Politecnico di Torino

Politecnico di Torino

Master's Degree in Computer Engineering

A.a. 2025/2026

Graduation Session March 2026

Topology-Aware and Multi-Target Transfer Learning for Vascular Graph Extraction

Supervisors:

Paolo Garza

Maria A. Zuluaga

Candidate:

Alessandro Scavone

Abstract

Accurate extraction of vascular graphs from medical images is essential for quantitative analysis of vascular morphology, enabling downstream tasks such as cognitive disease assessment, hemodynamic modeling, and temporal analysis of structural changes. Despite the importance of this task, several challenges remain, including the sparsity of graph annotations, variability across imaging modalities, and the need to preserve global topological consistency. Recent transformer-based architectures enable end-to-end inference of structured representations but remain highly sensitive to domain shifts and data scarcity, particularly when transferring across imaging modalities or dimensionalities. This work builds on a single-stage image-to-graph transformer and presents a semantic-aware transfer-learning framework for vascular graph extraction that leverages biologically meaningful structural priors. Rather than relying on artificial structures for auxiliary supervision, botanical branching structures are used as a source domain, motivated by shared growth principles and fractal topologies observed in plant venation and animal vasculature. In addition, a multi-target training strategy across heterogeneous vessel datasets is adopted to promote robust relational reasoning and enable zero-shot generalization to unseen domains. The proposed framework is evaluated across multiple experimental settings, including cross-domain transfer, multi-modal training, and zero-shot generalization, using established graph similarity and detection metrics. Overall, the results highlight biologically grounded transfer learning as a principled direction for scalable and generalizable vascular graph extraction from medical images.

Acknowledgements

I would like to express my gratitude to Prof. Paolo Garza for his assistance, availability, and support throughout my time abroad and the organization of this thesis. A special thanks to Maria A. Zuluaga, who taught me a great deal during these six months and was always available, providing extraordinary guidance and mentorship.

I would also like to thank my lifelong friends Matteo, Gianluca, and Andrea for always being there, for believing in what I was doing, and for their constant support over the years. I am also grateful to all my friends who have been part of my life over the years, even without knowing it, you were always there during difficult times, making those days more enjoyable. Thank you all. I could not imagine going through this journey again without you.

A big thank you to the AI4Health team, Daniele, Vincenzo, Matteo, Xiaoming, Luisa, and Natalia, for sharing your time, knowledge, lunches, Pizza Thursdays, and Pollo Fridays with me. You made the whole experience even more special. Vincenzo, you have been a constant example through your help and guidance. Thank you for all the time you spent (hopefully not wasted) helping me solve problems or suggesting films and music. You taught me more than you know.

I am really grateful to my girlfriend Emna, without whom these six months would not have been the same. You were always there to support me when I needed it, with love, and a lot of patience too. Thank you for sharing both the joys and the struggles of this experience with me, and for making every moment lighter by going through it together.

Infine vorrei ringraziare tutta la mia famiglia. Grazie mamma e papà per essere sempre stati presenti e per non avermi mai fatto mancare l'amore e il supporto durante tutto il mio percorso di studi e nella vita quotidiana. Grazie per tutti i sacrifici che avete fatto per permettermi di vivere la vita migliore possibile e per avermi aiutato anche quando volevo fare tutto da solo. Grazie infinite.

Table of Contents

List of Figures VII

1 Introduction 1

2 Background and Related Work 3
- 2.1 Image-to-Graph Transformers . . . 3
- 2.2 Transfer Learning and Domain Adaptation . . . 6
  - 2.2.1 Cross-Domain and Adversarial Transfer Learning . . . 6
  - 2.2.2 Cross-Dimension Transfer Learning . . . 8
  - 2.2.3 Sparse Relational Supervision and Topology Preservation . . . 9
- 2.3 Biological and Semantic Priors . . . 11
- 2.4 Multi-Target Scaling and Zero-Shot Generalization . . . 12
  - 2.4.1 Data Aggregation . . . 12
  - 2.4.2 Topology-Preserving Pipelines . . . 13
  - 2.4.3 Scaling and Zero-Shot Generalization . . . 14

3 Methods 15
- 3.1 Base Architecture: Relationformer . . . 15
  - 3.1.1 Object Prediction . . . 16
  - 3.1.2 Relation Prediction . . . 18
  - 3.1.3 Architecture . . . 19
  - 3.1.4 Relationformer Loss Function . . . 20
- 3.2 Berger's Framework and Setup . . . 21
  - 3.2.1 Problem Setting and Notation . . . 22
  - 3.2.2 Regularized Edge Sampling Loss . . . 23
  - 3.2.3 Supervised Domain Adaptation . . . 25
  - 3.2.4 Combined Training Objective . . . 28
  - 3.2.5 2D-to-3D Transfer Learning Framework . . . 30
- 3.3 Botanical Source Domain Selection . . . 33
  - 3.3.1 Biological Transport Networks as Structural Priors . . . 33
  - 3.3.2 From Internal Vasculature to External Branching . . . 34

3.3.3 Implications for Graph-Based Representation Learning . . . 35
3.3.4 Role in the Transfer Learning Framework . . . 35
3.3.5 Degree-2 Node Retention . . . 36
3.4 Multi-Target Diversity for Dataset Zero-Shot Learning . . . 37
3.4.1 Problem Setting . . . 37
3.4.2 Capped Per-Dataset Contribution . . . 38
3.4.3 Two Target-Scale Regimes . . . 39
3.4.4 Sampling Within Training Batches . . . 40

