_λ selects the corresponding product state:

$$|\partial_{a_\lambda}\Psi\rangle = |\rho_\lambda\rangle = \bigotimes_{k=0}^{L-1} |B_{k\lambda}\rangle \,. \tag{5.20}$$

Derivative with respect to θ_i^λ. Changing θ_i^λ only affects site i of configuration λ. Define

$$|M_{i\lambda}\rangle = \frac{\partial}{\partial\theta_i^\lambda} |B_{i\lambda}\rangle = -\sin\theta_i^\lambda \,|\uparrow_i\rangle + \cos\theta_i^\lambda \,|\downarrow_i\rangle \,. \tag{5.21}$$

Then

$$|\partial_{\theta_i^\lambda}\Psi\rangle = a_\lambda \,|\partial_{\theta_i^\lambda}\rho_\lambda\rangle = a_\lambda \bigotimes_{k\neq i} |B_{k\lambda}\rangle \otimes |M_{i\lambda}\rangle \,. \tag{5.22}$$

5.3.3 Metric tensor $S_{\bar{w}}$

The TDVP "metric" (also called the covariance matrix in SR language) is

$$(S_{\bar{w}})_{j,k} = \langle\partial_{w_j}\Psi|\partial_{w_k}\Psi\rangle - \langle\partial_{w_j}\Psi|\Psi\rangle\,\langle\Psi|\partial_{w_k}\Psi\rangle \,, \tag{5.23}$$

where we have dropped hats for readability (i.e. we work with the state $|\Psi\rangle$ assuming normalization is taken into account in the subtraction terms).

Due to the linear combination of product states nature of our ansatz, these matrix elements can be computed analytically, block by block.

In what follows, we employ the approximation delineated in Appendix A.3. The comprehensive pseudo-code utilized in our implementation is articulated in Appendix A.4.

Case A: $w_j = \theta_i^\lambda$, $w_k = \theta_j^\mu$. The first term gives

$$\langle\partial_{\theta_i^\lambda}\Psi|\partial_{\theta_j^\mu}\Psi\rangle = a_\lambda a_\mu \prod_{k=0}^{L-1} \cos(\theta_k^\lambda - \theta_k^\mu) \times \begin{cases} -\tan(\theta_i^\lambda - \theta_i^\mu)\,\tan(\theta_j^\lambda - \theta_j^\mu), & i \neq j, \\ 1, & i = j \,, \end{cases} \tag{5.24}$$

while

$$\langle\partial_{\theta_i^\lambda}\Psi|\Psi\rangle = \sum_\nu a_\nu a_\lambda \prod_{k=0}^{L-1} \cos(\theta_k^\lambda - \theta_k^\nu) \left[-\tan(\theta_i^\lambda - \theta_i^\nu)\right], \tag{5.25}$$

$$\langle\Psi|\partial_{\theta_j^\mu}\Psi\rangle = \sum_\eta a_\eta a_\mu \prod_{k=0}^{L-1} \cos(\theta_k^\eta - \theta_k^\mu)\tan(\theta_j^\eta - \theta_j^\mu). \tag{5.26}$$

Combining these,

$$(S_{\bar{w}})_{\partial_{\theta_i^\lambda},\partial_{\theta_j^\mu}} = a_\lambda a_\mu \prod_{k=0}^{L-1} \cos(\theta_k^\lambda - \theta_k^\mu) \times \begin{cases} -\tan(\theta_i^\lambda - \theta_i^\mu)\ \tan(\theta_j^\lambda - \theta_j^\mu), & i \neq j, \\ 1, & i = j \end{cases}$$
$$- \left[\sum_\nu a_\nu a_\lambda \prod_{k=0}^{L-1} \cos(\theta_k^\lambda - \theta_k^\nu) \left(-\tan(\theta_i^\lambda - \theta_i^\nu) \right) \right]$$
$$\times \left[\sum_\eta a_\eta a_\mu \prod_{k=0}^{L-1} \cos(\theta_k^\eta - \theta_k^\mu) \tan(\theta_j^\eta - \theta_j^\mu) \right]. \tag{5.27}$$

Case B: $w_j = a_\lambda$**,** $w_k = a_\mu$**.** We have

$$\langle \partial_{a_\lambda} \Psi | \partial_{a_\mu} \Psi \rangle = \langle \rho_\lambda | \rho_\mu \rangle = \prod_{k=0}^{L-1} \cos(\theta_k^\lambda - \theta_k^\mu), \tag{5.28}$$

and

$$\langle \partial_{a_\lambda} \Psi | \Psi \rangle = \sum_\nu a_\nu \prod_{k=0}^{L-1} \cos(\theta_k^\lambda - \theta_k^\nu), \tag{5.29}$$

$$\langle \Psi | \partial_{a_\mu} \Psi \rangle = \sum_\eta a_\eta \prod_{k=0}^{L-1} \cos(\theta_k^\eta - \theta_k^\mu). \tag{5.30}$$

Therefore

$$(S_{\bar{w}})_{\partial_{a_\lambda},\partial_{a_\mu}} = \prod_{k=0}^{L-1} \cos(\theta_k^\lambda - \theta_k^\mu)$$
$$- \left(\sum_\nu a_\nu \prod_{k=0}^{L-1} \cos(\theta_k^\lambda - \theta_k^\nu) \right) \left(\sum_\eta a_\eta \prod_{k=0}^{L-1} \cos(\theta_k^\eta - \theta_k^\mu) \right). \tag{5.31}$$

Case C: $w_j = a_\lambda$**,** $w_k = \theta_j^\mu$**.** Here

$$\langle \partial_{a_\lambda} \Psi | \partial_{\theta_j^\mu} \Psi \rangle = a_\mu \prod_{k=0}^{L-1} \cos(\theta_k^\lambda - \theta_k^\mu)\ \tan(\theta_j^\lambda - \theta_j^\mu), \tag{5.32}$$

and with

$$\langle \partial_{a_\lambda} \Psi | \Psi \rangle = \sum_\nu a_\nu \prod_{k=0}^{L-1} \cos(\theta_k^\lambda - \theta_k^\nu), \tag{5.33}$$

$$\langle \Psi | \partial_{\theta_j^\mu} \Psi \rangle = \sum_\eta a_\eta a_\mu \prod_{k=0}^{L-1} \cos(\theta_k^\eta - \theta_k^\mu)\ \tan(\theta_j^\eta - \theta_j^\mu), \tag{5.34}$$

we obtain

$$(S_{\bar{w}})_{\partial_{a_\lambda},\partial_{\theta_j^\mu}} = a_\mu \prod_{k=0}^{L-1} \cos(\theta_k^\lambda - \theta_k^\mu)\ \tan(\theta_j^\lambda - \theta_j^\mu)$$
$$- \left(\sum_\nu a_\nu \prod_{k=0}^{L-1} \cos(\theta_k^\lambda - \theta_k^\nu) \right) \left(\sum_\eta a_\eta a_\mu \prod_{k=0}^{L-1} \cos(\theta_k^\eta - \theta_k^\mu)\ \tan(\theta_j^\eta - \theta_j^\mu) \right). \tag{5.35}$$

Case D: $w_j = \theta_i^\lambda$, $w_k = a_\mu$. By symmetry we find

$$\langle \partial_{\theta_i^\lambda} \Psi | \partial_{a_\mu} \Psi \rangle = a_\mu \prod_{k=0}^{L-1} \cos(\theta_k^\lambda - \theta_k^\mu) \, [- \tan(\theta_i^\lambda - \theta_i^\mu)], \tag{5.36}$$

and

$$\langle \partial_{\theta_i^\lambda} \Psi | \Psi \rangle = \sum_\nu a_\nu a_\lambda \prod_{k=0}^{L-1} \cos(\theta_k^\lambda - \theta_k^\nu) \, [- \tan(\theta_i^\lambda - \theta_i^\nu)], \tag{5.37}$$

$$\langle \Psi | \partial_{a_\mu} \Psi \rangle = \sum_\eta a_\eta \prod_{k=0}^{L-1} \cos(\theta_k^\eta - \theta_k^\mu), \tag{5.38}$$

so that

$$\begin{aligned}(S_{\bar{w}})_{\partial_{\theta_i^\lambda}, \partial_{a_\mu}} = a_\mu & \prod_{k=0}^{L-1} \cos(\theta_k^\lambda - \theta_k^\mu) \, [- \tan(\theta_i^\lambda - \theta_i^\mu)] \\ & - \left(\sum_\nu a_\nu a_\lambda \prod_{k=0}^{L-1} \cos(\theta_k^\lambda - \theta_k^\nu) \, [- \tan(\theta_i^\lambda - \theta_i^\nu)] \right) \left(\sum_\eta a_\eta \prod_{k=0}^{L-1} \cos(\theta_k^\eta - \theta_k^\mu) \right).\end{aligned} \tag{5.39}$$

Taken together, these blocks define the full TDVP metric $S_{\bar{w}}$ for our ansatz.

5.3.4 Force vector $F_{\bar{w}}$

The TDVP driving term (also known as SR "force") is

$$F_j(\bar{w}) = \langle \partial_{w_j} \Psi | \hat{H} | \Psi \rangle - \langle \partial_{w_j} \Psi | \Psi \rangle \, \langle \Psi | \hat{H} | \Psi \rangle \, . \tag{5.40}$$

To evaluate $F_{\bar{w}}$ we need: (i) $\langle \Psi | \hat{H} | \Psi \rangle$, (ii) $\langle \partial_{w_j} \Psi | \Psi \rangle$ (already given above), and (iii) $\langle \partial_{w_j} \Psi | \hat{H} | \Psi \rangle$.

First note that

$$\hat{H} \, |\Psi\rangle = \sum_\lambda a_\lambda \sum_{c:(i,j,h)} \bigotimes_{k \neq i,j,h}^{L-1} |B_{k\lambda}\rangle \bigotimes_{q=i,j,h} |S_{q\lambda}\rangle \, , \tag{5.41}$$

where, for each site q appearing in the clause c,

$$|S_{q\lambda}\rangle = \begin{cases} \cos\theta_q^\lambda \, |0_q\rangle \, , & s_q = -1, \\ \sin\theta_q^\lambda \, |1_q\rangle \, , & s_q = +1. \end{cases} \tag{5.42}$$

Using this structure, one obtains

$$\langle \Psi | \hat{H} | \Psi \rangle = \sum_{\eta,\nu} a_\eta a_\nu \prod_{k=0}^{L-1} \cos(\theta_k^\eta - \theta_k^\nu) \sum_{c:(y,z,h)} \prod_{q=y,z,h} \begin{cases} \dfrac{\cos\theta_q^\eta \cos\theta_q^\nu}{\cos(\theta_q^\eta - \theta_q^\nu)}, & s_q = -1, \\ \dfrac{\sin\theta_q^\eta \sin\theta_q^\nu}{\cos(\theta_q^\eta - \theta_q^\nu)}, & s_q = +1 \, . \end{cases} \tag{5.43}$$

The overlaps $\langle \partial_{\theta_i^\lambda} \Psi | \Psi \rangle$ and $\langle \partial_{a_\lambda} \Psi | \Psi \rangle$ were already given in Section 5.3.3. For completeness:

$$\langle \partial_{\theta_i^\lambda} \Psi | \Psi \rangle = \sum_\nu a_\nu a_\lambda \prod_{k=0}^{L-1} \cos(\theta_k^\lambda - \theta_k^\nu) \, [- \tan(\theta_i^\lambda - \theta_i^\nu)], \tag{5.44}$$

$$\langle \partial_{a_\lambda} \Psi | \Psi \rangle = \sum_\nu a_\nu \prod_{k=0}^{L-1} \cos(\theta_k^\lambda - \theta_k^\nu). \tag{5.45}$$

The mixed matrix elements $\langle \partial_{w_j} \Psi | \hat{H} | \Psi \rangle$ then follow by inserting the explicit form of $\hat{H} \, |\Psi\rangle$ and again using tensor-product factorization. For example:

- If $w_j = \theta_i^\lambda$ and i *not* one of the three spins in the clause (y, z, h), one finds

$$\langle \partial_{\theta_i^\lambda} \Psi | \hat{H} | \Psi \rangle = \sum_\nu a_\lambda a_\nu \prod_{k=0}^{L-1} \cos(\theta_k^\lambda - \theta_k^\nu) \, [- \tan(\theta_i^\lambda - \theta_i^\nu)] \times \sum_{c:(y,z,h)} \prod_{q=y,z,h} \begin{cases} \dfrac{\cos\theta_q^\lambda \cos\theta_q^\nu}{\cos(\theta_q^\lambda - \theta_q^\nu)}, & s_q = -1, \\ \dfrac{\sin\theta_q^\lambda \sin\theta_q^\nu}{\cos(\theta_q^\lambda - \theta_q^\nu)}, & s_q = +1 \, . \end{cases} \tag{5.46}$$

- If $w_j = \theta_i^\lambda$ and i *is* one of the spins hit by a clause (y, z, h), say $i = y$ (or analogously z or h), then the corresponding site is replaced by the derivative of $|S_{q\lambda}\rangle$. Writing $c/\{i\}$ for the other two spins of the clause, this gives

$$\begin{aligned} \langle \partial_{\theta_i^\lambda} \Psi | \hat{H} | \Psi \rangle = \sum_\nu a_\lambda a_\nu \prod_{k=0}^{L-1} \cos(\theta_k^\lambda - \theta_k^\nu) \\ \times \sum_{c:(y,z,h)} \left[\prod_{q \in c/\{i\}} \begin{cases} \dfrac{\cos\theta_q^\lambda \cos\theta_q^\nu}{\cos(\theta_q^\lambda - \theta_q^\nu)}, & s_q = -1, \\ \dfrac{\sin\theta_q^\lambda \sin\theta_q^\nu}{\cos(\theta_q^\lambda - \theta_q^\nu)}, & s_q = +1 \end{cases} \right] \\ \times \left[\prod_i \begin{cases} \dfrac{-\sin\theta_i^\lambda \cos\theta_i^\nu}{\cos(\theta_i^\lambda - \theta_i^\nu)}, & s_i = -1, \\ \dfrac{\cos\theta_i^\lambda \sin\theta_i^\nu}{\cos(\theta_i^\lambda - \theta_i^\nu)}, & s_i = +1 \end{cases} \right]. \end{aligned} \tag{5.47}$$

- If $w_j = a_\lambda$, the derivative simply projects out $|\rho_\lambda\rangle$:

$$\langle \partial_{a_\lambda} \Psi | \hat{H} | \Psi \rangle = \sum_\nu a_\nu \prod_{k=0}^{L-1} \cos(\theta_k^\lambda - \theta_k^\nu) \sum_{c:(y,z,h)} \prod_{q=y,z,h} \begin{cases} \dfrac{\cos\theta_q^\lambda \cos\theta_q^\nu}{\cos(\theta_q^\lambda - \theta_q^\nu)}, & s_q = -1, \\ \dfrac{\sin\theta_q^\lambda \sin\theta_q^\nu}{\cos(\theta_q^\lambda - \theta_q^\nu)}, & s_q = +1 \, . \end{cases} \tag{5.48}$$

The final force entries are then

$$(F_{\bar{w}})_{\theta_i^\lambda} = \langle \partial_{\theta_i^\lambda} \Psi | \hat{H} | \Psi \rangle - \langle \partial_{\theta_i^\lambda} \Psi | \Psi \rangle \, \langle \Psi | \hat{H} | \Psi \rangle \, , \tag{5.49}$$

$$(F_{\bar{w}})_{a_\lambda} = \langle \partial_{a_\lambda} \Psi | \hat{H} | \Psi \rangle - \langle \partial_{a_\lambda} \Psi | \Psi \rangle \, \langle \Psi | \hat{H} | \Psi \rangle \, . \tag{5.50}$$

Thus both $S_{\bar{w}}$ and $F_{\bar{w}}$ can be computed from analytic overlaps of tensor-product states, with no stochastic sampling.

5.3.5 TDVP equations of motion for the ansatz

For our ansatz, the TDVP parameter update is obtained by solving

$$S_{\bar{w}} \, \delta\bar{w} = -\gamma \, F_{\bar{w}}. \tag{5.51}$$

Here:

- $S_{\bar{w}}$ is the metric/covariance tensor assembled in Section 5.3.3;
- $F_{\bar{w}}$ is the force vector from Section 5.3.4;
- $\delta\bar{w}$ is the update of all parameters $(a_\lambda, \theta_i^\lambda)$ at that imaginary-time step.

Geometrically, $\delta\bar{w}$ is the displacement in parameter space that best approximates a short imaginary-time step $e^{-\gamma \hat{H}}$ within the variational manifold. This is the deterministic analogue of SR in neural-network VMC [30].

5.4 Regularization and Krylov solution of the TDVP equation

The linear system (5.51) is not automatically well behaved. The matrix $S_{\bar{w}}$ can be nearly singular or extremely ill-conditioned for several reasons:

- **Redundant directions.** Different parameter combinations can represent almost the same physical state, making some tangent vectors nearly linearly dependent. Those directions correspond to very small eigenvalues of $S_{\bar{w}}$.
- **Numerical imprecision.** Finite precision (and, in Monte Carlo approaches, sampling noise [30]) can spoil exact positivity and introduce spurious tiny or even slightly negative eigenvalues.

If we naively used $\delta\bar{w} = -\gamma\, S_{\bar{w}}^{-1} F_{\bar{w}}$, small eigenvalues of $S_{\bar{w}}$ would blow up those directions and produce unphysically large parameter updates. To avoid this, we stabilize $S_{\bar{w}}$ by regularization and solve the resulting linear system with a Krylov method designed for ill-conditioned symmetric problems.

5.4.1 Metric regularization

Following the strategy discussed in [30], we introduce a small positive regularization parameter λ_{reg} and define the regularized metric

$$S_{\bar{w}}^{(\text{reg})} = S_{\bar{w}} + \lambda_{\text{reg}}\, \mathbb{I}. \tag{5.52}$$

This uniform diagonal shift has two effects: (i) it opens up near-zero eigenvalues (lifting almost-null directions), and (ii) it penalizes very large steps in parameter space, acting like a Tikhonov / ridge regularizer.

An alternative is to rescale only the diagonal entries:

$$\left(S_{\bar{w}}^{(\text{reg})}\right)_{k,k} = (1 + \lambda_{\text{reg}})\, (S_{\bar{w}})_{k,k}, \qquad \left(S_{\bar{w}}^{(\text{reg})}\right)_{j\neq k} = (S_{\bar{w}})_{j\neq k}. \tag{5.53}$$

This preserves the off-diagonal correlations between parameters while mildly inflating each parameter's self-overlap. In practice, both (5.52) and (5.53) suppress runaway updates along directions where $S_{\bar{w}}$ would otherwise be nearly singular.

After regularization, the TDVP step becomes

$$S_{\bar{w}}^{(\text{reg})}\, \delta\bar{w} = -\gamma\, F_{\bar{w}}. \tag{5.54}$$

This linear system is now better conditioned, but it can still be far from trivial: it is large, symmetric, and can remain close to singular.

5.4.2 Solving the linear system with MINRES-QLP

After regularization, the TDVP update is obtained by solving the symmetric linear system in Eq.(5.54) where $S_{\bar{w}}^{(\text{reg})}$ is large, dense in principle, and only guaranteed to be *symmetric*, not strictly positive definite. In practice, $S_{\bar{w}}^{(\text{reg})}$ can still be ill-conditioned or effectively singular due to redundant or flat directions in the variational manifold. Direct inversion or naive linear solvers are therefore unreliable both numerically and computationally.

For these reasons we use MINRES-QLP [25]. This is a Krylov subspace method designed specifically for solving large symmetric systems (and symmetric least-squares problems) that may be indefinite or nearly singular. In our case, it lets us compute a physically meaningful update $\delta\bar{w}$ without ever inverting or factorizing $S_{\bar{w}}^{(\text{reg})}$ explicitly.

Krylov viewpoint

Given a symmetric matrix A and a right-hand side b, Krylov methods construct approximate solutions in a sequence of subspaces

$$\mathcal{K}_k(A, b) = \mathrm{span}\{b,\ Ab,\ A^2b,\ \ldots,\ A^{k-1}b\}. \tag{5.55}$$

At iteration k, the solver chooses $x_k \in \mathcal{K}_k(A, b)$ that (in some sense) best solves $Ax = b$. The key point is that Krylov methods only ever need matrix–vector products Av. This is crucial for us because

- we can compute the action $v \mapsto S_{\bar{w}}^{(\mathrm{reg})} v$ using the matrix-free application of $S_{\bar{w}}$ described in Appendix A.5, without ever forming $S_{\bar{w}}$ explicitly;
- but we do *not* want to build or store $S_{\bar{w}}^{(\mathrm{reg})}$ as a dense matrix, which would scale quadratically in the number of parameters.

Therefore, from a cost and memory standpoint, Krylov is exactly the right regime for our TDVP equations.

Classical MINRES vs. MINRES-QLP

Classical MINRES (Minimum Residual) applies the Lanczos process to A: it builds an orthonormal basis $\{v_1, \ldots, v_k\}$ for $\mathcal{K}_k(A, b)$ together with a small $k \times k$ *tridiagonal* matrix T_k such that

$$AV_k \ \approx\ V_k T_k, \tag{5.56}$$

where $V_k = [v_1 \cdots v_k]$. The method then solves a reduced least-squares problem involving T_k (which is tiny compared to A) to pick $x_k = V_k y_k$ that minimizes the residual norm $\|Ax_k - b\|$ over $\mathcal{K}_k(A, b)$. Because T_k is tridiagonal and symmetric, MINRES can update x_k and the residual norm with short recurrences, keeping memory usage modest (growing essentially linearly with the number of parameters).

However, standard MINRES has two key weaknesses:

1. If A is singular or nearly singular, MINRES still tries to minimize the residual, but it does *not* guarantee that the vector x_k it returns is the *minimum-norm* solution. In fact, it can "wander" into almost-null directions of A and pick a very large $\|x_k\|_2$, which for us means a huge and physically meaningless parameter update $\delta\bar{w}$ along a nearly flat direction of $S_{\bar{w}}^{(\mathrm{reg})}$.
2. If A is strongly ill-conditioned, the small tridiagonal problem T_k becomes numerically delicate. Plain MINRES uses a QR-like processing of T_k, which can lose stability when T_k is close to rank-deficient. The resulting x_k may then be dominated by numerical noise in those almost-null directions.

MINRES-QLP is engineered to fix exactly these issues for symmetric (possibly indefinite) systems. It does so by upgrading the way the reduced problem in the Krylov space is solved.

QLP refinement and minimum-norm solutions

In MINRES-QLP, the Lanczos tridiagonal T_k is not only processed by left-orthogonal transformations (like QR), but also by *right* orthogonal transformations. More precisely, MINRES-QLP computes a decomposition of the form

$$Q_k^\top\, T_k\, P_k \ =\ L_k, \tag{5.57}$$

where Q_k and P_k are orthogonal (products of Givens/Householder rotations), and L_k is lower (tri)angular. This is called a QLP decomposition. Conceptually:

- The left orthogonal factor Q_k plays a similar role as in QR, gradually driving T_k towards triangular form.

- The additional right orthogonal factor P_k then "rebalances" that triangular form into L_k, whose diagonal elements behave like robust approximations to the singular values of T_k.

Why is this important? Because the diagonal of L_k immediately reveals whether T_k (and hence, effectively, A restricted to $\mathcal{K}_k$) is close to rank-deficient. In other words, (5.57) tells us which directions in the Krylov space are nearly in the nullspace of A.

MINRES-QLP then solves the reduced problem in such a way that the corresponding Krylov approximation $x_k = V_k y_k$ is *the minimum Euclidean-norm solution* among those with minimal residual. In the singular or nearly singular case, this is mathematically the pseudoinverse solution $x_k \approx A^\dagger b$ [25]. Physically, for TDVP, this does exactly what we want:

- If $S_{\bar{w}}^{(\mathrm{reg})}$ has very soft/flat directions (tiny eigenvalues), there are infinitely many $\delta\bar{w}$ that almost solve $S_{\bar{w}}^{(\mathrm{reg})}\delta\bar{w} = -\gamma F_{\bar{w}}$.

- MINRES-QLP picks *the one with the smallest* $\|\delta\bar{w}\|_2$, i.e., the most conservative change of the parameters consistent with the imaginary-time descent.

This is crucial: in our variational algorithm, we prefer a gentle, well-controlled step in parameter space over an arbitrarily large jump in a nearly redundant direction.

Another advantage of using the QLP decomposition of T_k is that it provides built-in conditioning diagnostics. Because L_k exposes approximate singular values of T_k, the algorithm can detect when the projected system is becoming numerically singular. Operationally, this means:

- We can monitor how ill-conditioned the effective linear system is at each Krylov iteration.

- If certain modes are essentially null, MINRES-QLP does not try to "chase" them by producing a giant $\delta\bar{w}$. Instead, it damps them automatically via the minimum-norm criterion.

Practically, MINRES-QLP keeps short recurrences like MINRES, so the memory cost grows linearly with the number of parameters (no dense factorizations are stored). Each iteration uses one application of $S_{\bar{w}}^{(\mathrm{reg})}$ to a vector, which in our case we can compute efficiently without explicitly forming $S_{\bar{w}}^{(\mathrm{reg})}$. The method naturally stops when either the residual $\|S_{\bar{w}}^{(\mathrm{reg})}\delta\bar{w} + \gamma F_{\bar{w}}\|$ is below a tolerance or the Krylov space stops producing meaningful improvement (e.g. the projected system is effectively rank-deficient and the minimum-norm solution has stabilized).

Why MINRES-QLP is the right solver for TDVP

To summarize why this particular solver is the best fit for our TDVP equation:

- **Symmetry-compatible:** $S_{\bar{w}}^{(\mathrm{reg})}$ is symmetric by construction, and MINRES-QLP explicitly targets symmetric systems (unlike generic GMRES-type solvers for non-Hermitian matrices).

- **Robust to near-singularity:** The ansatz often has redundant directions in parameter space; these show up as almost-null eigenvalues of $S_{\bar{w}}^{(\mathrm{reg})}$. MINRES-QLP was designed to handle exactly this regime and produce stable updates.

- **Minimum-norm update direction:** Among all $\delta\bar{w}$ that (approximately) solve $S_{\bar{w}}^{(\mathrm{reg})}\delta\bar{w} = -\gamma F_{\bar{w}}$, MINRES-QLP returns the one with smallest $\|\delta\bar{w}\|_2$. This is physically meaningful: we move in parameter space as little as possible while still descending in energy.

- **Matrix-free:** Only matrix–vector products with $S_{\bar{w}}^{(\mathrm{reg})}$ are required. We avoid storing or inverting the full metric, which would be prohibitive for large systems.
- **Numerical diagnostics :** The QLP step (the $Q_k^\top T_k P_k = L_k$ factorization) exposes rank structure and effective singular values of the reduced problem, giving us direct feedback on whether the TDVP step is well-conditioned or dominated by nearly flat directions.

In the language of variational quantum dynamics, MINRES-QLP therefore implements a stable, geometry-aware update: it solves the regularized TDVP equation in a way that respects the (symmetrized) quantum-geometric metric $S_{\bar{w}}$, avoids exploding along redundant directions, and produces the "least aggressive" descent step $\delta\bar{w}$ consistent with energy minimization [25, 30].

5.5 Scaling of the TDVP metric and force with frontier growth

The MINRES-QLP step described above implicitly assumes that we can evaluate the regularized metric $S_{\bar{w}}^{(\mathrm{reg})}$ and the force vector $F_{\bar{w}}$ for the current variational state. In practice, the computational cost of a TDVP update is dominated by the *size* of these objects, which in turn is controlled by the number of product replicas generated by the preceding frontier-based ITP stage.

As discussed in Sec. 4.6 and in the mean-field analysis of Appendix A.2, after k steps of frontier imaginary-time propagation the ansatz contains

$$D(k) + 1 = \left|\mathcal{S}^{(k)}\right|$$

distinct product states. The total number of variational parameters at that depth is

$$N_{\mathrm{params}}(k) \;\sim\; \big(D(k)+1\big)(L+1),$$

since for each product replica we store one amplitude a_λ and L Bloch-disk angles $\{\theta_i^\lambda\}_{i=0}^{L-1}$. The TDVP metric $S_{\bar{w}}$ and force vector $F_{\bar{w}}$ live in this parameter space: $S_{\bar{w}}$ is an $N_{\mathrm{params}}(k) \times N_{\mathrm{params}}(k)$ matrix and $F_{\bar{w}}$ is a vector of length $N_{\mathrm{params}}(k)$. In other words, their dimension grows linearly with $D(k)$ and L, while any explicit representation of $S_{\bar{w}}$ scales *quadratically* in $N_{\mathrm{params}}(k)$.

The mean-field frontier model of Appendix A.2 shows that $D(k)$ itself exhibits a very unfavourable growth pattern. For fixed problem size (L, M), the number of distinct product states displays a short but violent proliferation regime in imaginary-time depth: during the first few ITP steps the effective branching factor is larger than one, and $D(k)$ increases almost exponentially with k before saturating to a plateau value $D_\star$ at a saturation depth $k_\star$.

Combining these observations, the scaling of the TDVP objects is particularly constraining. At depth k we have

$$N_{\mathrm{params}}(k) \sim D(k)\,(L+1),$$

so a dense representation of the metric would require storing

$$\mathcal{O}\big(N_{\mathrm{params}}(k)^2\big) \sim \mathcal{O}\big(D(k)^2(L+1)^2\big)$$

floating-point entries. Because $D(k)$ grows nearly exponentially with the number of ITP steps before saturation, the memory footprint of an explicit $S_{\bar{w}}$ would blow up even more rapidly. This makes it clear that we *cannot* afford to perform many ITP steps before handing the resulting manifold to TDVP: the exponential burst of $D(k)$ would translate into an even more dramatic explosion in the size of $S_{\bar{w}}$.

The force vector $F_{\bar{w}}$ is cheaper to store, but its evaluation exhibits a similar dependence on $D(k)$. As seen in Sec. 5.3.4, each component $F_j(\bar{w})$ is built from mixed matrix elements

$\langle \partial_{w_j} \Psi | \hat{H} | \Psi \rangle$, which in turn involve double sums over product replicas and clause-local factors. The total cost of assembling $F_{\bar{w}}$ therefore scales at least linearly in $N_{\text{params}}(k)$ and inherits an implicit $\mathcal{O}(D(k)^2)$ dependence through the pairwise overlaps between replicas.

The combination of large N_{params} and rapid growth of $D(k)$ has two distinct impacts on the MINRES-QLP:

- **Memory footprint.** If we form $S_{\bar{w}}$ explicitly as a dense matrix (the S_{matrix} route of Appendix A.5.2), the storage requirement scales as $\mathcal{O}(D{+}DL)^2$. This is only feasible for "small" and "medium" instances, where D is still modest. As soon as the frontier manifold gets close to its saturated size $D_\star(L)$, the required memory becomes prohibitive even for relatively small L, and the explicit-matrix strategy breaks down.

- **Iteration count and wall-clock time.** The matrix-free, operator-based implementation S_{op} described in Appendix A.5 circumvents the memory problem by providing MINRES-QLP only with the action $v \mapsto S_{\bar{w}} v$, without ever materializing a $\mathcal{O}(D{+}DL)^2$ matrix. However, the dimension of parameter space and the ill-conditioning of $S_{\bar{w}}$ are unchanged: the Krylov solver still operates in an N_{params}-dimensional space riddled with almost-null, gauge-like directions. As discussed in Appendix A.5.2, this leads to a dramatic increase in the number of MINRES-QLP iterations needed to reach a given tolerance, and the solver frequently hits the iteration cap in the fully matrix-free regime. Thus, the operator form alleviates the *memory* cost but does not eliminate the *time* cost of the TDVP step.

In summary, the near-exponential proliferation of product states in the frontier ITP (before saturation) severely limits how deep we can push the imaginary-time dynamics *before* applying TDVP. Each additional ITP step in the growth regime increases $D(k)$, which in turn enlarges the TDVP metric and force and either (i) overwhelms memory if $S_{\bar{w}}$ is assembled explicitly, or (ii) forces MINRES-QLP to work in a very high-dimensional, ill-conditioned space when $S_{\bar{w}}$ is used in matrix-free form. In the numerical experiments of the next chapter we therefore restrict ourselves to a small number of ITP steps, well before the saturation plateau of $D(k)$, and use the explicit S_{matrix} formulation whenever possible, resorting to the operator form S_{op} only when memory constraints make it unavoidable.

Chapter 6

Variational Imaginary-Time Dynamics and TDVP Refinement

In this chapter, we test and interpret the projected imaginary-time procedure introduced in Secs. 4.4–4.6, and show how the time-dependent variational principle (TDVP) can subsequently drive the ansatz toward low-energy states within the fixed variational manifold generated by that procedure.

First, we benchmark the variational imaginary-time propagation (ITP) against exact imaginary-time evolution on small instances, where full diagonalization of H is feasible. This provides a ground-truth trajectory against which to validate our frontier-based construction. Second, after the frontier has saturated, we switch from ITP to TDVP and study how it further reshapes amplitudes and local angles inside the resulting manifold.

6.1 Benchmarking Variational ITP on Small Instances

In this section, we benchmark our frontier-based variational imaginary-time propagation on small instances where exact quantum evolution is still accessible. The aim is twofold: to clarify the physical interpretation of frontier ITP as a dynamics that explicitly tracks clause violations and to quantify how faithfully this picture reproduces true imaginary-time evolution.

6.1.1 Setup: frontier ITP and clause-violation dynamics

As introduced in chapter 4.4, our variational ITP evolves a state of the form

$$|\Psi^{(k)}\rangle = \sum_{\lambda \in \mathcal{S}^{(k)}} a_\lambda^{(k)} \, |\rho_\lambda\rangle \tag{6.1}$$

where each $|\rho_\lambda\rangle$ is a *product state* in the Bloch-disk ansatz [Eq. (4.18)], and $\mathcal{S}^{(k)}$ is the pool of distinct product states accumulated up to discrete ITP step k.

Every new $|\rho_\lambda\rangle$ that enters $\mathcal{S}^{(k)}$ during imaginary time is generated by the action of a clause projector O_c [Eq. (4.25)] on some parent product state. Consequently, **all newly generated $|\rho_\lambda\rangle$ are clause-violating configurations.** Each such state encodes the statement "this clause is violated in exactly this way on these three spins," while the remaining spins are left (possibly) in superposition. The frontier pool is, thus, a collection of partial or total *invalid* assignments. No $|\rho_\lambda\rangle$ produced by frontier ITP is, at the moment of its creation, a satisfying assignment for all clauses. The algorithm is not discovering solutions and storing them explicitly; instead, it systematically collects and labels structured patterns of clause violations.

This picture becomes more transparent if we recall that the cost Hamiltonian is

$$H = \sum_c O_c, \tag{6.2}$$

where each O_c is a projector onto the unique falsifying assignment of clause c. For any normalized state $|\psi\rangle$,

$$E(\psi) := \langle\psi|H|\psi\rangle = \sum_c \langle\psi|O_c|\psi\rangle = \sum_c \Pr_\psi[\text{clause } c \text{ is violated}], \tag{6.3}$$

so the energy is exactly the *total violation probability*.

A first-order (Euler) imaginary-time step reads

$$|\psi\rangle \longmapsto |\psi'\rangle \;=\; (\mathbf{1} - \delta\tau\, H)\,|\psi\rangle\,. \tag{6.4}$$

The correction term $-\delta\tau\, H\,|\psi\rangle$ subtracts, clause by clause, the component of $|\psi\rangle$ living *within* each violating subspace $\mathrm{Ran}(O_c)$. After renormalization, configurations that violate fewer clauses are relatively enhanced, and $E(\psi)$ decreases monotonically as imaginary time flows.

Our frontier algorithm essentially *implements this subtraction explicitly* in a compressed basis. Starting from a pool of product states, for each parent $|\rho_\lambda\rangle$ and each clause c we apply O_c and obtain a child product state that pins exactly the falsifying triple for c (and leaves the other spins unchanged). That child is, by definition, an invalid state that certifies a particular violation. New children are inserted into the pool $\mathcal{S}^{(k)}$ only if they are distinct, and their contribution to the wavefunction enters with the appropriate signed amplitude inherited from the discrete update.

The crucial point is that the *superposition* Eq.(6.1) can nevertheless drift toward the satisfying sector of Hilbert space. This is enabled by the coefficients $a_\lambda^{(k)}$, which can interfere destructively on clause-violating subspaces and thereby suppress their weight and lower the energy. In concise form:

> **frontier ITP builds a clause-structured manifold of violating configurations, and imaginary time rebalances amplitudes.**

Individual basis states remain "bad", but their coherent combination becomes increasingly concentrated on satisfying assignments.

To test how accurately this clause-violation picture reproduces true imaginary-time dynamics, we now turn to a fully controlled setting where exact evolution is available. For small problem sizes L (so that 2^L is still tractable), we can compute the "true" imaginary-time state

$$|\Psi_{\mathrm{exact}}(\tau)\rangle \;=\; \frac{e^{-\tau H}\,|\Psi_0\rangle}{\|e^{-\tau H}\,|\Psi_0\rangle\,\|}, \tag{6.5}$$

by diagonalizing H and applying the propagator $e^{-\tau H}$ directly in the full 2^L-dimensional Hilbert space. This provides a gold-standard imaginary-time trajectory $|\Psi_{\mathrm{exact}}(\tau)\rangle$ against which we benchmark the frontier-based variational state $|\Psi^{(k)}\rangle$.

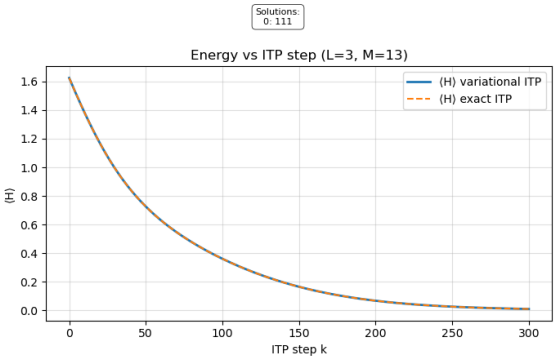
6.1.2 Validation observables and numerical comparison

The numerical data shown in Fig. 6.1 ,6.2 and in Fig. 6.3 correspond to a representative satisfiable instance with $L = 3$, $M = 13$ clauses and 111 as a satisfying assignment for the first and $100,011$ for which both the exact and the variational trajectories can be computed up to $k = 300$ ITP steps. In these simulations we fix the imaginary-time step to $\tau = 0.01$, which is of the order of $1/M$ for $M = 13$. This choice is consistent with the general scaling $d\tau, d\gamma \propto 1/M$, motivated by the extensive nature of the cost function. Since the Hamiltonian has the form $H = \sum_{c=1}^{M} \Pi_c$ keeping the parameter update in a regime where the Euler discretization remains reliable requires $d\tau = \mathcal{O}(1/M)$. We adopt the same criterion for the TDVP $d\gamma$ below. This scaling will be used consistently throughout the thesis.

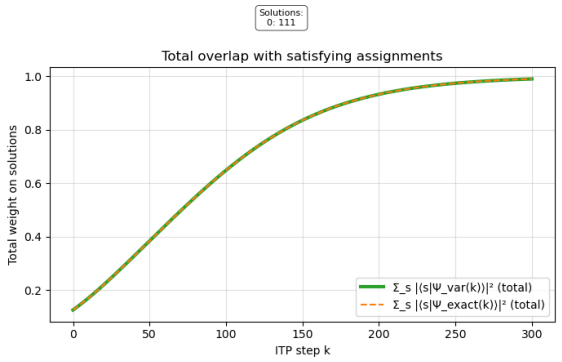
1. Energy decay. We first track the mean clause-violation energy

$$E(k) = \langle\Psi(k)|H|\Psi(k)\rangle$$

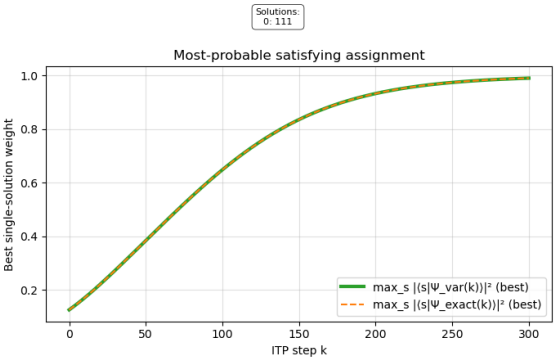
for both the exact imaginary-time state and the frontier-based variational state at each discrete ITP step k. In Fig. 6.1a and 6.2a the solid blue curve (frontier ITP) and the dashed orange curve (exact ITP) lie almost on top of each other over the entire evolution. Both exhibit



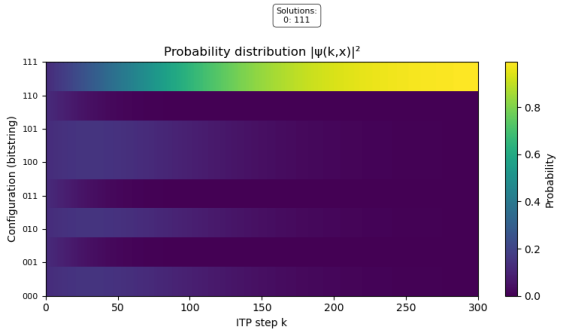
(a) Mean clause-violation energy $\langle H \rangle$ as a function of ITP step k for a small satisfiable instance with $L = 3$ and $M = 13$. Solid blue: frontier-based variational ITP. Dashed orange: exact ITP obtained by evolving in the full 2^L Hilbert space.



(b) Total overlap $P_{\text{tot}}(k) = \sum_\alpha |\langle S_\alpha | \Psi(k) \rangle|^2$ with all satisfying assignments as a function of ITP step k. Solid green: variational frontier ITP. Dashed orange: exact ITP.



(c) Best-assignment weight $P_{\text{best}}(k) = \max_\alpha |\langle S_\alpha | \Psi(k) \rangle|^2$ as a function of ITP step k. The frontier ITP (solid green) tracks the exact ITP (dashed orange) very closely.



(d) Heat map of $p_k(b) = |\langle b | \Psi(k) \rangle|^2$ across imaginary-time steps k. Each row corresponds to a classical bitstring b, each column to an ITP step. Bright bands indicate configurations that accumulate probability mass over time.

Figure 6.1: Diagnostics of imaginary-time dynamics, comparing exact ITP and frontier-based variational ITP, shown in a 2×2 panel layout for a representative small instance with $L = 3$, $M = 13$ clauses and 111 as a satisfying assignment.