4 Datasets Construction and Preprocessing Pipeline 41
4.1 Ground-Truth Graph Construction Framework . . . 41
4.1.1 Voreen Graph Extraction Tool . . . 42
4.1.2 Centerline-Based Graph Generation . . . 45
4.1.3 Dataset-Specific Graph Post-Processing . . . 47
4.2 Patch Extraction Framework . . . 47
4.2.1 Botanical and 2D Datasets . . . 48
4.2.2 3D Vessels Patch Extraction . . . 50
4.2.3 Patch Filtering and Constraints . . . 52
4.3 Botanical Data Preparation Pipeline . . . 53
4.3.1 Graph-Derived Segmentation Reconstruction . . . 53
4.3.2 Procedural Generation of 3D Input Images . . . 54

5 Experiments and Results 56
5.1 Datasets . . . 56
5.2 Data Preprocessing and Graph Construction . . . 57
5.3 Implementation Details . . . 59
5.4 Evaluation Metrics . . . 62
5.4.1 Object Detection Metrics . . . 62
5.4.2 Topological Metrics . . . 63
5.4.3 Graph Distance Metric . . . 63
5.5 Experiments . . . 64
5.5.1 Structural Source-Domain Comparison . . . 64
5.5.2 Sample Efficiency Analysis . . . 65
5.5.3 Multi-Target Training for Generalization . . . 66
5.6 Results . . . 67
5.6.1 Structural Source-Domain Comparison . . . 68
5.6.2 Sample Efficiency Analysis . . . 70
5.6.3 Multi-Target Training for Generalization . . . 71

6 Discussion 78
6.1 Summary of Contributions and Main Findings 78
6.2 Structural Congruence as an Inductive Bias 79
6.3 Sample Efficiency and the Annotation Tax 81
6.4 Multi-Target Scaling and Zero-Shot Generalization 82
6.5 Limitations 85

7 Conclusions and Future Work 88
7.1 Conclusions 88
7.2 Future Work 89

A Additional Experimental Results 92
A.1 Qualitative Comparison of Pretraining Domains 92
A.2 Statistical Significance Analysis 92
A.3 Metric Distributions for Scale-Expanded Training 93

Bibliography 97

List of Figures

2.1 Traditional multi-stage pipeline for vascular graph extraction. The input image is first segmented to obtain a binary vessel mask, followed by skeletonization to derive centerlines, and finally graph construction through iterative pruning and post-processing. Errors introduced at each stage accumulate across the pipeline, often leading to fragmented connectivity and loss of topological consistency. 4

2.2 End-to-end image-to-graph transformer framework. The input image is processed by a CNN backbone for feature extraction, followed by a vision transformer that jointly predicts nodes and edges through dedicated object and relation heads. Unlike traditional multi-stage pipelines, segmentation and skeletonization are bypassed, enabling direct inference of the vascular graph and reducing error propagation across intermediate steps. 5

2.3 Overview of cross-domain and cross-dimension transfer learning for image-to-graph transformation. Structural knowledge learned from a source domain is encoded into a shared latent representation and transferred to a visually distinct target domain to predict graph topology. The framework emphasizes domain-invariant connectivity and branching patterns rather than low-level image appearance. . . 7

2.4 Overview of the cross-dimension image-to-graph learning framework. The model jointly processes 2D and 3D inputs within a shared structural learning setup, enabling knowledge transfer across dimensionalities and producing a unified 3D graph prediction. 8

2.5 Illustration of relation imbalance in graph prediction. Left: all possible unordered node pairs forming the set of candidate edges, which grows quadratically with the number of nodes. Right: the sparse subset of true edges representing valid graph connectivity. The large discrepancy between candidate and true relations (candidate edges $\gg$ true edges) leads to severe class imbalance during relation learning. 10

3.1 Overview of the proposed topology-aware transfer learning framework. A botanical source domain (2D data) and multiple vascular target domains (2D or 3D) are processed through a unified preprocessing and graph construction pipeline, including patch extraction, deterministic graph extraction, and 2D-to-3D projection. The resulting image-graph pairs are used to train a Relationformer-based model that jointly predicts nodes and relations, enabling structurally aligned cross-domain and cross-dimension learning. 16

3.2 Relationformer architecture from Shit et al. [13], composed of a CNN backbone for feature extraction, a transformer to extract rich object tokens, and two heads for object and relation prediction, yielding the output relational graph. 17

3.3 Supervised domain adaptation framework adopted from Berger et al. [2]. Image-level features extracted by the CNN backbone and graph-level embeddings produced by the transformer are each passed through a Gradient Reversal Layer (GRL) and corresponding domain classifiers. Adversarial training encourages domain-invariant representations at both visual and relational levels, while a consistency regularization term aligns image- and graph-level domain predictions. 28

3.4 Image-to-graph transformer with set-based prediction and Hungarian matching. The input image is encoded into global feature representations, which are decoded using a fixed set of learnable object queries. Each query produces a node hypothesis with coordinates and class probabilities (node vs. no-object). Hungarian matching performs a global one-to-one assignment between predictions and ground-truth nodes by minimizing a joint classification and localization cost. Colored arrows indicate matched correspondences. The greyed node and crossed arrow illustrate an unmatched prediction, which is supervised as no-object during training and removed at inference. 29

3.5 Schematic illustration of the 2D-to-3D projection function used for cross-dimension transfer learning in the considered image-to-graph framework. A 2D image with its associated graph is first resized and embedded as a planar slice within the center of an empty cubic 3D volume, while preserving the graph topology and relative node positions. The embedded image-graph pair is then subjected to a shared random 3D rotation, resulting in a rotated planar representation inside the volume. This simple projection enables the use of labeled 2D image-graph data for training 3D image-to-graph models without requiring handcrafted projections or model modifications, following the framework introduced by Berger et al [2]. 32

3.6 Illustration of degree-2 node retention. Intermediate nodes are preserved along vessel centerlines to maintain branch curvature and geometric continuity, providing a more faithful representation of the underlying vascular topology. 37