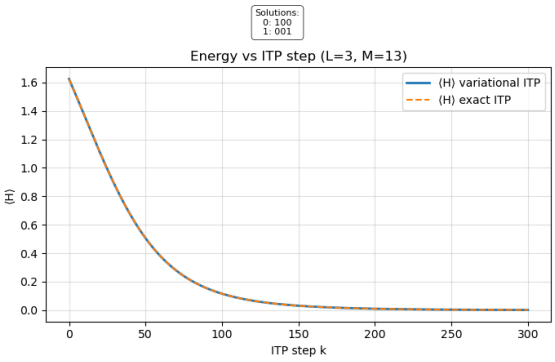
a smooth exponential-like decay of $\langle H \rangle$ towards zero, reflecting the suppression of clause-violating configurations. Any discrepancy between the two curves is at the level of the line thickness, indicating that the compressed frontier manifold already reproduces the energy of exact imaginary-time evolution with very high fidelity.

2. Overlap with satisfying assignments. Let $\{|S_\alpha\rangle\}$ denote all satisfying assignments for the instance. We monitor the total and best overlaps

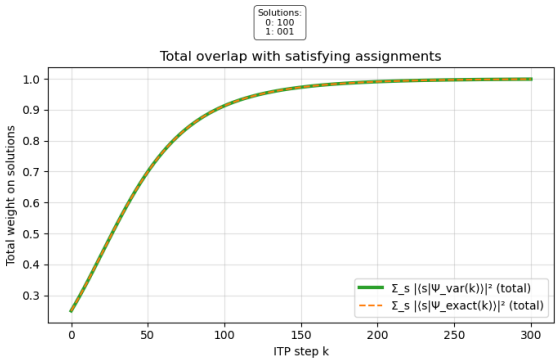
$$P_{\text{tot}}(k) = \sum_\alpha \left| \langle S_\alpha | \Psi(k) \rangle \right|^2, \tag{6.6}$$

$$P_{\text{best}}(k) = \max_\alpha \left| \langle S_\alpha | \Psi(k) \rangle \right|^2, \tag{6.7}$$

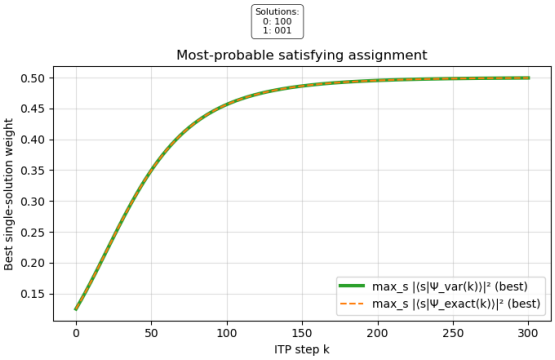
In Fig. 6.1b and 6.2c we plot $P_{\text{tot}}(k)$. Starting from a small initial weight ~ 0.1 on the solution manifold, both curves grow smoothly and monotonically, approaching $P_{\text{tot}} \simeq 1$ by $k \simeq 300$. Thus, at late times essentially all probability mass resides on satisfying assignments. The exact (dashed orange) and variational (solid green) curves are visually indistinguishable, showing that frontier ITP drives the wavefunction into the same solution subspace as exact ITP.



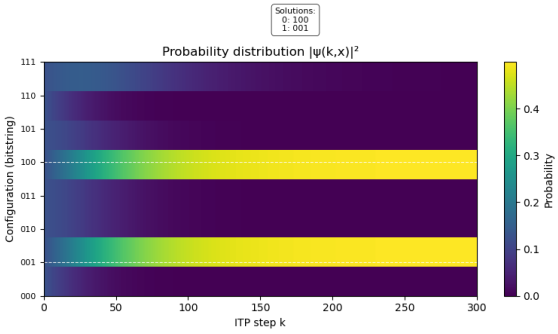
(a) Mean clause-violation energy $\langle H \rangle$ as a function of ITP step k for a second satisfiable instance with $L = 3$ and $M = 13$, now with two satisfying assignments $|001\rangle$ and $|100\rangle$. Solid blue: frontier-based variational ITP. Dashed orange: exact ITP.



(b) Total overlap $P_{\text{tot}}(k) = \sum_\alpha |\langle S_\alpha|\Psi(k)\rangle|^2$ with all satisfying assignments for the second instance.



(c) Best-assignment weight $P_{\text{best}}(k) = \max_\alpha |\langle S_\alpha|\Psi(k)\rangle|^2$ for the second instance. Frontier ITP (solid green) again tracks exact ITP (dashed orange) within line thickness.



(d) Heat map of $p_k(b) = |\langle b|\Psi(k)\rangle|^2$ for the second instance. The two satisfying bitstrings $|001\rangle$ and $|100\rangle$ (dashed lines) progressively brighten and end up carrying almost all probability, while clause-violating configurations are exponentially suppressed by imaginary-time evolution.

Figure 6.2: Same diagnostics as in Fig. 6.1, now for a second small satisfiable instance with $L = 3$, $M = 13$ and two satisfying assignments $|001\rangle$ and $|100\rangle$. Frontier ITP and exact ITP remain in excellent agreement for all observables.

Figure 6.1c and 6.2c shows the behaviour of $P_{\text{best}}(k)$, the weight of the single most probable satisfying assignment. This quantity also increases monotonically in both evolutions: after a roughly linear rise at early times, the most-probable solution gradually takes over and ends up with probability very close to 1. The frontier ITP again tracks the exact imaginary-time evolution step by step, demonstrating that not only the total weight on solutions but also the shape of the peak in configuration space is accurately reproduced.

3. Computational-basis redistribution. To obtain a more microscopic view of the dynamics, we inspect the evolution of the full probability distribution over classical bitstrings. At each step k we consider

$$p_k(b) = |\langle b|\Psi(k)\rangle|^2, \qquad b \in \{0,1\}^L,$$

and visualize it as a heat map with bitstrings b on the y-axis and imaginary-time steps k on the x-axis (Fig. 6.1d,6.2d). Dark regions correspond to configurations with negligible weight, while bright regions indicate bitstrings that accumulate probability mass.

For the first instance, one satisfying assignment quickly emerges as the dominant configuration (the bright band near the top of the panel), with probability saturating close to unity by the end

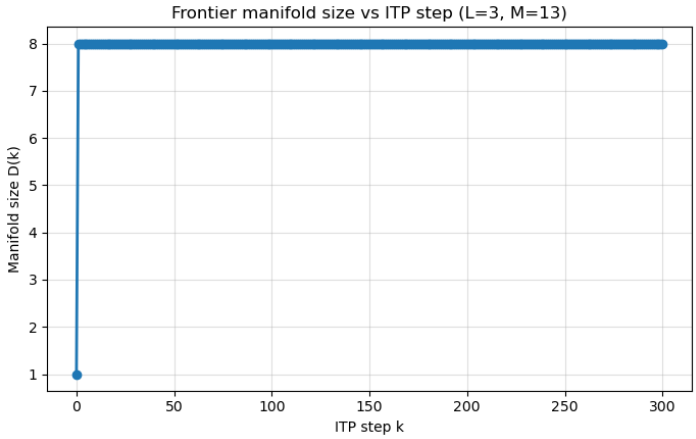


Figure 6.3: Size of the frontier manifold $D(k) = |\mathcal{S}^{(k)}|$ as a function of ITP step k .

of the evolution. While for the second one the value saturates close to $1/2$, reflecting the nearly equal sharing of probability between the two satisfying assignments. Configurations that violate more clauses are progressively darkened as imaginary time flows, illustrating the interpretation of imaginary-time propagation as systematic suppression of clause-violating patterns. For the single-solution instance, this redistribution can also be viewed directly in variational parameter space: Fig. 6.4 shows that the modulus of the frontier amplitudes $|a_\lambda(k)|$ evolves from a strongly polarized initial configuration towards a pattern in which all non-solution replicas settle to a small, nearly uniform weight while the replica aligned with the solution remains dominant but slowly decreases, consistently with the flow of probability mass in configuration space.

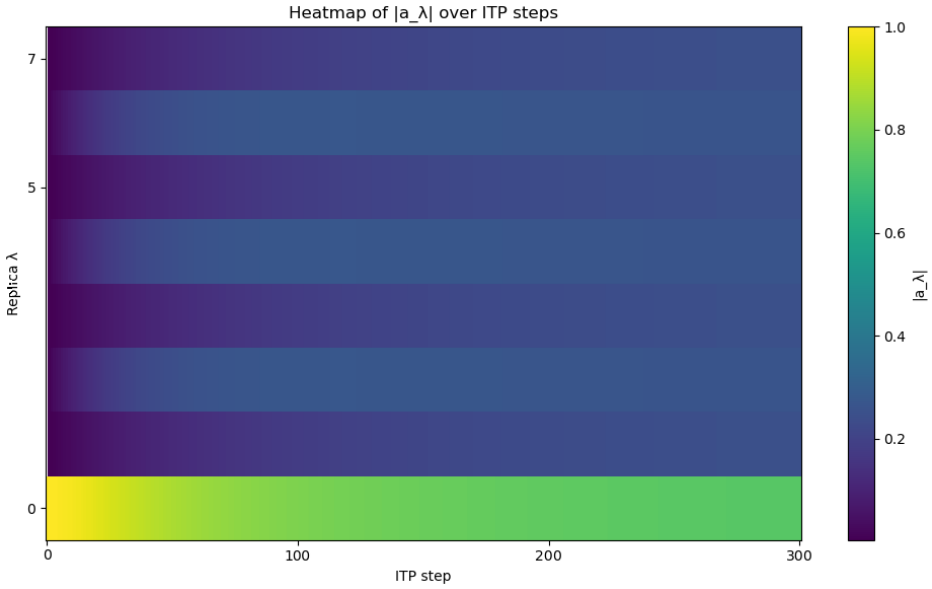


Figure 6.4: Heat map of the absolute value of the variational amplitudes $|a_\lambda(k)|$ across imaginary-time steps k for the single-solution instance of Fig. 6.1. Each row corresponds to a frontier replica λ, while columns are ITP steps. The plot highlights how the modulus of the amplitudes is gradually reshaped during the evolution, with non-solution replicas acquiring a small but nonzero stationary weight and the solution-aligned replica remaining the most prominent component of the frontier manifold.

4. Growth and saturation of the frontier manifold. An additional diagnostic, specific to the variational algorithm, is the size of the frontier manifold itself,

$$D(k) = \left|\mathcal{S}^{(k)}\right|,$$

i.e. the number of distinct product states accumulated up to step k. This is shown in Fig. 6.3. Starting from a single product state at $k = 0$, the frontier update rule rapidly generates new clause-violating configurations, and $D(k)$ jumps to $D(1) = 8$ after the first ITP step. For $L = 3$ this already equals the full Hilbert-space dimension 2^L, so the manifold saturates immediately and remains of constant size thereafter. Obviously, this is true for both instances.

This small-L behavior concretely illustrates the general picture developed in 4.4 frontier ITP quickly builds a clause-structured basis of product states, and all subsequent imaginary-time evolution consists of reweighting these basis states via their amplitudes $a_\lambda^{(k)}$. The fact that the energy and overlaps continue to evolve significantly even after $D(k)$ has saturated shows that most of the "work" of imaginary-time cooling is performed by amplitude rearrangement on a fixed manifold rather than by continued growth of the basis.

The excellent agreement observed on small instances might suggest that one could simply continue frontier ITP for as many steps as needed. This is not the case. For larger L, exact evolution in the full 2^L Hilbert space is impossible, and even the frontier manifold itself can grow very large before saturation (Sec. 4.6 and App. A.2). In particular, the number $D(k)$ of distinct product configurations generated by the branching process increases rapidly with both k and L, so that the variational parameter count $N_{\text{params}}(k) \sim (D(k)+1)(L+1)$ becomes prohibitive well before exact convergence in imaginary time would be reached. As we will discuss, it could also exceed 2^L, making our variational description ineffective.

The limitation is not expressive power, but the cost of explicitly storing and updating all $O(D(k)L)$ variational parameters when $D(k)$ itself scales poorly with system size.

This is precisely where TDVP enters. In the TDVP stage, we allow the same product states $|\rho_\lambda\rangle$ to *deform* continuously, together with their amplitudes a_λ: the update changes both Bloch angles and amplitudes in a way that attempts to heal the pinned violations. The role of frontier ITP is, therefore, to build a structured, clause-aware manifold of product states, while TDVP moves within that manifold and bends individual configurations toward satisfiability instead of relying purely on interference between fixed violating patterns.

6.2 Benchmarking TDVP on Small Instances

Throughout this section, we study two planted 3-SAT instances with $L = 5$, $M = 22$, and $\alpha = M/L \approx 4.3$, right at the SAT/UNSAT transition window. The first instance has three satisfying assignments: `00111,00101, and 10101`, while the second instance is uniquely satisfiable with the planted solution `11101`. For both instances, the frontier ITP stage after a single step produces an effective variational manifold of size $D(1) = 23$ product states (replicas). We then *freeze* this label set and switch to TDVP, evolving the internal parameters of the ansatz according to the projected imaginary-time flow

$$S_{\bar{w}}\,\delta\bar{w} = -\gamma\,F_{\bar{w}}, \tag{6.8}$$

see Sec. 5.

Numerically, we will see that: (i) the energy $\langle H\rangle$ continues to decrease under TDVP and approaches zero; (ii) the wavefunction weight concentrates on replicas that align with the planted solution; and (iii) the Krylov solver MINRES–QLP used to solve Eq. (6.8) remains stable (residual $\lesssim 10^{-10}$) even though the metric $S_{\bar{w}}$ is nearly singular, cf. Sec. 5.4.

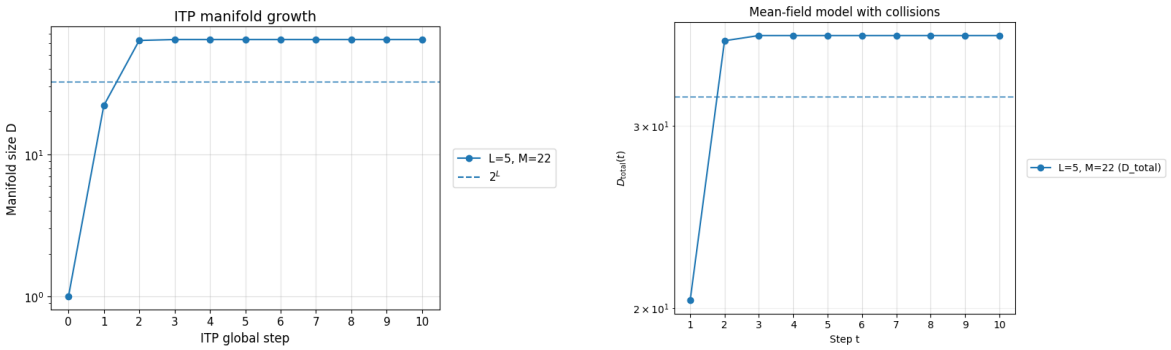


Figure 6.5: Growth of the ITP manifold. Left: number $D(t)$ of distinct pinned product states generated by ITP for the $L = 5$ reference instance. Right: mean-field prediction $D_{\text{total}}(t)$. The horizontal dashed line marks 2^L.

Manifold growth. For the size of our reference instance, we could run the full frontier ITP procedure up to saturation. Figure 6.5 compares three manifolds: (a) the frontier growth measured directly for this instance, obtained by running $k_{\max} = 10$ steps of frontier ITP;(b) the mean-field model. In panel (a) the total number of distinct replicas $D_{\text{total}}(k)$ increases from 1 to about 20 after the first frontier step and then rapidly saturates around $D_{\text{total}} \approx 54$, already larger than the Hilbert-space dimension $2^L = 32$. The mean-field curve in panel (b) reproduces the same qualitative behaviour. This is the reason why, in this case, we chose to perform only a single step of ITP. This choice is not motivated by computational considerations (these will become relevant in the next chapter, when the system size increases), but rather by the fact that additional steps would invalidate the effectiveness of our variational description.

Numerical stabilizers. We regularize and solve this linear system using MINRES–QLP, as described in Sec. 5.4. In particular, we use:

- a diagonal regularization $S_{\bar{w}} \to S_{\bar{w}} + \lambda_{\text{reg}} \mathbb{1}$ with $\lambda_{\text{reg}} \sim 10^{-8}$,
- MINRES–QLP tolerance $\texttt{rtol} = 10^{-10}$ and cap on Krylov steps $\texttt{maxit} = 10^4$.

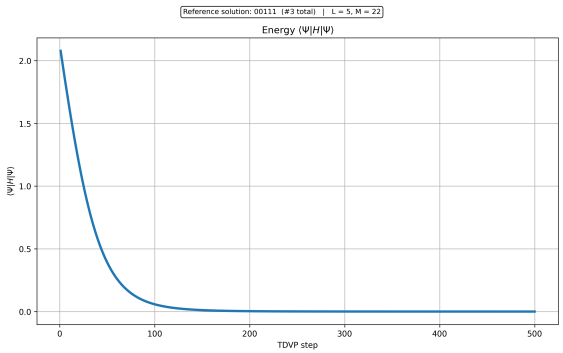
These choices keep the dynamics numerically stable without visibly affecting the physical evolution.

6.2.1 Global TDVP behaviour on hard instances

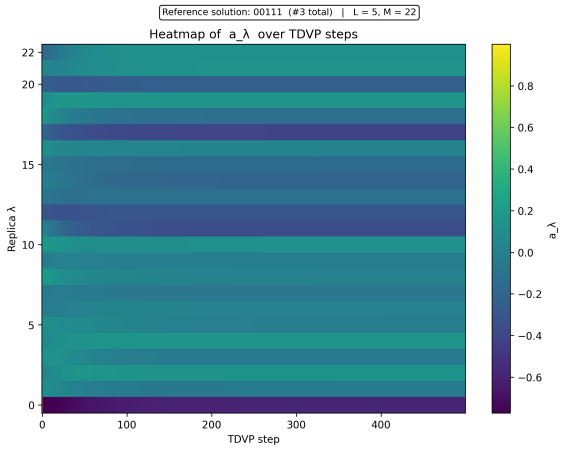
Figures 6.6 and 6.7 collect four complementary observables for the multi-solution and single-solution instances, respectively. The top-left panels show the imaginary-time energy $\langle\Psi|H|\Psi\rangle$, while the other three panels probe the internal structure of the replica manifold: the coefficients a_λ, the overlap $|\langle\rho_\lambda|\text{solution}\rangle|$ with the chosen reference solution, and the replica–replica overlaps $|\langle\rho_\lambda|\rho_\mu\rangle|$ at the end of the run.

For the three-solution instance (Fig. 6.6a), the energy drops quickly and essentially reaches zero within a few hundred TDVP steps, signaling that the evolving state has reached the solutions. The single-solution case (Fig. 6.7a) shows a similar trend: $\langle H\rangle$ decays smoothly and approaches zero on a slightly longer time scale. We attribute this slower convergence to the presence of only one solution, which makes it harder for the state to find the path to zero energy. In both instances, TDVP, therefore, does not merely lower the energy but actually drives some states towards configurations that satisfy almost all clauses.

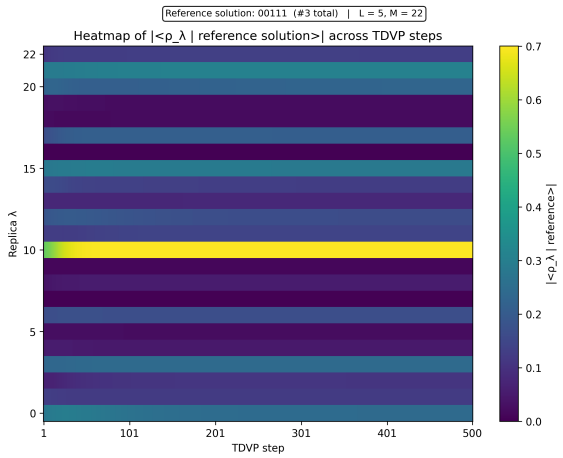
Figs. 6.6c and 6.7c reveal how this energy reduction is achieved. The rotations of the internal Bloch angles θ_i^λ, expressed via the overlap with the most probable solution, show that only a handful of replicas develop a large overlap with the reference satisfying assignment. This



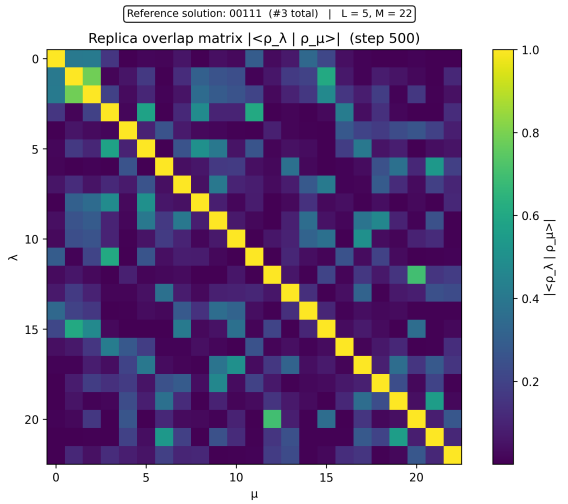
(a) Energy $\langle H \rangle$.



(b) Magnitude $|a_\lambda|$.



(c) Overlap with reference solution.



(d) Replica–replica overlaps at step 500.

Figure 6.6: TDVP refinement on the planted instance with three solutions; we monitor the reference bitstring `00111`. The four panels show the energy, the flow of amplitudes $|a_\lambda|$, the overlap of each replica with the reference solution, and the replica–replica overlap matrix at the end of the run.

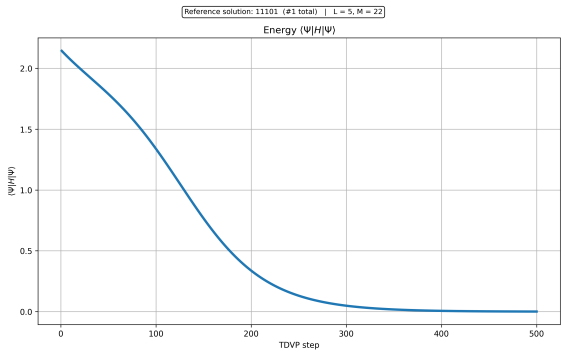
demonstrates that TDVP both *repairs* the internal configuration of a few promising replicas and *amplifies* them in the global superposition Figs. 6.6b and 6.7b .

The final replica–replica overlap matrices in Figs. 6.6d and 6.7d complete this picture. Apart from the trivial diagonal, most off-diagonal overlaps remain well below 1: even after TDVP has concentrated weight on a small subset of replicas, these survivors are not simply copies of one another. Instead, the algorithm ends with a tiny cluster of distinct product states, all aligned with the solution manifold but still exploring slightly different directions in configuration space.

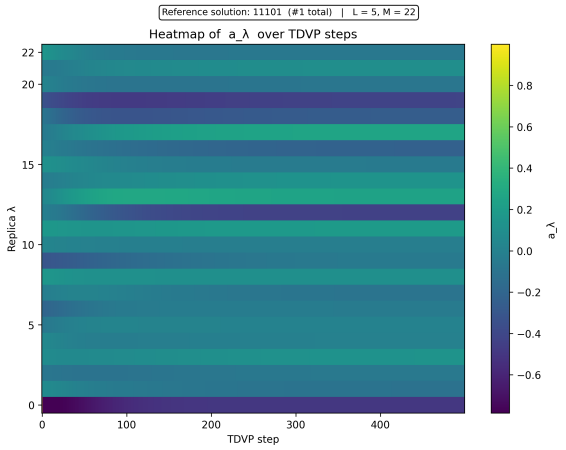
6.2.2 Target probabilities and Krylov diagnostics

The global observables above show how TDVP reshapes the replica manifold. To assess actual solution reconstruction and numerical stability, we now inspect the diagnostic plots in Figs. 6.8 and 6.9.

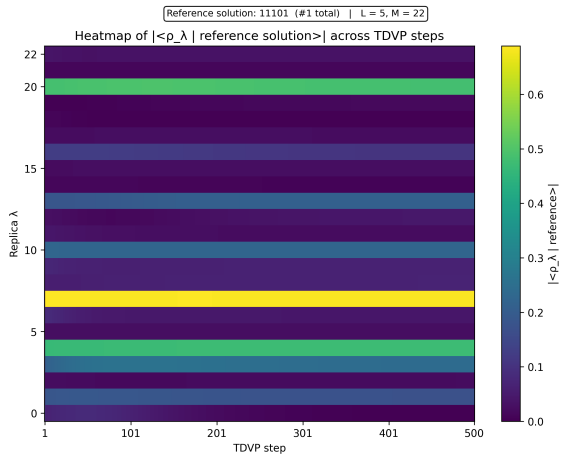
For the three-solution instance (Fig. 6.8, top panel) the probability weight on the most probable bitstring `00111` increases monotonically and saturates at a value of order 0.6, while the *total* probability carried by all satisfying assignments approaches 1. This is consistent



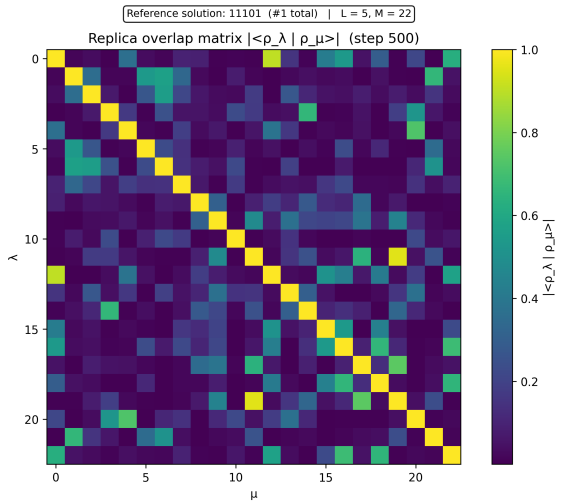
(a) Energy $\langle H \rangle$.



(b) a_λ vs steps.



(c) Overlap with reference solution.



(d) Replica–replica overlaps at step 500.

Figure 6.7: Same diagnostics as in Fig. 6.6, now for the planted instance with a unique satisfying assignment `11101`.

with the multi-solution structure of the instance: TDVP identifies several exact solutions and distributes weight among them but not equally. In the unique-solution case (Fig. 6.9, top panel) the behaviour is even sharper: the probability of `11101` rises smoothly and eventually coincides with the total solution weight, which also saturates close to 1. Here TDVP has no alternative satisfying assignment to populate, and the algorithm effectively collapses onto the planted bitstring.

The middle and bottom panels of Figs. 6.8 and 6.9 report the performance of the MINRES–QLP solver used to implement the TDVP update $S_{\bar{w}}\delta\bar{w} = -\gamma F_{\bar{w}}$ (Eq. 6.8). In both instances there is an initial spike in the residual and in the iteration count, reflecting the fact that the TDVP metric $S_{\bar{w}}$ is most ill-conditioned near the starting point of the refinement. After this transient the residual stabilizes well below 10^{-10} and the iteration number settles to $\mathcal{O}(10^2)$ per TDVP step. The detailed traces differ slightly between the two instances, but the overall numerical cost and stability are comparable: the regularization scheme of Sec. 5.4 is therefore sufficient to keep the Krylov solver well behaved even in the glassy regime.

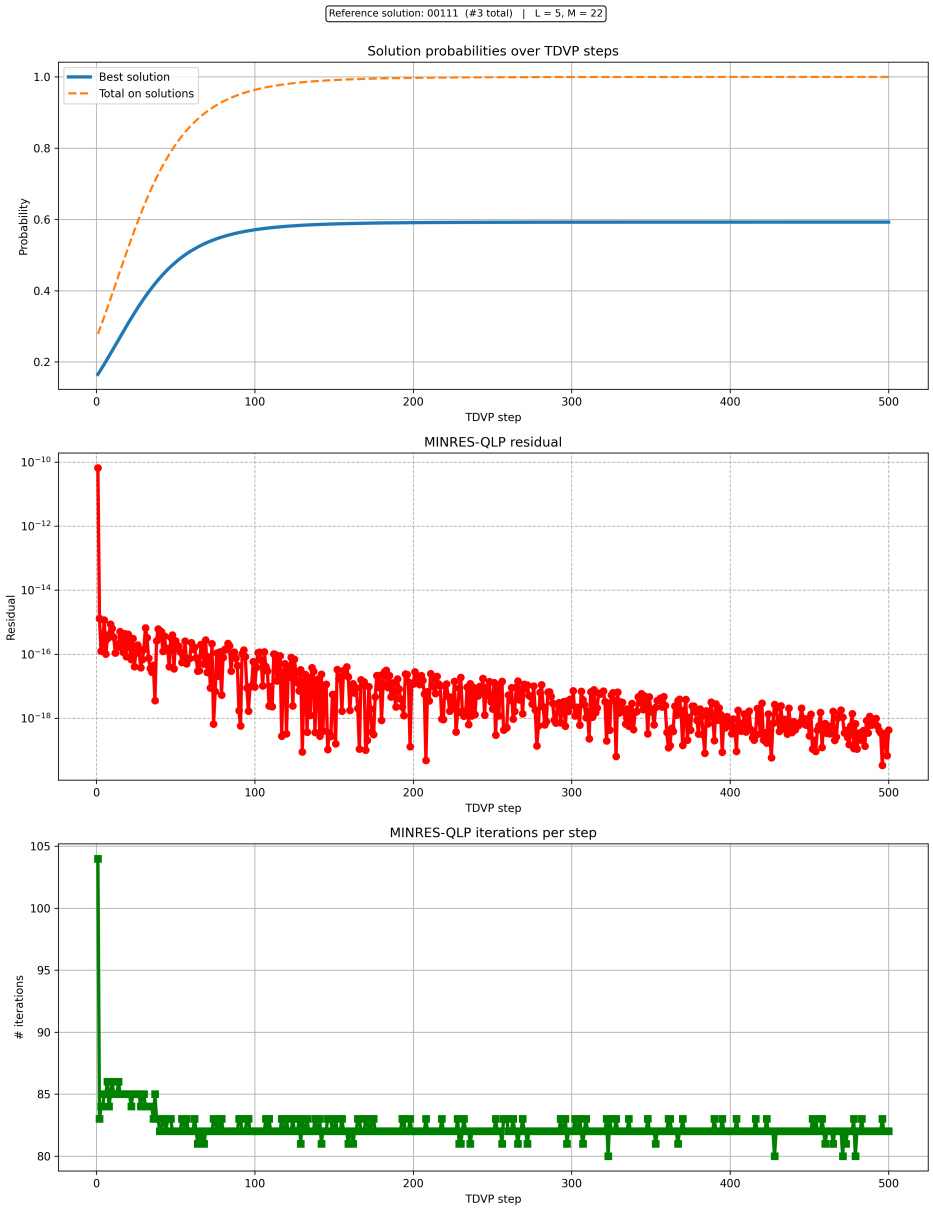


Figure 6.8: Target probability and MINRES–QLP diagnostics for the three-solution instance with reference bitstring `00111`. The top panel shows the probability of the reference solution and of the full solution manifold; the middle and bottom panels show the residual and iteration count of MINRES–QLP at each TDVP step.

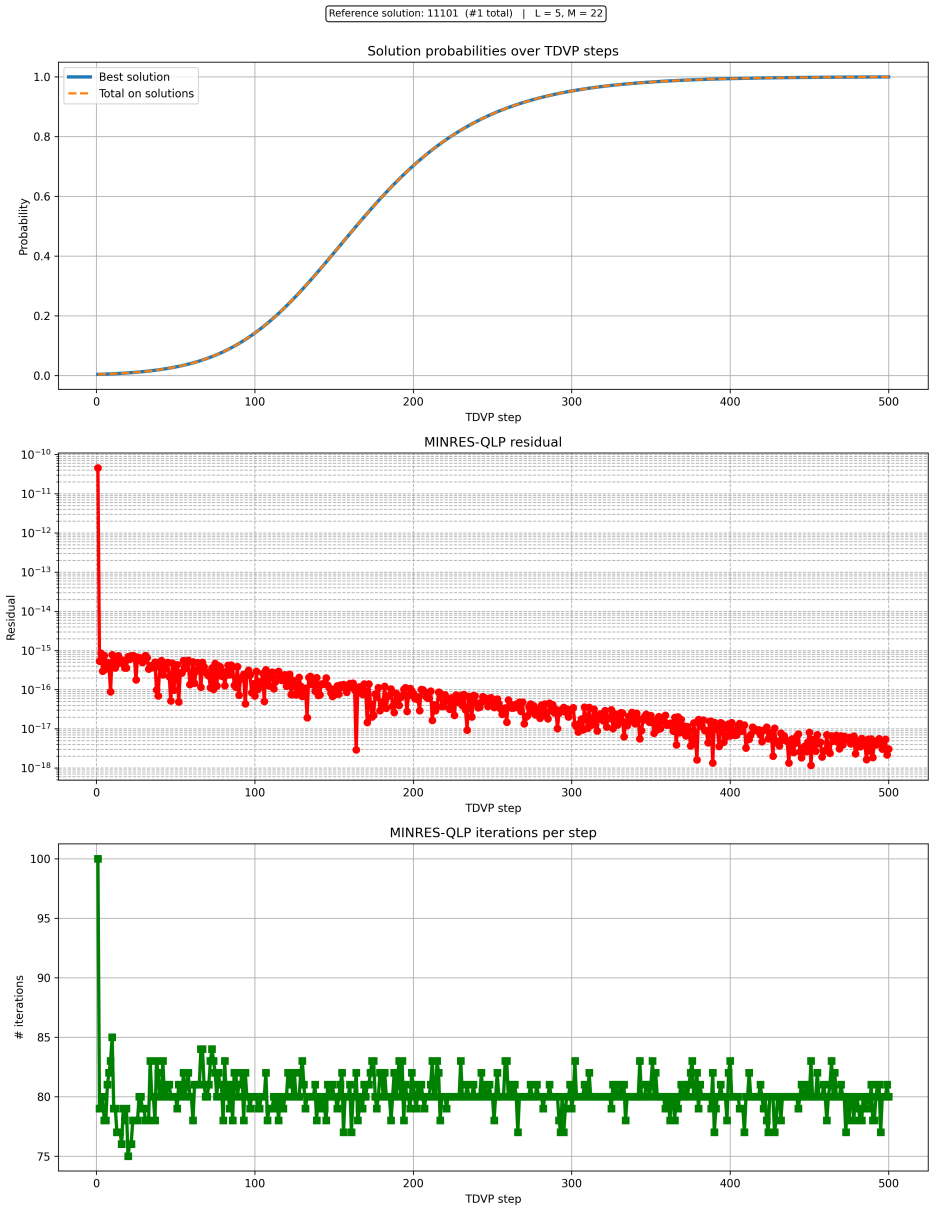


Figure 6.9: Same diagnostics as in Fig. 6.8, now for the instance with a unique solution `11101`.

Chapter 7

Sensitivity-driven manifold compression

In this chapter we move from small-scale benchmarks to the question of how far our algorithm can be pushed into the hard-SAT regime as the number of spins increases, and what ultimately limits this scaling. Our starting point is an ITP–TDVP scheme that already solves a considerable fraction of random 3-SAT instances, including many with $L = 50$, using only a small number of replicas. Sensitivity-driven compression is then introduced as a refinement: its goal is to speed up convergence and to recover (some of) the relatively small subset of hard-regime instances that remain unsolved within 1000 TDVP steps.

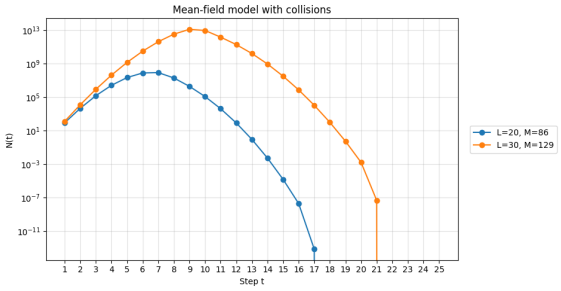
The first part of the chapter introduces a sensitivity-based active-set strategy for TDVP. We define an energy-sensitivity score for each replica and use it, at every step, to select a small subset of "active" replicas on which the TDVP update is explicitly performed. This reduces the effective size and density of the metric S_{eff}, keeps the associated linear system tractable, and allows us to retain good convergence properties even when the full manifold contains many more replicas. We benchmark this scheme on single instances and on finite-size statistics at critical clause density, and we show that local metric rescaling (LMR) is a crucial ingredient for stabilising the dynamics. We also analyse cases where the TDVP energy fails to fully converge, linking metastable plateaux and slow freezing windows to features of the underlying glassy landscape, and we briefly explore the trade-off between accuracy and runtime when the TDVP selection fraction is increased.

The second part (Sec. 7.2) addresses the state-space wall posed by multi-step ITP frontiers. Naively extending the frontier in imaginary time leads to an almost exponential growth of the manifold dimension, which quickly makes both memory usage and TDVP updates impractical. To partially overcome this, we reuse the same sensitivity score as a *compression* tool: short blocks of frontier growth are interleaved with sensitivity-based pruning and a simple diversity filter, producing compressed frontiers that are much smaller than the raw multi-step manifold but more structured and informative than a single-step frontier. We compare "soft" and "aggressive" compression schemes on hard instances at $L = 30$, $L = 40$, and $L = 50$, showing that aggressive compression can recover instances that the baseline one-step pipeline fails to solve, while keeping the final manifold size of order $M+1$ and a small TDVP active set. At the same time, we identify a subset of especially difficult $L = 50$ formulas in the hard-SAT regime for which all pipelines (uncompressed, softly compressed, and aggressively compressed) stall within our time budget: in these cases, the combination of glassy landscape complexity and computational cost prevents the variational manifold from escaping deep metastable basin.

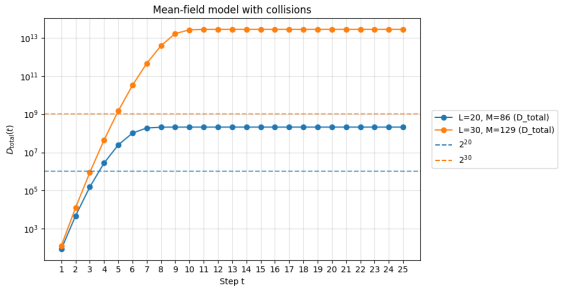
7.1 Sensitivity-based active replica selection during TDVP

As discussed in Sec. 4.6 and Appendix A.2, the frontier-based ITP dynamics can generate a manifold whose size $D(k)$ grows very rapidly with both the ITP step k and the system size L. Before saturation, the variational parameter count $N_{\text{params}}(k) \sim (D(k) + 1)(L + 1)$ becomes prohibitive well before exact imaginary-time convergence is reached. This is the "cost wall" that prevents us from running many ITP steps on large, hard instances.

The severity of this cost wall is illustrated in Fig. 7.1, which shows the mean-field upper bounds (including clone collisions) for the frontier proliferation at $L = 20$ and $L = 30$ obtained



(a) Mean-field estimate of the number of new frontier states $N(t)$ for $L = 20$ and $L = 30$ (log scale).



(b) Mean-field cumulative number of distinct product states $D_{\mathrm{tot}}(t)$ for the same L, with dashed lines indicating 2^{20} and 2^{30}.

Figure 7.1: Mean-field upper bounds on frontier proliferation for larger system sizes, based on the layer–resolved model of Appendix A.2. For $L = 20$ and $L = 30$ the predicted saturated manifold sizes $D_\star(L)$ already lie many orders of magnitude above the Hilbert-space dimensions 2^L, illustrating the "cost wall" associated with the scaling of D.

from the model of Appendix A.2. Panel (a) displays the expected number of new frontier states $N(t)$ per ITP step, while panel (b) shows the corresponding cumulative number of distinct product states $D_{\mathrm{tot}}(t)$ compared with the Hilbert-space sizes 2^{20} and 2^{30}. Even for these moderate system sizes, $D_{\mathrm{tot}}(t)$ grows by many orders of magnitude in just a few steps and quickly saturates to values that vastly exceed 2^L.

For TDVP refinement, we therefore deliberately stay in the regime where only *one* ITP–frontier step is taken on the planted instance. In our implementation, this implies that at the TDVP starting point, the number of replicas D scales at most linearly with the number of clauses (roughly $D \sim M + 1$), rather than exploding combinatorially. Nevertheless, even after a single ITP–frontier step D can still be a few hundred for challenging instances at $L \sim 20$. The TDVP parameter vector then has $\sim D(1+L)$ entries, and the associated metric remains large. Building and solving (iteratively) the *full* TDVP metric at each step is still the dominant cost, and the metric can be ill-conditioned enough to slow down MINRES–QLP despite the regularization strategies of Sec. 5.4.

A natural idea would be to diagonalize the metric and drop the directions associated with its smallest eigenvalues. However, this is dangerous for our manifold: each replica can correspond to a *different* glassy basin in configuration space, and by the Overlap Gap Property, these basins can be well separated in Hamming distance (cf. Chapter 3). Permanently discarding a replica risks throwing away our only foothold in an entire basin. What we want is to *downweight* replicas that are unimportant at the *current* TDVP step without erasing them from the manifold.

In this section, we introduce a first strategy to partially circumvent this cost wall while preserving the full manifold generated after the first ITP–frontier step. The central idea is to adaptively select, at each TDVP step, only the replicas that have the largest instantaneous "energetic leverage" on the cost function and to solve the TDVP linear system *only* in the subspace spanned by those active replicas. All replicas remain available and can become active again in later steps.

7.1.1 Replica energy sensitivity

We now define a scalar sensitivity score that quantifies, for each replica, how much it is currently "holding onto" energy.

Consider the instantaneous (unnormalized) energy expectation

$$\mathcal{E} = \langle \Psi | \, H \, | \Psi \rangle = \sum_{\lambda,\mu=1}^{D} a_\lambda a_\mu \, H_{\lambda\mu}, \tag{7.1}$$

where $H_{\lambda\mu} = \langle\rho_\lambda| H |\rho_\mu\rangle$ denote the projected Hamiltonian matrix elements.

Viewing $\mathcal{E}$ as a quadratic form in the amplitudes $\{a_\lambda\}$, we first differentiate with respect to a_λ:

$$\frac{\partial \mathcal{E}}{\partial a_\lambda} = 2 \sum_{\mu=1}^{D} H_{\lambda\mu}\, a_\mu. \tag{7.2}$$

We then define, for each replica, the scalar *energy sensitivity*

$$s_\lambda := a_\lambda \sum_{\mu=1}^{D} H_{\lambda\mu}\, a_\mu = \frac{1}{2}\, a_\lambda\, \frac{\partial \mathcal{E}}{\partial a_\lambda}. \tag{7.3}$$

By construction, s_λ measures the contribution associated with the *single* variational parameter a_λ; it is therefore a replica-resolved (parameter-wise) energy sensitivity, not a global measure of the overall sensitivity of $\mathcal{E}$ with respect to all parameters.

Interpretation:

- $|s_\lambda|$ measures how much replica λ is currently "carrying" the energy, with respect to variations of its own amplitude a_λ while keeping all other amplitudes fixed. If $|s_\lambda|$ is large, a change in a_λ can strongly affect $\mathcal{E}$; replica λ has high energetic leverage.

- If $|s_\lambda|$ is tiny, tweaking replica λ barely moves the energy at the present step; that replica is, momentarily, a spectator.

From the implementation point of view, we never form the dense $D \times D$ matrix $H_{\lambda\mu}$ explicitly. Instead, for each pair (λ, μ) we compute their product-state overlap and their clause-resolved contributions analytically:

- The overlap factor is

$$\langle\rho_\lambda|\rho_\mu\rangle = \prod_{i=1}^{L} \cos(\theta_i^{(\lambda)} - \theta_i^{(\mu)}),$$

  obtained from precomputed $\cos\theta_i^{(\lambda)}$ and $\sin\theta_i^{(\lambda)}$.

- Since $H = \sum_c \Pi_c$ is a sum of 3-local clause projectors, each matrix element $H_{\lambda\mu}$ factorizes into this global overlap times a clause term that depends only on the three spins involved by clause c and on whether each literal is negated or not.

In practice, we loop over replicas λ and μ, reuse cached trigonometric values, and accumulate directly the combinations $a_\lambda \sum_\mu H_{\lambda\mu} a_\mu$ appearing in Eq. (7.3). This yields all $\{s_\lambda\}$ in $\mathcal{O}(D^2(L + C))$ floating-point operations, where C is the number of clauses.

For the regime of interest here — one frontier step, with D scaling roughly like M and $D \lesssim \text{few} \times 10^2$ at $L \sim 20$ — this additional cost is modest compared to assembling and applying the full TDVP metric S, whose dimension scales as $\mathcal{O}\big(D^2(L+1)^2\big)$ and whose Krylov solution dominates the runtime. However, the quadratic scaling in D means that if the frontier manifold itself were allowed to grow much larger, sensitivity evaluation would also become a bottleneck; the present strategy is therefore tailored to the "moderately large D" regime unlocked by frontier saturation.

Physically, s_λ aggregates detailed clause information. Since $H = \sum_c \Pi_c$ and each Π_c penalizes a specific falsifying triple, $H_{\lambda\mu}$ encodes how the pair of replicas (λ, μ) jointly violates or satisfies the instance. A large $|s_\lambda|$ signal indicates that replica λ is aligned with energetically expensive directions that TDVP should try to update next. In the following, we will simply refer to the quantity in Eq. (7.3) and to the associated replica-selection procedure as the *sensitivity*.

7.1.2 Active subset, reduced TDVP solve, and scatter-back

We now use the sensitivity scores $\{s_\lambda\}$ to define an active subset of replicas and restrict the TDVP solve to that subset.

At each TDVP step, we proceed as follows:

1. Compute all s_λ from Eq. (7.3).
2. Rank replicas by $|s_\lambda|$ and select the top fraction (for instance, the most sensitive $\sim$20%). Denote this index set by $\mathcal{K}_{\text{active}}$ and its size by $D_{\text{eff}} = |\mathcal{K}_{\text{active}}| \ll D$.
3. Build a *restricted* parameter vector
$$\bar{w}_{\text{eff}} = \big(\{a_\lambda\}_{\lambda\in\mathcal{K}_{\text{active}}}, \{\theta_i^{(\lambda)}\}_{\lambda\in\mathcal{K}_{\text{active}},\, i=1,\dots,L}\big),$$
so that the working dimension is reduced from $\mathcal{O}(DL)$ to $\mathcal{O}(D_{\text{eff}}L)$.
4. Assemble *only on that subset*:
   - the reduced TDVP metric $S_{\bar{w}}^{(\text{eff})}$, i.e. the block of $S_{\bar{w}}$ involving $\bar{w}_{\text{eff}}$,
   - the reduced force vector $F_{\bar{w}}^{(\text{eff})}$.
5. Solve the smaller linear system restricted to the active parameters with MINRES–QLP (plus the usual small diagonal regularization of Sec. 5.4 to tame conditioning). Because $S_{\bar{w}}^{(\text{eff})}$ is much smaller and typically less ill-conditioned than the full $S_{\bar{w}}$, Krylov convergence is drastically faster and cheaper here.
6. Apply the update only on the active replicas:
$$a_\lambda \to a'_\lambda, \qquad \theta_i^{(\lambda)} \to \theta_i^{(\lambda)\prime} \quad \text{for } \lambda \in \mathcal{K}_{\text{active}}.$$
We then scatter these updated parameters back into the full ansatz, keep all other replicas unchanged, renormalize the global amplitudes, and move on to the next imaginary-time step.

Two conceptual remarks:

- **No replica is ever deleted.** Replicas that are inactive at a given step remain part of the manifold. At the next TDVP step, we recompute the scores $\{s_\lambda\}$ from scratch; if a previously inactive replica becomes energetically important, it enters $\mathcal{K}_{\text{active}}$ again. In this sense, the selection is *dynamic* and reversible, unlike a rigid pruning of the basis.
- Algorithmically, this behaves like a greedy coordinate descent in a very high-dimensional curved space: at each step, we only "push" those coordinates $(a_\lambda, \theta_i^{(\lambda)})$ that currently promise the largest immediate drop in energy, as diagnosed by $|s_\lambda|$.

In the next sections, we will see empirically that this active-set strategy significantly lowers the TDVP cost wall while preserving the ability of the ansatz to explore and refine multiple glassy basins in parallel.

7.1.3 Sensitivity-Based TDVP at $L = 20, M = 86$

To highlight the effect of the sensitivity-based TDVP layer alone, we now show a representative run at $L = 20$, $M = 86$. We start from the one-step ITP frontier produced by the same branching procedure as in Sec. 4.4, which here has size $D_{\text{init}} = 87$ replicas, and we apply TDVP with the active-set rule of Sec. 7.1.2, keeping only the most sensitive $\approx$20% replicas at each step.

Global convergence, solver behaviour, and active-set size. Figure 7.2 summarizes the run. The top panel shows the probability weight on the planted solution, $|\langle s|\Psi(k)\rangle|^2$, as a function of the TDVP step k. Even though at each step we update only a small active subset of replicas, the target probability grows smoothly and reaches $O(0.5)$ by $k = 500$, with a clearly accelerating trend in the late-time regime.

The middle and bottom panels report the MINRES–QLP residual and the number of Krylov iterations per TDVP step. The residual decays rapidly down to $\sim 10^{-15}$ and stays there, indicating that the reduced TDVP systems are solved essentially to machine precision. The number of iterations is largest at the beginning of the evolution and then drops to a few hundred, confirming that restricting to the energetically relevant subspace greatly stabilizes the linear solver.

The effective size of the TDVP subspace at each step is shown in Fig. 7.4(c). The dashed line marks the initial manifold size $D_{\text{init}} = 87$, while the solid line sits at $D_{\text{eff}} \approx 17$ throughout the evolution. Thus each TDVP update operates on only about 20% of the replicas, yet still drives the state toward the planted solution.

Amplitude flow, solution overlaps, and selection frequencies. Figure 7.3 focuses on how individual replicas behave.

The amplitude heatmap in Fig. 7.3(a) shows $a_\lambda(k)$ for all replicas λ and TDVP steps k. A small number of bright horizontal bands emerge: these are the replicas that end up carrying most of the probability weight. Most replicas retain small but nonzero amplitudes, reflecting the fact that the state remains a coherent superposition rather than collapsing to a single product state.

Figure 7.3(b) plots the overlaps $|\langle \rho_\lambda|\text{solution}\rangle|$ versus k. Only a *handful* of replicas acquire a large overlap with the planted solution; one of them eventually reaches overlap close to unity. These are the "winning" replicas that the TDVP flow has learned to align with the solution basin.

The histogram in Fig. 7.3(c) shows how often each replica index λ was included in the active set $\mathcal{K}_{\text{active}}$ over the whole run. Every bar is nonzero: **every single replica is selected at least once**. This validates the design choice of *never* throwing replicas away: even those with small final amplitudes are temporarily important at some stage of the evolution and contribute to steering the flow. Interestingly, the replica that ends up with the largest solution overlap is *not* the one with the highest selection frequency. The active-set strategy is therefore not simply "always pick the eventual winner"; rather, it redistributes attention over time across many replicas that are useful in different phases of the descent.

Finally, Fig. 7.3(d) shows the absolute overlap matrix $|\langle \rho_\lambda|\rho_\mu\rangle|$ at the final TDVP step. The strong diagonal and mostly dark off-diagonal entries indicate that the replicas remain relatively distinct; a few brighter off-diagonals reflect near-duplicate product states. The active-set evolution thus preserves a diverse, nontrivial manifold while gradually amplifying the replicas aligned with the planted solution.

Energy, entropy, coherence, and subspace dimension. Figure 7.4 collects additional scalar diagnostics.

The energy expectation $\langle\Psi|H|\Psi\rangle$ decreases steadily throughout the run [Fig. 7.4(a)], dropping by more than a factor of five. The decay is not perfectly monotone—as expected for a discrete, approximate TDVP integrator—but the overall trend is clearly downward.

The Shannon entropy of the replica weights $|a_\lambda|^2$, shown in Fig. 7.4(b), rises sharply during the early transient as probability mass is spread more evenly across replicas, then slowly increases and finally bends down again. This late-time decrease reflects the concentration of weight onto the few solution-aligned replicas visible in Figs. 7.3(a,b).

Panel (c) shows the already-mentioned effective TDVP subspace size D_{eff} as a function of step. It remains close to 17 for the entire evolution, so each step only touches about 20% of the full manifold. The fact that the target probability and energy continue to improve under this

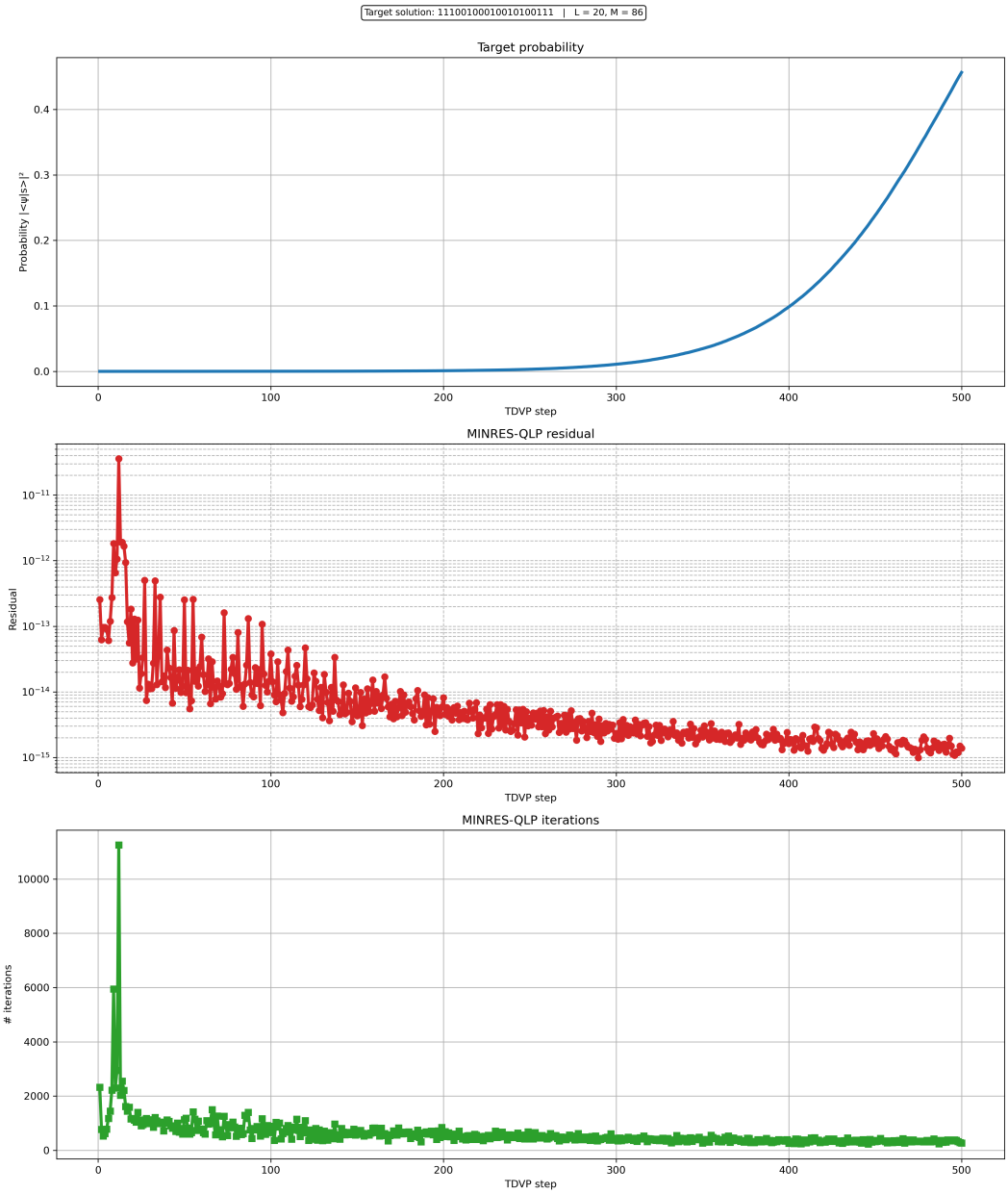
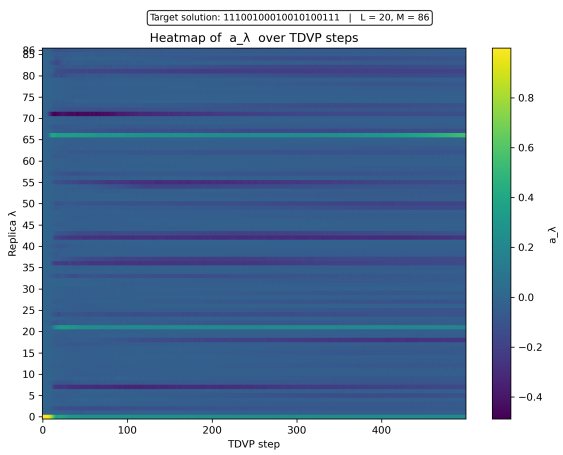


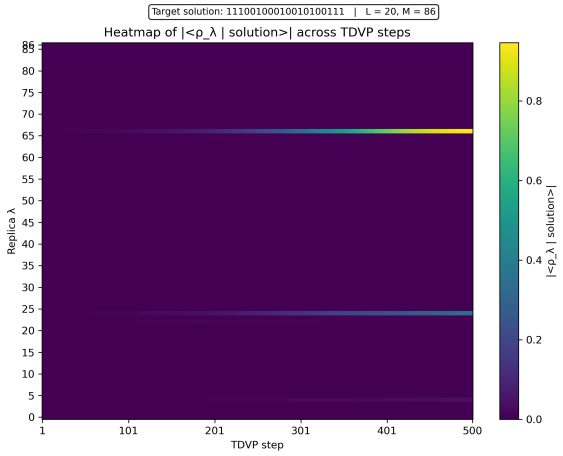
Figure 7.2: Summary of sensitivity-based TDVP at $L = 20$, $M = 86$ (no PCA, one-step ITP frontier). Top: probability weight on the planted solution versus TDVP step. Middle: MINRES–QLP residual for the reduced TDVP linear system. Bottom: number of MINRES–QLP iterations per step. Even though only a small active subset of replicas is updated at each step, the state flows toward the planted solution while the linear solves remain well behaved.

constraint demonstrates that fine-grained control of all parameters at every step is unnecessary: what matters is to *cycle* through replicas according to their current sensitivity, not to update them all simultaneously.

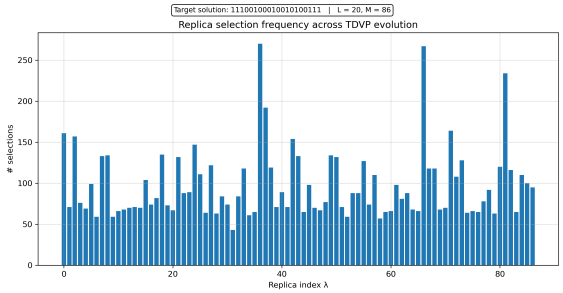
Finally, Fig. 7.4(d) reports the mean, minimum, and maximum values of a per-replica coherence score (a scalar diagnostic that quantifies consistency between clause projectors and local replica configurations). The mean coherence gradually decreases, the minimum rises toward zero, and the maximum collapses early in the evolution. Taken together, these trends indicate that the manifold becomes progressively more self-consistent and less dominated by strongly incoherent replicas as the TDVP flow proceeds.



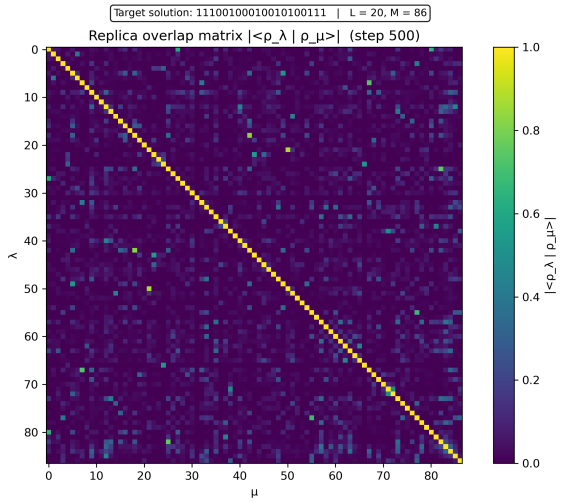
(a) a_λ vs TDVP step.



(b) $|\langle\rho_\lambda|\text{solution}\rangle|$ vs step.



(c) Active-set selection frequency for each λ.



(d) Final replica overlap matrix $|\langle\rho_\lambda|\rho_\mu\rangle|$.

Figure 7.3: Replica-level behaviour in the sensitivity-based TDVP run at $L = 20$, $M = 86$ (no PCA). A few replicas acquire large amplitudes and very high overlap with the planted solution, while all replicas are selected at least once by the active-set rule and remain part of a diverse manifold.

7.1.4 Sensitivity-based TDVP at $L = 30, M = 129$ and Local Metric Rescaling

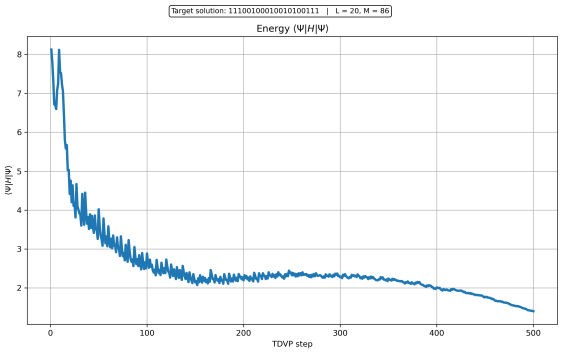
Figure 7.5 shows a representative run at

$$L = 30, \qquad M = 129,$$

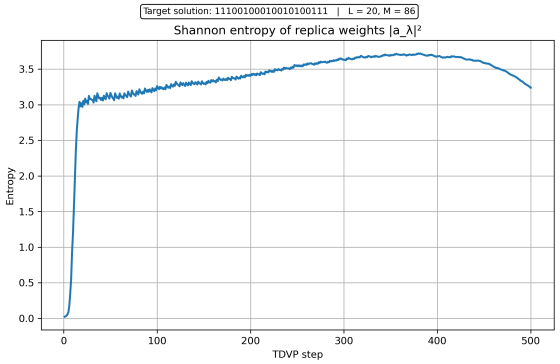
using the same sensitivity-based active-set TDVP as for $L = 20$, but with smaller imaginary-time steps

$$\tau = \gamma = 10^{-3}.$$

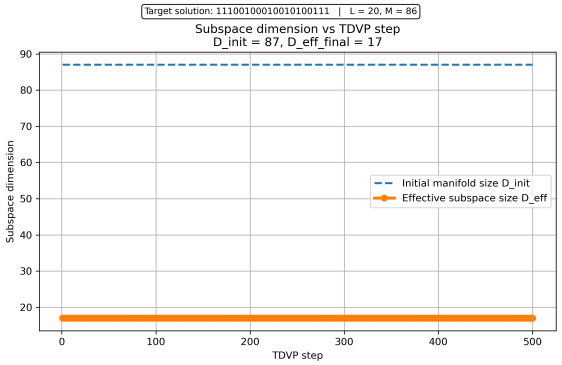
The choice $\tau, \gamma \propto 1/M$ is motivated by the extensive nature of the cost function. Since the Hamiltonian has the form $H = \sum_{m=1}^{M} h_m$, and the TDVP force components scale roughly as M, we keep the parameter update $\delta\bar{w} = -\gamma S_{\bar{w}}^{-1} F_{\bar{w}}$ in a regime where the Euler discretization is reliable, it requires $\gamma = \mathcal{O}(1/M)$. For $M = 129$, we therefore adopt the conservative value $\gamma = 10^{-3}$ (and analogously for the ITP step τ), so that each TDVP step is guaranteed to be "small" on the variational manifold.



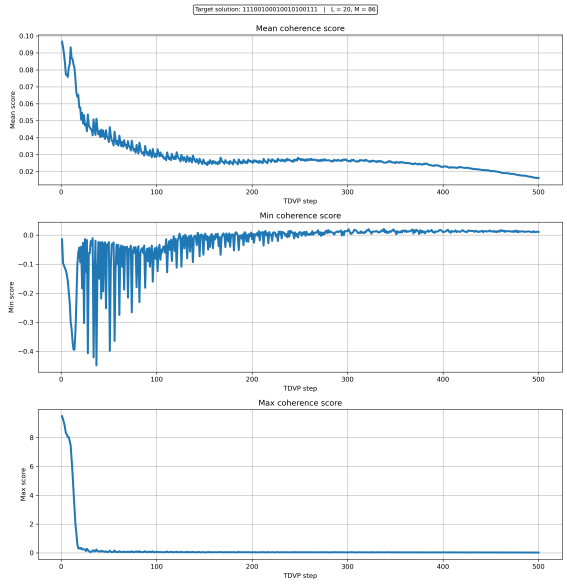
(a) Energy $\langle\Psi|H|\Psi\rangle$ vs TDVP step.



(b) Shannon entropy of $|a_\lambda|^2$.



(c) Initial manifold size D_{init} and active-set size D_{eff}.



(d) Mean, minimum, and maximum coherence score vs step.

Figure 7.4: Scalar diagnostics for the sensitivity-based TDVP run at $L = 20$, $M = 86$ (no PCA). The energy decreases, the entropy first rises and then falls as the state focuses on a few replicas, the active-set dimension stays at about 20% of the full manifold, and coherence metrics become more regular over time.

As expected, the convergence is much slower than at $L = 20$: the probability weight on the planted solution starts growing only after $k \simeq 7{,}000$ TDVP steps and reaches $|\langle s|\Psi(k)\rangle|^2 \simeq 0.2$ only by $k \simeq 12{,}000$, while the total probability on the solution manifold gradually approaches one. This confirms that the combined ITP FRONTIER + ACTIVE TDVP ansatz is *expressive enough* to solve at least some 30-bit instances, but doing so with a globally fixed tiny step size quickly becomes computationally unattractive.

Local Metric Rescaling (LMR). To make the TDVP evolution more efficient we borrow the geometric idea of Local Metric Rescaling introduced by Kaubruegger *et al.* for SR[30], but we apply it directly to our imaginary–time TDVP update. At every TDVP step, for the current parameters $\bar{w}$ we first solve the linear system

$$S_{\bar{w}}\,\bar{\delta w} = -\gamma F_{\bar{w}}, \tag{7.4}$$

where $S_{\bar{w}}$ is the TDVP metric and $F_{\bar{w}}$ is the projected force. The vector $\delta\bar{w}$ is therefore the natural–gradient direction that TDVP would follow for a standard fixed step.

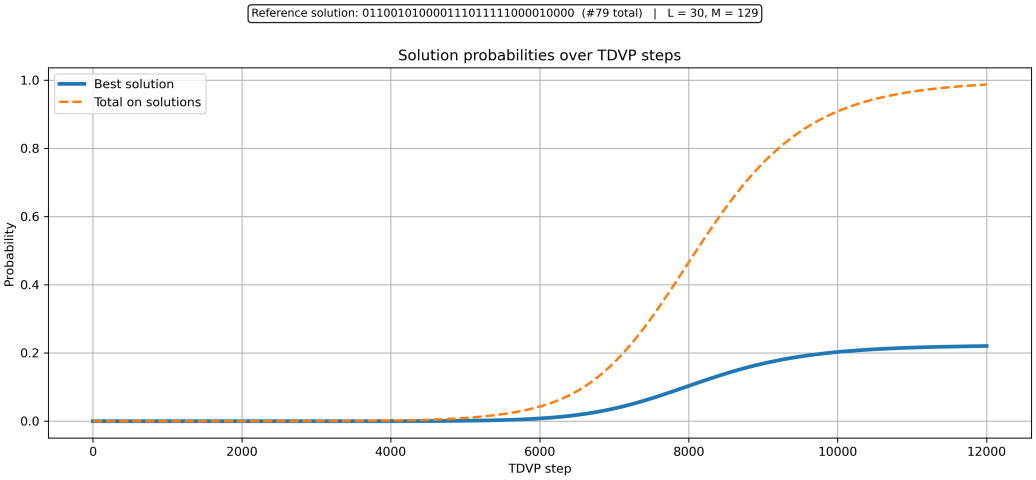


Figure 7.5: Sensitivity-based TDVP at $L = 30$, $M = 129$ with fixed small step size $\tau = \gamma = 10^{-3}$. Top: probability weight on the best discovered solution (blue) and on the whole solution manifold (orange) versus TDVP step. Middle: MINRES–QLP residual for the TDVP linear system. Bottom: number of MINRES–QLP iterations per step. The algorithm eventually concentrates almost all probability on satisfying assignments and identifies the planted solution, but only after $\sim 10^4$ TDVP steps, with clearly slower convergence than in Fig. 7.2.

The important observation is that the *geometric* length of this step on the variational manifold,

$$\ell^2 = \delta\bar{w}^\dagger S_{\bar{w}} \delta\bar{w}, \tag{7.5}$$

can fluctuate by many orders of magnitude from one iteration to the next. When $S_{\bar{w}}$ is badly conditioned or very anisotropic, some updates become extremely long (and can overshoot), while others are ineffectively small. LMR keeps the direction $\delta\bar{w}$ but rescales it so that the metric displacement is approximately constant. We choose a target squared step length $\varepsilon_{\mathrm{LMR}}$ and define the effective scaling factor

$$\gamma_{\mathrm{eff}} = \sqrt{\frac{\varepsilon_{\mathrm{LMR}}}{\delta\bar{w}^\dagger S_{\bar{w}} \delta\bar{w}}}.$$

The actual TDVP update is then

$$\delta\bar{w}_{\mathrm{LMR}} = \gamma_{\mathrm{eff}}\, \delta\bar{w}, \tag{7.6}$$

so that $\delta\bar{w}_{\mathrm{LMR}}^\dagger S_{\bar{w}} \delta\bar{w}_{\mathrm{LMR}} \approx \varepsilon_{\mathrm{LMR}}$. If the bare step is very long in the $S_{\bar{w}}$–metric, γ_{eff} is reduced and the update is damped; if it is very short, γ_{eff} is increased and the update is amplified. In practice we also impose a small lower bound on $\delta\bar{w}^\dagger S_{\bar{w}} \delta\bar{w}$ to avoid instabilities due to numerical noise when this quantity becomes negative or almost zero. Overall, LMR behaves like a metric–aware line search that automatically adapts the step size to the local conditioning of $S_{\bar{w}}$, preventing catastrophic jumps while avoiding unnecessarily tiny moves.

Crucially, this rescaling does not change the fixed points of the imaginary–time evolution: we always move along the same natural–gradient direction $\delta\bar{w}$, so the stationary states of the TDVP equations are untouched. What LMR modifies is only the *parametrization in time* of the trajectory: because γ_{eff} varies from step to step, the iteration index k can no longer be interpreted as a discretized imaginary time with a single constant $\Delta\tau$. TDVP steps should therefore be viewed as optimization iterations rather than physical time units, but the payoff is a much faster and more regular convergence at fixed numerical accuracy.

Impact on the 30-bit instance. Figure 7.6 shows the evolution on the same $L = 30$, $M = 129$ instance as in Fig. 7.5, but now with LMR enabled. The nominal step size is still $\gamma = 10^{-3}$, yet convergence is now dramatically accelerated: both the total probability on

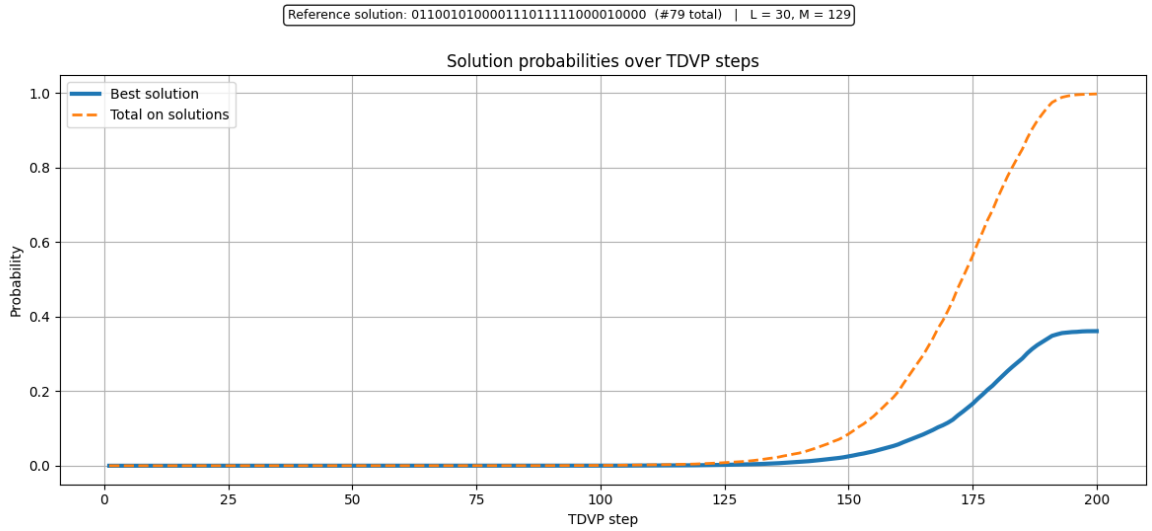


Figure 7.6: Sensitivity-based TDVP with Local Metric Rescaling on the same $L = 30$, $M = 129$ instance. Top: probability weight on the best solution (blue) and on all solutions (orange).

solutions and the weight on the planted solution reach their asymptotic values within $\mathcal{O}(10^2)$ TDVP steps. The price to pay is conceptual rather than computational: the TDVP step index k can no longer be interpreted as a discretized imaginary time, since each step corresponds to a different effective $\gamma_{\text{eff}}(k)$. From an algorithmic perspective, however, we primarily care about iteration count, and LMR yields an enormous speed-up at fixed numerical accuracy.

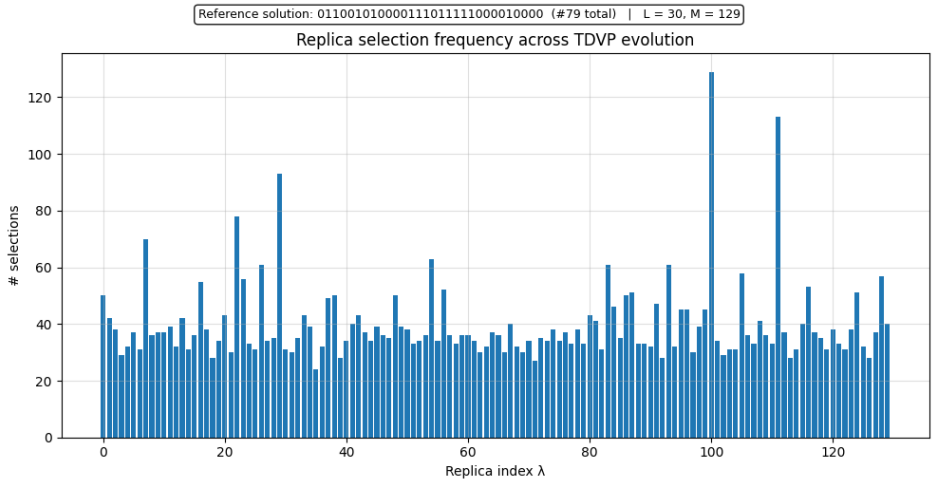
Internal structure at $L = 30$. Finally, we check whether the internal structure of the TDVP manifold at $L = 30$ remains similar to the picture developed in Sec. 6.2.1 for the $L = 5$ instance. Two diagnostics are collected in Fig. 7.7 for the LMR run:

- Panel 7.7a shows how often each replica index λ is selected in the active set over the full TDVP evolution. Only a relatively small subset of replicas is picked very frequently, while many others are touched only sporadically. This mirrors the $L = 5$ behaviour of Fig. 7.3: TDVP learns to focus computational effort on a handful of product states.
- Panel 7.7b tracks the overlap $|\langle\rho_\lambda|\text{solution}\rangle|$ as a function of TDVP step and replica index. Again, only a few replicas ever develop a large overlap with the planted solution, and these are precisely the ones that appear most often in the active set. The majority of replicas remain at small overlap and are gradually pushed to the periphery of the dynamics, even though they are never explicitly discarded.

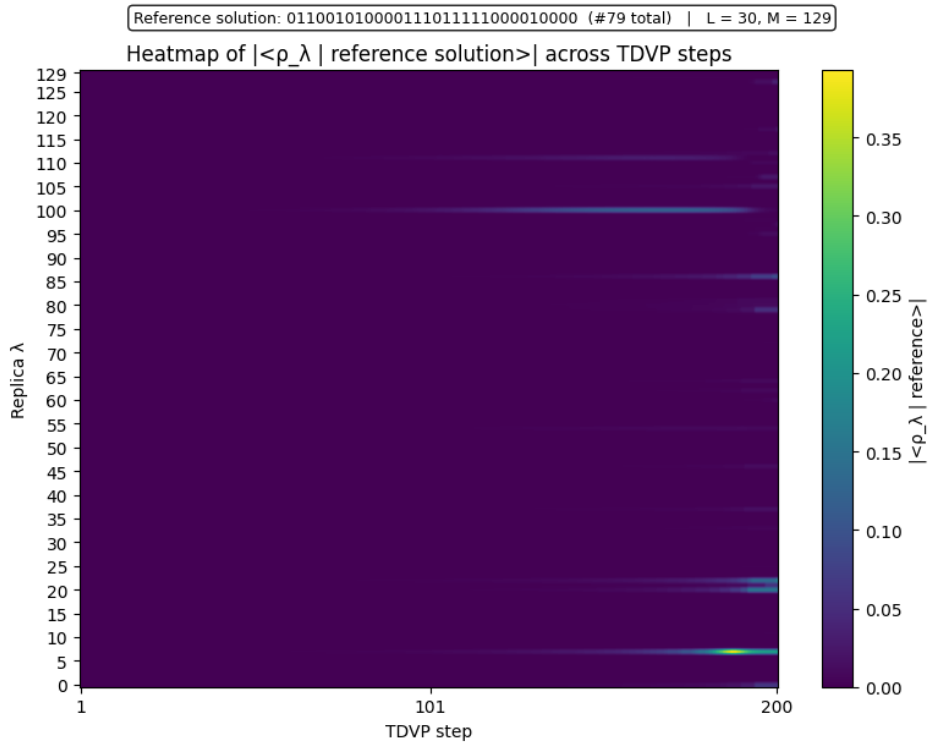
Qualitatively, the 30-bit run thus exhibits the same mechanism already identified in Chapter 5: TDVP uses the variational manifold generated by frontier ITP as a "violation reservoir", then gradually repairs and amplifies a small cluster of solution-aligned product states. The new ingredient is that, in high dimension, Local Metric Rescaling becomes essential to navigate this manifold efficiently without having to reduce the global step size to prohibitively small values.

7.1.5 Finite-size statistics at critical clause density with LMR

Thanks to the speed-up provided by Local Metric Rescaling (LMR), we can carry out a systematic benchmark of the ITP+TDVP algorithm at fixed clause density. For each problem size $L \in \{10,20,30,40,50\}$ we generate 100 random satisfiable 3-SAT instances at critical density $\alpha = M/L \approx 4.3$. All runs use a single frontier-ITP step, so that the uncompressed manifold



(a) Replica selection frequency across the TDVP evolution.



(b) Overlap $|\langle \rho_\lambda | \text{solution} \rangle|$ versus TDVP step for each replica λ.

Figure 7.7: Internal structure of the TDVP manifold with LMR at $L = 30$, $M = 129$. Only a small subset of replicas is selected very frequently and develops a large overlap with the planted solution, in close analogy with the $L = 5$ diagnostics of Sec. 6.2.1.

has dimension $D_{\text{frontier}} = M + 1$. The imaginary-time evolution is implemented with the sensitivity-based active-set TDVP of Sec. 7.1.2, where at every update we retain only the 20% most sensitive replicas and apply LMR to precondition the effective metric. In practice, the TDVP dynamics therefore explores an effective variational space whose dimension is only a fraction of the full frontier manifold, while still growing linearly in M. This scaling is illustrated in Fig. 7.8, which shows both D_{frontier} and the size of the active set as a function of L for a single ITP step.

The linear systems in the TDVP update are solved with a MINRES_QLP Krylov solver. Table 7.1 summarizes the numerical parameters used for all system sizes. In particular, we

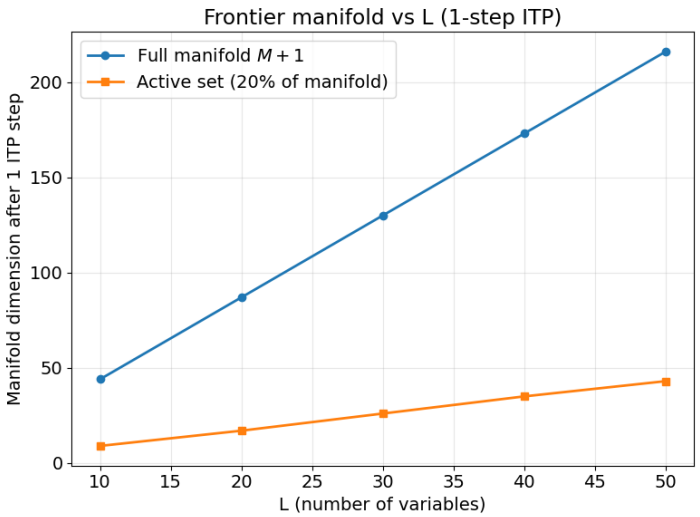


Figure 7.8: Frontier manifold size after a single ITP step. For each L at critical density $\alpha \simeq 4.3$ the full uncompressed manifold has dimension $D_{\text{frontier}} = M+1$ (blue), while the TDVP evolution is restricted at each update to the 20% most sensitive replicas (orange). Even with this truncation, the effective variational space grows linearly with L, whereas the underlying Hilbert space grows exponentially.

L	τ	γ	`reg`	`rtol`	`max_iter`
$10, 20$	10^{-2}	10^{-2}	10^{-8}	10^{-16}	10^4
$30, 40, 50$	10^{-3}	10^{-3}	10^{-7}	10^{-16}	10^4

Table 7.1: Numerical parameters for the TDVP updates with LMR. The regularization `reg` is added to the effective metric S_{eff} entering the MINRES_QLP linear solve.

work with a stringent tolerance `rtol` $= 10^{-16}$ and allow up to `max_iter` $= 10^4$ iterations for MINRES_QLP. The parameter `reg` denotes the diagonal regularization added to the effective metric S_{eff}; for small systems a weaker regularization is sufficient, while for $L \geq 30$ we slightly increase it to improve numerical stability. The parameter γ is the bare TDVP step size, which is then rescaled by LMR according to the local spectrum of S_{eff}.

Figure 7.9 summarizes the overall success probability as a function of L. An instance is counted as "solved" if the final TDVP state has total probability at least 0.9 on satisfying assignments. We allow up to 1000 TDVP steps for $L \leq 40$, while for $L = 50$ we cap the evolution at 500 steps. Each data point in Fig. 7.9 therefore represents the fraction of solved instances out of 100 runs for the corresponding L. All $L = 10$ and $L = 20$ instances are solved, while the success rate gradually decreases to $\sim$95% at $L = 30$, $\sim$88% at $L = 40$, and $\sim$70% at $L = 50$. This degradation with system size is expected: both the Hilbert space and the variational manifold grow exponentially with L, and even with LMR and the active-set truncation (to 20% of the $M + 1$ frontier replicas shown in Fig. 7.8) the TDVP flow must navigate an increasingly complex energy landscape.

Besides success probability, it is also instructive to quantify the actual computational cost of the TDVP evolution. Figure 7.10 collects two complementary runtime diagnostics: the mean wall-clock time per TDVP step and the average number of steps required for convergence.

To understand which formulas are missed by the algorithm it is useful to inspect the distribution of the number of satisfying assignments. Figure 7.11 shows, for each L, the histogram of the ground-state degeneracy (binned on a logarithmic scale) over *all* instances. At critical density the formulas display a broad spread of degeneracies: for $L \geq 20$ many instances already have tens or hundreds of solutions, and for $L = 50$ the distribution extends to very

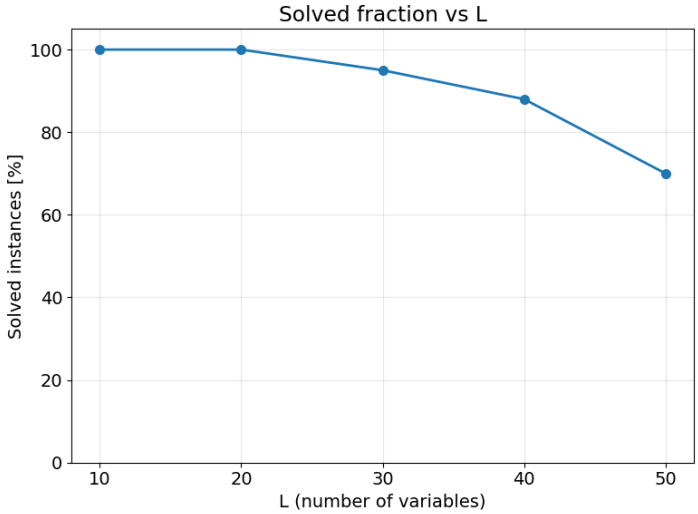
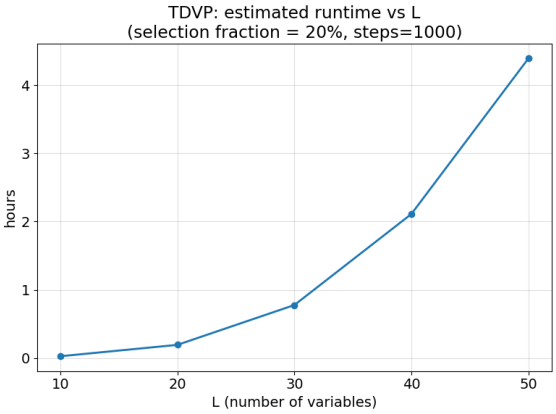
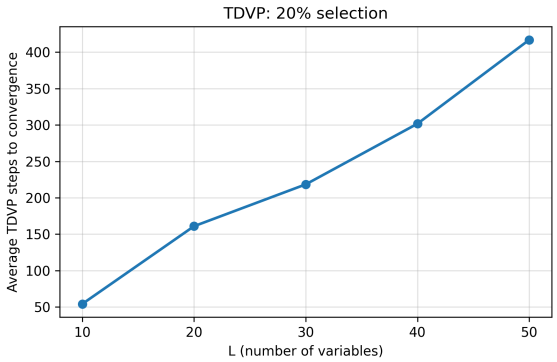


Figure 7.9: Solved fraction versus problem size with LMR. Fraction of instances for which the TDVP dynamics finds at least one satisfying assignment, as a function of L at fixed clause density $\alpha = M/L \approx 4.3$. The maximum number of time steps is 1000 for all system sizes; an instance is counted as solved when the total success probability exceeds 0.9.