4.1 Graph cropping with edge clipping to the borders. On the left, the case where one of the two termination points falls outside of the patch; in this case one node is added in the intersection point and the edge updated with the new node. On the right, the case where both termination points fall outside the patch but the edge still intersects the patch; in this case two nodes are added and the edge updated with the two new terminations. 49

4.2 Botanical data preparation pipeline. A raw RGB plant image is first associated with its annotated branching graph, which encodes node positions and connectivity. The graph structure is then rasterized to reconstruct a corresponding segmentation mask, enabling successive generation of the 3D input images for supervision, while preserving the underlying topology. 54

4.3 Procedural generation of input images from binary segmentations. The original mask is iteratively convolved with a 3×3 kernel to produce thicker and smoother tubular structures used as model inputs. 55

5.1 Qualitative comparison of graph predictions for a representative 3D patch from syntheticMRI dataset [25]. From left to right: ground-truth graph, prediction from the road-pretrained model, and prediction from the plant-pretrained model. The plant-based model better preserves the underlying branching topology. 69

5.2 Edge mean Average Recall (mAR) on the IXI test set [58] for plant- and road-pretrained models evaluated using 30%, 50%, and 100% of the available training annotations. 71
5.3 Qualitative comparison of graph predictions on a representative sample from an unseen dataset. From left to right: ground truth, syntheticMRI baseline, and the E4 configuration obtained under the scale-expanded multi-domain training regime. The baseline is close to random predictions, while E4 model more closely matches the ground-truth topology, with improved node detection and branch connectivity. 74
5.4 Performance across cumulative training configurations (syntheticMRI [25], E1-E4, as reported in table 5.4) for fixed-scale and scale-expanded multi-domain regimes. The top row reports results on OCTA [51] and the bottom row on SMILE-UHURA [50]. Columns show Structural Matching Distance (SMD), node mAP, and edge mAP. Shaded regions indicate variability across samples. 75
5.5 Paired t-test results comparing the single-domain baseline (syntheticMRI [25]) and the E4 configuration under the fixed-scale multi-domain training regime. Results are shown for OCTA [51] and SMILE-UHURA [50] across SMD, node mAP, node mAR, edge mAP, and edge mAR. The plotted values represent the mean paired difference, with error bars indicating the corresponding 95% confidence intervals. 76
5.6 Violin plots showing the distribution of performance metrics on SMILE-UHURA [50] under the fixed-scale multi-domain training regime. Results are reported for the single-domain baseline (syntheticMRI [25]) and the E4 configuration across edge mAP, edge mAR, node mAP, node mAR, and SMD. Each violin represents the distribution across test samples. 77

A.1 Qualitative comparison of predictions from the road-pretrained and plant-pretrained models across multiple samples. The plant-based model generally produces graph structures that more closely follow the underlying branching topology. 94
A.2 Distribution of performance metrics on the OCTA [51] dataset under the diversity-expanded training regime. 95
A.3 Distribution of performance metrics on the OCTA [51] and SMILE-UHURA [50] dataset under the scale-expanded training regime. . . 95
A.4 Qualitative comparison between the syntheticMRI baseline and the E4 scale-expanded multi-domain configuration on SMILE-UHURA [50]. From left to right: ground truth, baseline prediction, and E4 prediction. 96

Chapter 1

Introduction

Vascular graph extraction converts medical images into structured representations of vessel topology and connectivity. Instead of predicting dense pixel labels, the model infers a sparse relational graph in which nodes represent centerline samples and edges encode anatomical connections. This representation enables quantitative analysis of branching topology, global connectivity, and flow organization. Such properties are central to the study of vascular remodeling, tumor angiogenesis, and neurovascular disease [1].

Despite advances in medical image segmentation, extracting topologically consistent vascular graphs remains difficult. Errors in node localization or edge prediction can fragment the network, alter branching statistics, or break connectivity. These failures affect downstream measurements and may invalidate clinical interpretation. Graph extraction therefore requires global relational reasoning rather than purely local predictions.

A major limitation is the *annotation tax*. High-quality vascular graph annotations require expert supervision and often depend on complex preprocessing pipelines. In practice, the number of annotated graphs is small compared to the amount of available imaging data. Transformer-based image-to-graph models address structured prediction in a unified manner, but they rely on large annotated datasets, which in medical imaging are rare.

Recent work has explored cross-domain pretraining to mitigate this constraint by using infrastructure networks, such as urban roads, as auxiliary supervision for vascular graph extraction [2]. This approach aligns with the foundation model paradigm, in which large-scale pretraining datasets provide transferable priors. However, general-purpose foundation models often underperform in medical settings, motivating the development of specialized Medical Foundation Models [3]. Moreover, scale alone does not guarantee structural alignment. Road networks are shaped by human planning constraints, whereas vascular systems develop under biological processes linked to flow efficiency and space-filling organization. As a result, the two

domains may share a graph structure, but they do not reflect the same branching logic.

This mismatch raises a central question: which priors transfer effectively to vascular topology? If the source domain encodes structural regularities that conflict with vascular organization, the learned inductive bias may limit generalization. For structured prediction tasks, representational congruence between source and target domains may matter more than dataset size [4].

Biological transport systems provide a more aligned source of structural supervision. Plant venation and tree branching networks arise under physical constraints similar to those governing vascular growth. They exhibit hierarchical branching, limited node degrees, curvature, and scale-invariant organization. These properties reflect optimization of transport and spatial coverage [5, 6]. Such structural principles are closer to vascular networks than anthropogenic infrastructure.

Building on this observation, this thesis investigates biologically grounded transfer learning for vascular graph extraction. Instead of transferring from infrastructure data, image-to-graph transformers are pretrained on large collections of botanical skeleton graphs. This nature-to-nature paradigm promotes relational priors that reflect growth-constrained branching rather than engineered layouts.

Beyond source-domain alignment, generalization also depends on the diversity of target domains encountered during training. Models trained on a single vascular dataset tend to internalize dataset-specific priors tied to acquisition protocol, modality, or anatomical region. When applied to new data, these priors may fail. Aggregating multiple heterogeneous vascular datasets can encourage the model to learn invariant structural patterns that persist across imaging conditions. However, naive aggregation risks introducing inconsistent topology due to differences in preprocessing pipelines and graph extraction rules. Controlled, topology-preserving dataset construction is therefore essential.

This thesis addresses vascular graph extraction through three interconnected directions:

1. **Structurally aligned biological pretraining.**
2. **Topology-aware supervision and graph construction.**
3. **Multi-target scaling for robust and zero-shot generalization.**

By aligning inductive biases with biological growth principles and by exposing the model to diverse yet consistently constructed vascular targets, the proposed framework aims to reduce annotation requirements while improving structural fidelity and cross-domain robustness.

Chapter 2

Background and Related Work

2.1 Image-to-Graph Transformation

Image-to-graph transformation refers to the task of mapping visual data to a structured graph representation, where nodes and edges encode semantically meaningful entities and their relationships. This topic has gained increasing attention across computer vision applications such as robotics, geographic information systems and document understanding, thanks to its ability to recognize relational and topological structures. More recently, image-to-graph learning has also been explored in the medical field, particularly for anatomical structures such as vessels.

Unlike common vision tasks such as classification or segmentation, image-to-graph transformation requires predicting a sparse and structured output with strong constraints. The number of possible edges increases quadratically with the number of nodes, while only a small subset corresponds to valid relationships. As a result, the output space is highly unbalanced and combinatorial. Moreover, errors in node localization or edge prediction do not remain local. Missing nodes, small displacements, or spurious edges can break connectivity, fragment structures, or alter the global topology of the graph. For applications such as vessel analysis, these failures can invalidate downstream measurements and clinical interpretation. This makes topology preservation a central requirement rather than a secondary objective.

Early approaches typically solved this task using multi-step pipelines consisting of image segmentation, skeletonization, and subsequent graph extraction through iterative pruning procedures [7, 8, 9]. While effective in constrained settings, such pipelines are sensitive to errors introduced at each stage, leading to uncertainty

accumulation and loss of structural information. Moreover, their reliance on domain-specific heuristics limits robustness and generalization, especially in cross-domain or transfer learning scenarios.

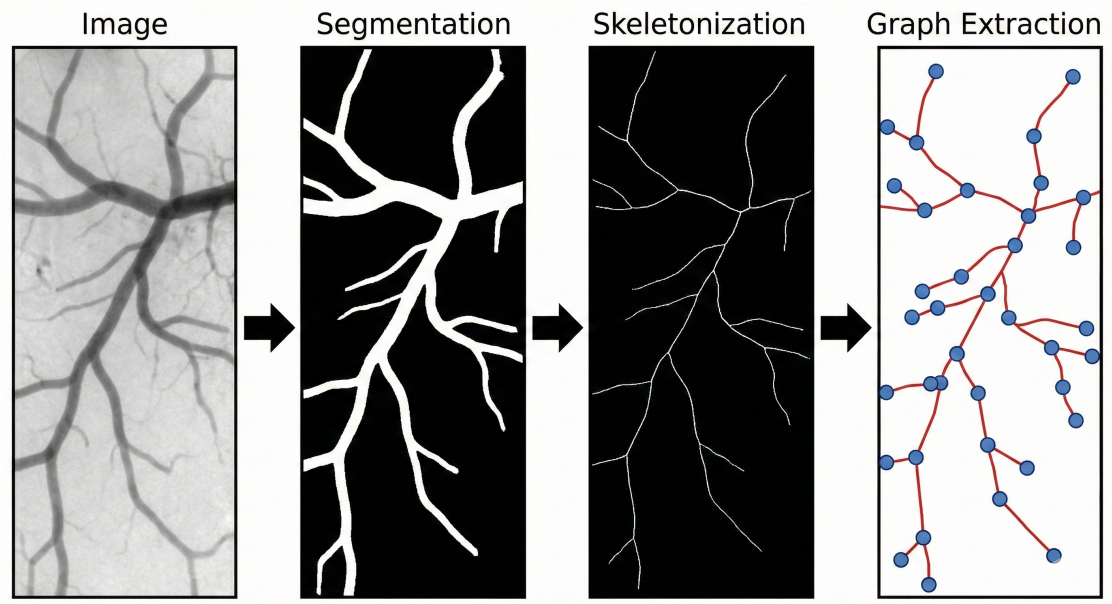

Figure 2.1: Traditional multi-stage pipeline for vascular graph extraction. The input image is first segmented to obtain a binary vessel mask, followed by skeletonization to derive centerlines, and finally graph construction through iterative pruning and post-processing. Errors introduced at each stage accumulate across the pipeline, often leading to fragmented connectivity and loss of topological consistency.

Traditional convolutional models and local prediction strategies [10] struggle in this setting because they rely mainly on local spatial evidence. Graph structures, however, often depend on long-range dependencies and context, such as topological continuity through tortuosity or consistency between distant branches. This mismatch between local supervision and global structure motivates models that can reason over the entire image and its relational content.

To overcome these limitations, recent learning-based methods have shifted toward end-to-end formulations. Several approaches employ deep neural networks to first detect graph nodes, followed by a separate relation prediction or edge inference stage [11, 12]. Even if these methods reduce manual design effort, they still have a sequential structure, which can still propagate prediction errors and constrain adaptability across domains.