(a) Mean TDVP time per step.



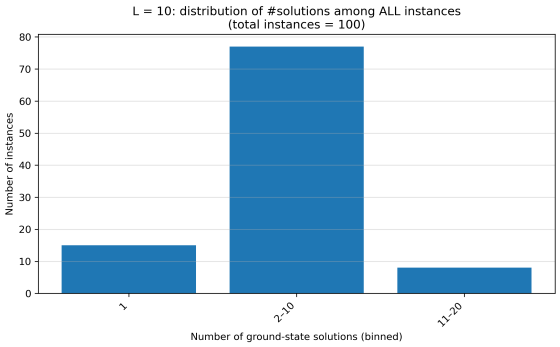
(b) Average TDVP steps to convergence.

Figure 7.10: TDVP runtime statistics with 20% selection. (a) Average wall-clock time per TDVP update for the runs at critical density, using a selection fraction of 20%. The cost per step grows strongly with L, reflecting the increasing dimension of the effective manifold and the larger linear systems in MINRES_QLP. (b) Mean number of TDVP iterations needed to reach the convergence criterion for the successful runs with $L \leq 50$.

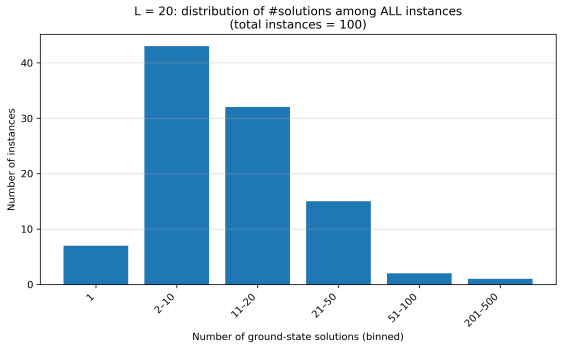
large ground-state counts.

The crucial observation is that the failures of the algorithm are not uniformly spread across these ensembles. Figure 7.12 repeats the same analysis, but restricted to the subset of failed runs (there are no failures at $L = 10$). For $L = 20$ all failures occur on formulas with at most three satisfying assignments. At $L = 30$ and $L = 40$ the majority of failed instances again has only a handful of solutions, and for $L = 50$ the failure set is strongly skewed towards the low-degeneracy tail, even though the overall success probability is lower. In other words, at fixed density the algorithm is most reliable when the formula admits many ground states, and it struggles precisely on instances with very few satisfying bit strings.

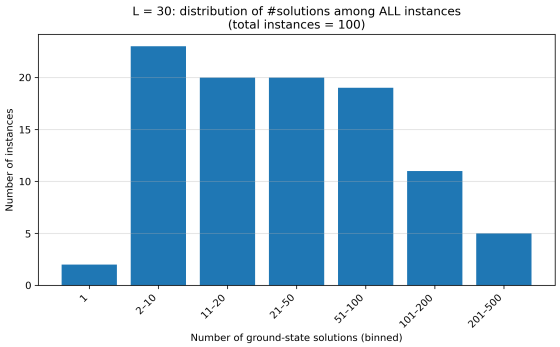
This ensemble-level behaviour is fully consistent with the microscopic picture developed in Chapter 5. In the small-L setting of that chapter we observed that, whenever multiple satisfying assignments are present, the TDVP dynamics tends to spontaneously select and amplify one



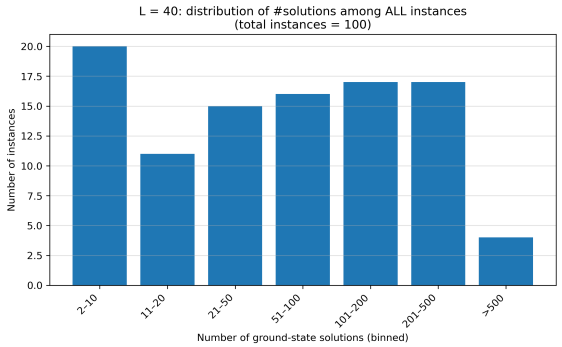
(a) $L = 10$



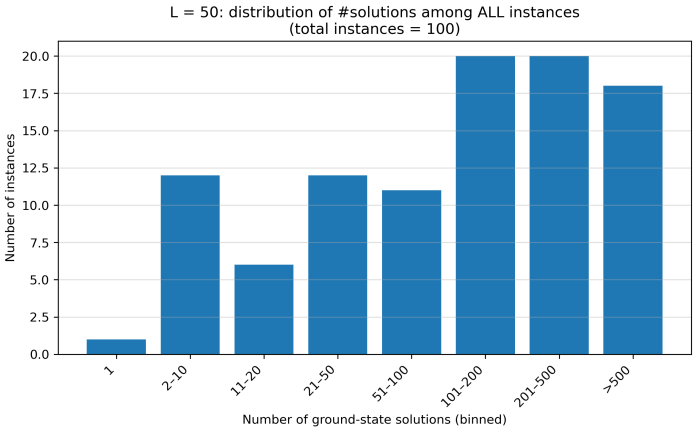
(b) $L = 20$



(c) $L = 30$



(d) $L = 40$

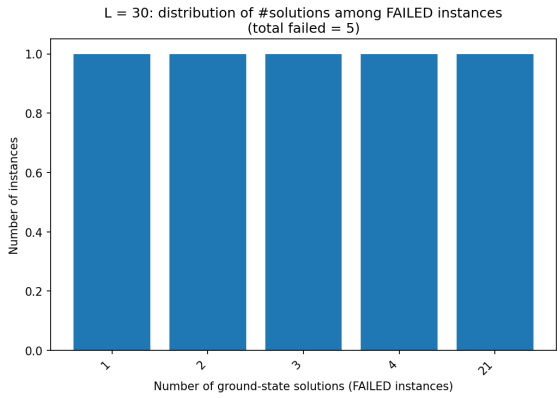


(e) $L = 50$

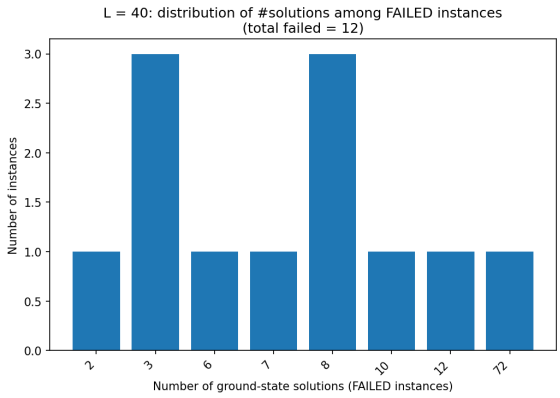
Figure 7.11: Distribution of the number of satisfying assignments at fixed clause density $\alpha \simeq 4.3$**.** Histograms over all random instances for each L; the x–axis is binned logarithmically in the number of ground states. For $L \geq 20$ many formulas already have tens or hundreds of solutions, and the $L = 50$ data show a tail with very large degeneracies.

of them, funnelling probability weight onto a narrow subset of replicas aligned with a specific solution. A larger ground-state degeneracy effectively provides many such "attractors" on the variational manifold, making it easier for the flow to reach at least one of them.

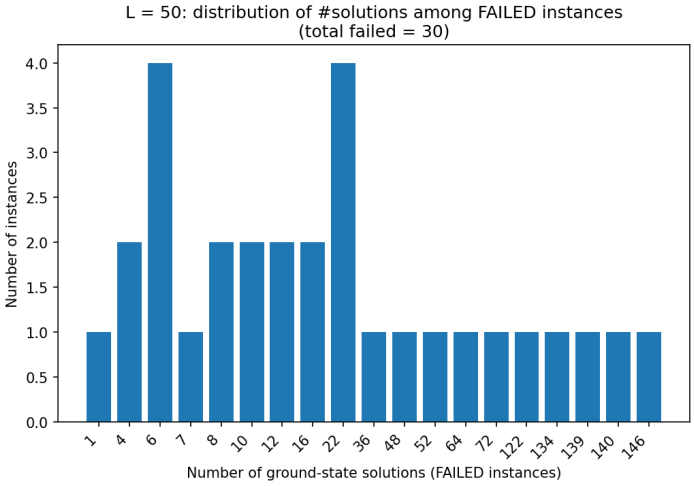
Conversely, formulas with very few solutions are highly frustrated and sharply localized in configuration space. As illustrated explicitly for a representative failed $L = 30$ instance in Sec. 7.1.6, the TDVP dynamics with LMR can still drive the energy expectation down to $\langle H \rangle \sim \mathcal{O}(1)$, but the probability on any exact solution remains tiny and does not accumulate over time. The most natural interpretation is that the variational flow converges to a low-lying *metastable* state, supported on configurations that violate only one or a few clauses on average,



(a) $L = 30$ (5 failed instances)



(b) $L = 40$ (12 failed instances)



(c) $L = 50$ (30 failed instances)

Figure 7.12: Number of satisfying assignments among the failed instances. For each L we show the histogram of ground-state degeneracies restricted to the instances where the TDVP dynamics with LMR does *not* reach a satisfying assignment. Across all sizes, failures are strongly concentrated on formulas with only a few solutions.

yet separated from any true ground state by large barriers. Reaching a satisfying assignment would require coordinated flips of many bits, while TDVP performs smooth, gradient-like parameter updates within a restricted product-replica manifold and cannot easily escape such minima. From this perspective, the statistics of Fig. 7.12 confirm that, at the critical density, the dominant failure mode of the ITP+TDVP algorithm with LMR is not a generic loss of expressivity, but the difficulty of locating the exact ground state in instances with very few satisfying assignments and a rugged energy landscape.

7.1.6 Lack of convergence and interpretation of the energy observable

In order to assess the convergence of the TDVP dynamics, I monitor both the expected energy $\langle\Psi|H|\Psi\rangle$ and the probability weight on the known satisfying assignment(s). Figure 7.13 shows the behaviour of the energy as a function of the TDVP step for a representative 3-SAT instance with $L = 30$ variables and $M = 129$ clauses, while Fig. 7.14 reports the corresponding probability of the reference solution, together with the MINRES–QLP residual and iteration count.

For this class of clause-penalty Hamiltonians, the exact ground-state energy is zero by

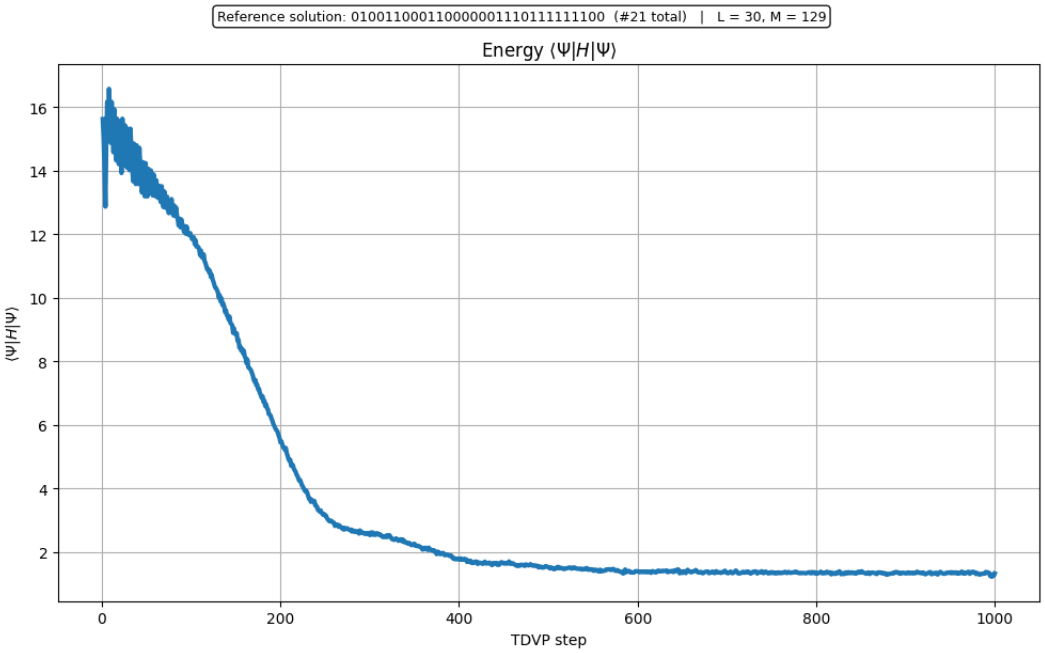


Figure 7.13: Expectation value of the Hamiltonian $\langle\Psi|H|\Psi\rangle$ as a function of the TDVP iteration for a satisfiable 3-SAT instance with $L = 30$ and $M = 129$.

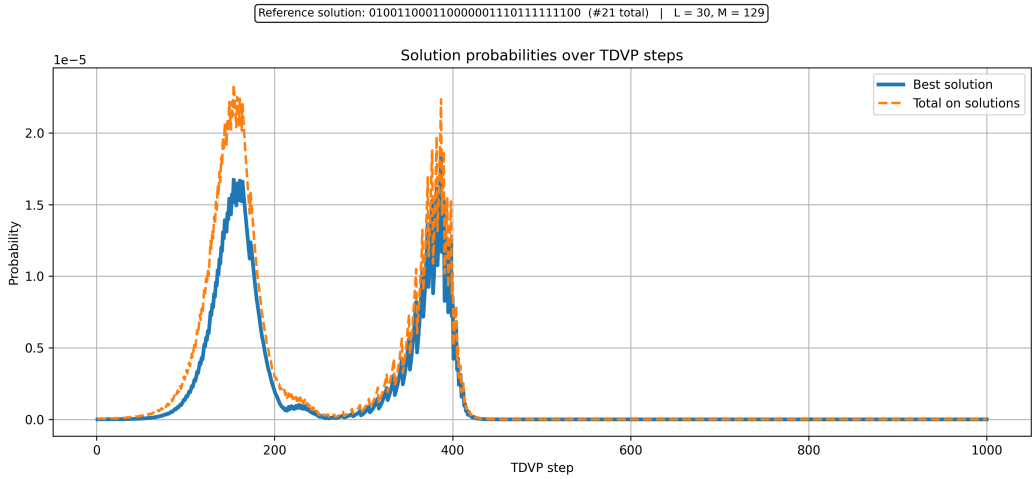


Figure 7.14: TDVP diagnostics for the same instance as in Fig. 7.13. Top: probability of the best solution and total probability on all satisfying assignments

construction: each satisfying assignment corresponds to an eigenstate with energy $E_0 = 0$, while every violated clause contributes a positive penalty. One would therefore expect a successful optimization to both drive $\langle H\rangle$ close to zero and concentrate probability on at least one satisfying bit string.

The numerical results point to a different scenario. Along the TDVP evolution the energy decreases monotonically and reaches values $\langle H\rangle \sim \mathcal{O}(1)$, small compared to the typical scale $\mathcal{O}(M)$ but still clearly non-zero (Fig. 7.13). At the same time, the probability of the reference solution, as well as the total probability weight on all known solutions, remains extremely small (of order 10^{-5} in this example) and only exhibits short-lived peaks before decaying again (Fig. 7.14, top panel). The evolved state $|\Psi\rangle$ thus does not develop a significant overlap with any exact ground state, despite its relatively low energy expectation value.

The most natural interpretation is that the TDVP flow within the chosen variational manifold

converges towards a low-lying *metastable* state rather than to a true ground state. Since the Hamiltonian spectrum encodes the number of violated clauses, a state with $\langle H \rangle \approx 1$ can be viewed as a superposition whose support is concentrated on classical configurations that violate only a few clauses on average, but still fail to satisfy the formula exactly. Reaching a satisfying assignment from such a state would require coordinated flips of a macroscopically large number of bits. The local, gradient-like parameter updates generated by the TDVP equations change the state only smoothly and therefore cannot easily overcome the corresponding energy barriers in the rugged 3-SAT landscape.

This picture mirrors the rigorous insights associated with the m-OGP (Sec. 3.2), which establishes provable barriers for broad classes of local or low-complexity algorithms on random k-SAT and related models. TDVP realizes a deterministic, geometry-aware local dynamics in parameter space and thus inherits analogous limitations in clustered regimes: it can substantially reduce the energy while remaining confined within a metastable basin whose classical support lies at large Hamming distance from any exact solution.

Finally, it is important to stress that a low value of $\langle \Psi | H | \Psi \rangle$ constitutes only a *variational* certificate. Imaginary-time TDVP guarantees that the evolution is optimal within the chosen ansatz manifold, but it does not guarantee that this manifold contains a state with exactly zero energy or that the dynamics will reach such a state. The combination of a highly non-convex cost landscape and a restricted variational family therefore naturally explains why the procedure can achieve $\langle H \rangle \ll M$ without ever concentrating probability on any satisfying assignment.

7.1.7 Preliminary tests with higher TDVP selection fraction

A natural way to try and increase the solved fraction is to enlarge the active set in the one–step frontier protocol, i.e. to keep a larger fraction of the frontier replicas in each TDVP update. As a first test we repeated the benchmark at critical density using exactly the same settings as in Sec. 7.1.5, but increasing the selection fraction from 20% to 30%. Because of the higher cost, these runs were limited to a maximum of 500 TDVP steps for all L (instead of 1000).

Figure 7.15 compares the success probability for the two choices of selection fraction. Even with only 500 steps, the 30% curve lies systematically above the 20% one: at $L = 30$ the solved fraction increases from about 85% to ∼92%, and at $L = 40$ from ∼72% to above 80%. This confirms that, as expected, giving TDVP access to a larger portion of the frontier manifold does help on the more challenging instances.

However, this gain comes at a substantial computational price. Increasing the selection fraction from 20% to 30% enlarges the active manifold dimension from roughly $0.2(M + 1)$ to $0.3(M + 1)$, and the effective metric S_{eff} used in TDVP grows accordingly. Since both the construction of S_{eff} and each MINRES_QLP solve scale superlinearly in the number of active replicas, the wall–clock time per TDVP step grows rapidly with L.

This is quantified in Fig. 7.16, which reports the estimated runtime for 1000 TDVP steps at a selection fraction 30% as a function of L. The curves should be compared with the 20%–selection runtime statistics in Fig. 7.10. Already at $L = 30$, the projected runtime is on the order of a few hours, rising to ∼7 hours at $L = 40$ and approaching a full day at $L = 50$ on our reference hardware. In other words, the apparently modest increase of the selection fraction by 10 percentage points leads to a dramatic growth of the total runtime for larger instances.

These preliminary tests therefore confirm the qualitative picture from Sec. 4.6: the true bottleneck is the scaling of the manifold dimension D and of the associated metric S_{eff}. While a higher selection fraction does improve the solved fraction, the resulting growth of S_{eff} makes the overall runtime explode, rendering this naive strategy prohibitive for larger instances. More sophisticated compression or multi-scale selection schemes would be needed to further increase performance without running into this cost wall.

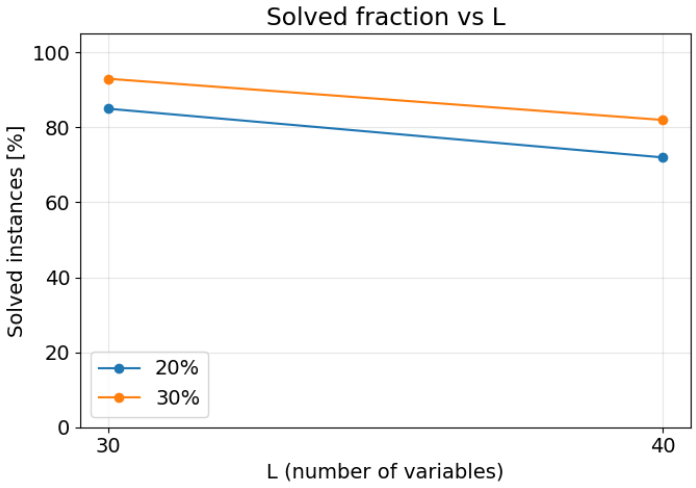


Figure 7.15: Effect of increasing the TDVP selection fraction. Solved fraction versus L at critical clause density for the one–step frontier protocol, comparing a 20% (blue) and a 30% (orange) selection of the most sensitive replicas. All runs are capped at 500 TDVP steps. For both system sizes shown, a larger active set improves the success probability.

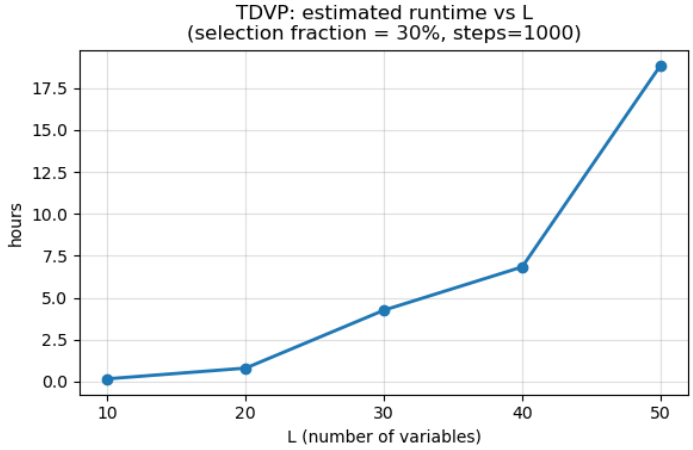


Figure 7.16: Estimated TDVP runtime with 30% selection. Mean wall–clock time required to perform 1000 TDVP steps as a function of L at critical density, using the same numerical parameters as in Table 7.1 but with a 30% active set. Compared with the 20%–selection statistics in Fig. 7.10, the cost grows much more steeply with L, making it impractical to increase the selection fraction even slightly for larger problem sizes.

7.2 Sensitivity-driven compression of the ITP frontier

In Sec. 7.1 we introduced the sensitivity score s_λ [Eq. (7.3)] as a way to identify, at a given TDVP step, the replicas that have the largest energetic leverage and to restrict the TDVP update to that active subset.

When we apply the one-step pipeline

$$1 \text{ ITP–frontier step} \longrightarrow \text{TDVP with } 20\% \text{ active replicas}$$

to ensembles of random 3-SAT instances (100 instances per system size L), we nevertheless see systematic failures on a subset of hard formulas. A plausible explanation is that the one-step frontier is sometimes too narrowly focused on a suboptimal glassy basin; TDVP with the active-set strategy of Sec. 7.1 then optimizes *within* that basin but never reaches a satisfying assignment, even though the energy can be pushed well below M.

A first, very natural, idea is to strengthen the TDVP stage by increasing the active fraction, e.g. from 20% to 30% of the $M+1$ frontier replicas. As shown in Sec. 7.1.7, this indeed raises the solved fraction, but it also makes the effective metric S_{eff} significantly larger and denser. The cost per TDVP update then grows steeply with L, as quantified in Fig. 7.16; in practice, even a modest 10% increase in selection becomes prohibitive for the largest sizes. In other words, pushing the active fraction higher runs into the same parameter-space cost wall discussed in Sec. 5.5.

Extending the frontier in imaginary time does not solve the problem either: Sec. 4.6 and App. A.2 show that multi-step frontiers suffer from a severe *state-space* cost wall (Fig. 7.1). The manifold dimension $D(k)$ grows nearly exponentially with the number of ITP steps, and the TDVP metric becomes increasingly ill conditioned. Thus both obvious knobs—more active replicas at fixed frontier depth, or a deeper frontier at fixed selection fraction—ultimately fail for the same reason: the scaling of D and of S_{eff}.

This motivates a different strategy: *can we grow the ITP frontier for more than one step, but periodically compress it to a smaller, more informative manifold before it explodes?* The goal is to end with a final frontier that is (i) much smaller than a raw multi-step frontier, yet (ii) more expressive and diversified than the purely one-step frontier. We can then feed this compressed manifold to TDVP and again use the sensitivity-based active set of Sec. 7.1.2, keeping the active fraction at about 20% of $M+1$ replicas and avoiding the 30% runtime blow-up seen in Sec. 7.1.7.

7.2.1 ITP compression procedure

The key observation is that the sensitivity score s_λ defined in Eq. (7.3) provides a scalable handle on the frontier. By construction,

$$s_\lambda = a_\lambda \sum_{\mu=1}^{D} H_{\lambda\mu}\, a_\mu, \tag{7.7}$$

so it depends only on the current amplitudes $\{a_\lambda\}$ and on the projected Hamiltonian matrix elements $H_{\lambda\mu} = \langle\rho_\lambda|\, H\, |\rho_\mu\rangle$, which we can evaluate analytically using the product-state structure of the ansatz and the clause decomposition $H = \sum_c \Pi_c$. As discussed in Sec. 7.1.1, $|s_\lambda|$ measures how strongly replica λ is currently "holding on" to energy.

In the TDVP setting we used $|s_\lambda|$ to decide which replicas to *update*. Here we reuse exactly the same quantity earlier in the pipeline: $|s_\lambda|$ becomes a criterion for *keeping or discarding* replicas during frontier growth. This leads to an iterative ITP scheme in which short bursts of frontier growth are interleaved with sensitivity-based compression:

1. **Frontier growth.** Run a block of B frontier-based ITP steps (typically $B = \mathcal{O}(1)$). At each imaginary-time step we branch the current replicas according to the clause projectors, keep successful children, and update the amplitudes $\{a_\lambda\}$.

2. **Sensitivity evaluation.** After B steps the frontier has size D_{curr}. We compute the sensitivity scores $\{s_\lambda\}_{\lambda=1}^{D_{\text{curr}}}$ via Eq. (7.3),

$$s_\lambda = a_\lambda \sum_{\mu=1}^{D_{\text{curr}}} H_{\lambda\mu}\, a_\mu,$$

   evaluating $H_{\lambda\mu}$ analytically as in Sec. 7.1.1.

3. **Selection of important replicas.** We rank replicas by $|s_\lambda|$ and retain only a target fraction $f_{\text{keep}} \in (0,1)$. If D_{curr} is the current frontier size, we keep $n_{\text{keep}} = \lceil f_{\text{keep}} D_{\text{curr}} \rceil$ replicas with largest $|s_\lambda|$. This filters out energetically irrelevant directions and focuses the manifold on those replicas that appear most promising for further energy reduction.

4. **Diversity filter.** A pure $|s_\lambda|$ ranking tends to keep many nearly identical replicas, especially once the frontier has clustered around a good local pattern. To avoid a compressed manifold that is small but highly redundant, we enforce a simple diversity constraint: as we scan the sorted list, a candidate is accepted only if its pattern of single-spin angles $\{\theta_i^{(\lambda)}\}$ is sufficiently different from those already kept (e.g. via a cosine-similarity threshold). Near-duplicates are discarded.

5. **Compression and restart.** The replicas that pass the sensitivity and diversity filters define the new frontier, with their original amplitudes and angles. All others are dropped. We then recast this compressed frontier into the data structures used by the ITP routine and start the next cycle: another block of B growth steps, followed by a new compression.

This alternating growth–compression scheme keeps the frontier dimension under control throughout the ITP while continuously pruning energetically irrelevant or redundant replicas. In practice, we also impose a final cap on the frontier size before entering TDVP: after the last ITP block we perform one last sensitivity-based compression enforcing

$$D_{\text{final}} \leq M + 1. \tag{7.8}$$

In the following, this final cap on the manifold dimension will be referred to as "aggressive" compression. The resulting compressed frontier then serves as the starting manifold $\{|\rho_\lambda\rangle\}$ for the TDVP stage.

7.2.2 ITP-compression on failed $L = 30$ instance

To gauge how much extra structure such a compressed frontier can provide, we revisit one of the hardest instances in our $L = 30$ ensemble, with $M = 129$ clauses and 21 satisfying assignments (the same instance used in Sec. 7.1.6).

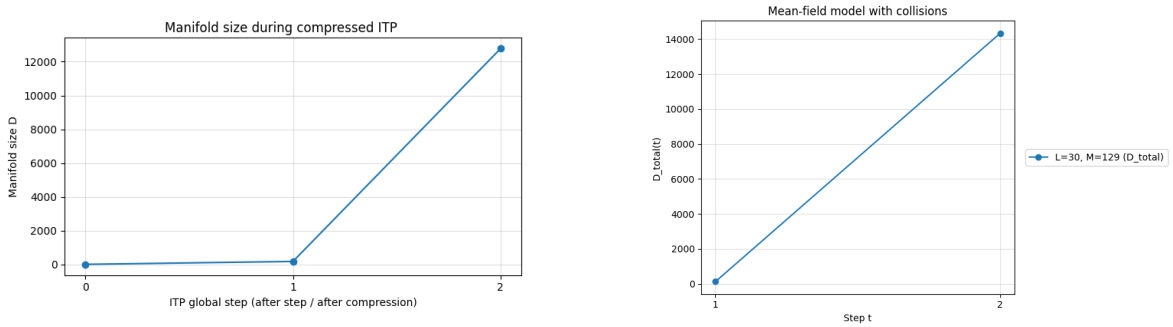
"Soft" sensitivity-driven ITP compression.

In the original pipeline—one ITP–frontier step followed by TDVP with a 20% active set—the algorithm essentially never builds appreciable weight on any solution: the probability of all satisfying assignments remains tiny , and the energy stalls far above zero.

We then rerun the instance with a *two-step* frontier interleaved with sensitivity-based compression after each step ($B = 1$). At each compression, we keep the most sensitive ∼20% of replicas and apply the diversity filter; however, we *do not* impose the final cap $D_{\text{final}} \leq M + 1$; we refer to this as "soft" compression. The resulting "deep but compressed" frontier has size $D_{\text{final}} = 501$, to be compared with $D = 130$ for the one-step frontier and $D \simeq 1.3 \times 10^4$ for the fully uncompressed two-step frontier (Fig. 7.17).

Feeding this compressed manifold into TDVP, we set the selection fraction to enforce the standard $20\%(M + 1)$ active-set rule based on $|s_\lambda|$. Thus, even if this manifold is larger, we work within effective subspaces of the same dimension (26) for both the uncompressed and "soft" compression, enabling a coherent comparison of the two procedures. On the compressed manifold, the TDVP dynamics change qualitatively: as shown in Figs. 7.19 and 7.20, the total probability of satisfying assignments rises sharply and approaches unity, while the best solution acquires a finite fraction of the total weight. Apart from a short initial bump, the energy decays monotonically towards values compatible with the ground-state energy. The selection histogram in Fig. 7.21 shows that the sensitivity-based rule heavily favors a small subset of replicas, but every replica is selected at least occasionally, as discussed in Sec. 7.1.2.

Thus, with a relatively modest increase of the final manifold size (from 130 to 501 replicas, still tiny compared to 2^L and to the uncompressed frontier) we solve a hard 3-SAT instance while keeping the same number of active replicas per TDVP step unchanged.



(a) ITP frontier (data).

(b) Mean-field with collisions.

Figure 7.17: Growth of the manifold size D as a function of the global ITP step for the *uncompressed* two-step frontier at $L = 30$, $M = 129$. Panel (a) shows the exact frontier dimension, which exhibits an exponential blow-up reaching $D \simeq 1.3 \times 10^4$ at the second step. Panel (b) displays the mean-field estimates discussed in Sec. 4.6 and App. A.2, confirming the quantitative accuracy of the analytical model.

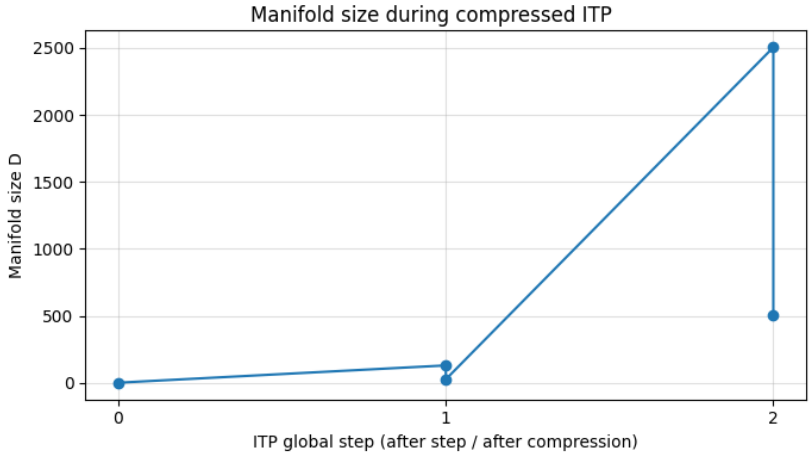


Figure 7.18: Manifold size D during the two-step frontier with sensitivity-based compression at each step for the same $L = 30$, $M = 129$ instance. After each ITP step only the most sensitive and diverse replicas are retained, leading to a final compressed frontier with $D_{\text{final}} = 501$ instead of the $\sim 1.3 \times 10^4$ replicas of Fig. 7.17a.

"Aggressive" sensitivity-driven ITP compression

The previous soft-compression experiment at $L = 30$ raises a key question: is the performance gain mainly due to the *larger* manifold (allowing selection from more candidates without the unchecked expansion of a 2-step ITP or a 30% selection) or to the manifold being more *informative* thanks to sensitivity-driven pruning?

To disentangle these effects we discuss the second variant of the pipeline, which we will call "aggressive" compression. We keep the same growth–compression loop as in the soft procedure (same block length B, same fraction f_{keep}, same diversity filter), but at the very end of the ITP stage we enforce a strict cap

$$D_{\text{final}} = \min(D_{\text{curr}}, M + 1), \tag{7.9}$$

by performing one last sensitivity-based truncation. This yields a compact frontier with at most $M+1$ replicas, constructed by the same multi-step, sensitivity-guided ITP dynamics as the larger compressed manifold, but ultimately constrained to have the *same* size as the one-step frontier used in the baseline pipeline. TDVP is then run with the usual $20\%(M + 1)$ active-set rule, so that in all comparisons (uncompressed, soft and aggressive) the TDVP update acts on

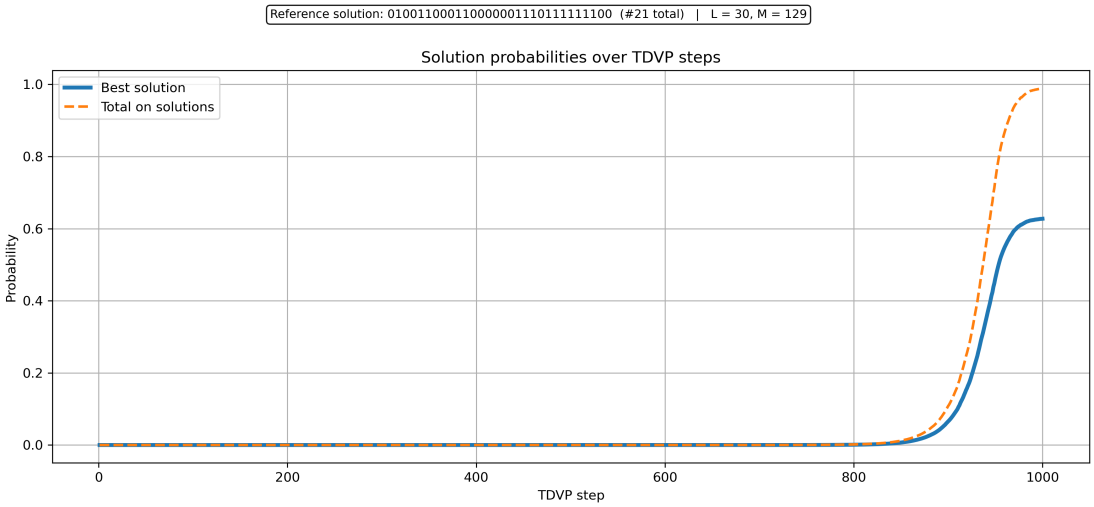


Figure 7.19: TDVP evolution of the probability on satisfying assignments for the softly compressed two-step frontier at $L = 30$, $M = 129$ with 21 solutions. The solid line shows the probability of the best solution; the dashed line shows the total probability on all satisfying assignments. Both rise sharply near the end of the trajectory, in contrast to the single-step frontier run where solution probabilities remain negligible.

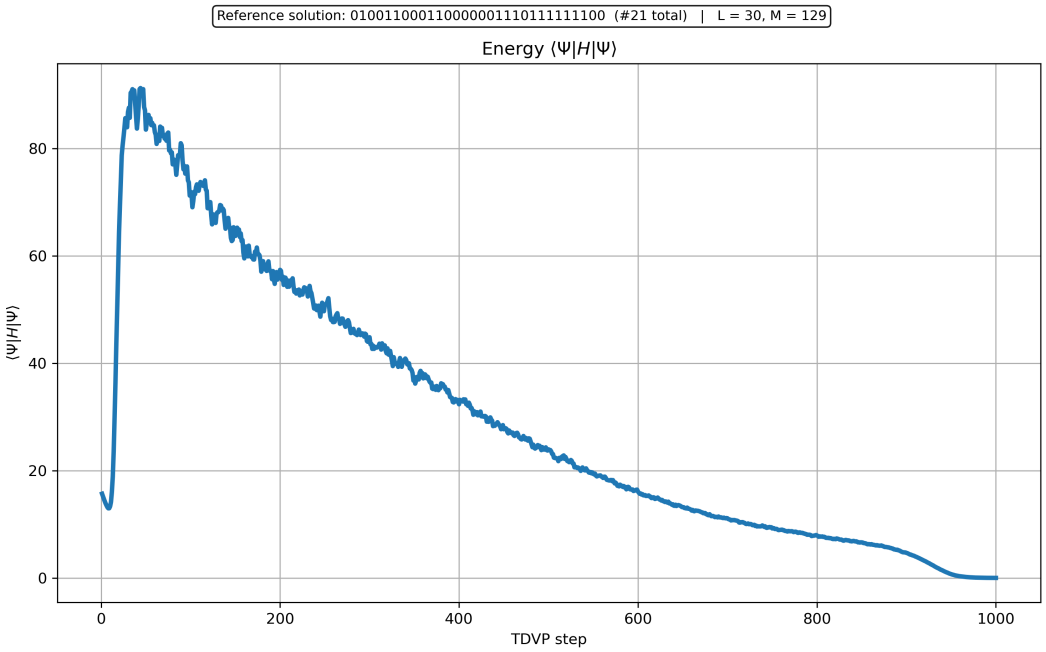


Figure 7.20: Expectation value $\langle\Psi|H|\Psi\rangle$ as a function of TDVP step for the softly compressed two-step frontier. After a short transient peak, the energy decreases monotonically and approaches zero, consistent with convergence to a satisfying assignment.

the same number of replicas.

Below, we summarize we complete our previous analysis introducing the soft compression on the same failed $L = 30$ instance.

Frontier growth and manifold size. The three pipelines are:

1. one-step frontier with $D = 130$ (uncompressed),

2. softly compressed two-step frontier with $D_{\text{final}} = 501$ (see Fig. 7.18),

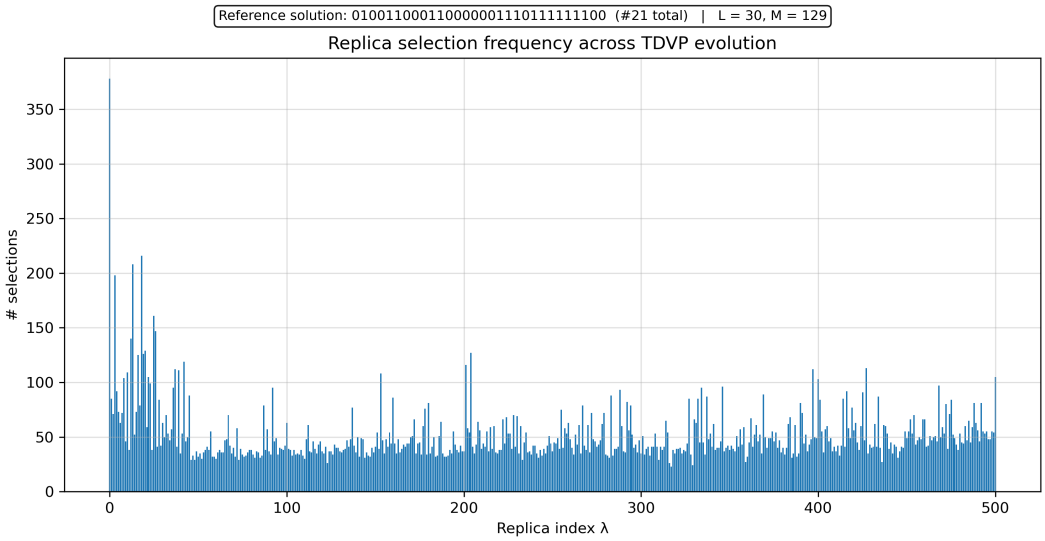


Figure 7.21: Histogram of replica-selection frequencies across the TDVP evolution for the softly compressed two-step frontier. Each bar counts how many TDVP steps a given replica λ belonged to the active set. The selection is strongly concentrated on a subset of replicas, but every replica is selected at least occasionally, reflecting the dynamic, reversible nature of the sensitivity-based active set discussed in Sec. 7.1.2.

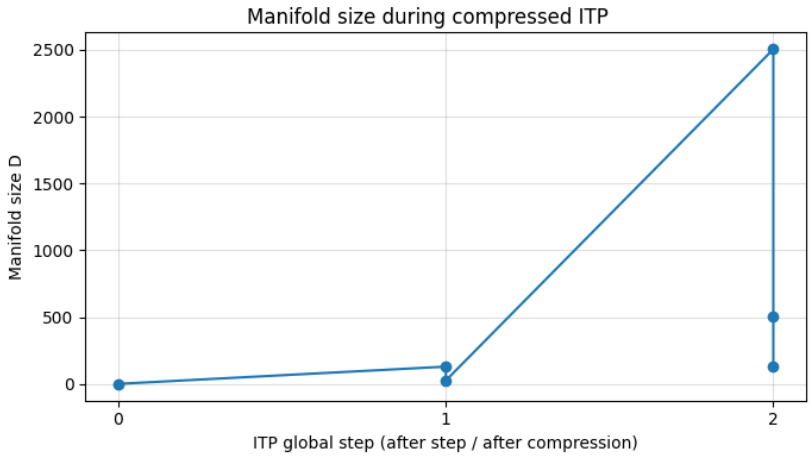


Figure 7.22: Manifold size D during the aggressively compressed two-step frontier for the $L = 30$, $M = 129$ instance. After growth similar to the soft-compression case, a final sensitivity-based truncation enforces $D_{\text{final}} = M + 1 = 130$, yielding a compact but more structured manifold.