By taking inspiration from vision transformers that are able to learn global spatial relationships directly from image patches, more recent work has explored

transformer-based architectures that directly infer graph structure from image features within a single model. By leveraging global self-attention mechanisms, these approaches enable joint reasoning over nodes and relationships, providing a unified framework for image-to-graph prediction across diverse domains [13, 14]. Despite their flexibility, existing image-to-graph transformers face challenges related to supervision availability, class imbalance, and domain shift, which are particularly pronounced in medical imaging applications.

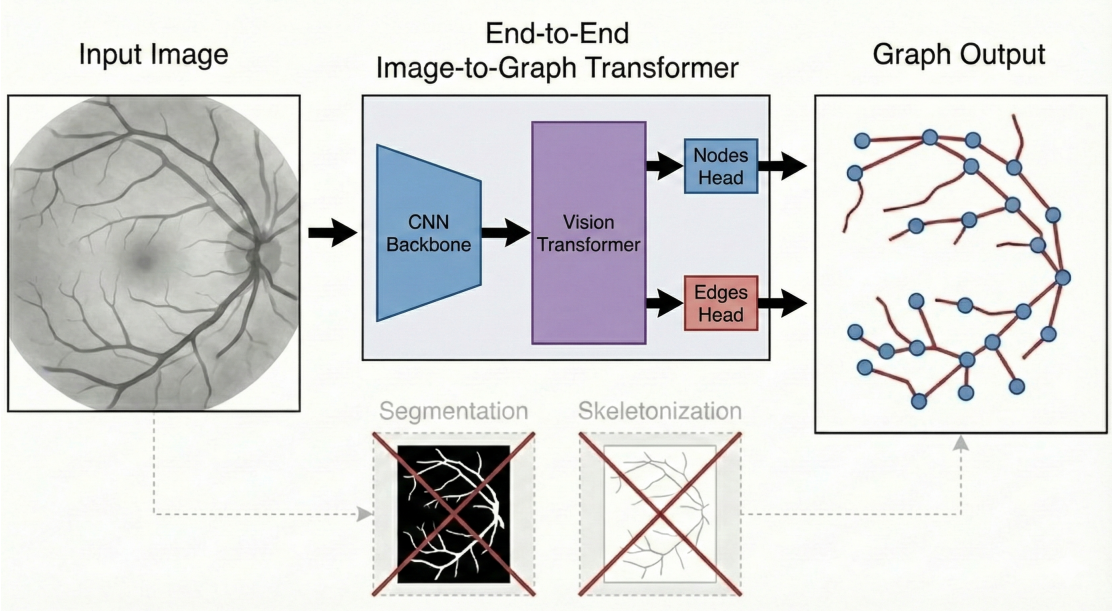

Figure 2.2: End-to-end image-to-graph transformer framework. The input image is processed by a CNN backbone for feature extraction, followed by a vision transformer that jointly predicts nodes and edges through dedicated object and relation heads. Unlike traditional multi-stage pipelines, segmentation and skeletonization are bypassed, enabling direct inference of the vascular graph and reducing error propagation across intermediate steps.

Supervision is particularly challenging. Graph annotations are expensive to retrieve and are usually derived from other predicted sources. In many medical imaging applications [2, 15], graph ground truths are extracted using tools that require segmentations; therefore, graph quality depends on segmentation quality. In contrast to dense pixel-wise labels, graph supervision is sparse and sensitive to annotation noise. These factors make transfer learning essential to overcome graph annotation scarcity and quality limitations.

2.2 Transfer Learning and Domain Adaptation

One of the main consequences of transformer-like architectures and deep models is that they require large amounts of annotated data to be trained. In many domains, such as medical imaging, such data is scarce, expensive to obtain, or unavailable. This is especially true for graph annotations, which are commonly derived from complex processing pipelines and expert knowledge [2, 15]. These limitations make transfer learning a central tool for image-to-graph transformation.

2.2.1 Cross-Domain and Adversarial Transfer Learning

Cross-domain transfer learning aims to reuse knowledge learned from a source domain, to improve performance in a different target domain [16]. The more the two domains differ, the harder the transfer will be. In image-to-graph tasks, it involves transferring both visual representations and structural reasoning. While image appearance can vary strongly across domains, many physical networks share common structural patterns [2]. Examples include road systems, vascular networks, plant branching systems and neural structures, which all form sparse graphs with branching and connectivity constraints. This underlying structural similarity motivates the use of information sharing across domains.

Standard transfer learning methods for vision models focus on image-level features [17, 10]. They often rely on pretraining on large natural image datasets and fine-tuning on the target task. While effective for classification or segmentation, these strategies are limited for image-to-graph transformation. Graph prediction depends not only on visual cues but also on global relational structure. As a result, transferring only image features is often insufficient to preserve graph connectivity and topology. To address this gap, recent works have explored transfer learning strategies that operate on both image and graph representation levels [2]. By reducing domain-specific biases, the model can focus on shared structural principles rather than appearance alone. This is particularly important when the source and target domains differ strongly in texture, signal-to-noise ratio, or imaging modality.

Adversarial Transfer Learning Adversarial learning is commonly used to reduce distribution shifts between datasets. It aims to learn feature representations that support the target task while remaining insensitive to the data source. A standard adversarial transfer learning setup is built around a Gradient Reversal Layer (GRL), which enables joint optimization of task performance and domain invariance [18]. The architecture consists of a feature extractor, a task predictor, and a domain classifier. The feature extractor maps the input data to a latent representation, the task predictor uses this representation to solve the main task, and the domain classifier attempts to identify the origin of the data.

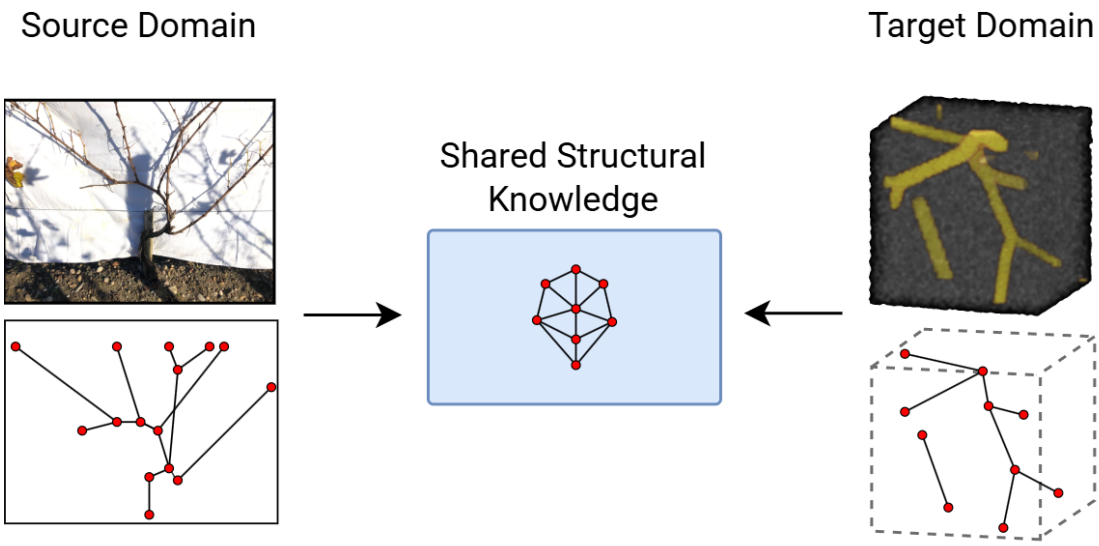

Figure 2.3: Overview of cross-domain and cross-dimension transfer learning for image-to-graph transformation. Structural knowledge learned from a source domain is encoded into a shared latent representation and transferred to a visually distinct target domain to predict graph topology. The framework emphasizes domain-invariant connectivity and branching patterns rather than low-level image appearance.

The GRL controls how gradients flow during training. During the forward pass, it acts as an identity operation and leaves the features unchanged. During the backward pass, it multiplies the gradients by a negative factor. This inversion causes the feature extractor to receive gradients that confuse the domain classifier [18]. As a result, the feature extractor is encouraged to remove domain-specific cues while retaining information relevant to the task. This process leads to features that are more stable across domains.

For tasks with structured outputs, adversarial learning is often applied at multiple representation levels. Aligning only low-level image features may reduce appearance differences, but it does not guarantee consistency in higher-level predictions. Prior work has shown that combining adversarial objectives at different stages of the model helps preserve spatial and relational properties [19]. This is particularly relevant for graph prediction tasks, where both global appearance and structural patterns must remain consistent across domains.

In image-to-graph transformation, Berger et al. applied adversarial alignment at both the image and graph representation levels [2]. Image-level alignment addresses

differences in texture, contrast, and noise, while graph-level alignment targets differences in structural properties such as node distribution and edge regularity. This multi-level strategy improves transfer when source and target domains share similar dimensionality and task structure.

Adversarial transfer learning is most effective when source and target data lie in the same dimensional space. In such cases, a model trained on annotated 2D plant images can transfer knowledge about branching and connectivity to 2D medical scans. While adversarial methods do not remove all domain gaps, they provide a principled way to leverage synthetic or auxiliary data and improve performance on real-world medical tasks.

2.2.2 Cross-Dimension Transfer Learning

Beyond domain shifts, cross-dimension transfer learning addresses transfer between data with different dimensionalities, such as from 2D images to 3D volumes. Fully annotated 3D datasets are rare, particularly for graph extraction tasks, and annotation retrieval is expensive and time-consuming. On the other hand, 2D datasets are more common. Cross-dimension transfer learning seeks to exploit this imbalance by using 2D data to support training in 3D settings. However, this introduces additional challenges, as both image representation and graph structure change with dimensionality.

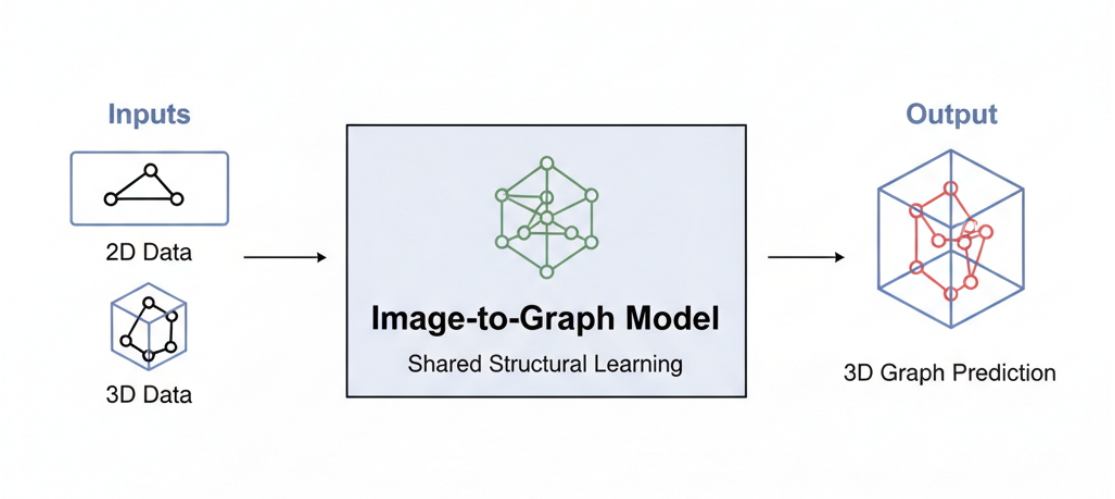

Figure 2.4: Overview of the cross-dimension image-to-graph learning framework. The model jointly processes 2D and 3D inputs within a shared structural learning setup, enabling knowledge transfer across dimensionalities and producing a unified 3D graph prediction.