3. aggressively compressed two-step frontier with $D_{\text{final}} = M + 1 = 130$ (see Fig. 7.22)

With TDVP always updating $20\%(M + 1)$ replicas per step.

Solution probabilities and energy decay. On the soft frontier, the TDVP dynamics manage to reach the satisfying manifold (see Fig.7.19) , but do so relatively late compared with the aggressive pipeline that produces a much sharper transition. (Fig. 7.23) Additionally, the energy decays more steeply, flattening at zero (Fig. 7.24).

Replica-selection patterns. The selection histogram for soft compression (Fig. 7.21) shows a small subset of very frequently selected replicas and a long tail of almost-inactive directions. Aggressive compression, instead, yields a markedly more balanced pattern (Fig. 7.25): many replicas are used often, and very few are completely dormant.

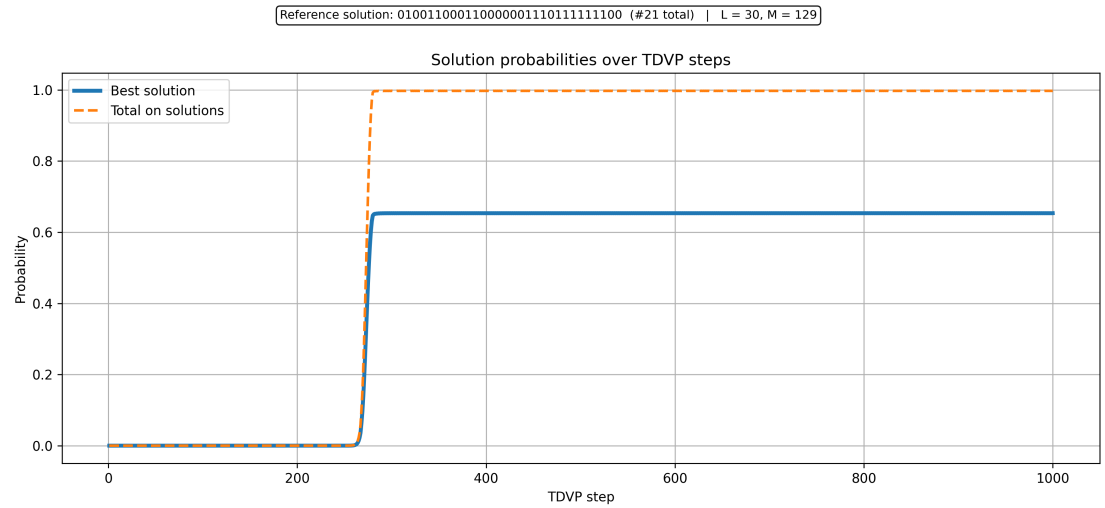


Figure 7.23: TDVP evolution of solution probabilities for the aggressively compressed two-step frontier at $L = 30$, $M = 129$. As in Fig. 7.19, the dashed line shows the total probability on all satisfying assignments and the solid line the probability of the best solution. Here the jump to unit total probability occurs significantly earlier and over a narrower time window.

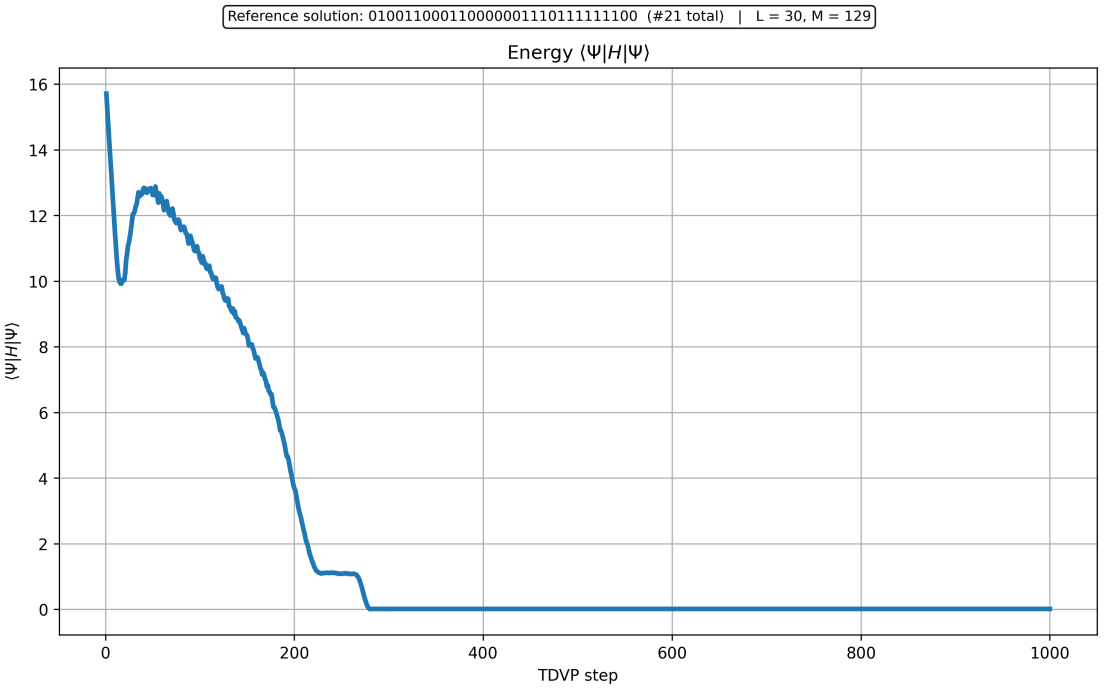


Figure 7.24: Energy expectation $\langle\Psi|H|\Psi\rangle$ for the aggressively compressed run. The decay is substantially steeper than in Fig. 7.20 and quickly reaches zero, where it stays for the rest of the evolution, indicating convergence to the satisfying subspace.

In summary, for this $L = 30$ instance the aggressively compressed frontier outperforms both the one-step frontier and the soft frontier, even though all three pipelines use the same TDVP active-set size and, in the aggressive case, the *same* total manifold size $D_{\text{final}} = M + 1$ as the one-step baseline. The improvement seems to be structural: sensitivity-driven compression is carving out a more informative manifold, not merely buying accuracy with more parameters.

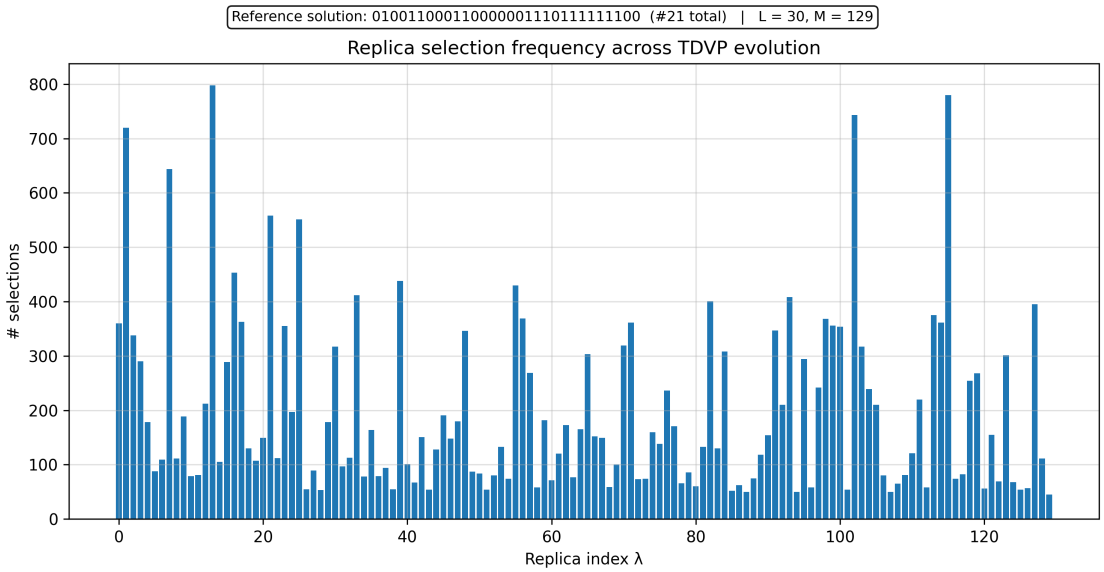


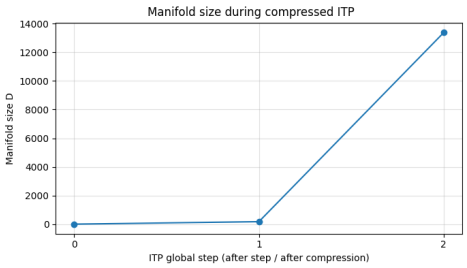
Figure 7.25: Replica-selection frequency across the TDVP evolution for the aggressively compressed frontier. Compared to Fig. 7.21, a larger fraction of replicas is selected many times, indicating that the compact manifold contains fewer "dead" directions that are almost never exploited by the TDVP dynamics.

7.2.3 ITP-compression for increasing system sizes

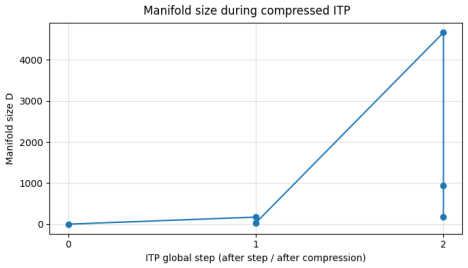
Since the aggressive compression procedure appears to provide a genuine advantage at $L = 30$, the next question is whether the same mechanism remains effective as the system size grows. In this subsection, we compare the aggressive compressed-frontier and uncompressed protocols for a failed $L = 40$ instance and a solved $L = 50$ instance taken from from the same 1000-step batch of Sec. 7.1.5. The full soft-compression data for $L = 40$ and $L = 50$ are reported in Appendices , A.6.1, and A.6.2. In both cases the qualitative message is the same: aggressive, sensitivity-driven compression allows us to exploit the extra expressivity of a multi-step frontier while keeping the final manifold size $D_{\text{final}} \sim M + 1$ and a small TDVP active set, leading to a cleaner and faster convergence than in the one-step baseline.

Frontier growth and manifold size. On the ITP side, both instances clearly illustrate the state-space cost wall of uncompressed multi-step frontiers. For the $L = 40$ formula, extending the frontier naïvely to a second step would already generate $D \simeq 1.3 \times 10^4$ distinct product states (Fig. 7.26, left panel), far beyond what the TDVP metric can handle. Similarly, for the $L = 50$ instance, a hypothetical uncompressed second step produces more than 2×10^4 replicas (Fig. 7.27, left panel), again signaling an impending blow-up of the manifold dimension. Aggressive compression tames this proliferation in essentially the same way at both sizes. (Fig. 7.26, right panel, and Fig. 7.27, right panel) In other words, it inherits information from a deeper frontier while preserving the scaling of the one-step metric: the TDVP stage never sees more than $M{+}1$ replicas, and the active set still contains only $0.2(M{+}1)$ replicas per update (about 35 at $L = 40$ and 43 at $L = 50$). Softly compressed two-step frontiers, where the final D_{final} is allowed to grow to $\sim 10^3$ replicas, sit between these two extremes. They keep the manifold size under control compared to the fully uncompressed frontier, but do not enforce the strict $D_{\text{final}} = M + 1$ cap. The detailed soft-compression data for $L = 40$ and $L = 50$ are reported in Appendix A.6.

Impact on TDVP dynamics. The effect of these different frontiers on the TDVP evolution is shown in Figs. 7.28–7.29 for $L = 40$ and in Figs. 7.31–7.32 for $L = 50$. Despite the different



(a) Hypothetical two-step uncompressed frontier.



(b) Aggressively compressed two-step frontier.

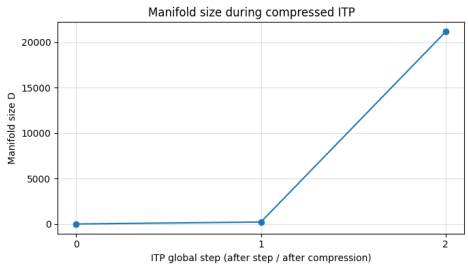
Figure 7.26: Manifold size during two-step ITP at $L = 40$, $M = 172$. (a) If we naively extended the uncompressed frontier to two steps, D would explode to $\sim 10^4$ states. (b) With aggressive sensitivity-based compression, the two-step frontier is distilled back to $D_{\text{final}} = M + 1 = 173$, matching the size of the one-step baseline.

sizes and clause densities, the two instances display a remarkably similar pattern. For the $L = 40$ formula, the aggressively compressed frontier turns a failing run into a successful one. With compression, both the total solution probability and the probability of the best solution exhibit a sharp freezing transition: after a long transient with negligible overlap on any satisfying assignment, they rapidly jump to 1 and remain there (Fig. 7.28a), while the energy decays to zero and stays pinned at the ground-state value (Fig. 7.29a). In stark contrast, the one-step frontier relaxes only to a metastable energy plateau above zero (Fig. 7.29b), and the solution probability never becomes appreciable (Fig. 7.28b). For the $L = 50$ instance all pipelines eventually succeed, but aggressive compression again leads to a noticeably cleaner TDVP flow. The compressed frontier drives the state into the satisfying manifold earlier and more decisively: the total probability on solutions and the probability of the best solution start to rise around 250 steps and quickly saturate close to 1 and 0.6, respectively (Fig. 7.31, left panel), while the energy drops steeply to zero with only a modest shoulder at intermediate times (Fig. 7.32, right panel). The baseline one-step frontier shows the same qualitative behaviour but shifted to later times and with a slightly smaller asymptotic weight on the best solution. Overall, these two instances suggest that aggressive compression is not just trading accuracy for a larger parameter count; even when $D_{\text{final}} = M + 1$ as in the one-step baseline, the TDVP flow on the compressed manifold converges faster and more sharply toward the satisfying subspace.

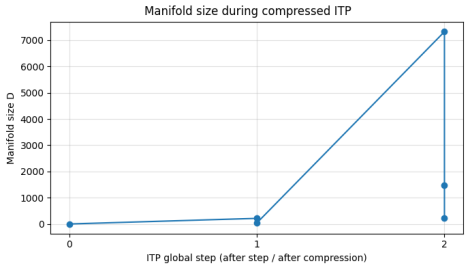
Replica-selection patterns. The selection histograms complete this picture. At $L = 40$, the aggressively compressed frontier yields a compact manifold in which most replicas are selected a substantial number of times (Fig. 7.30a), whereas the one-step frontier leaves more directions effectively unused (Fig. 7.30b). The TDVP active set therefore spends most of its time on genuinely useful directions rather than redistributing weight within redundant clusters.

A similar effect is visible at $L = 50$ (Fig. 7.33). In the one-step baseline many replicas are selected only rarely, while in the aggressively compressed pipeline the pattern of "few very active, many occasionally active" replicas becomes more pronounced and essentially no direction is completely dormant. The limited active set is concentrated on a small backbone of energetically important replicas, but still occasionally explores the rest of the manifold.

In summary, across these $L = 40$ and $L = 50$ examples, aggressive, sensitivity-driven compression consistently distils more informative frontiers out of multi-step ITP. It avoids the state-space blow-up of uncompressed frontiers, preserves the favourable $D_{\text{final}} \sim M + 1$ scaling of the one-step pipeline, and steers TDVP away from redundant directions. In several challenging cases, including the $L = 40$ instance that fails in the baseline algorithm, this structural advantage is enough to reach the satisfying manifold without increasing either the total manifold size or the TDVP active-set fraction.

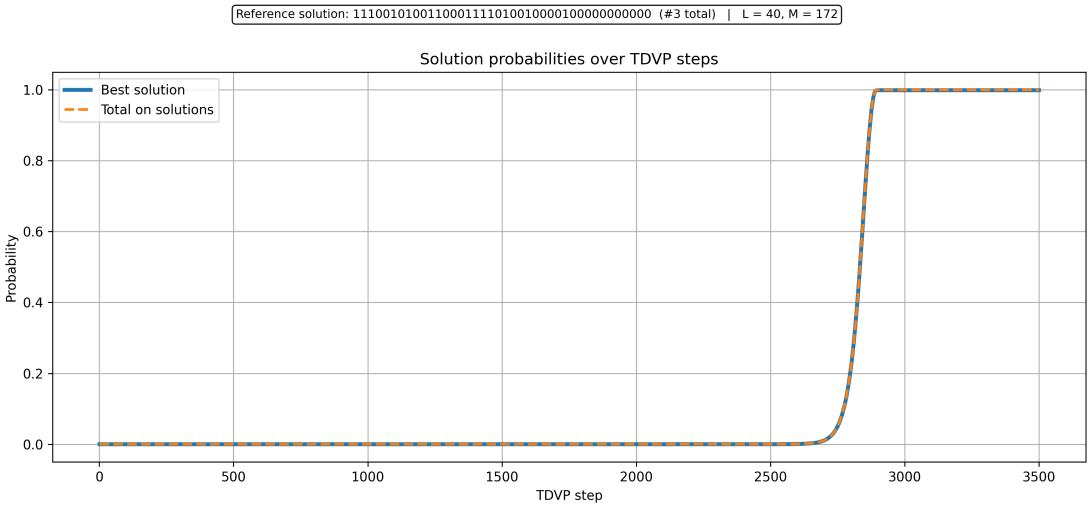


(a) Hypothetical two-step uncompressed frontier.

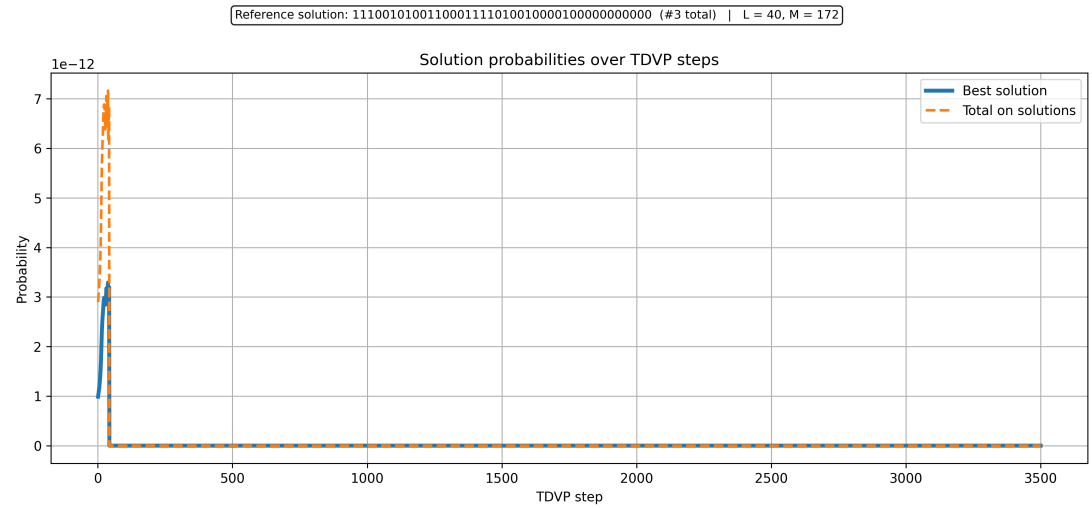


(b) Aggressively compressed two-step frontier.

Figure 7.27: Manifold size D during two-step ITP for a solved $L = 50$, $M = 215$ instance. Without compression a second ITP step would generate more than 2×10^4 replicas, whereas aggressive compression keeps D of order M and enforces $D_{\text{final}} = M + 1 = 216$, matching the one-step baseline.

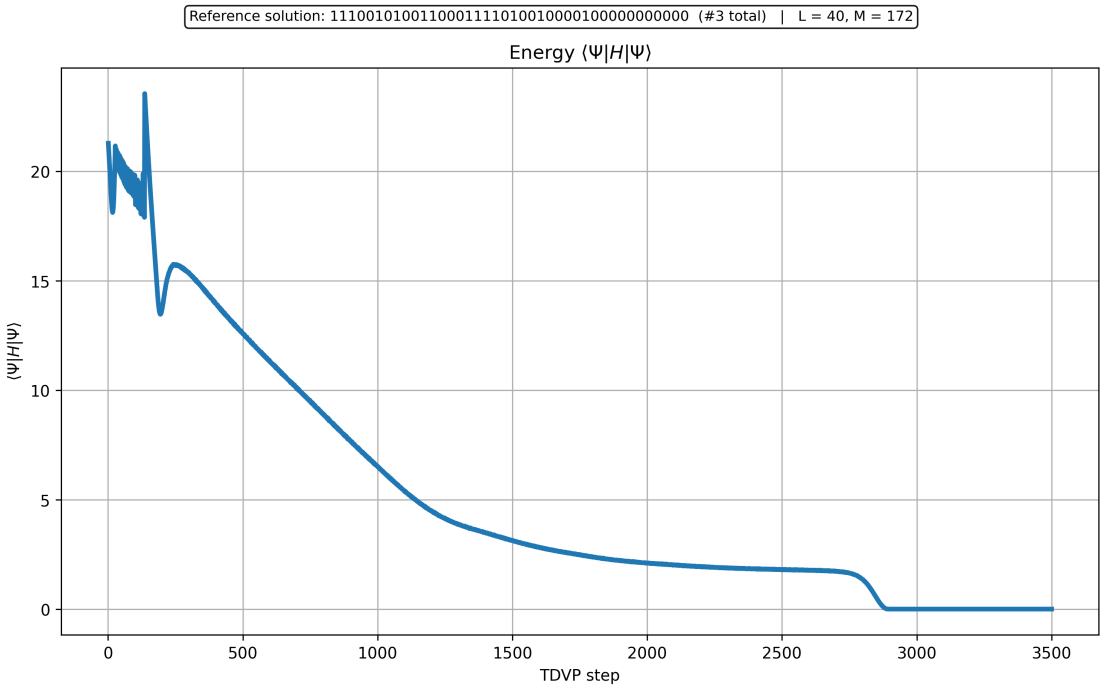


(a) Aggressively compressed two-step frontier.

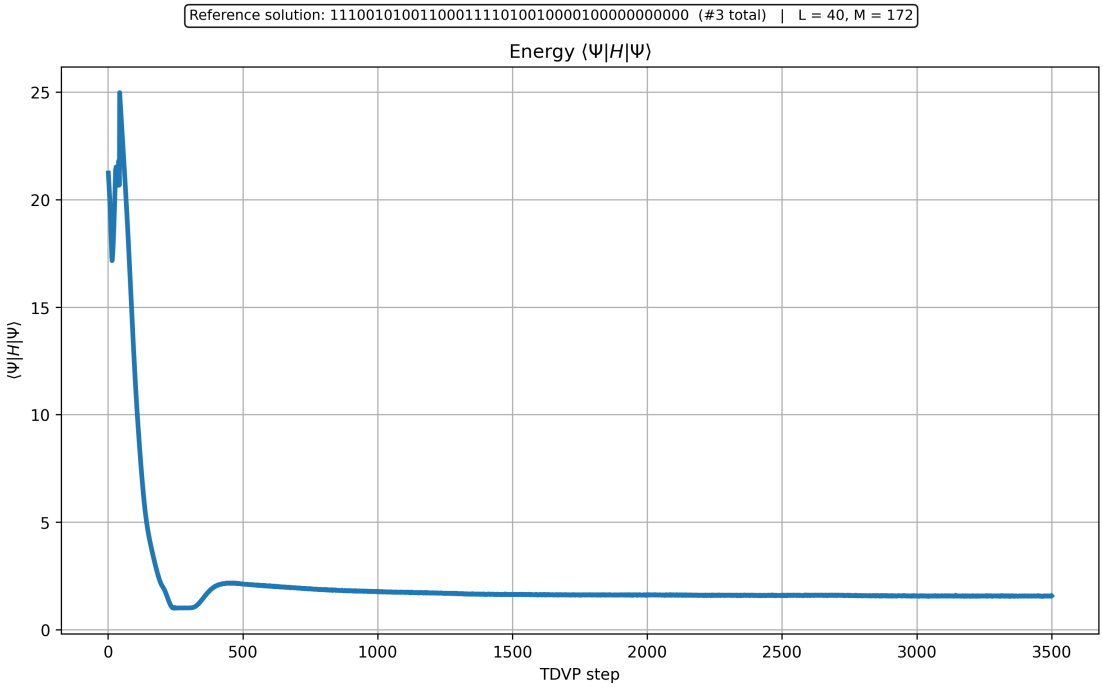


(b) Baseline one-step frontier.

Figure 7.28: Solution probabilities for the $L = 40$, $M = 172$ instance. Solid lines: probability of the best solution; dashed lines: total probability on all three satisfying assignments. Aggressive compression (a) drives the state to unit success probability, whereas the one-step frontier (b) never develops significant weight on any solution.

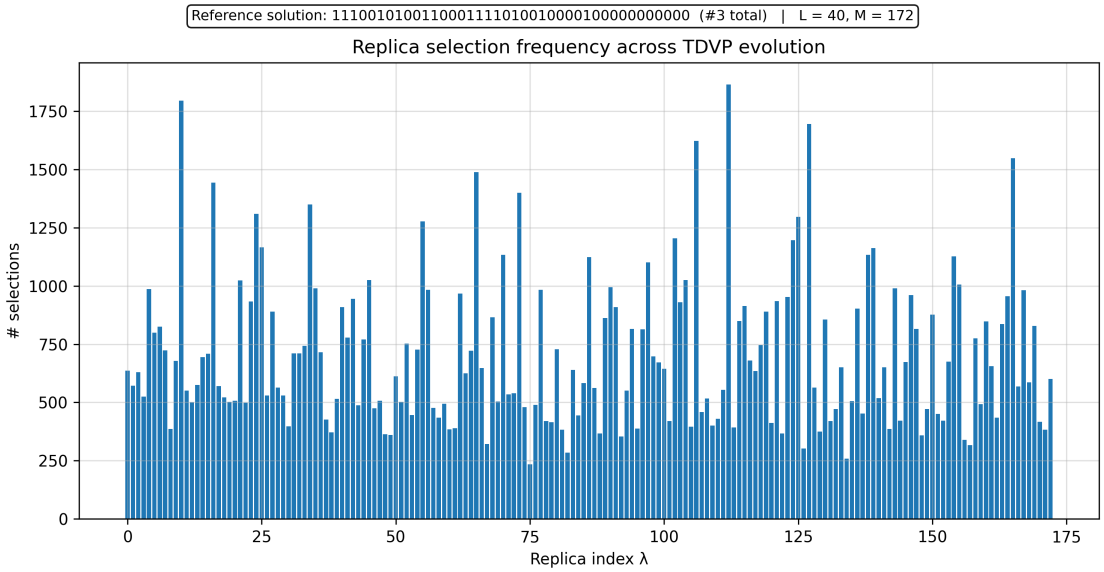


(a) Aggressively compressed two-step frontier.

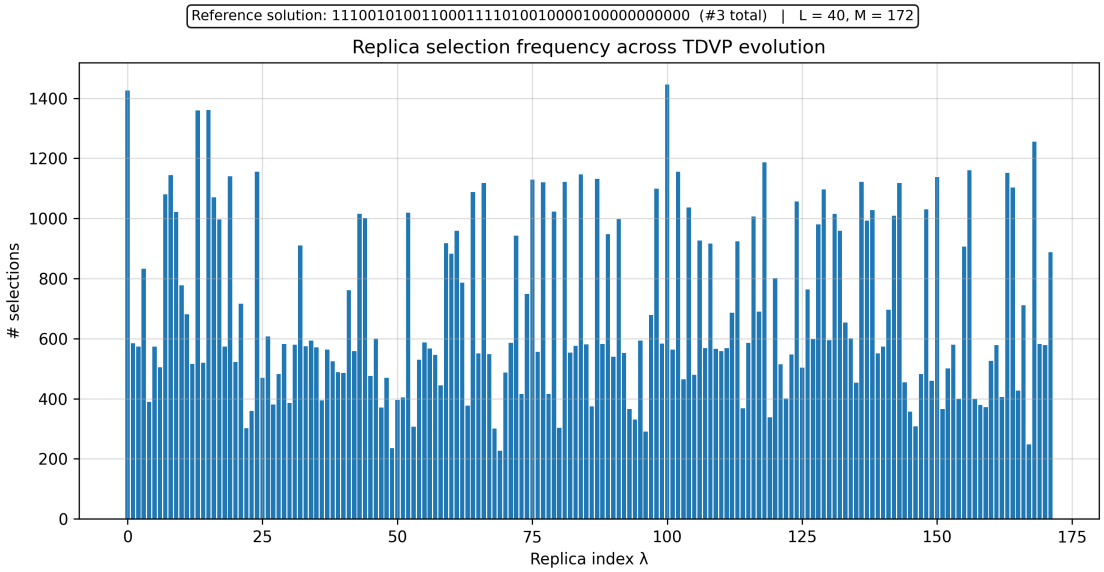


(b) Baseline one-step frontier.

Figure 7.29: Energy expectation $\langle\Psi|H|\Psi\rangle$ for the $L = 40$, $M = 172$ instance. With aggressive compression (a) the energy decays all the way to zero, while the one-step frontier (b) relaxes only to a metastable plateau above the ground-state energy.

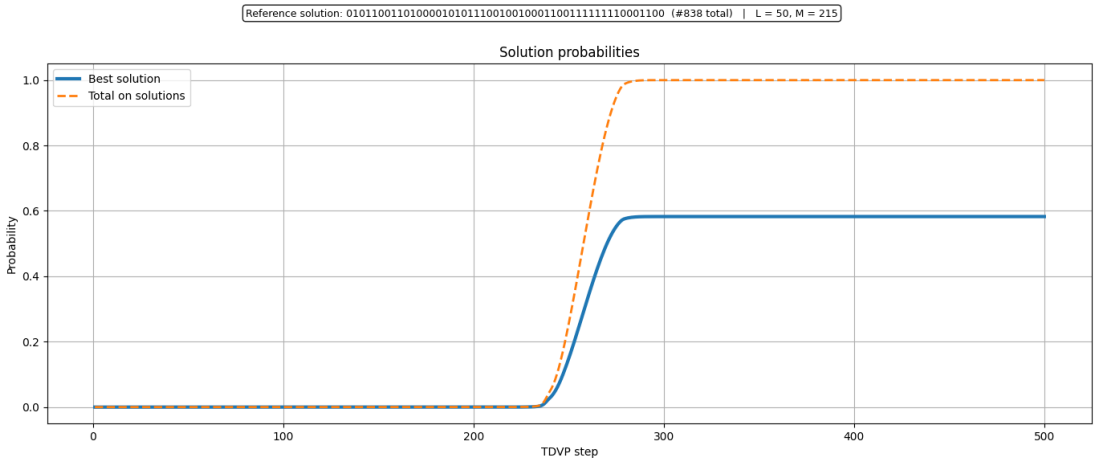


(a) Aggressively compressed two-step frontier.

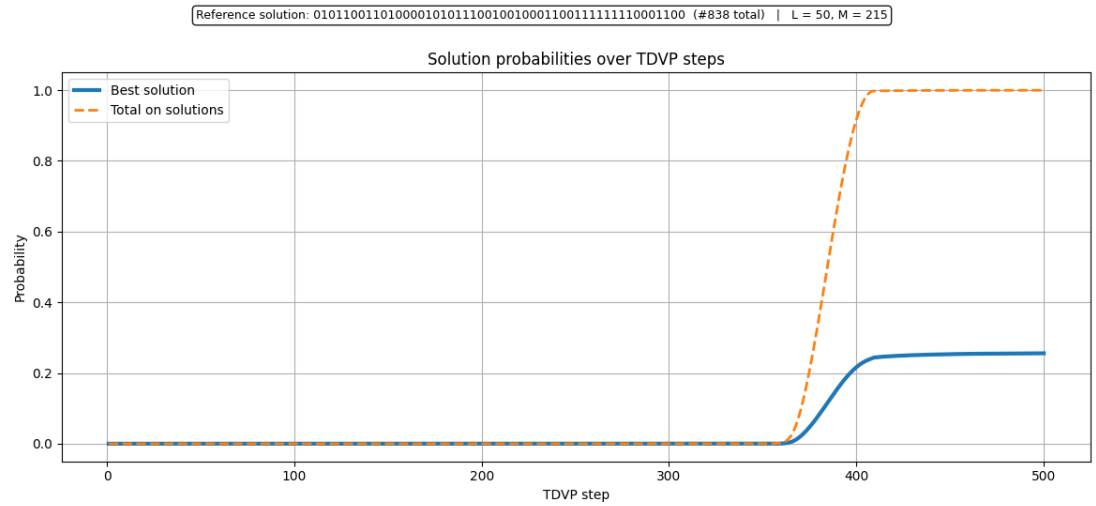


(b) Baseline one-step frontier.

Figure 7.30: Replica-selection frequency for the $L = 40$, $M = 172$ instance. Each bar counts the number of TDVP steps in which replica λ was included in the active set. Aggressive compression (a) leads to a more uniformly exploited set of replicas, whereas the one-step frontier (b) leaves more directions effectively unused.

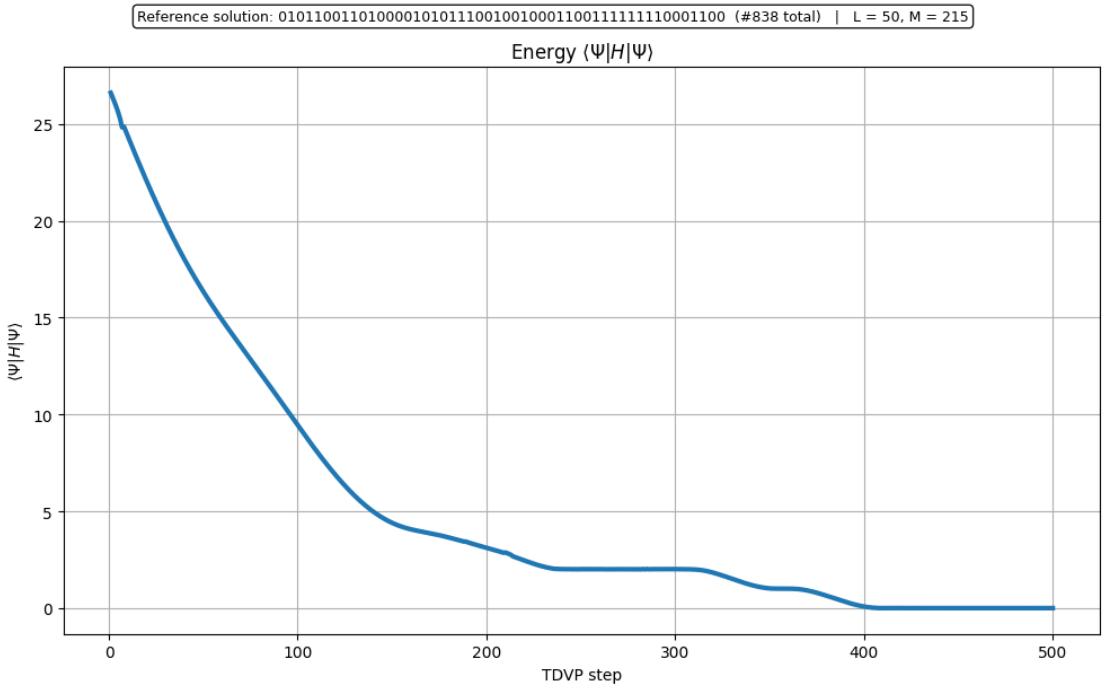


(a) Aggressively compressed two-step frontier.

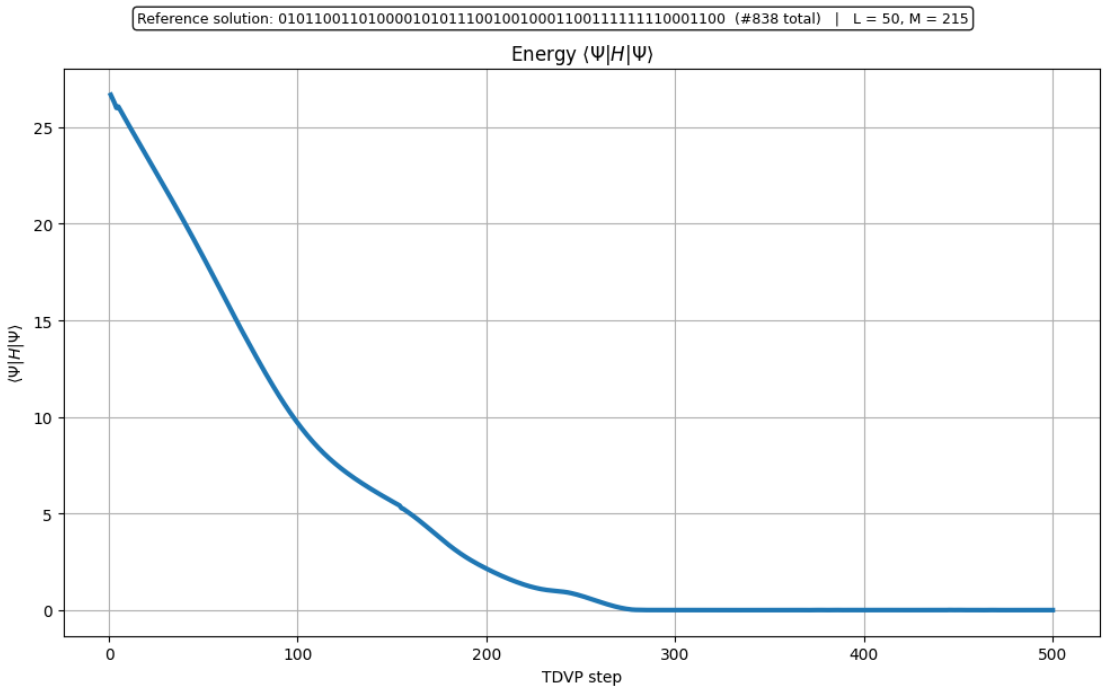


(b) Baseline one-step frontier.

Figure 7.31: Solution probabilities for the solved $L = 50$, $M = 215$ instance. Solid lines: probability of the best solution; dashed lines: total probability on all satisfying assignments. Both pipelines succeed, but the aggressively compressed frontier freezes earlier and with a higher asymptotic weight on the best solution.

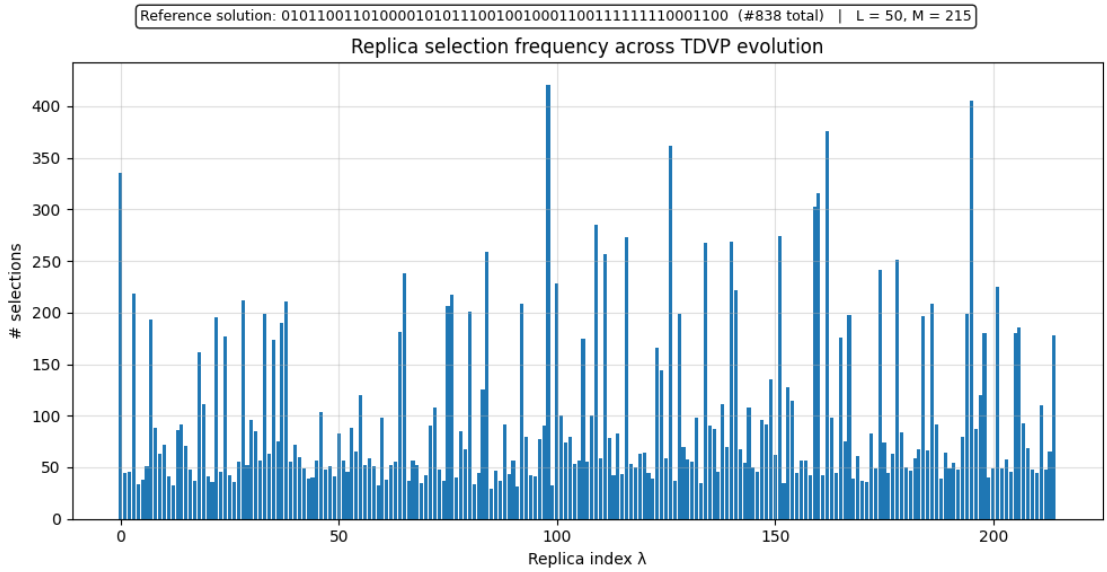


(a) Softly compressed frontier (for reference).

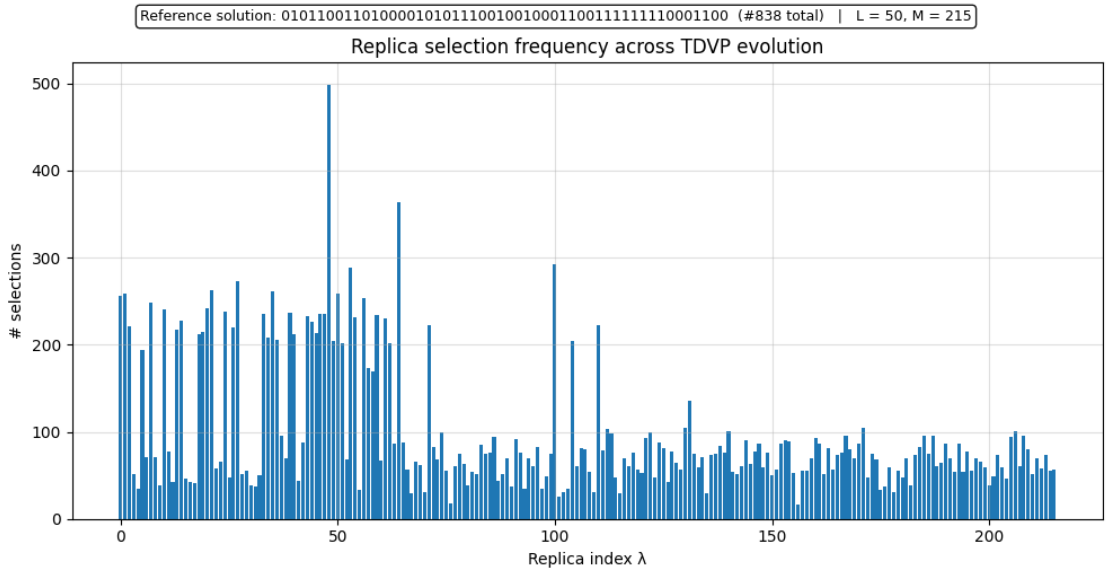


(b) Aggressively compressed two-step frontier.

Figure 7.32: Energy expectation $\langle\Psi|H|\Psi\rangle$ for the solved $L = 50$ instance. Both compressed pipelines reach the ground-state energy, but the aggressive scheme (b) does so earlier and with a steeper, more decisive decay. The baseline one-step frontier follows a trend similar to (a), with a slightly slower relaxation.