Data-based methods focus on projecting or augmenting data between dimensions.

For instance, Shen et al. [20] propose a technique that projects 3D point clouds into pseudo-2D RGB images. Conversely, Liu et al. [21] introduce a pixel-to-point knowledge transfer method. Their approach pretrains a 3D model by generating 3D point cloud data from 2D images using a learned projection function.

Model-based approaches, on the other hand, adjust the architecture to handle multi-dimensional inputs. Xie et al. [22] utilize dimension-specific feature extractors paired with a dimension-independent Transformer. Similarly, Wang et al. [23] implement a specialized tokenizer for 3D input data that creates 2D-shaped patch embeddings within a standard 2D Vision Transformer (ViT) model.

Current methods, even though effective, have some limitations. Most of them require alterations to the target model, depend on complex and carefully crafted projection functions, or demand additional training steps to learn these projections. Recent work by Berger et al. [2] demonstrates the effectiveness of this approach for image-to-graph transformers. They propose a simple framework that transfers knowledge from 2D road networks to 3D vascular graphs. The core of this method is a projection function that transforms 2D source data into a representation that mimics 3D target space. Specifically, they:

1. Resize the image to the target domain's spatial patch size by a linear downsampling operator.
2. Initialize an empty 3D volume, place the 2D image in the middle of it, and augment node coordinates to have three axes.
3. Apply a random three-dimensional rotation matrix to the 2D image, applying the same rotation to node coordinates.

This allows the model to process 2D images as if they were 3D inputs during the pretraining phase. Berger et al. validated this approach on two datasets, including whole-brain vessel graphs [24] and a synthetic MRI dataset [25]. Their results show that pretraining on 2D road maps significantly improves performance on 3D medical tasks. This method consistently outperforms standard transfer learning and self-supervised strategies. It effectively enables data-efficient training for complex 3D problems where labeled data is otherwise insufficient.

2.2.3 Sparse Relational Supervision and Topology Preservation

Graph prediction tasks operate under strong class imbalance. In most graphs, only a small subset of all possible node pairs corresponds to true and existing edges. As the number of nodes increases, the number of negative pairs grows quadratically, while the number of positive relations remains limited. This imbalance can dominate the learning signal, biasing models toward predicting sparse or empty graphs.

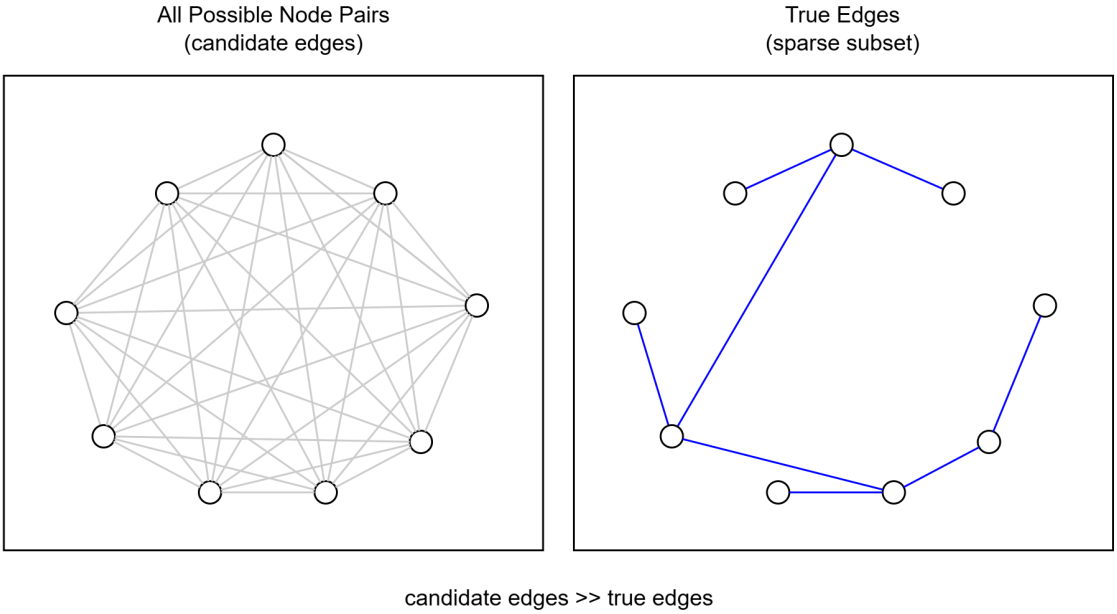

Figure 2.5: Illustration of relation imbalance in graph prediction. Left: all possible unordered node pairs forming the set of candidate edges, which grows quadratically with the number of nodes. Right: the sparse subset of true edges representing valid graph connectivity. The large discrepancy between candidate and true relations (candidate edges $\gg$ true edges) leads to severe class imbalance during relation learning.

Beyond class imbalance, sparse supervision also affects the preservation of graph topology. In image-to-graph transformation, edges mainly encode object connectivity and global structure, rather than being independent labels. Missing or incorrect edges can fragment the graph or alter its topology, even when node predictions are accurate. As a result, learning reliable relations requires supervision strategies that emphasize meaningful connections without overwhelming the model with background relations.

A common way to address these issues is to restrict the set of edges used for supervision during training [13, 2]. Instead of computing the loss over all possible node pairs, models sample a subset of relations that includes all positive edges and a controlled number of negative ones [13]. This reduces computational cost and limits the influence of the dominant background class while maintaining a learning signal for true relations.