(a) Baseline one-step frontier.



(b) Aggressively compressed two-step frontier.

Figure 7.33: Replica-selection frequency for the solved $L = 50$, $M = 215$ instance. Each bar counts how many TDVP steps a given replica belongs to the active set. In the aggressively compressed case the pattern of "few very active, many occasionally active" replicas is more pronounced than in the baseline run, but virtually no direction is completely dormant.

7.2.4 A failed $L = 50$ instance: when all pipelines stall

To probe the limits of frontier compression, we now turn to a hard $L = 50$, $M = 215$ instance from the same 1000-step batch of Sec. 7.1.5, for which *none* of our pipelines finds a satisfying assignment. We again compare: (i) the uncompressed one-step frontier, (ii) the softly compressed frontier, and (iii) the aggressively compressed frontier , all followed by TDVP with a $20\%(M+1)$ active-set fraction.

Frontier growth and manifold size. The frontier dynamics during ITP (not shown) is quantitatively similar to the $L = 50$ solved instance discussed in the previous subsection. The uncompressed two-step frontier reaches a manifold size of order 10^4–10^5, the soft compression keeps an intermediate number of replicas of order 10^3, and the aggressive compression enforces $D_{\text{final}} = M + 1$ at the end of the second step. Thus, as before, the two compression pipelines differ mainly in how aggressively they prune the raw frontier, not in the basic pattern of growth.

Solution probabilities and energy decay. Figure 7.34 shows the probability on satisfying assignments as a function of TDVP step for the three pipelines, while Fig. 7.35 displays the corresponding energy trajectories.

For the uncompressed and aggressively compressed frontiers, Figs. 7.34(a)–(b), the solution probability exhibits a small peaks at intermediate times and then decays back to essentially zero. The TDVP flow seems to briefly approach the true ground-state region before relaxing into a metastable basin with negligible overlap on any exact solution.

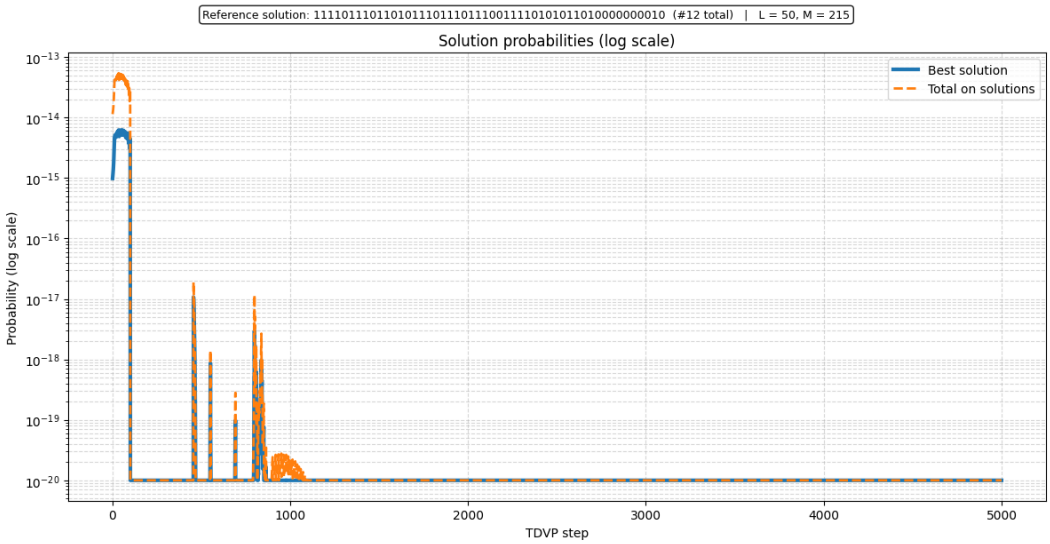
The softly compressed frontier, Fig. 7.34(c), reaches a noticeably larger maximum total solution probability than the other two. However, it is unclear whether it fully converges, and verifying this would require many more steps, which is infeasible with our current implementation.

The energy plots in Fig. 7.35 clarify the nature of these failures. For the uncompressed and aggressively compressed frontiers [Figs. 7.35(a)–(b)] the energy initially decreases rapidly and then saturates to a nonzero plateau, indicating convergence to a low-lying but metastable state. In contrast, the soft-compression run [Fig. 7.35(c)] shows a much more dramatic behaviour: after an initial decay the energy climbs back up to a relatively high value and only then starts slowly decreasing again, without ever approaching zero. In other words, the configuration that achieves the largest (still tiny) solution probability is associated with a much higher energy plateau than the metastable minima reached by the other two pipelines.

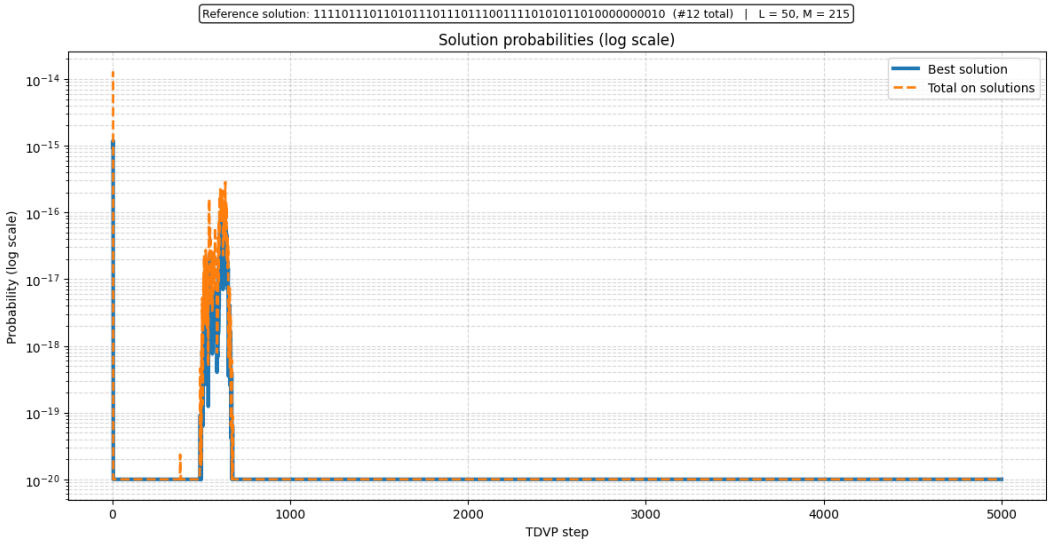
Replica-selection patterns. Finally, Fig. 7.36 compares the replica-selection frequencies. For the uncompressed frontier the active set is spread thinly over a very large number of replicas, with many directions touched only rarely. The aggressively compressed manifold, by construction, contains only $M+1$ replicas and exhibits a more structured pattern, but several replicas are still selected relatively infrequently, consistent with the system being stuck in a single basin.

The soft-compression run shows an even more uneven distribution: a handful of replicas are selected very often, while a large tail of states is barely used. Combined with the energy behaviour, this suggests that the enlarged manifold gives TDVP enough freedom to wander away from low-energy regions into a broad high-energy "plateau" where the sensitivity-based selection keeps revisiting a small set of replicas without ever building significant solution weight.

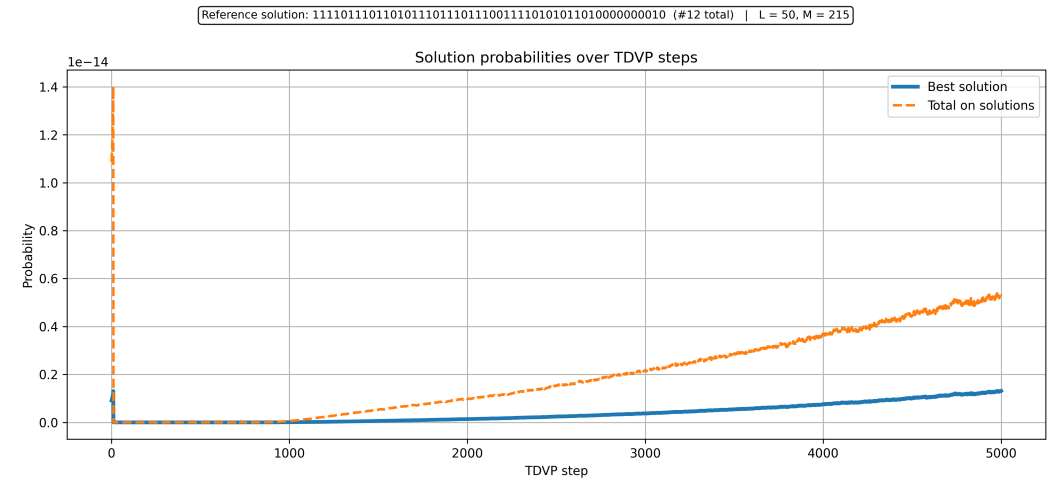
In summary, this $L = 50$ instance illustrates a genuine limitation of our current sensitivity-driven compression strategy. At this scale, and for instances with very few ground states, the variational manifold and our local TDVP updates appear insufficient to escape deep glassy basins, despite the structural advantages observed at smaller L and on the "solved" $L = 50$.



(a) Uncompressed frontier.

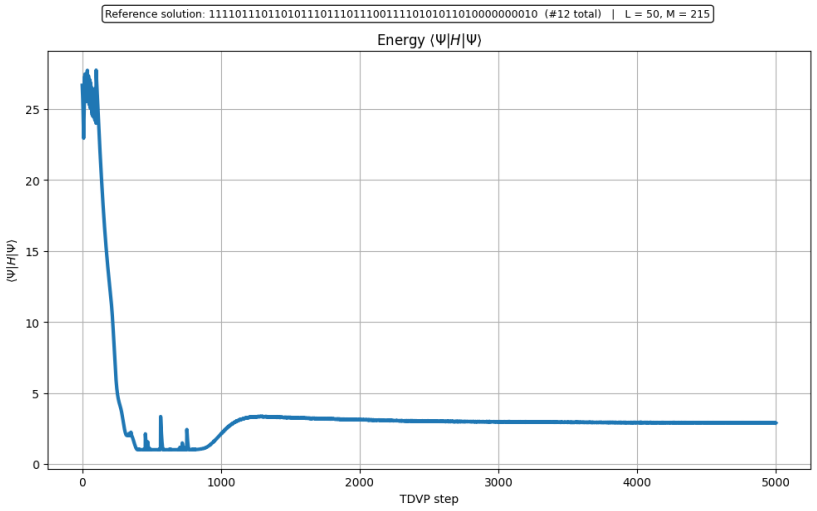


(b) Aggressively compressed frontier.

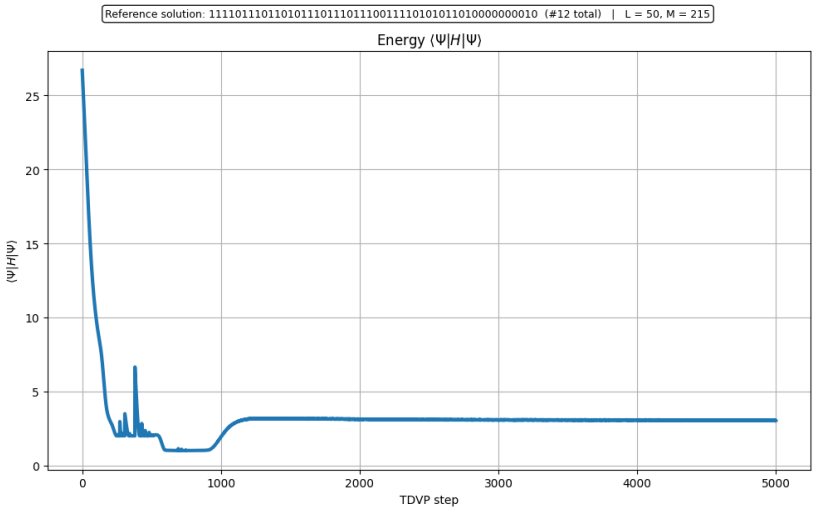


(c) Softly compressed frontier.

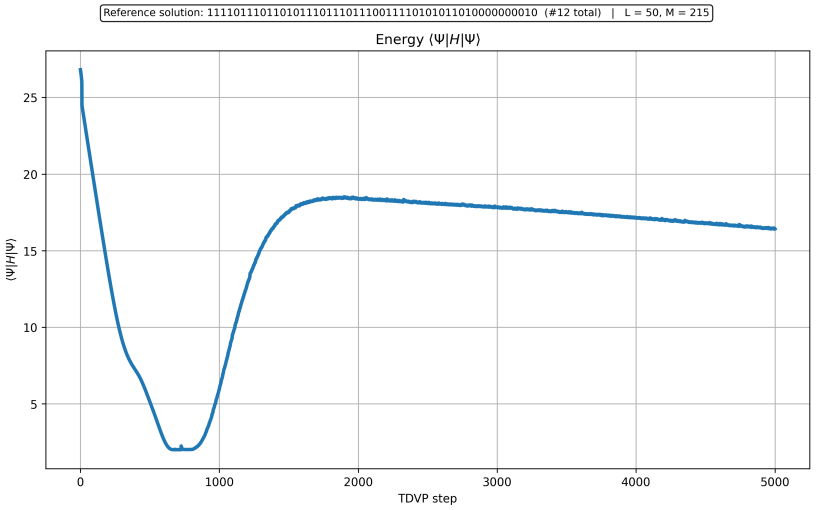
Figure 7.34: Solution probabilities for a failed $L = 50$ instance. Total probability on satisfying assignments (dashed) and probability of the best solution (solid) as a function of TDVP step for the uncompressed frontier (a), the aggressively compressed frontier (b), and the softly compressed frontier (c). None of the pipelines develops macroscopic solution weight; the soft compression reaches the largest peak probability, but it remains tiny in absolute terms.



(a) Uncompressed frontier.

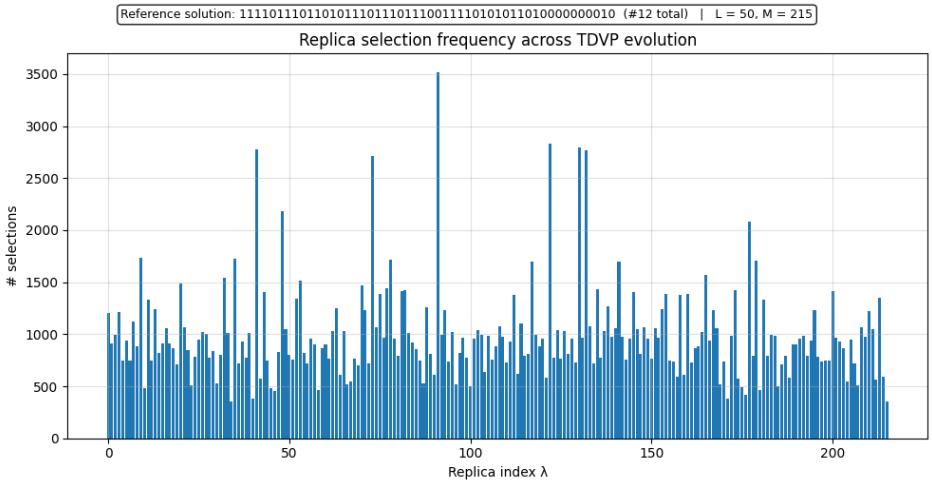


(b) Aggressively compressed frontier.

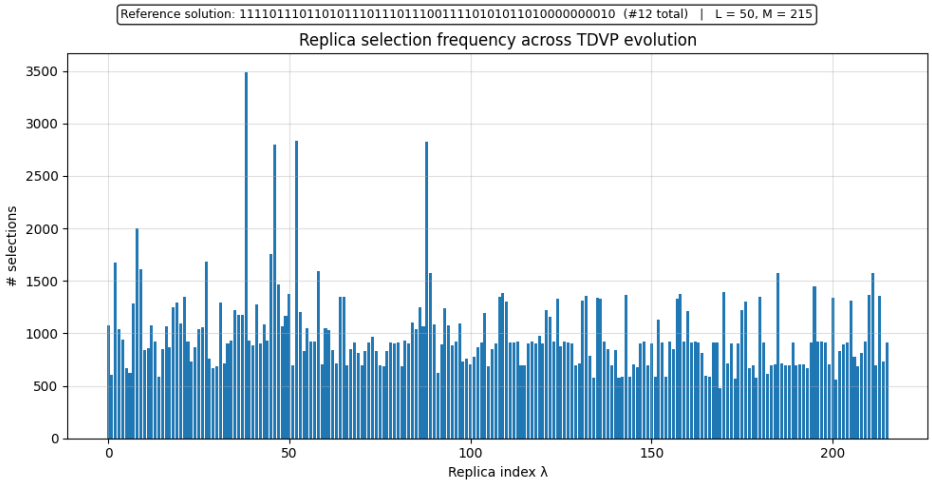


(c) Softly compressed frontier.

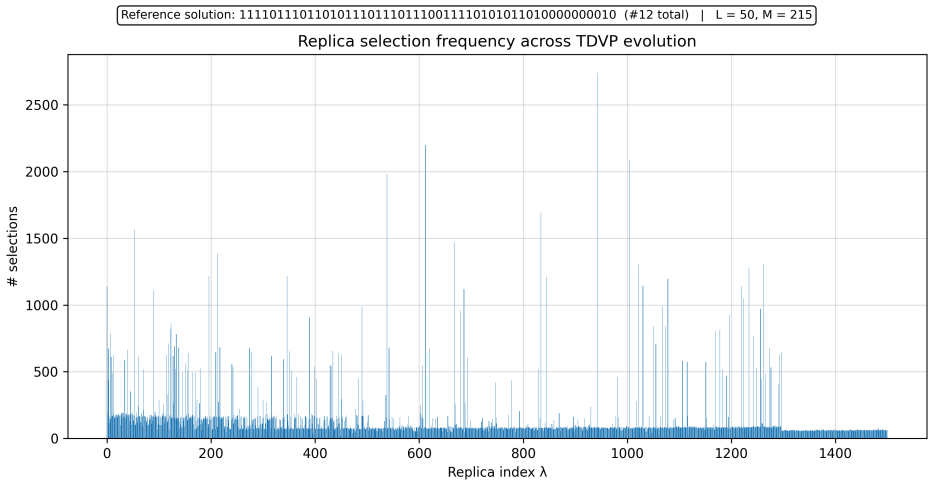
Figure 7.35: Energy trajectories for the failed $L = 50$ instance. Expectation value $\langle\Psi|H|\Psi\rangle$ versus TDVP step for the three pipelines. The uncompressed and aggressive runs relax to nonzero plateaux, while the soft-compression run first drops to a relatively low energy, then *climbs* to a much higher value before slowly decreasing again, signalling a qualitatively different (and higher-energy) metastable state.



(a) Uncompressed frontier.



(b) Aggressively compressed frontier.



(c) Softly compressed frontier.

Figure 7.36: Replica-selection frequencies for the failed $L = 50$ instance. Histogram of how many TDVP steps each replica belongs to the active set, for the uncompressed, aggressively compressed, and softly compressed frontiers. The soft-compression manifold shows a particularly strong imbalance, with a small number of replicas dominating the updates while many are almost never used.

7.2.5 Interpretation of the $L = 50$ failure

The failed $L = 50$ instance should be viewed as representative of the whole subset of $L = 50$ formulas that remain unsolved after 1000 TDVP steps in the baseline one-step pipeline of Sec. 7.1.5. These formulas are generated in the hard-SAT regime, at clause densities close to the critical threshold, and form only a relatively small fraction of the 100-instance batches considered at each size; the majority of instances (including many at $L = 50$) are solved by the baseline scheme without any compression. For all *unsolved* instances that we could check in this hard subset, we have not found a single case in which either the softly or the aggressively compressed frontier turns a failed run into a successful one, even after scanning over the regularization parameters of the metric S and reducing the TDVP time step γ within the stability window of the algorithm. In this sense, the behavior in Figs. 7.34 and 7.35 is characteristic of the $L = 50$ formulas that remain unsolved within our time budget, rather than a pathological outlier.

A first, conservative interpretation is that at $L = 50$ we are simply under-sampling the TDVP dynamics. In the $L = 30$ and $L = 40$ case studies, the onset of the "solving" transition—where the solution probability suddenly becomes macroscopic—systematically shifts to later and later TDVP times as L grows. Nothing in our data rules out the possibility that the same phenomenon persists at $L = 50$ but on even longer time scales. Consistent with this picture, the softly compressed $L = 50$ frontier shows a very slowly increasing but strictly positive trend in the total solution probability [Fig. 7.34(c)], even though the absolute values remain tiny at the time where we stop the run.

Pushing the soft-compression runs much further in time is, however, computationally prohibitive with our current implementation. For the failed $L = 50$ instance, the softly compressed frontier reaches $D_{\text{final}} \sim 10^3$ replicas, i.e. a manifold size several times larger than $M{+}1$. At this point, the dominant cost in each TDVP step comes from building and regularizing the effective metric S_{eff} and from solving the corresponding linear system with MINRES–QLP, together with the repeated evaluation of the sensitivity scores s_λ for active-set selection. Both operations become serious bottlenecks when D enters the 10^3 range, making it unfeasible to extend the soft runs well beyond 1000 TDVP steps with our current implementation. The aggressively compressed frontier avoids this state-space blow-up and thus remains computationally cheaper, but in the failed $L = 50$ instance it no longer exhibits the clear positive trend in solution probability seen at smaller L.

We emphasize that we have restricted ourselves to a single family of compression protocols: two-step ITP growth blocks followed by compression at every step back to a manifold of size $M{+}1$, with either the soft or the aggressive variant of sensitivity-driven pruning. Within each TDVP update, we also fix a given choice of active-set selection fraction. In principle, one could imagine overcoming some of the $L = 50$ failures by systematically varying these hyperparameters—e.g. increasing the TDVP selection fraction, relaxing the compression target below $M{+}1$, changing the number of ITP steps between pruning events, or allowing the frontier to grow to larger intermediate dimensions before compression. Exploring this high-dimensional space of schedules and selection fractions would amount to a large-scale benchmarking and optimization effort that we have *not* carried out here, both because it would substantially increase the total running time of our study and because our primary goal is to evaluate whether this compression protocol is promising at all. A systematic optimization of these design choices, starting from the present implementation, is therefore a natural direction for future work.

Beyond these finite-time and computational considerations, there is also a more structural limitation tied to the underlying energy landscape. In the glassy phase of random constraint-satisfaction problems and mean-field spin glasses, the number of metastable states (or clusters of solutions) grows exponentially with system size, as described by the complexity or configurational-entropy function $\Sigma(e) = \frac{1}{L} \log \mathcal{N}(e)$, which remains positive over a finite energy

window.[1] Working at clause densities close to the satisfiability threshold, our instances are precisely in this regime, where an exponential number of low-lying metastable basins competes with a very small set of true ground states. By construction, our procedures only retain $\mathcal{O}(M)$ product states; even after multi-step growth and sensitivity-driven pruning, the effective manifold explores a vanishingly small fraction of the relevant configurations. It is therefore natural that, as L increases, the ability of the TDVP flow on this low-dimensional manifold to mimic the true imaginary-time dynamics and to escape deep metastable basins becomes increasingly limited.

Finally, the near-exponential proliferation of frontier states with the number of ITP steps (Sec. 4.6 and App. A.2) implies that any attempt to push the method to larger sizes or deeper frontiers will run into a severe state-space and runtime wall unless the underlying implementation is substantially optimized. More efficient linear solvers, better exploitation of sparsity in S_{eff} and in the Hamiltonian, GPU-accelerated routines for both the TDVP update and the sensitivity-based selection, and a systematic benchmarking of different active-set fractions and compression schedules could all change the practical regime of accessibility. These directions constitute a natural line of future work starting from this thesis.

With these caveats in mind, the main takeaway of this section remains positive. Sensitivity-driven frontier compression *does* extend the reach of our algorithm in the hard-SAT regime: it allows us to solve a subset of random 3-SAT instances near the critical clause density that remain unsolved within the same time budget when using the uncompressed one-step frontier, while still working with a relatively small number of replicas and keeping the growth of the variational manifold under tight control. At the same time, it is important to stress that the underlying ITP–TDVP scheme without compression already solves a large fraction of random instances, including many with $L = 50$, within a relatively small number of iterations; compression should therefore be viewed as a structural refinement designed to speed up convergence and to recover part of the *relatively small* set of hard-regime instances that remain unsolved after 1000 steps, rather than as an indispensable ingredient for all formulas. From the viewpoint of state-of-the-art classical SAT solvers, problem sizes such as $L = 50$ are of course modest, and our method is not intended to be competitive as a large-scale practical solver. In the context of deterministic variational and quantum-inspired algorithms, however, reaching $L = 50$ at critical density with a full ITP–TDVP pipeline and explicit metric handling is already sufficient to expose genuine scaling bottlenecks rather than toy-model artefacts. At $L = 50$ and beyond, the combination of glassy landscape complexity and computational cost therefore currently prevents a definitive assessment of the asymptotic performance of this strategy on the most challenging instances.

[1] See, e.g., Refs. [1, 2, 7, 8] for classical analyses of random k-SAT and p-spin models.

Conclusions

In this thesis we have introduced and tested a deterministic, variational approach to random 3-SAT near the SAT/UNSAT threshold, based on imaginary-time dynamics and the time-dependent variational principle. Each instance is mapped to a classical spin Hamiltonian, and the search for satisfying assignments is recast as an approximate ground-state problem on a structured manifold of product states generated by frontier ITP. On top of this manifold, an imaginary-time TDVP flow refines both local parameters and amplitudes, while sensitivity-based selection and compression protocols are used to control the manifold growth and focus computational effort on the most relevant replicas.

Benchmark results on random instances at critical clause density show a mixed but overall encouraging picture. On the one hand, the ITP–TDVP pipeline without compression already solves a large fraction of instances up to $L = 50$, demonstrating that imaginary-time dynamics and variational geometry can be turned into an effective deterministic heuristic in the hard-SAT regime. On the other hand, the same experiments clearly expose the main scaling bottlenecks: the near-exponential proliferation of frontier states with ITP depth, the limited expressive power of an $\mathcal{O}(M)$-dimensional ansatz in a glassy landscape with exponentially many relevant configurations, and the rapidly increasing cost and numerical fragility of the TDVP metric. Sensitivity-driven compression strategies partially mitigate these issues by trading manifold size for depth and by recovering some hard instances that the baseline scheme fails to solve.

These findings suggest several natural directions for future work. Algorithmically, a more systematic benchmark of compression strategies is needed, exploring different sensitivity measures, diversity filters, and selection fractions, and scanning over frontier depths and regularization schemes for the metric in order to quantify the best trade-offs between expressivity, stability, and cost. On the implementation side, further gains are likely to come from optimized linear algebra, better exploitation of sparsity, and GPU-accelerated routines for both TDVP updates and sensitivity evaluation, which would relax current runtime and memory limitations and allow larger systems, deeper frontiers, and higher active-set fractions to be explored. Conceptually, the framework developed here could be extended to more expressive variational families (for example, tensor-network-inspired or weakly entangled ansätze), alternative imaginary-time discretizations, and other classes of constraint-satisfaction problems. In the longer term, these extensions may help clarify whether suitably compressed variational imaginary-time dynamics can lead to scalable, structure-aware heuristics for a broader range of NP-hard optimization problems and how such classical methods might interface with or inspire future quantum and hybrid algorithms.

Appendix A

A.1 Frontier-based ITP: implementation details

In Sec. 4.4 we introduced the frontier-based imaginary-time propagation (ITP) scheme acting on the variational superposition of product states $|\rho_\lambda\rangle$. In this appendix we summarize the concrete implementation used in the numerics, highlighting the bitmask encoding for product states and providing pseudocode for one frontier-based ITP step and for the outer ITP loop.

A.1.1 Efficient bitmask encoding

Rather than storing each $|\rho_\lambda\rangle$ as the full list of angles $\{\theta_k^\lambda\}_{k=0}^{L-1}$, we store two L-bit masks per state:

- `fixed_mask`$_\lambda[q] = 1$ if qubit q is already "pinned" by previous clauses (so $\theta_q^\lambda \in \{0, \pi/2\}$), and 0 if it is still unfixed at $\theta_q^\lambda = \pi/4$;
- `value_mask`$_\lambda[q] = 0$ if the pinned value is $|0_q\rangle$ ($\theta = 0$), and 1 if it is $|1_q\rangle$ ($\theta = \pi/2$).

This encoding lets us:

- reconstruct $\theta_q^\lambda \in \{0, \pi/2, \pi/4\}$ on demand (via `decode_masks_to_theta_matrix`);
- compare two product states for equality in constant time (two 64-bit integer comparisons);
- detect contradictions instantly (trying to pin the same qubit to 0 and 1);
- merge clones efficiently by summing their amplitude contributions.

A.1.2 Frontier-based ITP step

The frontier-based variant of the imaginary-time propagation concentrates the "create-or-merge" logic on a dynamically selected subset $\mathcal{F}^{(k)} \subseteq \mathcal{S}^{(k)}$ (the frontier), while older states are updated in a cheaper "no-new-state" mode. The current state pool is denoted $\mathcal{S}^{(k)}$, with amplitudes $a_\lambda^{(k)}$ and bitmasks encoding each product state. The frontier mask $\mathcal{F}^{(k)}$ is implemented as a boolean array pointing to indices in the state pool.

The core routine, `itp_step_optimized`, realizes one ITP step $k \to k{+}1$ in the spirit of the three-phase description given in Sec. 4.4. Its inputs are:

- `state_fixed_masks[i]`, `state_value_masks[i]`: the bitmasks encoding the i-th product state in $\mathcal{S}^{(k)}$;
- `a_vector[i]`: the corresponding amplitudes $a_\lambda^{(k)}$;
- `frontier_mask[i]`: a boolean flag selecting the frontier $\mathcal{F}^{(k)}$;
- `clauses`: the list of 3-SAT clauses (literals) defining $\hat{H}$;
- $d\tau$ and the problem size L.

Each clause application is handled by `apply_clause_to_state_optimized`, which enforces the local projector action of Eq. (4.26), computes the trigonometric weight $w_c(\lambda)$ (cf. Eq. (4.27)), constructs the child masks, and returns the unnormalized amplitude increment $\Delta a^{(c)}_{\lambda\to\lambda'} = -d\tau\, a^{(k)}_\lambda w_c(\lambda)$.

Algorithm 1 summarizes the logic of one step in pseudocode form. Phase 1 copies all states and adds the identity contribution; Phase 2 performs the "create-or-merge" expansion of the frontier; Phase 3 performs a "merge-only" expansion of the non-frontier states; and finally all amplitudes are globally normalized and the new frontier is identified as the set of states that were created for the first time in Phase 2.

Algorithm 1 Frontier-based ITP step $k \to k{+}1$

Require: Current pool $\mathcal{S}^{(k)}$ (bitmasks and amplitudes), frontier mask $\mathcal{F}^{(k)}$, clause list, time step $d\tau$, size L.
Ensure: Updated pool $\mathcal{S}^{(k+1)}$, amplitudes $a^{(k+1)}_\lambda$, new frontier $\mathcal{F}^{(k+1)}$.

Copy all states in $\mathcal{S}^{(k)}$ into a working pool.
Initialize provisional amplitudes $\tilde{a}^{(k+1)}_\lambda \leftarrow a^{(k)}_\lambda$ for all $\lambda \in \mathcal{S}^{(k)}$ (identity channel).

Frontier phase
for each parent $\lambda \in \mathcal{F}^{(k)}$ **do**
  for each clause c in the clause list **do**
    Apply c to $|\rho_\lambda\rangle$ and generate all children λ'.
    for each child λ' **do**
      if branch is incompatible **then**
        Discard λ'.
      else if λ' already exists in the working pool **then**
        Update its provisional amplitude $\tilde{a}^{(k+1)}_{\lambda'} \leftarrow \tilde{a}^{(k+1)}_{\lambda'} + \Delta a^{(c)}_{\lambda\to\lambda'}$.
  (clone merging).
      else
        Append λ' as a new state in the working pool.
        Initialize its provisional amplitude $\tilde{a}^{(k+1)}_{\lambda'} \leftarrow \Delta a^{(c)}_{\lambda\to\lambda'}$.
        Mark λ' as "new" in this step.

Non-frontier phase
for each parent $\lambda \in \mathcal{S}^{(k)} \setminus \mathcal{F}^{(k)}$ **do**
  for each clause c in the clause list **do**
    Apply c to $|\rho_\lambda\rangle$ and generate all children λ'.
    for each child λ' **do**
      if branch is incompatible **then**
        Discard λ'.
      else if λ' matches an existing state in the working pool **then**
        Update its provisional amplitude $\tilde{a}^{(k+1)}_{\lambda'} \leftarrow \tilde{a}^{(k+1)}_{\lambda'} + \Delta a^{(c)}_{\lambda\to\lambda'}$.
      else
        Discard λ'.
  (non-frontier parents cannot enlarge the basis).

Normalize all provisional amplitudes $\tilde{a}^{(k+1)}_\lambda$ to obtain $a^{(k+1)}_\lambda$.
Define the new frontier $\mathcal{F}^{(k+1)}$ as the set of states that were marked as "new" during the frontier phase.

A.1.3 Outer ITP loop

The outer loop initializes the single unfrozen state and applies the frontier-based ITP step a fixed number of times. Bitmasks are finally decoded into angles $\theta_q^\lambda \in \{0, \pi/2, \pi/4\}$ by `decode_masks_to_theta_matrix`. Algorithm 2 summarizes this driver routine.

Algorithm 2 Outer frontier-based ITP loop

Require: Number of qubits L, clause list, time step $d\tau$, number of ITP steps n_{steps}.

Ensure: Final amplitudes $a_\lambda^{(n_{\text{steps}})}$ and angles θ_q^λ for all states in $\mathcal{S}^{(n_{\text{steps}})}$.

1: Initialize the state pool $\mathcal{S}^{(0)}$ with a single fully unfrozen product state: all masks equal to zero, amplitude $a_0^{(0)} \leftarrow 1$.
2: Initialize the frontier $\mathcal{F}^{(0)} \leftarrow \mathcal{S}^{(0)}$.
3: **for** $k = 0, \ldots, n_{\text{steps}} - 1$ **do**
4: Apply Algorithm 1 to perform one frontier-based ITP step $k \to k{+}1$ and update $(\mathcal{S}^{(k+1)}, a_\lambda^{(k+1)}, \mathcal{F}^{(k+1)})$.
5: Decode the final bitmasks into angles $\theta_q^\lambda \in \{0, \pi/2, \pi/4\}$ via `decode_masks_to_theta_matrix`.
6: **return** the final amplitude vector $\{a_\lambda^{(n_{\text{steps}})}\}$ and the corresponding angle matrix $\{\theta_q^\lambda\}$.

A.2 Mean-field growth of the frontier manifold

In this appendix we refine the qualitative discussion of Sec. 4.6 by giving a fully explicit *mean-field* model for how the number of distinct product states

$$D(k) + 1 = |\mathcal{S}^{(k)}|$$

grows with the number of ITP steps k, the number of variables L, and the number of clauses M, for the frontier-based imaginary-time propagation (ITP) scheme of Sec. 4.4. The mean-field recursions derived below are exactly the ones used to generate the curves shown in Fig. 4.4(b,c) of the main text.

We do *not* re-derive the frontier algorithm itself: we assume the reader is familiar with the notation used there, in particular with

- the pool $\mathcal{S}^{(k)}$ of distinct product states after k ITP steps,
- the frontier $\mathcal{F}^{(k)} \subseteq \mathcal{S}^{(k)}$ of states first created at step k,
- the rule that only $\mathcal{F}^{(k)}$ is allowed to create new product states at step k+1.

Our goal is to build a stochastic model that predicts, on average, how many *new* states are generated at each step and how the total number of distinct states $D(k) + 1$ saturates to a finite plateau, in a way that mirrors the layer-resolved Python implementation used to generate the mean-field curves in Fig. 4.4(b,c).

Throughout this appendix we work purely at the level of the *combinatorics* of state creation. Amplitudes $a_\lambda^{(k)}$ and the normalization of $(\mathbb{I} - d\tau H)$ are irrelevant here and will be ignored.

Layer-resolved state counting

Each product state in the ansatz can be characterized by how many spins are already pinned to σ^z eigenvalues (i.e. at Bloch angles $\theta = 0$ or $\theta = \pi/2$) and how many remain unfrozen at $\theta = \pi/4$. We therefore introduce a layer index $k = 0,1,\ldots,L$:

- the layer k consists of all product states with exactly k pinned spins and $L - k$ unfrozen spins.

The total number of distinct configurations in layer k is

$$S_k = \binom{L}{k} 2^k, \tag{A.1}$$

because we choose which k sites are pinned ($\binom{L}{k}$ choices), and for each pinned site we choose between $|0\rangle$ and $|1\rangle$ (2^k choices). One can check that $\sum_{k=0}^{L} S_k = (1+2)^L = 3^L$, consistent with having three logical statuses per spin (unfrozen, pinned to 0, pinned to 1).

We now track, for each layer k and each ITP step t:

- $F_k(t)$ = expected frontier size in layer k at step t, i.e. the expected number of states with k pinned spins that are *first created* at step t;
- $D_k(t)$ = expected total number of distinct states with k pinned spins that have been discovered up to and including step t.

We initialize at imaginary-time step $t = 0$ with a single unfrozen state:

$$F_0(0) = 1, \quad D_0(0) = 1, \qquad F_k(0) = 0, \quad D_k(0) = 0 \ \text{ for } k \geq 1. \tag{A.2}$$

At step t, the total frontier size and the total number of distinct states are then

$$N(t) := \sum_{k=0}^{L} F_k(t), \qquad D_{\text{tot}}(t) := \sum_{k=0}^{L} D_k(t). \tag{A.3}$$

Note that $D_{\text{tot}}(t)$ includes the initial $t = 0$ state, so the pool size after t steps is $|\mathcal{S}^{(t)}| \approx D_{\text{tot}}(t)$.

A.2.1 Local clause statistics within a fixed layer

Consider a typical *frontier parent* in layer k at step $t-1$. It has k pinned spins and $L-k$ unfrozen spins. A 3-SAT clause acts on three distinct variables drawn uniformly from $\{1,\dots,L\}$. We classify the local intersection by the pair (u,f), where u is the number of unfrozen and f the number of frozen spins touched by the clause:

- $(u,f)=(3,0)$: three unfrozen, zero frozen,
- $(u,f)=(2,1)$: two unfrozen, one frozen,
- $(u,f)=(1,2)$: one unfrozen, two frozen,
- $(u,f)=(0,3)$: zero unfrozen, three frozen.

The corresponding combinatorial weights are

$$N_{3,0}(k)=\binom{L-k}{3}, \tag{A.4}$$

$$N_{2,1}(k)=\binom{k}{1}\binom{L-k}{2}, \tag{A.5}$$

$$N_{1,2}(k)=\binom{k}{2}\binom{L-k}{1}, \tag{A.6}$$

$$N_{0,3}(k)=\binom{k}{3}, \tag{A.7}$$

and $\binom{L}{3}=N_{3,0}(k)+N_{2,1}(k)+N_{1,2}(k)+N_{0,3}(k)$ is the total number of three-site subsets.

We now introduce a compatibility parameter

$$\rho\in[0,1], \tag{A.8}$$

defined as the probability that a literal acting on a *pinned* spin requests the same computational-basis value as the one already pinned.

If one considers a completely unbiased clause ensemble where positive and negative literals occur with equal probability and are independent of the current state, a single literal on a pinned spin is true with probability $\frac{1}{2}$: this would give a natural "bare" value $\rho_{\mathrm{bare}}=\frac{1}{2}$. However, in the actual frontier-based ITP dynamics the clauses are not applied to independent random spin configurations: the frontier states have already survived several imaginary-time steps, correlations between different clauses build up, and configurations that are strongly incompatible with many clauses tend to disappear early. As a result the effective probability that a clause which touches pinned spins produces a *useful* child is substantially lower than the naive ρ_{bare}.

We therefore treat ρ as a *phenomenological* parameter that summarizes, in a single number, all those effects that the present mean-field model does not resolve explicitly:

- the fact that the ITP clause set is *fixed* and reused at every step, rather than freshly resampled at each application;
- nontrivial correlations between different pinned spins in the frontier states, which make some parents much "harder" to extend than others;
- the strong concentration of candidate children in a small region of the layer space, which effectively enhances clone production.

In Sec. A.2.4 below we determine this effective value ρ_{eff} directly from small system sizes, finding a nearly L–independent estimate $\rho_{\text{eff}} \approx 0.1$ that yields an excellent quantitative match with the measured frontier growth, as shown in Fig. 4.4.

With this in mind, and under a mean-field independence ansatz, we classify the local effect of a clause on a parent in layer k as follows:

- a (3,0) clause is always compatible and freezes three new spins,
- a (2,1) clause is compatible with probability ρ and freezes two new spins when compatible,
- a (1,2) clause is compatible with probability ρ^2 and freezes one new spin when compatible,
- a (0,3) clause freezes no new spins and is irrelevant for growth.

We therefore define

$$p_0(k; L, \rho) = \frac{N_{3,0}(k)}{\binom{L}{3}}, \tag{A.9}$$

$$p_1(k; L, \rho) = \frac{\rho\, N_{2,1}(k)}{\binom{L}{3}}, \tag{A.10}$$

$$p_2(k; L, \rho) = \frac{\rho^2\, N_{1,2}(k)}{\binom{L}{3}}, \tag{A.11}$$

as the probabilities that a random clause produces a locally valid child with $\Delta k = 3,2,1$ new pins, respectively, when acting on a parent in layer k. The total local success probability is

$$p_{\text{loc}}(k; L, \rho) = p_0(k; L, \rho) + p_1(k; L, \rho) + p_2(k; L, \rho) = \frac{\binom{L-k}{3} + \rho\binom{k}{1}\binom{L-k}{2} + \rho^2\binom{k}{2}\binom{L-k}{1}}{\binom{L}{3}}. \tag{A.12}$$

Given a parent in layer k at step $t-1$, the expected numbers of *candidate* children it produces in layers $k+3$, $k+2$, $k+1$ (before resolving global clones) are

$$Y^{(t)}_{k\to k+3} = F_k(t-1)\, M\, p_0(k; L, \rho), \tag{A.13}$$

$$Y^{(t)}_{k\to k+2} = F_k(t-1)\, M\, p_1(k; L, \rho), \tag{A.14}$$

$$Y^{(t)}_{k\to k+1} = F_k(t-1)\, M\, p_2(k; L, \rho), \tag{A.15}$$

where M is the total number of clauses and $F_k(t-1)$ the expected number of frontier parents in layer k at step $t-1$.

Summing over all possible parents, the total number of candidate children landing in layer k' at step t is

$$Y^{(t)}_{k'} = \sum_{k=0}^{L} \Big[Y^{(t)}_{k\to k'} \Big], \tag{A.16}$$

where $Y^{(t)}_{k\to k'}$ is nonzero only when $k' = k+1, k+2$, or $k+3$ and $k' \le L$.

A.2.2 Global collisions via occupancy in each layer

The counting in the previous subsection treats each locally valid candidate child as if it were automatically a *new* state. In reality, many candidates coincide with product states that have already appeared at earlier steps, or even earlier within the same step. To capture this global *clone* effect in a controlled way, we apply a standard *occupancy* (balls-in-boxes) argument *within each layer k' separately.*

At the beginning of step t, layer k' already contains, on average, $D_{k'}(t-1)$ distinct product states, out of the total $S_{k'}$ possibilities given by Eq. (A.1). Thus the number of configurations in layer k' that have *not* yet been seen is

$$U_{k'}(t-1) = S_{k'} - D_{k'}(t-1). \tag{A.17}$$

During step t, the clauses acting on the frontier generate on average $Y_{k'}^{(t)}$ candidate children that fall into layer k' [Eq. (A.16)]. In the mean-field model we treat these candidates as $Y_{k'}^{(t)}$ independent draws from a uniform distribution over the $S_{k'}$ possible configurations in that layer (with replacement).

Consider one particular configuration $\mathcal{C}$ in layer k' that is still unseen at the beginning of step t. In a single draw, the probability that the candidate equals $\mathcal{C}$ is

$$\Pr(\text{hit } \mathcal{C} \text{ in one draw}) = \frac{1}{S_{k'}}.$$

Therefore, the probability of *not* hitting $\mathcal{C}$ in a single draw is $1-1/S_{k'}$, and after $Y_{k'}^{(t)}$ independent draws the probability that $\mathcal{C}$ has never been selected is

$$\Big(1 - \frac{1}{S_{k'}}\Big)^{Y_{k'}^{(t)}}.$$

The complementary event is that $\mathcal{C}$ is hit at least once, with probability

$$\Pr(\mathcal{C} \text{ is selected at least once}) = 1 - \Big(1 - \frac{1}{S_{k'}}\Big)^{Y_{k'}^{(t)}}.$$

By symmetry, every unseen configuration in layer k' has the same probability to be selected at least once during step t. Let us denote by $\mathrm{New}_{k'}(t)$ the number of *distinct* unseen configurations in layer k' that are hit at least once. If we label the $U_{k'}(t-1)$ unseen configurations by an index i and introduce indicator random variables

$$X_i = \begin{cases} 1, & \text{if configuration } i \text{ is hit at least once in step } t, \\ 0, & \text{otherwise,} \end{cases}$$

then

$$\mathrm{New}_{k'}(t) = \sum_{i=1}^{U_{k'}(t-1)} X_i.$$

All X_i have the same expectation,

$$\mathbb{E}[X_i] = 1 - \Big(1 - \frac{1}{S_{k'}}\Big)^{Y_{k'}^{(t)}}.$$

Using linearity of expectation we obtain

$$\mathbb{E}[\mathrm{New}_{k'}(t)] = U_{k'}(t-1)\Big[1 - \Big(1 - \frac{1}{S_{k'}}\Big)^{Y_{k'}^{(t)}}\Big]. \tag{A.18}$$

In the mean-field evolution we identify this expectation with the deterministic quantity

$$\mathrm{New}_{k'}(t) = U_{k'}(t-1)\Big[1 - \big(1 - \tfrac{1}{S_{k'}}\big)^{Y_{k'}^{(t)}}\Big], \tag{A.19}$$

which is precisely the occupancy formula used in our recursion. Intuitively, $\mathrm{New}_{k'}(t)$ corrects the naive candidate count $Y_{k'}^{(t)}$ for the fact that many candidates fall on configurations that were already visited or collide with each other within the same step.

We now interpret

$$F_{k'}(t) = \mathrm{New}_{k'}(t) \tag{A.20}$$

as the expected size of the frontier in layer k' at step t (i.e. the number of states first created at step t in that layer), and update the cumulative distinct count in that layer according to

$$D_{k'}(t) = D_{k'}(t-1) + \mathrm{New}_{k'}(t). \tag{A.21}$$

Together with the candidate counts $Y_{k'}^{(t)}$ from Eq. (A.16), Eqs. (A.19) and (A.21) completely specify the mean-field evolution at the level of each layer k'.

Finally, the total frontier size and total distinct count at step t are

$$N(t) = \sum_{k=0}^{L} F_k(t), \qquad D_{\mathrm{tot}}(t) = \sum_{k=0}^{L} D_k(t), \tag{A.22}$$

so that, by construction, $D_{\mathrm{tot}}(t) \le \sum_{k=0}^{L} S_k = 3^L$ for all t.

It is convenient to define an effective *average* proliferation factor between steps $t-1$ and t as

$$\beta_{t-1} := \begin{cases} N(t)/N(t-1), & N(t-1) > 0, \\ 0, & N(t-1) = 0, \end{cases} \tag{A.23}$$

so that, at the coarse-grained level, $N(t) \approx \beta_{t-1}\, N(t-1)$. The detailed layer-resolved dynamics is however fully encoded in Eqs. (A.1), (A.12), (A.16), and (A.19).

A.2.3 Asymptotic behaviour and saturated basis size

The qualitative behaviour of the coupled recursions (A.16)–(A.21) can be summarized as follows.

Early steps: almost ideal branching. At very small t we have $D_k(t) \approx 0$ for $k > 0$, with $D_0(0) = 1$, and almost all frontier weight is in the $k = 0$ layer. In this regime $U_k(t) \approx S_k$ for all k, and the occupancy factor $1 - (1 - 1/S_k)^{Y_k^{(t)}}$ is close to 1 whenever $Y_k^{(t)}$ is not extremely small. Thus, almost every locally valid candidate child is also globally new, and the layer-resolved growth is close to the "ideal" law where each frontier parent produces M distinct children.

Intermediate steps: local pruning and layer filling. As t increases, frontier weight flows to larger k because each successful clause adds 1, 2, or 3 pins. Two effects then suppress growth:

1. Increasing k reduces the local success probability $p_{\mathrm{loc}}(k; L, \rho)$ in Eq. (A.12), because clauses are increasingly likely to encounter frozen spins with incompatible literals or to touch only already-pinned spins. This effect reduces $Y_{k'}^{(t)}$.

2. Within each layer k', the occupancy factor $U_{k'}(t-1)/S_{k'}$ decreases as $D_{k'}(t-1)$ grows and the layer fills up. Even when $Y_{k'}^{(t)}$ is large, the expected number of *new* configurations $\mathrm{New}_{k'}(t)$ saturates once $D_{k'}(t-1)$ is comparable with $S_{k'}$.

Both mechanisms shrink $F_k(t)$ and hence $N(t)$; correspondingly, β_t decreases from its initial value of order M.

Late steps: frontier extinction and plateau. For sufficiently large t, essentially all layers with nonzero $F_k(t)$ have either: (i) almost all spins pinned (k close to L), so that $p_{\mathrm{loc}}(k; L, \rho) \approx 0$, or (ii) are already densely populated ($D_k(t) \approx S_k$), so that $U_k(t) \approx 0$. In both cases, Eq. (A.19) yields $\mathrm{New}_k(t) \approx 0$, and therefore $F_k(t) \approx 0$ for all k. At that point $N(t) \to 0$ and $D_{\mathrm{tot}}(t)$ saturates to a finite limit

$$D_\star := \lim_{t \to \infty} D_{\mathrm{tot}}(t), \tag{A.24}$$

which we call the *saturated basis size*.

Beyond the corresponding saturation depth $t_\star$, the frontier-based ITP no longer generates new product states; it only redistributes amplitudes among the $D_\star$ states in the variational manifold. Since each product state carries L single-qubit angles and one global amplitude, the number of variational parameters for $t \geq t_\star$ scales as

$$N_{\text{params}}(t \geq t_\star) \;\sim\; D_\star(L{+}1). \tag{A.25}$$

Scaling with system size and number of steps. For a coarse-grained scaling analysis it is useful to neglect the layer structure and treat $D_{\text{tot}}(t)$ as a single population. In this approximation, an average frontier state produces λ candidate children per step, and the global occupancy correction acts on the total configuration space of size 3^L. The evolution of $D(t) \equiv D_{\text{tot}}(t)$ is then schematically

$$D(t{+}1) - D(t) \;\approx\; \lambda\, D(t) \left[1 - \frac{D(t)}{3^L}\right], \tag{A.26}$$

which is a discrete logistic equation. Passing to a continuum variable $t \in \mathbb{R}$ and writing $a \simeq \log(1+\lambda)$ we obtain

$$\frac{dD}{dt} = a\, D \left(1 - \frac{D}{3^L}\right). \tag{A.27}$$

The solution with initial condition $D(0) = D_0$ is

$$D(t; L) = \frac{3^L}{1 + B\, e^{-at}}, \qquad B = \frac{3^L}{D_0} - 1. \tag{A.28}$$

This form makes the interplay between L and t explicit. Introducing the rescaled depth $\theta = t/L$ and using $3^L = e^{L \ln 3}$ we can rewrite Eq. (A.28) as

$$\frac{D(t; L)}{3^L} \;\approx\; \frac{1}{1 + \exp\big[L(\ln 3 - a\theta)\big]} \qquad (L \gg 1). \tag{A.29}$$

Three regimes emerge:

- For fixed t and $L \to \infty$, the denominator is large and $D(t; L)$ grows only as $\mathcal{O}(1)$ with L: the system has not had time to explore the exponentially large configuration space.
- For t proportional to L, $t = \theta L$, we obtain a scaling form

  $$D(t; L) \sim \exp\big[L f(\theta)\big], \qquad f(\theta) = \min\big(a\theta, \ln 3\big). \tag{A.30}$$

  In particular, for $\theta < \theta_c = \ln 3/a$ the growth is *quasi-exponential* in the number of steps, $D(t) \propto e^{at}$, while for $\theta > \theta_c$ the curve saturates to $D_\star \sim 3^L$.
- For fixed L and $t \to \infty$, Eq. (A.28) recovers the plateau $D_\star(L) \propto 3^L$.

Even though this scalar approximation ignores the detailed layer structure, it captures the key qualitative message: for physically relevant depths $t = O(L)$ the number of distinct product states generated by the frontier dynamics grows exponentially in L up to a saturation value

$$D_\star(L) \sim \exp(\gamma L), \qquad \ln 2 < \gamma \leq \ln 3, \tag{A.31}$$

with γ set by the effective growth rate a and hence, in practice, by ρ_{eff} and the clause density M/L. The numerical results of Sec. A.2.4 below confirm this picture and allow us to determine γ more precisely for the parameter regime of interest.

A.2.4 Numerical mean-field curves and comparison with ITP data

The layer–resolved recursions derived above were implemented numerically for system sizes $L \in \{6,8,10\}$, with clause counts $M \approx 4.3\,L$ (rounded to the nearest integer). For each L we initialize

$$F_0(0) = 1, \quad D_0(0) = 1, \qquad F_k(0) = D_k(0) = 0 \text{ for } k \geq 1,$$

iterate the layer equations up to $T = 10$ steps, and record

$$N(t) = \sum_{k=0}^{L} F_k(t), \qquad D_{\text{tot}}(t) = \sum_{k=0}^{L} D_k(t).$$

Effective compatibility parameter. As discussed in Sec. A.2.1, the bare value $\rho_{\text{bare}} = 1/2$ that one would assign to an isolated literal on a pinned spin leads to a mean-field manifold size that overshoots the measured ITP frontier by roughly one order of magnitude already at $L = 10$. To obtain a quantitatively faithful description we therefore treat ρ as an effective parameter ρ_{eff} and fix it by matching the mean-field curves to the ITP data for small system sizes.

Concretely, for each trial value of ρ we evaluate $D_{\text{tot}}^{\text{mf}}(L,t;\rho)$ from the recursions and compare it to the manifold size $D^{\text{ITP}}(L,t)$ measured in the full frontier simulation, for $L = 6,8,10$ and $t = 1,2,3$. We then minimize the sum of squared deviations in $\log_{10} D$,

$$\mathcal{L}(\rho) = \sum_{L\in\{6,8,10\}} \sum_{t=1}^{3} \Big[\log_{10} D_{\text{tot}}^{\text{mf}}(L,t;\rho) - \log_{10} D^{\text{ITP}}(L,t)\Big]^2,$$

over a grid of ρ values. This procedure yields a nearly L–independent optimum

$$\rho_{\text{eff}} \simeq 0.12, \tag{A.32}$$

which we use in all the numerical mean-field curves shown here and in Fig. 4.4(b,c) of the main text.

The comparison between the calibrated mean-field prediction for $D_{\text{tot}}(t)$ and the measured ITP manifold growth for $L = 3,5,8,10$ is displayed in Fig. 4.4: panel (a) corresponds to $D^{\text{ITP}}(t)$, while panels (b) and (c) show $N(t)$ and $D_{\text{tot}}(t)$ obtained from the recursions with $\rho_{\text{eff}} \approx 0.1$. On the logarithmic scale used there the agreement is essentially perfect: both the initial exponential burst and the saturation plateaux are captured quantitatively by the mean-field model once the effective compatibility parameter is fixed. This supports the interpretation of ρ as a phenomenological parameter that absorbs the combined effects of clause correlations, heterogeneous parents, and strong concentration of the frontier in a small region of the layer space.

With ρ fixed once and for all to $\rho_{\text{eff}} = 0.1$, the mean-field model becomes a genuine predictive tool for larger instances where a full ITP simulation would be costly. Figure 7.1 in chapter 6 shows $D_{\text{tot}}(t)$ obtained from the recursions for $L = 20,30$ with $M \simeq 4.3L$ and the same value of ρ_{eff}. In all cases, we observe the same qualitative pattern as at small L and in Fig. 4.4:

- a short burst regime where the manifold size grows almost exponentially in the number of steps, $D_{\text{tot}}(t) \propto e^{at}$ with an L–independent rate $a > 0$;
- a crossover to a plateau at a depth $t_\star(L) = O(L)$, in line with the scaling analysis of Sec. A.2.3;
- a saturated manifold size $D_\star(L)$ that scales exponentially with L and, for the parameters considered here, lies *well above* the Hilbert-space dimension 2^L while still remaining below the full product-state count 3^L.

The last point is crucial for our variational strategy. Because the frontier-based ansatz is an *overcomplete* set of product states, there is no fundamental obstruction to having $D_\star(L) > 2^L$: one can always express the same subspace with more redundant vectors. However, once $D_{\mathrm{tot}}(t)$ becomes comparable to or larger than 2^L the number of variational parameters $N_{\mathrm{params}} \sim D_{\mathrm{tot}}(t)(L+1)$ ceases to be competitive with a direct parametrization in the computational basis. The mean-field results of Fig. 4.4 and Fig. 7.1 therefore support the practical choice, used throughout this work, of restricting the ITP depth to a small number of steps: this keeps the manifold firmly in the regime $D_{\mathrm{tot}}(t) \ll 2^L$ where the frontier-based ansatz provides a genuine compression of the relevant Hilbert space.

Summary. The layer–resolved mean-field model developed in this appendix is a direct coarse-grained counterpart of the frontier-based ITP dynamics discussed in Sec. 4.4. Local clause statistics — encoded in the probabilities $p_0(k), p_1(k), p_2(k)$ and in the derived quantities $p_{\mathrm{loc}}(k; L, \rho)$ and $\Delta k(k; L, \rho)$ — capture how often a clause can consistently add new pins to a typical parent state and by how much. Global clone effects are then incorporated layer by layer via the occupancy factor $U_k(t-1)\big[1-(1-1/S_k)^{Y_k^{(t)}}\big]$, which reduces the naive candidate count $Y_k^{(t)}$ to the expected number of *truly new* configurations in each layer.

Two additional ingredients are essential for quantitative accuracy. First, we recognize that the compatibility parameter ρ entering $p_1(k)$ and $p_2(k)$ must be interpreted as an effective quantity ρ_{eff} that folds into a single number the correlations, heterogeneity and enhanced clone production that the simple mean-field model does not resolve explicitly. Second, we fix $\rho_{\mathrm{eff}} \approx 0.1$ empirically by matching the mean-field plateau heights to the ITP manifold sizes for small systems $L = 6{,}8{,}10$ and using exactly the same value to generate the curves in Fig. 4.4(b,c). With this calibration in place, the mean-field model not only reproduces the measured frontier growth for those sizes, but also predicts that for larger instances the total number of distinct product states generated by the frontier-based ITP grows quasi-exponentially with depth and eventually saturates at a value $D_\star(L) \sim e^{\gamma L}$ that, for our parameters, far exceeds 2^L. Beyond this saturation depth, increasing the number of ITP steps only enlarges an already overcomplete manifold and offers no practical advantage over working in the full Hilbert space, justifying our focus on shallow ITP schedules in the main text.

A.3 Efficient evaluation of the TDVP metric and force

The explicit expressions of the TDVP metric tensor $S_{\bar{w}}$ and force vector $F_{\bar{w}}$ for our multi–product ansatz, as derived in Chapter 4, are algebraically simple but computationally expensive. They contain, for each pair of replicas (λ, μ) and indices (i, j), products of the form

$$\prod_{k\neq i,j} \cos(\theta_{\lambda k} - \theta_{\mu k}) \times \sin(\theta_{\mu i} - \theta_{\lambda i})\, \sin(\theta_{\lambda j} - \theta_{\mu j}), \tag{A.33}$$

together with sums over all replicas weighted by the amplitudes a_λ. A naive implementation would recompute similar products repeatedly for each component of $S_{\bar{w}}$ and $F_{\bar{w}}$, leading to a very unfavorable scaling in both the number of replicas D and the number of qubits L.

A.3.1 Algebraic factorization of trigonometric products

The key idea behind our implementation is to factor out, whenever possible, the full product of cosine overlaps between replicas,

$$C_{\lambda\mu} \;\equiv\; \prod_{k=0}^{L-1} \cos(\theta_{\lambda k} - \theta_{\mu k}), \tag{A.34}$$

and to express all remaining terms in terms of ratios such as

$$\tan(\theta_{\lambda i} - \theta_{\mu i}) = \frac{\sin(\theta_{\lambda i} - \theta_{\mu i})}{\cos(\theta_{\lambda i} - \theta_{\mu i})}. \tag{A.35}$$

For instance, the "new" expression for the angular–angular block of $S_{\bar{w}}$, already reported in *Ansantz_updated* as

$$(S_{\bar{w}})_{\partial_{\theta_{\lambda i}}, \partial_{\theta_{\mu j}}} = a_\lambda a_\mu\, C_{\lambda\mu} \times \begin{cases} -\tan(\theta_{\lambda i} - \theta_{\mu i})\, \tan(\theta_{\lambda j} - \theta_{\mu j}), & i \neq j, \\ 1, & i = j, \end{cases} \;-\; (\text{projected terms}), \tag{A.36}$$

is exactly obtained from the original product form by writing

$$\begin{aligned} \prod_{k\neq i,j} \cos(\theta_{\lambda k} - \theta_{\mu k})\, \sin(\theta_{\mu i} - \theta_{\lambda i})\, \sin(\theta_{\lambda j} - \theta_{\mu j}) &= \left[\prod_{k=0}^{L-1} \cos(\theta_{\lambda k} - \theta_{\mu k})\right] \frac{\sin(\theta_{\mu i} - \theta_{\lambda i})}{\cos(\theta_{\lambda i} - \theta_{\mu i})}\, \frac{\sin(\theta_{\lambda j} - \theta_{\mu j})}{\cos(\theta_{\lambda j} - \theta_{\mu j})} \\ &= C_{\lambda\mu}\, \tan(\theta_{\mu i} - \theta_{\lambda i})\, \tan(\theta_{\lambda j} - \theta_{\mu j}), \end{aligned} \tag{A.37}$$

and similarly for the mixed (a, θ) and (a, a) blocks, where $C_{\lambda\mu}$ appears naturally as the overlap $\langle \rho_\lambda | \rho_\mu \rangle$ of product states.

The same strategy is applied to the force vector $F_{\bar{w}}$: the expectation values $\langle \Psi | H_c | \Psi \rangle$ and $\langle \Psi | H_T | \Psi \rangle$ (and their derivatives) are reorganized so that all dependence on (λ, μ) outside the clause structure factorizes through $C_{\lambda\mu}$ and a small number of additional precomputed quantities. Concretely, for the clause Hamiltonian H_c we write

$$\langle \Psi | H_c | \Psi \rangle = \sum_{\lambda,\nu} a_\lambda a_\nu\, C_{\lambda\nu} \sum_{c:(y,z,h)} B^{(c)}_{\lambda\nu}, \tag{A.38}$$

where $B^{(c)}_{\lambda\nu}$ contains only local trigonometric factors on the three qubits in clause c. For the driver term H_T we similarly use a precomputed product

$$C^{(T)}_{\lambda\nu} = \prod_{k=0}^{L-1} \langle B_{k,\lambda} | A_{k,\nu} \rangle = \prod_{k=0}^{L-1} \Big[\cos\theta_{\lambda k}\, \sin\theta_{\nu k} + \sin\theta_{\lambda k}\, \cos\theta_{\nu k}\Big], \tag{A.39}$$

together with its derivative with respect to $\theta_{\lambda i}$.

From a mathematical point of view, these manipulations are *exact* whenever $\cos(\theta_{\lambda i} - \theta_{\mu i}) \neq 0$ for all replicas and qubits. What changes is the way the expressions are organized: instead of evaluating many products with one or two factors removed, we precompute the full product $C_{\lambda\mu}$ and reconstruct the "excluded" products by division.

A.3.2 Impact on computational complexity

Let D be the number of product states (replicas) and L the number of qubits. The naive evaluation of $S_{\bar{w}}$ scales roughly as

$$\mathcal{O}\big(D^2L^3\big) \tag{A.40}$$

for the angular–angular block alone: for each pair (λ, μ) and each pair of sites (i, j) one has to form a product over all $k \neq i, j$, which is $\mathcal{O}(L)$. The mixed and (a, a) blocks introduce additional contributions of comparable cost.

With the factorized form we instead proceed in two stages:

1. Precompute all angle differences and their trigonometric functions,

$$\Delta\theta_{\lambda\mu k} = \theta_{\lambda k} - \theta_{\mu k}, \qquad \cos\Delta_{\lambda\mu k}, \quad \sin\Delta_{\lambda\mu k}, \tag{A.41}$$

   together with the full products

$$P_{\lambda\mu} = \prod_{k=0}^{L-1} \cos\Delta_{\lambda\mu k}. \tag{A.42}$$

   This stage costs $\mathcal{O}(D^2L)$ operations.

2. Reconstruct all needed "excluded" products as

$$P_{\lambda\mu}^{(-i)} = \frac{P_{\lambda\mu}}{\cos\Delta_{\lambda\mu i}}, \qquad P_{\lambda\mu}^{(-i,-\ell)} = \frac{P_{\lambda\mu}}{\cos\Delta_{\lambda\mu i}\,\cos\Delta_{\lambda\mu \ell}}, \tag{A.43}$$

   and build the blocks of $S_{\bar{w}}$ and $F_{\bar{w}}$ using these precomputed arrays. Each matrix element is then obtained with a small, fixed number of arithmetic operations.

As a result, the overall cost of constructing $S_{\bar{w}}$ scales as

$$\mathcal{O}\big(D^2L^2\big), \tag{A.44}$$

which is parametrically faster than the naive $\mathcal{O}(D^2L^3)$ behaviour for moderate and large L. A similar reduction is obtained for the computation of the force vector $F_{\bar{w}}$, where the precomputed products $C_{\lambda\nu}$ and $C_{\lambda\nu}^{(T)}$ are reused in all derivatives with respect to both the amplitudes a_λ and the angles $\theta_{\lambda i}$.

In practice this reorganization drastically reduces the runtime of each TDVP step and makes it feasible to work with significantly larger values of (D, L) than would be accessible with a direct implementation of the analytic formulas.

A.3.3 Division by small cosines and numerical regularization

The price to pay for factoring out the full product $C_{\lambda\mu}$ is that all the "excluded" products are obtained by dividing by one or two factors $\cos(\theta_{\lambda i} - \theta_{\mu i})$. Analytically, this is completely harmless as long as these cosines are nonzero: the simplified formulas are exactly equivalent to the original ones. Numerically, however, two issues may appear:

- For generic parameters, $\cos(\theta_{\lambda i} - \theta_{\mu i})$ can become very small, amplifying rounding errors and potentially producing very large entries in $S_{\bar{w}}$ and $F_{\bar{w}}$.

- In the pathological case where a difference hits exactly a point of the form $\theta_{\lambda i} - \theta_{\mu i} = \pi/2 + n\pi$, the denominator would vanish, and the naive floating–point division would produce overflows or `NaN`.

To tame these issues, we adopt two complementary strategies:

1:In the `NumPy/Numba` implementation, we introduce a tiny regularization parameter $\varepsilon = 10^{-12}$ and replace

$$\frac{1}{\cos \Delta_{\lambda\mu i}} \longrightarrow \frac{1}{\cos \Delta_{\lambda\mu i} + \varepsilon}, \tag{A.45}$$

and similarly for products of two cosines. This prevents divisions by zero and limits the growth of the entries when $\cos \Delta_{\lambda\mu i}$ becomes extremely small. Since we work in double precision and typical values of $\cos \Delta_{\lambda\mu i}$ are far larger than 10^{-12}, the induced bias is negligible at the scale of the statistical and systematic errors discussed in the main text.

2: The linear system

$$S_{\bar{w}}\, \delta\bar{w} = -\gamma\, F_{\bar{w}} \tag{A.46}$$

is solved with the MINRES–QLP algorithm [25], which is specifically designed for symmetric, possibly indefinite, and nearly singular systems. MINRES–QLP is robust against moderate ill–conditioning of $S_{\bar{w}}$ and allows us to work safely even when some directions in parameter space are weakly constrained.

Together, these two ingredients ensure that the algebraic simplification based on the $C_{\lambda\mu}$ factors leads to a numerically stable and efficient implementation of the TDVP update.

A.4 Imaginary time TDVP: implementation details

A.4.1 Pseudocode for the metric $S_{\bar{w}}$

For clarity, we summarize the main steps of the optimized computation of the TDVP metric here. The parameter vector is ordered as

$$\bar{w} = \big(a_0, \dots, a_{D-1}; \theta_{0,0}, \dots, \theta_{0,L-1}; \dots; \theta_{D-1,0}, \dots, \theta_{D-1,L-1}\big), \tag{A.47}$$

so that the metric $S_{\bar{w}}$ is a matrix of size $(D + DL) \times (D + DL)$.

Algorithm 3 Efficient computation of the TDVP metric $S_{\bar{w}}$ (precomputations)

Require: Amplitudes a_λ; angles $\theta_{\lambda i}$; regularization parameter ε
Ensure: Intermediate arrays for building $S_{\bar{w}}$

$D \leftarrow$ number of replicas; $L \leftarrow$ number of qubits
Initialize S as a $(D + DL) \times (D + DL)$ zero matrix

Precompute angle differences and trigonometric factors
for $\lambda = 0, \dots, D-1$ **do**
  for $\mu = 0, \dots, D-1$ **do**
    for $k = 0, \dots, L-1$ **do**
      $\Delta\theta_{\lambda\mu k} \leftarrow \theta_{\lambda k} - \theta_{\mu k}$
      $\cos \Delta_{\lambda\mu k} \leftarrow \cos(\Delta\theta_{\lambda\mu k})$
      $\sin \Delta_{\lambda\mu k} \leftarrow \sin(\Delta\theta_{\lambda\mu k})$

Precompute full and excluded cosine products
for $\lambda = 0, \dots, D-1$ **do**
  for $\mu = 0, \dots, D-1$ **do**
    $P_{\lambda\mu} \leftarrow \prod_{k=0}^{L-1} \cos \Delta_{\lambda\mu k}$
    for $i = 0, \dots, L-1$ **do**
      $P^{(-i)}_{\lambda\mu} \leftarrow \dfrac{P_{\lambda\mu}}{\cos \Delta_{\lambda\mu i} + \varepsilon}$
    for $i = 0, \dots, L-1$ **do**
      for $\ell = 0, \dots, L-1$ **do**
        $P^{(-i,-\ell)}_{\lambda\mu} \leftarrow \dfrac{P_{\lambda\mu}}{\cos \Delta_{\lambda\mu i} \cos \Delta_{\lambda\mu \ell} + \varepsilon}$

Precompute auxiliary sums
for $j = 0, \dots, D-1$ **do**
  $A2_j \leftarrow \sum_{\nu=0}^{D-1} a_\nu P_{j\nu}$
for $\lambda = 0, \dots, D-1$ **do**
  for $i = 0, \dots, L-1$ **do**
    $S2_{\lambda i} \leftarrow a_\lambda \sum_{\nu=0}^{D-1} a_\nu P^{(-i)}_{\lambda\nu} \sin\big(\theta_{\nu i} - \theta_{\lambda i}\big)$
for $\mu = 0, \dots, D-1$ **do**
  for $i = 0, \dots, L-1$ **do**
    $S3_{\mu i} \leftarrow a_\mu \sum_{\eta=0}^{D-1} a_\eta P^{(-i)}_{\eta\mu} \sin\big(\theta_{\eta i} - \theta_{\mu i}\big)$
return $P_{\lambda\mu}$, $P^{(-i)}_{\lambda\mu}$, $P^{(-i,-\ell)}_{\lambda\mu}$, $A2_j$, $S2_{\lambda i}$, $S3_{\mu i}$, and the zero matrix S

Algorithm 4 Efficient computation of the TDVP metric $S_{\bar{w}}$ (assembly)

Require: Precomputed $P_{\lambda\mu}$, $P^{(-i)}_{\lambda\mu}$, $P^{(-i,-\ell)}_{\lambda\mu}$, $A2_j$, $S2_{\lambda i}$, $S3_{\mu i}$; zero matrix S
Ensure: Metric matrix $S \in \mathbb{R}^{(D+DL)\times(D+DL)}$

$D \leftarrow$ number of replicas; $L \leftarrow$ number of qubits

Fill the (a, a) **block**
for $j = 0, \ldots, D-1$ **do**
  for $k = 0, \ldots, D-1$ **do**
    $S_{j,k} \leftarrow P_{jk} - A2_j\, A2_k$

Fill the (θ, θ) **block**
for $\lambda = 0, \ldots, D-1$ **do**
  for $\mu = 0, \ldots, D-1$ **do**
    for $i = 0, \ldots, L-1$ **do**
      for $\ell = 0, \ldots, L-1$ **do**
        if $i = \ell$ **then**
          $\text{term1} \leftarrow a_\lambda a_\mu\, P^{(-i)}_{\lambda\mu}\, \cos(\theta_{\lambda i} - \theta_{\mu i})$
        else
          $\text{term1} \leftarrow a_\lambda a_\mu\, P^{(-i,-\ell)}_{\lambda\mu}\, \sin(\theta_{\mu i} - \theta_{\lambda i})\, \sin(\theta_{\lambda \ell} - \theta_{\mu \ell})$
        $\text{corr} \leftarrow S2_{\lambda i}\, S3_{\mu \ell}$
        Map (λ, i) and (μ, ℓ) to global indices $p, q \in \{0, \ldots, D + DL - 1\}$
        $S_{p,q} \leftarrow \text{term1} - \text{corr}$

Fill the mixed (a, θ) **and** (θ, a) **blocks**
for $j = 0, \ldots, D-1$ **do**
  for $\mu = 0, \ldots, D-1$ **do**
    for $i = 0, \ldots, L-1$ **do**
      $\text{term} \leftarrow a_\mu\, P^{(-i)}_{j\mu}\, \sin(\theta_{ji} - \theta_{\mu i})$
      $S_{\text{block}} \leftarrow \text{term} - A2_j\, S3_{\mu i}$
      Map (μ, i) to global index q
      $S_{j,q} \leftarrow S_{\text{block}}$
      $S_{q,j} \leftarrow S_{\text{block}}$ ▷ S is symmetric
return S

A.4.2 Efficient computation of the TDVP force for the clause Hamiltonian

In this appendix we summarize the optimized implementation used to evaluate the TDVP force $F_{\bar{w}}$ associated with the clause Hamiltonian H_c. The state $|\Psi(\bar{w})\rangle$ depends on a set of real amplitudes $\{a_\lambda\}_{\lambda=0}^{D-1}$ and angles $\{\theta_{\lambda k}\}_{\lambda=0,\ldots,D-1}^{k=0,\ldots,L-1}$, where D is the number of replicas and L the number of qubits. The TDVP force components are given by

$$F_j = \langle \partial_{w_j} \Psi | H_c | \Psi \rangle - \langle \partial_{w_j} \Psi | \Psi \rangle \, \langle \Psi | H_c | \Psi \rangle \,, \tag{A.48}$$

with parameters $w_j \in \{a_\lambda\} \cup \{\theta_{\lambda k}\}$.

The classical Hamiltonian H_c is constructed from a 3-SAT instance, encoded as a list of clauses $\mathcal{C}$. Each clause is first mapped to two integer arrays: one containing the qubit indices of its literals and one storing their signs. We then precompute all trigonometric factors that enter the overlaps and the expectation value $\langle \Psi | H_c | \Psi \rangle$. The same precomputed tables are reused to evaluate all derivatives with respect to a_λ and $\theta_{\lambda k}$ and, finally, to assemble the force vector.

Algorithm 5 Precomputations for the TDVP force $F_{\bar{w}}$ (classical Hamiltonian H_c)

Require: Amplitudes a_λ; angles $\theta_{\lambda k}$; clause list $\mathcal{C}$
Ensure: Clause arrays, precomputed trigonometric data and energy $\langle\Psi|H_c|\Psi\rangle$

$D \leftarrow$ number of replicas; $L \leftarrow$ number of qubits
Convert the clause list $\mathcal{C}$ to arrays `c_q`, `c_s` (qubit indices and literal signs ± 1)

Precompute trigonometric tables

for $\lambda = 0, \ldots, D-1$ **do**
  for $k = 0, \ldots, L-1$ **do**
    $\cos\theta_{\lambda k} \leftarrow \cos(\theta_{\lambda k})$
    $\sin\theta_{\lambda k} \leftarrow \sin(\theta_{\lambda k})$
for $\lambda = 0, \ldots, D-1$ **do**
  for $\nu = 0, \ldots, D-1$ **do**
    for $k = 0, \ldots, L-1$ **do**
      $\Delta\theta_{\lambda\nu k} \leftarrow \theta_{\lambda k} - \theta_{\nu k}$
      $\cos\Delta_{\lambda\nu k} \leftarrow \cos(\Delta\theta_{\lambda\nu k})$
      $\tan\Delta_{\lambda\nu k} \leftarrow \begin{cases} \tan(\Delta\theta_{\lambda\nu k}), & |\cos\Delta_{\lambda\nu k}| > \varepsilon, \\ 0, & \text{otherwise} \end{cases}$
    $C_{\lambda\nu} \leftarrow \prod_{k=0}^{L-1} \cos\Delta_{\lambda\nu k}$

Compute the classical energy $\langle\Psi|H_c|\Psi\rangle$

$\langle\Psi|H_c|\Psi\rangle \leftarrow 0$
for $\lambda = 0, \ldots, D-1$ **do**
  for $\nu = 0, \ldots, D-1$ **do**
    clause_sum $\leftarrow 0$
    for each clause c in $\mathcal{C}$ **do**
      val_clause $\leftarrow 1$
      for each literal index j in clause c **do**
        $q \leftarrow$ `c_q`$[c, j]$ ▷ qubit index
        $s \leftarrow$ `c_s`$[c, j]$ ▷ sign $s \in \{+1, -1\}$
        denom $\leftarrow \cos\Delta_{\lambda\nu q}$
        if |denom| $< \varepsilon$ **then**
          val_clause $\leftarrow 0$; **break**
        if $s = -1$ **then**
          val_clause $\leftarrow$ val_clause $\cdot \dfrac{\cos\theta_{\lambda q}\,\cos\theta_{\nu q}}{\text{denom}}$
        else
          val_clause $\leftarrow$ val_clause $\cdot \dfrac{\sin\theta_{\lambda q}\,\sin\theta_{\nu q}}{\text{denom}}$
      clause_sum $\leftarrow$ clause_sum + val_clause
    $\langle\Psi|H_c|\Psi\rangle \leftarrow \langle\Psi|H_c|\Psi\rangle + a_\lambda a_\nu\, C_{\lambda\nu}$ clause_sum
return `c_q`, `c_s`, $\cos\theta$, $\sin\theta$, $\cos\Delta$, $\tan\Delta$, $C_{\lambda\nu}$, $\langle\Psi|H_c|\Psi\rangle$

Algorithm 6 Efficient computation of the TDVP force $F_{\bar{w}}$ (derivatives and assembly for H_c)

Require: Amplitudes a_λ; precomputed $\cos\theta$, $\sin\theta$, $\cos\Delta$, $\tan\Delta$, overlaps $C_{\lambda\nu}$, clause arrays `c_q`, `c_s`, and energy $E_c = \langle\Psi|H_c|\Psi\rangle$

Ensure: Force vector $F_{\bar{w}}$ of length $D + DL$

$D \leftarrow$ number of replicas; $L \leftarrow$ number of qubits

Initialize F as a zero vector of length $D + DL$

Derivatives w.r.t. a_λ

for $\lambda = 0, \ldots, D-1$ **do**

$\partial_{a_\lambda} \langle\Psi|\Psi\rangle \leftarrow \sum_{\nu=0}^{D-1} a_\nu \, C_{\lambda\nu}$

$\partial_{a_\lambda} \langle\Psi|H_c|\Psi\rangle \leftarrow \sum_{\nu=0}^{D-1} a_\nu \, C_{\lambda\nu}$ [clause factor from H_c as in Alg. 5]

Derivatives w.r.t. $\theta_{\lambda i}$

for $\lambda = 0, \ldots, D-1$ **do**

  for $i = 0, \ldots, L-1$ **do**

$\partial_{\theta_{\lambda i}} \langle\Psi|\Psi\rangle \leftarrow \sum_{\nu=0}^{D-1} a_\lambda a_\nu \, C_{\lambda\nu} \, [-\tan\Delta_{\lambda\nu i}]$

$\partial_{\theta_{\lambda i}} \langle\Psi|H_c|\Psi\rangle \leftarrow$ sum over ν and clauses c using $\cos\theta$, $\sin\theta$, $\cos\Delta$, $\tan\Delta$ and the local clause structure (`c_q`, `c_s`), as in the analytic TDVP formulas

Assemble the force vector

for $\lambda = 0, \ldots, D-1$ **do**

$F_{a_\lambda} \leftarrow \partial_{a_\lambda} \langle\Psi|H_c|\Psi\rangle - \partial_{a_\lambda} \langle\Psi|\Psi\rangle \; E_c$

for $\lambda = 0, \ldots, D-1$ **do**

  for $i = 0, \ldots, L-1$ **do**

$F_{\theta_{\lambda i}} \leftarrow \partial_{\theta_{\lambda i}} \langle\Psi|H_c|\Psi\rangle - \partial_{\theta_{\lambda i}} \langle\Psi|\Psi\rangle \; E_c$

return F (first the D entries F_{a_λ}, then the DL entries $F_{\theta_{\lambda i}}$)

A.5 Action of the TDVP metric tensor as a linear operator

A.5.1 S as an Operator

In the TDVP (or SR) equation

$$S_{\bar{w}}\,\delta\bar{w} = -\gamma\,F_{\bar{w}}, \tag{A.49}$$

the matrix $S_{\bar{w}}$ is defined component-wise by

$$(S_{\bar{w}})_{j,k} = \langle\partial_{w_j}\Psi|\partial_{w_k}\Psi\rangle - \langle\partial_{w_j}\Psi|\Psi\rangle\,\langle\Psi|\partial_{w_k}\Psi\rangle\,, \qquad w_j, w_k \in \{a_\lambda, \theta_i^\lambda\}. \tag{A.50}$$

In practice we never form $S_{\bar{w}}$ explicitly as a dense matrix. Krylov solvers such as MINRES-QLP only need the *action* of $S_{\bar{w}}$ on a trial vector v. This appendix derives closed-form formulas for $y = S_{\bar{w}}v$, written in terms of analytically computable overlaps of our variational ansatz.

Parameter indexing

We split the variational parameters into:

- the amplitudes a_λ,
- the Bloch angles θ_i^λ (site i in configuration λ).

A generic vector v in parameter space is then

$$v \equiv \{v_\mu^{(a)}\}_\mu \;\cup\; \{v_{j,\mu}^{(\theta)}\}_{j,\mu},$$

where $v_\mu^{(a)}$ is the component along a_μ and $v_{j,\mu}^{(\theta)}$ is the component along θ_j^μ. The output $y = S_{\bar{w}}v$ will have the same block structure:

$$y \equiv \{y_\lambda^{(a)}\}_\lambda \;\cup\; \{y_{i,\lambda}^{(\theta)}\}_{i,\lambda}.$$

Shorthand definitions

For compactness we define the following quantities, all determined by the current parameters $\{a_\lambda, \theta_i^\lambda\}$ of the ansatz

$$|\Psi\rangle = \sum_\lambda a_\lambda \bigotimes_{k=0}^{L-1} \Big(\cos\theta_k^\lambda\,|\uparrow_k\rangle + \sin\theta_k^\lambda\,|\downarrow_k\rangle\Big).$$

Configuration overlaps.

$$C_{\lambda\mu} := \prod_{k=0}^{L-1} \cos(\theta_k^\lambda - \theta_k^\mu), \tag{A.51}$$

$$S_{\lambda\mu} := a_\lambda a_\mu\, C_{\lambda\mu}, \tag{A.52}$$

$$t_i^{\lambda\mu} := \tan(\theta_i^\lambda - \theta_i^\mu). \tag{A.53}$$

Mixed amplitude–overlap factors.

$$R_{\lambda\mu} := a_\mu\, C_{\lambda\mu}. \tag{A.54}$$

Note that $R_{\lambda\mu}$ is not symmetric in (λ, μ).

Projected overlaps with $|\Psi\rangle$. These combinations appear in the projector terms $\langle\partial\Psi|\Psi\rangle\,\langle\Psi|\partial\Psi\rangle$:

$$A_\lambda := \sum_\nu R_{\lambda\nu} = \sum_\nu a_\nu\, C_{\lambda\nu} = \langle \partial_{a_\lambda}\Psi|\Psi\rangle\,, \tag{A.55}$$

$$E_\mu := \sum_\eta R_{\eta\mu} = \sum_\eta a_\eta\, C_{\eta\mu} = \langle \Psi|\partial_{a_\mu}\Psi\rangle\,, \tag{A.56}$$

$$B_{\lambda i} := \sum_\nu S_{\lambda\nu}\,[-t_i^{\lambda\nu}] = \sum_\nu a_\lambda a_\nu\, C_{\lambda\nu}\,[-\tan(\theta_i^\lambda - \theta_i^\nu)] = \langle \partial_{\theta_i^\lambda}\Psi|\Psi\rangle\,, \tag{A.57}$$

$$D_{\mu j} := \sum_\eta S_{\eta\mu}\,[+t_j^{\eta\mu}] = \sum_\eta a_\eta a_\mu\, C_{\eta\mu}\,\tan(\theta_j^\eta - \theta_j^\mu) = \langle \Psi|\partial_{\theta_j^\mu}\Psi\rangle\,. \tag{A.58}$$

Important: $B_{\lambda i}$ carries a *minus* in front of $t_i^{\lambda\nu}$, while $D_{\mu j}$ carries a *plus* in front of $t_j^{\eta\mu}$. This sign structure is what ensures the correct projector subtraction.

Angle components of the output: $y^{(\theta)}_{i,\lambda}$

The component $y^{(\theta)}_{i,\lambda}$ corresponds to the parameter θ_i^λ. It receives contributions from both angle-type inputs $v^{(\theta)}$ and amplitude-type inputs $v^{(a)}$:

$$y^{(\theta)}_{i,\lambda} = y^{(\theta)}_{i,\lambda}\big|_{\theta\leftarrow\theta} + y^{(\theta)}_{i,\lambda}\big|_{a\leftarrow a}.$$

(i) Angle–angle block ($\theta \leftarrow \theta$). The matrix element for two angle parameters $(\theta_i^\lambda, \theta_j^\mu)$ is

$$S^{(\theta,\theta)}_{(i,\lambda),(j,\mu)} = \langle \partial_{\theta_i^\lambda}\Psi|\partial_{\theta_j^\mu}\Psi\rangle - \langle \partial_{\theta_i^\lambda}\Psi|\Psi\rangle\,\langle\Psi|\partial_{\theta_j^\mu}\Psi\rangle \tag{A.59}$$

$$= S_{\lambda\mu}\Big[\delta_{ij} - (1-\delta_{ij})\, t_i^{\lambda\mu}\, t_j^{\lambda\mu}\Big] \;-\; B_{\lambda i}\, D_{\mu j}. \tag{A.60}$$

Acting on the "angle part" $v^{(\theta)}$ gives

$$y^{(\theta)}_{i,\lambda}\big|_{\theta\leftarrow\theta} = \sum_\mu \sum_j S^{(\theta,\theta)}_{(i,\lambda),(j,\mu)}\; v^{(\theta)}_{j,\mu}$$

$$= \sum_\mu S_{\lambda\mu}\left[v^{(\theta)}_{i,\mu}\Big(1 + (t_i^{\lambda\mu})^2\Big) \;-\; t_i^{\lambda\mu} \sum_j t_j^{\lambda\mu}\; v^{(\theta)}_{j,\mu}\right] \;-\; B_{\lambda i} \sum_\mu \sum_j D_{\mu j}\, v^{(\theta)}_{j,\mu}. \tag{A.61}$$

The square bracket is the "direct" piece $\langle\partial\Psi|\partial\Psi\rangle$, while the last term comes from the projector subtraction $-\langle \partial_{\theta_i^\lambda}\Psi|\Psi\rangle\,\langle\Psi|\partial_{\theta_j^\mu}\Psi\rangle$.

(ii) Angle–amplitude block ($a \leftarrow a$). For parameters $(\theta_i^\lambda, a_\mu)$ we find

$$S^{(\theta,a)}_{(i,\lambda),(\,a,\mu)} = \langle \partial_{\theta_i^\lambda}\Psi|\partial_{a_\mu}\Psi\rangle - \langle \partial_{\theta_i^\lambda}\Psi|\Psi\rangle\,\langle\Psi|\partial_{a_\mu}\Psi\rangle \tag{A.62}$$

$$= \Big[- a_\mu\, C_{\lambda\mu}\, t_i^{\lambda\mu}\Big] \;-\; B_{\lambda i}\, E_\mu \tag{A.63}$$

$$= -R_{\lambda\mu}\, t_i^{\lambda\mu} \;-\; B_{\lambda i}\, E_\mu, \tag{A.64}$$

using $R_{\lambda\mu} = a_\mu C_{\lambda\mu}$. Its action on $v^{(a)}$ is

$$y^{(\theta)}_{i,\lambda}\big|_{a\leftarrow a} = \sum_\mu S^{(\theta,a)}_{(i,\lambda),(\,a,\mu)}\; v^{(a)}_\mu$$

$$= \sum_\mu \Big[- R_{\lambda\mu}\, t_i^{\lambda\mu} \;-\; B_{\lambda i}\, E_\mu\Big] v^{(a)}_\mu. \tag{A.65}$$

(iii) Total angle output. Combining the two contributions:

$$y^{(\theta)}_{i,\lambda} = y^{(\theta)}_{i,\lambda}\big|_{\theta\leftarrow\theta} + y^{(\theta)}_{i,\lambda}\big|_{a\leftarrow a}, \tag{A.66}$$

with $y^{(\theta)}_{i,\lambda}\big|_{\theta\leftarrow\theta}$ and $y^{(\theta)}_{i,\lambda}\big|_{a\leftarrow a}$ given in Eqs. (A.61) and (A.65).

Amplitude components of the output: $y^{(a)}_\lambda$

Now consider the component $y^{(a)}_\lambda$, associated to parameter a_λ. It is sourced by both angle-type inputs and amplitude-type inputs:

$$y^{(a)}_\lambda = y^{(a)}_\lambda\big|_{\theta\leftarrow\theta} + y^{(a)}_\lambda\big|_{a\leftarrow a}.$$

(i) Amplitude–angle block ($\theta \leftarrow \theta$). For parameters $(a_\lambda, \theta^\mu_j)$:

$$S^{(a,\theta)}_{(a,\lambda),(j,\mu)} = \langle \partial_{a_\lambda}\Psi | \partial_{\theta^\mu_j}\Psi\rangle - \langle \partial_{a_\lambda}\Psi|\Psi\rangle\ \langle\Psi|\partial_{\theta^\mu_j}\Psi\rangle \tag{A.67}$$

$$= \Big[a_\mu\, C_{\lambda\mu}\, t^{\lambda\mu}_j\Big]\ -\ A_\lambda\, D_{\mu j} \tag{A.68}$$

$$= R_{\lambda\mu}\, t^{\lambda\mu}_j\ -\ A_\lambda\, D_{\mu j}. \tag{A.69}$$

Acting on $v^{(\theta)}_{j,\mu}$:

$$y^{(a)}_\lambda\big|_{\theta\leftarrow\theta} = \sum_\mu \sum_j S^{(a,\theta)}_{(a,\lambda),(j,\mu)}\ v^{(\theta)}_{j,\mu}$$

$$= \sum_{\mu,j} \Big[R_{\lambda\mu}\, t^{\lambda\mu}_j\ -\ A_\lambda\, D_{\mu j}\Big] v^{(\theta)}_{j,\mu}. \tag{A.70}$$

(ii) Amplitude–amplitude block ($a \leftarrow a$). For parameters (a_λ, a_μ):

$$S^{(a,a)}_{(a,\lambda),(a,\mu)} = \langle \partial_{a_\lambda}\Psi | \partial_{a_\mu}\Psi\rangle - \langle \partial_{a_\lambda}\Psi|\Psi\rangle\ \langle\Psi|\partial_{a_\mu}\Psi\rangle \tag{A.71}$$

$$= C_{\lambda\mu}\ -\ A_\lambda\, E_\mu. \tag{A.72}$$

Its action on $v^{(a)}_\mu$ is

$$y^{(a)}_\lambda\big|_{a\leftarrow a} = \sum_\mu S^{(a,a)}_{(a,\lambda),(a,\mu)}\ v^{(a)}_\mu$$

$$= \sum_\mu \Big[C_{\lambda\mu}\ -\ A_\lambda\, E_\mu\Big] v^{(a)}_\mu. \tag{A.73}$$

(iii) Total amplitude output. Thus

$$y^{(a)}_\lambda = y^{(a)}_\lambda\big|_{\theta\leftarrow\theta} + y^{(a)}_\lambda\big|_{a\leftarrow a}, \tag{A.74}$$

with $y^{(a)}_\lambda\big|_{\theta\leftarrow\theta}$ and $y^{(a)}_\lambda\big|_{a\leftarrow a}$ given in Eqs. (A.70) and (A.73).

Summary

Eqs. (A.66) and (A.74) together provide the full matrix-free action

$$y = S_{\bar{w}} v,$$

i.e. they tell us how to compute the product of the TDVP metric tensor $S_{\bar{w}}$ with an arbitrary parameter-space vector v without ever assembling or inverting $S_{\bar{w}}$ as a dense matrix. This is precisely what is required by Krylov solvers such as MINRES-QLP, which only rely on repeated evaluations of $v \mapsto S_{\bar{w}} v$.

A.5.2 Explicit $S_{\bar{w}}$ vs. operator form, and Krylov convergence

At each TDVP step we solve the linear system

$$S_{\bar{w}}\,\delta\bar{w} = -\gamma\,F_{\bar{w}}, \qquad \bar{w} = \{a_\lambda, \theta_i^\lambda\}, \tag{A.75}$$

where $S_{\bar{w}}$ is the TDVP (stochastic reconfiguration) metric tensor, and $F_{\bar{w}}$ is the corresponding force vector. We have implemented and benchmarked two numerically distinct ways of applying this update:

1. **Dense-matrix approach (S_{matrix}).** We explicitly build the restricted TDVP metric $S_{\bar{w}}$ as a dense matrix in the effective subspace spanned by the currently "active" replicas $\{\rho_\lambda\}$ (the top-coherence configurations we keep after pruning). We then regularize it with a small diagonal shift $S_{\bar{w}} \to S_{\bar{w}} + \epsilon\,\mathbb{I}$, and solve the linear system either by a direct dense solver or by a Krylov method on that explicit matrix.

2. **Matrix-free / operator approach (S_{op}).** We never construct $S_{\bar{w}}$ as a matrix. Instead, we provide MINRES-QLP with a *callable linear operator*

   $$v \longmapsto S_{\bar{w}} v,$$

   implemented analytically from the closed-form TDVP metric expressions $\langle\partial_{w_j}\Psi|\partial_{w_k}\Psi\rangle - \langle\partial_{w_j}\Psi|\Psi\rangle\,\langle\Psi|\partial_{w_k}\Psi\rangle$. This completely avoids storing a $(D_{\text{eff}}{+}D_{\text{eff}}L)\times(D_{\text{eff}}{+}D_{\text{eff}}L)$ dense matrix, which is essential when D_{eff} and L become large.

Both approaches are *formally correct*: they generate TDVP updates that drive the parameters $\{a_\lambda, \theta_i^\lambda\}$ along a projected imaginary-time descent direction that lowers the cost function in the restricted variational manifold.

However, in practice we observe a striking numerical difference: *MINRES-QLP needs dramatically more iterations to converge when using the operator form S_{op} than when using the explicit matrix S_{matrix}, even with the same tolerance (`rtol` $= 10^{-10}$), the same maximum iteration cap (`maxit` $= 10^4$), and the same effective replica subspace.* Concretely, with S_{matrix} the Krylov solve typically converges in a relatively small number of iterations. With S_{op} under identical conditions, MINRES-QLP often either takes orders of magnitude more iterations, or hits the iteration cap before fully reducing the residual.

This behaviour is consistent with three numerical effects:

(i) Near-singular conditioning due to gauge-like directions. In our ansatz the amplitudes a_λ and the angles θ_i^λ are not all physically independent. There are directions in parameter space that change the overall normalization of $|\Psi\rangle$, or reshuffle weight among replicas without altering the physical state very much. Those directions show up as very small eigenvalues of $S_{\bar{w}}$, i.e. $S_{\bar{w}}$ is extremely ill-conditioned. When we build S_{matrix} on the *already-pruned* effective set of replicas, we are implicitly removing a lot of this redundancy, which improves the effective conditioning. By contrast, in the pure operator version S_{op} the Krylov solver still "sees" essentially all those almost-null (gauge-like) directions. MINRES-QLP then interprets the system as nearly singular and keeps iterating in an attempt to find a numerically stable solution.

(ii) Imperfect symmetry / projector enforcement at the operator level. By construction,

$$S_{\bar{w}} = \langle\partial\Psi|\partial\Psi\rangle - \langle\partial\Psi|\Psi\rangle\,\langle\Psi|\partial\Psi\rangle$$

is supposed to be real-symmetric and (after projection orthogonal to $|\Psi\rangle$) positive semi-definite. In the explicit S_{matrix} path, we assemble all blocks (aa, $a\theta$, θa, $\theta\theta$), subtract the projector terms, and then *symmetrize* numerically. We also add a small diagonal regularizer $\epsilon\mathbb{I}$, which both enforces symmetry and lifts tiny eigenvalues.

In the S_{op} path, symmetry and the projector subtraction are reproduced "on the fly" inside the matvec routine $v \mapsto S_{\bar{w}} v$. Tiny floating-point differences (loop ordering, rounding in the $\tan(\theta^{\lambda} - \theta^{\mu})$ terms, etc.) mean that the operator that MINRES-QLP actually experiences is only *approximately* symmetric and positive-definite. Krylov methods like MINRES-QLP are very sensitive to this: even a small violation of symmetry or a slight drift in the null space projection can drastically worsen the effective conditioning seen by the solver, causing the residual to decay much more slowly.

In summary, the matrix-free S_{op} implementation is algorithmically correct; however, in practice, it is *much harder* for MINRES-QLP to solve to a high tolerance, and the solver tends to hit the maximum number of iterations. The explicit S_{matrix} approach converges in far fewer Krylov iterations under the same stopping criteria because (a) it works in a pruned subspace with fewer gauge-like directions, (b) we can explicitly symmetrize and regularize the matrix, and (c) the resulting linear system is effectively better conditioned.

Because of this, our pragmatic strategy is:

- For "small" and "medium" instances, where the effective subspace size D_{eff} is still numerically manageable and the dense S_{matrix} fits in memory, we **prefer the explicit matrix solve**. It is simply faster in wall-clock time because Krylov converges in a handful of iterations.

- We reserve the fully matrix-free S_{op} route for **larger systems**, where storing S_{matrix} becomes prohibitive in memory. In that regime, even if MINRES-QLP requires many iterations, S_{op} is still our only viable option: it allows us to apply the TDVP update without ever materializing a $(D_{\text{eff}} + D_{\text{eff}} L)^2$ matrix.

In other words, S_{op} is the scalable path, but it comes with a strong numerical penalty in terms of the Krylov iteration count. S_{matrix} is numerically kinder and converges quickly, but it does not scale in memory once the manifold (i.e. the number of replicas ρ_{λ} times the number of angles per replica) becomes too large.

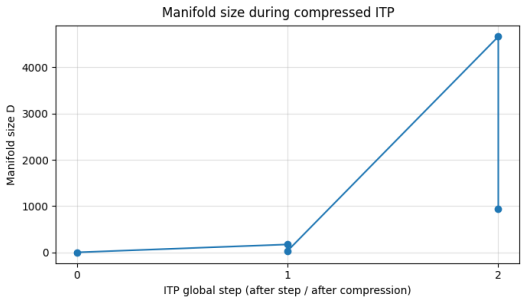


Figure A.1: Manifold size D during two-step ITP with soft compression for the $L = 40$, $M = 172$ instance.

A.6 Soft-compressed frontiers at $L = 40$ and $L = 50$

In this appendix, we collect the diagnostics for the *softly* compressed two-step frontiers used in Sec. 7.2.3, focusing on the same $L = 40$ and $L = 50$ instances discussed there. The goal is not to provide a detailed case-by-case analysis, but simply to contrast soft compression with the aggressively compressed frontiers shown in the main text.

Overall, the picture is consistent across both sizes. For the hard $L = 40$ instance, the soft-compressed frontier manages to lower the energy but keeps an extremely small probability on the satisfying assignments, which is much smaller than in the aggressively compressed run. For the solved $L = 50$ instance, soft and aggressive compression lead to essentially the same convergence behavior in both energy and solution probability, but the soft scheme does so with a manifold that is *much* larger than M+1. In both regimes, therefore, the soft procedure is dominated by the aggressive one: it is either less effective at building solution weight (at $L = 40$), or no more accurate but substantially more expensive (at $L = 50$).

A.6.1 $L = 40$, $M = 172$: failed instance with soft compression

Figure A.1 shows the growth of the manifold size during two-step ITP with soft sensitivity-based compression for the hard $L = 40$, $M = 172$ instances. The final frontier remains of order 10^3 replicas, much smaller than the hypothetical uncompressed two-step frontier but considerably larger than the aggressively compressed manifold of Fig. 7.26.

The corresponding TDVP diagnostics are reported in Figs. A.4–A.2. The total probability of solutions remains extremely small throughout the run, despite a clear overall decay of the energy. Compared to the aggressive case in Figs. 7.28–7.29, soft compression yields a lower solution probability at comparable or higher computational costs.

A.6.2 $L = 50$, $M = 215$: solved instance with soft compression

We now turn to the solved $L = 50$, $M = 215$ instance considered in Sec. 7.2.3. Figure A.3 shows the manifold size during ITP with soft compression: the two-step frontier reaches $D_{\text{final}} \sim 2 \times 10^3$ replicas before TDVP, to be compared with the aggressively compressed frontier in Fig. 7.27, which enforces $D_{\text{final}} = M + 1 = 216$.

Despite this large difference in manifold size, the TDVP dynamics is qualitatively very similar for soft and aggressive compression. As shown in Fig. A.6, the total probability on solutions and the probability of the best solution undergo a sharp freezing transition and saturate close to one, while the energy decays rapidly to the ground-state value. These trends mirror those of Fig. 7.31 and Fig. 7.32 for the aggressively compressed frontier.

The replica-selection histogram in Fig. A.5 confirms that the TDVP active set concentrates on a relatively small subset of energetically important replicas. However, because the soft manifold is much larger than M+1, each TDVP step is significantly more expensive than in

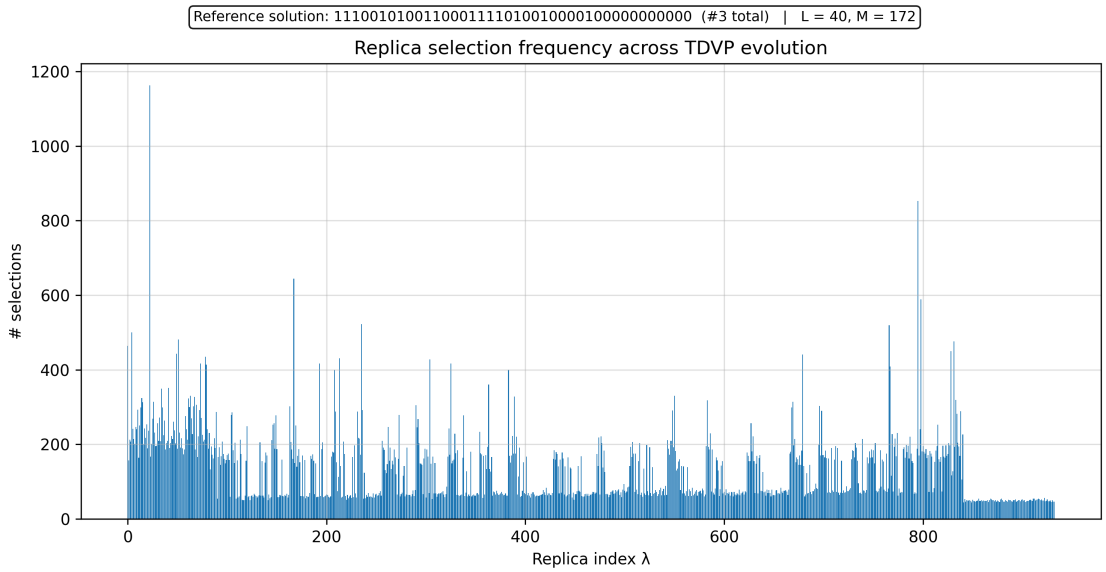


Figure A.2: Replica-selection frequency across the TDVP evolution for the softly compressed $L = 40$ frontier. The active set keeps revisiting a relatively small subset of replicas, but without ever producing a macroscopic solution weight.

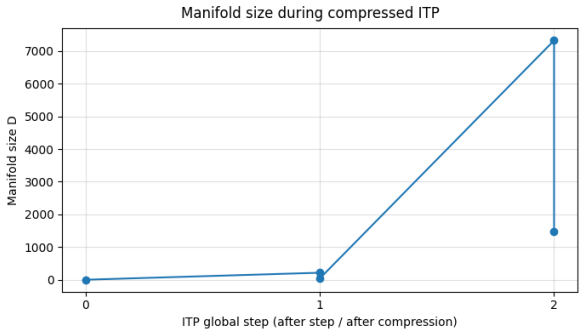
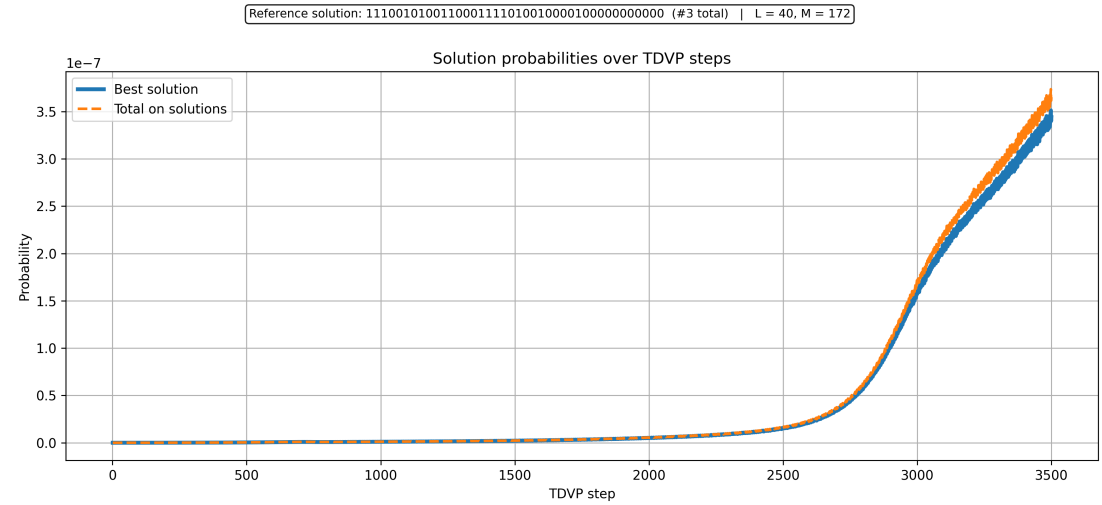
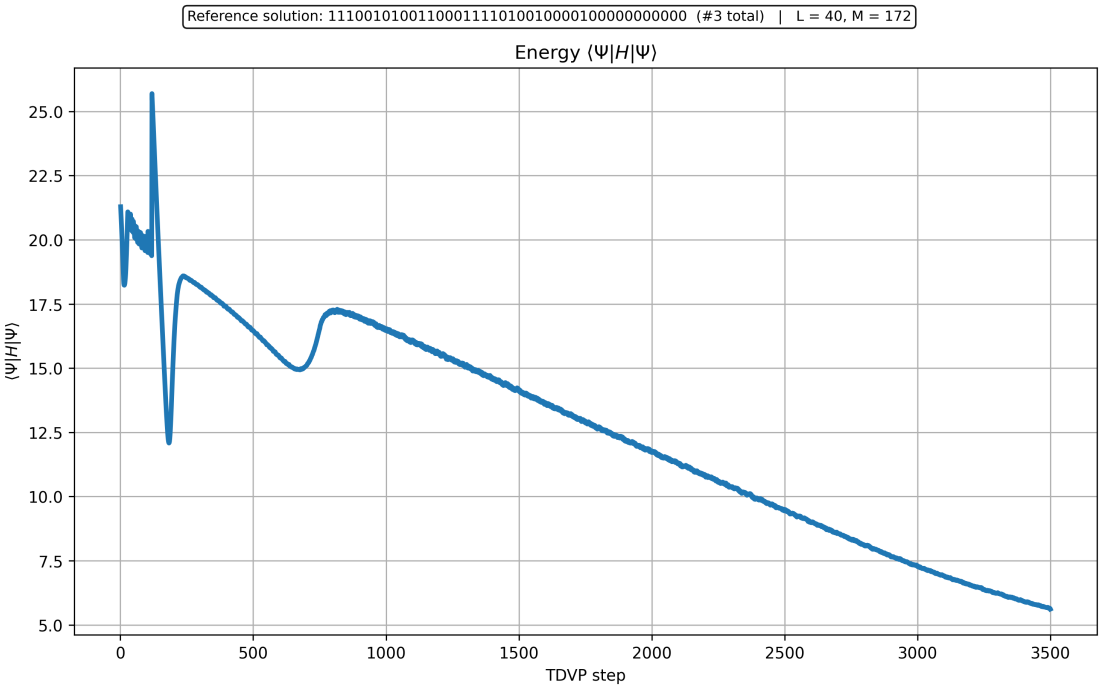


Figure A.3: Manifold size D during two-step ITP with soft compression for the solved $L = 50$, $M = 215$ instance.

the aggressive case. In this sense, the soft scheme offers no clear accuracy advantage for this instance while incurring a much higher computational cost.



(a) Solution probabilities.



(b) Energy expectation $\langle\Psi|H|\Psi\rangle$.

Figure A.4: TDVP diagnostics for the softly compressed two-step frontier at $L = 40$, $M = 172$. The energy decreases significantly, but the total probability on satisfying assignments remains orders of magnitude below the aggressively compressed case of Figs. 7.28 and 7.29.

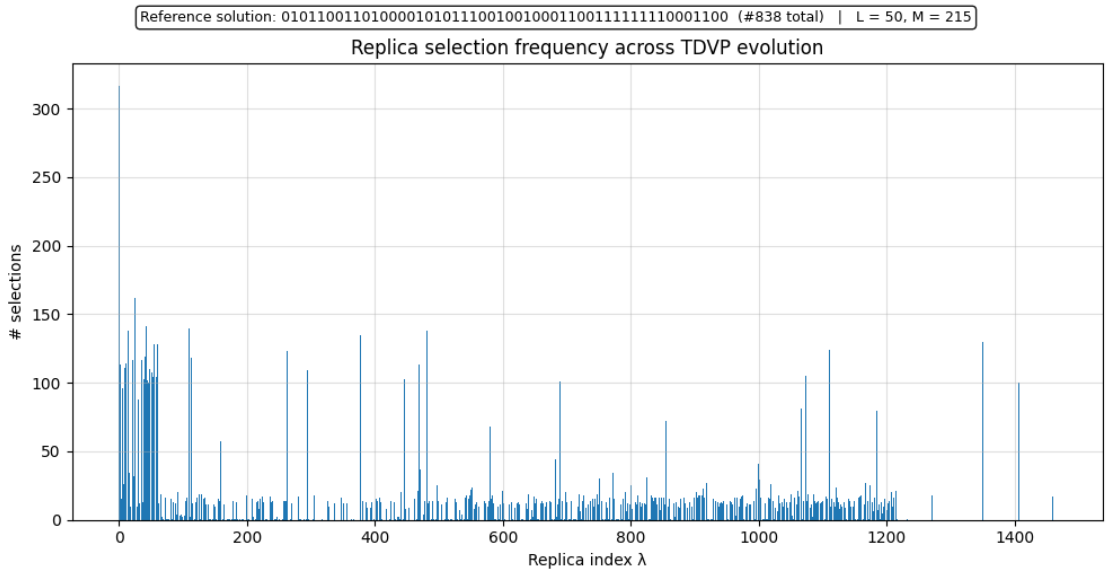
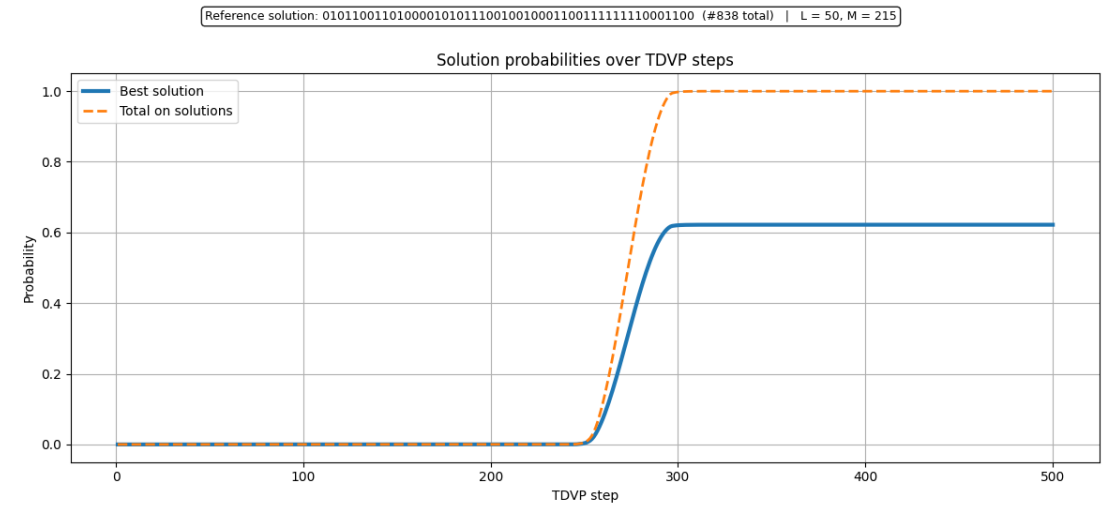
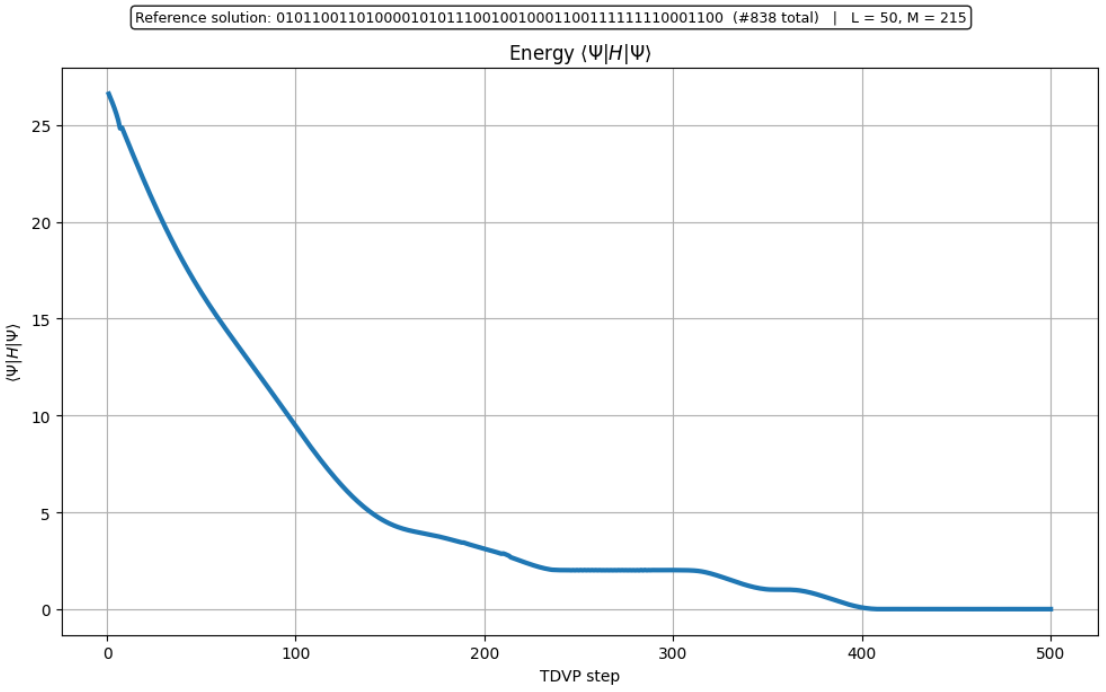


Figure A.5: Replica-selection frequency across the TDVP evolution for the softly compressed $L = 50$ frontier. As in Fig. 7.33, a small backbone of replicas is selected very often, but here it is embedded in a much larger manifold.



(a) Solution probabilities.



(b) Energy expectation $\langle\Psi|H|\Psi\rangle$.

Figure A.6: TDVP diagnostics for the softly compressed two-step frontier at $L = 50$, $M = 215$. The convergence to the satisfying manifold and to the ground-state energy is essentially equivalent to that of the aggressively compressed frontier in Figs. 7.31 and 7.32, but achieved with a much larger manifold.

Bibliography

[1] Rémi Monasson, Riccardo Zecchina, Scott Kirkpatrick, Bart Selman, and Lidror Troyansky. «2+p-SAT: Relation of Typical-Case Complexity to the Nature of the Phase Transition». In: *Random Structures & Algorithms* 15.3–4 (1999), pp. 414–435 (cit. on pp. 1, 7–9, 93).

[2] Marc Mézard, Giorgio Parisi, and Riccardo Zecchina. «Analytic and Algorithmic Solution of Random Satisfiability Problems». In: *Science* 297.5582 (2002), pp. 812–815 (cit. on pp. 1, 7–12, 14, 93).

[3] Rémi Monasson and Riccardo Zecchina. «Statistical mechanics of the random K-satisfiability model». In: *Physical Review E* 56.2 (1997), pp. 1357–1370 (cit. on pp. 1, 7, 12, 14).

[4] Frédéric Lafitte, Jorge Nakahara Jr., and Dirk Van Heule. «Applications of SAT Solvers in Cryptanalysis: Finding Weak Keys and Preimages». In: *Journal on Satisfiability, Boolean Modeling and Computation* 9 (2014), pp. 1–25 (cit. on p. 1).

[5] Vassiliy Lubchenko and Peter G. Wolynes. «Theory of Structural Glasses and Supercooled Liquids». In: *Annual Review of Physical Chemistry* 58 (2007), pp. 235–266 (cit. on pp. 1, 12).

[6] Bernd R. Schuh. «Easy/Hard Transition in k-SAT». In: *arXiv preprint arXiv:1411.2901* (2014) (cit. on pp. 1, 9).

[7] Florent Krzakala, Andrea Montanari, Federico Ricci-Tersenghi, Guilhem Semerjian, and Lenka Zdeborová. «Gibbs States and the Set of Solutions of Random Constraint Satisfaction Problems». In: *Proceedings of the National Academy of Sciences of the United States of America* 104.25 (2007), pp. 10318–10323 (cit. on pp. 1, 8, 9, 11, 12, 93).

[8] Theodore R. Kirkpatrick and Devarajan Thirumalai. «p-spin-interaction spin-glass models: Connections with the structural glass problem». In: *Physical Review B* 36.10 (1987), pp. 5388–5397 (cit. on pp. 1, 8, 93).

[9] T. Kadowaki and H. Nishimori. «Quantum annealing in the transverse Ising model». In: *Physical Review E* 58.5 (1998), pp. 5355–5363. DOI: 10.1103/PhysRevE.58.5355 (cit. on p. 1).

[10] M. Cerezo et al. «Variational quantum algorithms». In: *Nature Reviews Physics* 3 (2021), pp. 625–644. DOI: 10.1038/s42254-021-00348-9 (cit. on pp. 1, 33).

[11] Arnab Das and Bikas K. Chakrabarti. «Colloquium: Quantum annealing and analog quantum computation». In: *Reviews of Modern Physics* 80 (2008), pp. 1061–1081. DOI: 10.1103/RevModPhys.80.1061 (cit. on p. 1).

[12] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, Joshua Lapan, Andrew Lundgren, and Daniel Preda. «A Quantum Adiabatic Evolution Algorithm Applied to Random Instances of an NP-Complete Problem». In: *Science* 292.5516 (2001), pp. 472–475. DOI: 10.1126/science.1057726 (cit. on p. 1).

[13] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. «A Quantum Approximate Optimization Algorithm». In: *arXiv preprint arXiv:1411.4028* (2014). arXiv: 1411.4028 [quant-ph] (cit. on p. 1).

[14] Jarrod R. McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. «The theory of variational hybrid quantum-classical algorithms». In: *New Journal of Physics* 18.2 (2016), p. 023023. DOI: 10.1088/1367-2630/18/2/023023 (cit. on pp. 1, 33).

[15] Sebastian Zielinski, Jonas Nüßlein, Jonas Stein, Thomas Gabor, Claudia Linnhoff-Popien, and Sebastian Feld. «Pattern QUBOs: Algorithmic construction of 3SAT-to-QUBO transformations». In: *arXiv preprint arXiv:2305.02659* (2023) (cit. on pp. 6, 12).

[16] M. Mézard and R. Zecchina. «The random K-satisfiability problem: From an analytic solution to an efficient algorithm». In: *Physical Review E* 66.5 (2002). Also available as arXiv:cond-mat/0207194, p. 056126. DOI: 10.1103/PhysRevE.66.056126 (cit. on pp. 7–12, 14).

[17] Haijun Zhou. «Long-range frustration in finite connectivity spin glasses: A mean-field theory and its application to the random K-satisfiability problem». In: *New Journal of Physics* 7 (2005). Also available as arXiv:cond-mat/0411079, p. 123. DOI: 10.1088/1367-2630/7/1/123 (cit. on pp. 8, 14).

[18] Eren C. Kızıldağ. «Sharp Thresholds for the Overlap Gap Property: Ising p-Spin Glass and Random k-SAT». In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2025)*. Vol. 353. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025, 48:1–48:16. DOI: 10.4230/LIPIcs.APPROX/RANDOM.2025.48. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.APPROX/RANDOM.2025.48 (cit. on pp. 8, 14).

[19] Wolfgang Barthel, Alexander K. Hartmann, Michele Leone, Federico Ricci-Tersenghi, Martin Weigt, and Riccardo Zecchina. «Hiding Solutions in Random Satisfiability Problems: A Statistical Mechanics Approach». In: *Physical Review Letters* 88.18 (2002), p. 188701 (cit. on pp. 10–12).

[20] Haixia Jia, Cristopher Moore, and Bart Selman. «From Spin Glasses to Hard Satisfiable Formulas». In: *Theory and Applications of Satisfiability Testing, SAT 2004: 7th International Conference on Theory and Applications of Satisfiability Testing, Vancouver, BC, Canada, May 10–13, 2004, Revised Selected Papers*. Vol. 3542. Lecture Notes in Computer Science. Springer, 2005, pp. 199–210 (cit. on pp. 12, 14).

[21] W. M. C. Foulkes, Lubos Mitas, R. J. Needs, and G. Rajagopal. «Quantum Monte Carlo simulations of solids». In: *Reviews of Modern Physics* 73.1 (2001), pp. 33–83. DOI: 10.1103/RevModPhys.73.33 (cit. on pp. 15–17).

[22] Peter van Nieuwenhuizen. *On Euclidean spinors and Wick rotations*. Phys. Lett. B. See also standard discussions of Wick rotation relating quantum dynamics to statistical mechanics. 1991 (cit. on pp. 15, 16).

[23] Sam McArdle, Tyson Jones, Suguru Endo, Ying Li, Simon C. Benjamin, and Xiao Yuan. «Variational ansatz-based quantum simulation of imaginary time evolution». In: *npj Quantum Information* 5.1 (2019), p. 75. DOI: 10.1038/s41534-019-0187-2 (cit. on pp. 15, 17).

[24] Xiao Yuan, Suguru Endo, Qi Zhao, Ying Li, and Simon C. Benjamin. «Theory of variational quantum simulation». In: *Quantum* 3 (2019), p. 191. DOI: 10.22331/q-2019-10-07-191 (cit. on pp. 15, 17).

[25] Sou-Cheng T. Choi, Christopher C. Paige, and Michael A. Saunders. «MINRES-QLP: A Krylov Subspace Method for Indefinite or Singular Symmetric Systems». In: *SIAM Journal on Scientific Computing* 33.4 (2011), pp. 1810–1836. DOI: 10.1137/100783504 (cit. on pp. 32, 40, 42, 43, 108).

[26] A. D. McLachlan. «A variational solution of the time-dependent Schrödinger equation». In: *Molecular Physics* 8.1 (1964), pp. 39–44. DOI: 10.1080/00268976400100041 (cit. on p. 32).

[27] Caroline Lasser and Chunmei Su. «Various variational approximations of quantum dynamics». In: *Journal of Mathematical Physics* 63.7 (2022), p. 072107. DOI: 10.1063/5.0088265 (cit. on p. 32).

[28] Sandro Sorella. «Green Function Monte Carlo with Stochastic Reconfiguration». In: *Physical Review Letters* 80.20 (1998), pp. 4558–4561. DOI: 10.1103/PhysRevLett.80.4558 (cit. on pp. 33, 35).

[29] Sandro Sorella and Luca Capriotti. «Green function Monte Carlo with stochastic reconfiguration: An effective remedy for the sign problem». In: *Physical Review B* 61.4 (2000), pp. 2599–2612. DOI: 10.1103/PhysRevB.61.2599 (cit. on pp. 33, 35).

[30] Raphael Kaubruegger, Lorenzo Pastori, and Jan Carl Budich. «Chiral Topological Phases from Artificial Neural Networks». In: *Physical Review B* 97.19 (2018), p. 195136. DOI: 10.1103/PhysRevB.97.195136. arXiv: 1710.04713 [cond-mat.str-el] (cit. on pp. 33, 35, 39, 40, 43, 63).

[31] Jutho Haegeman, J. Ignacio Cirac, Tobias J. Osborne, Iztok Pizorn, Henri Verschelde, and Frank Verstraete. «Time-Dependent Variational Principle for Quantum Lattices». In: *Physical Review Letters* 107.7 (2011), p. 070601. DOI: 10.1103/PhysRevLett.107.070601 (cit. on p. 33).

[32] Jutho Haegeman, Tobias J. Osborne, and Frank Verstraete. «Post-matrix product state methods: To tangent space and beyond». In: *Physical Review B* 88.7 (2013), p. 075133. DOI: 10.1103/PhysRevB.88.075133 (cit. on p. 33).

[33] Jutho Haegeman, Christian Lubich, Ivan Oseledets, Bart Vandereycken, and Frank Verstraete. «Unifying time evolution and optimization with matrix product states». In: *Physical Review B* 94.16 (2016), p. 165116. DOI: 10.1103/PhysRevB.94.165116 (cit. on p. 33).

[34] Giuseppe Carleo and Matthias Troyer. «Solving the quantum many-body problem with artificial neural networks». In: *Science* 355.6325 (2017), pp. 602–606. DOI: 10.1126/science.aag2302 (cit. on pp. 33, 35).

[35] Giuseppe Carleo et al. «NetKet: A machine learning toolkit for many-body quantum systems». In: *SoftwareX* 10 (2019), p. 100311. DOI: 10.1016/j.softx.2019.100311 (cit. on p. 33).

[36] Irene López Gutiérrez and Christian B. Mendl. «Real time evolution with neural-network quantum states». In: *Quantum* 6 (2022), p. 627. DOI: 10.22331/q-2022-01-20-627 (cit. on p. 33).