Berger et al. [2] propose a regularized edge sampling approach that enforces a fixed ratio between positive and negative edges, rather than relying on a fixed total number of sampled relations. In their formulation, the maximum number

of considered relations is set to a large constant, which includes all available negative edges while replicating positive ones to reach the desired ratio. This avoids discarding background relations while ensuring that positive edges contribute sufficiently to the loss. As a result, the supervision remains stable across datasets with different graph densities.

Sampling-based supervision is well suited for sparse and structured prediction tasks, where the goal is to learn global connectivity patterns from a limited number of positive evidences. By controlling class imbalance, these strategies support more reliable topology learning without introducing task-specific heuristics. For these reasons, the same edge sampling strategy is adopted to handle strong class imbalance in graph learning.

2.3 Biological and Semantic Priors

Deep learning approaches reduced the need for explicit, hand-crafted semantic rules, but they remain dependent on prior assumptions. In modern models, such priors are learned implicitly from data and constrain the space of plausible predictions. As a result, the choice of training data plays a central role. Models trained on a specific domain tend to internalize its structural regularities and statistical patterns, which can strongly influence behavior when transferred to new domains.

This dependency is particularly pronounced in graph extraction tasks. Unlike dense prediction problems, graph prediction requires consistency at both local and global scales. Local decisions, such as node placement or edge connectivity, directly affect global properties such as connectivity, topology, and overall structure. For natural networks, these requirements extend beyond accuracy and completeness to include biological plausibility. Prior work has shown that image-to-graph models benefit from training data that encodes not only visual similarity, but also meaningful structural properties [2].

Berger et al. [2] exploit this idea through cross-domain transfer, assuming that physical networks share transferable graph structure and using urban road networks as a source-domain prior for vessel graph extraction. While roads and vessels exhibit graph-like organization, road networks are shaped by design and planning constraints that differ from the generative processes governing biological systems. This semantic mismatch can affect the suitability of the learned priors for vascular topology.

Biological transport networks provide an alternative class of semantically grounded graph priors. Vascular systems, plant venation, and tree branching structures arise from growth processes driven by transport efficiency, energy minimization, and space-filling constraints. These systems exhibit recursive branching, limited node degrees, and scale-invariant organization. Classical models such as Murray's

law [26, 5] and allometric scaling [6] formalize these properties by linking local geometric features, such as vessel radius, to global optimization principles.

Comparative studies in plant physiology and morphology report strong similarities between plant and animal vascular architectures. Shared branching statistics and scaling laws have been observed across botanical and animal transport networks [27, 26]. These findings suggest that biological graphs can serve as meaningful source domains for learning relational structure. Unlike man-made networks such as roads, biological networks are shaped by physical and metabolic constraints rather than design conventions, resulting in higher curvature, variable branch angles, and non-uniform spacing. These characteristics are commonly observed in medical vessel data.

Recent image-to-graph transformers learn relational structure directly from images [13], but they remain sensitive to the priors induced by their training domains. When these priors are learned from semantically mismatched sources, such as urban road networks, conflicts with vascular topology may arise. This sensitivity can limit transfer performance, particularly in scarce-data and zero-shot settings [13, 2].

2.4 Multi-Target Scaling and Zero-Shot Generalization

2.4.1 Data Aggregation

Beyond architectural design, the generalization behavior of models can be strongly influenced by the scale and diversity of target domains observed during training [28, 29]. Diversity can be introduced through data augmentation, the use of synthetic data, or training on datasets that cover different imaging modalities, resolutions, and acquisition settings [30, 31, 28, 29].

When trained on a single dataset, models often learn patterns tied to that dataset. These patterns may not hold in new settings [32, 33]. Recent work has shown that data aggregation across heterogeneous target domains can partially alleviate this limitation. With respect to the medical field, by combining multiple datasets with different imaging characteristics, anatomical regions, and acquisition protocols into a unified training distribution, models are exposed to a broader range of structural variability [28, 29].

This strategy shifts the learning objective from fitting a single dataset to learning invariant features that remain stable across domains. In vascular graph extraction, this means learning structures that persist across imaging modalities, branching densities, tissue types, organisms, and spatial scales.

Large-scale data aggregation has been shown to improve robustness and transferability in vessel segmentation and representation learning, particularly when training on diverse, multi-center datasets [28]. These results suggest that exposing models to broad structural variability encourages more stable and transferable representations.

Extending this paradigm to image-to-graph transformers requires additional care. Unlike dense prediction tasks, graph supervision depends on the consistency of node placement and connectivity. When datasets are aggregated without control, differences in graph density, annotation protocols, or extraction pipelines can introduce inconsistencies that act as spurious cues and reduce generalization.

In this setting, curated multi-target training serves two purposes. It increases the effective amount of training data and acts as an implicit regularizer that limits reliance on dataset-specific priors. This motivates the transition from single-target training to controlled, topology-aware aggregation of multiple vascular datasets, as discussed in the following sections.

2.4.2 Topology-Preserving Pipelines

Data aggregation increases diversity, but its effectiveness depends on the consistency of the underlying supervision. For graph prediction tasks, naively combining datasets preprocessed with different preprocessing pipelines can lead to incompatible topological representations, undermining the benefits of scale. Small discrepancies in centerline extraction, pruning criteria, or connectivity rules can produce graphs with different node degrees and edge statistics, even when derived from visually similar structures [12].

To address this issue, recent approaches emphasize topology-preserving preprocessing pipelines that enforce consistent structural representations across datasets. Rather than relying on dataset-specific heuristics, these pipelines aim to extract graphs in a reproducible manner that preserves connectivity, branching structure, and geometric continuity [2, 13]. Patch-based extraction strategies with explicit handling of boundary-crossing edges further ensure that local samples remain topologically valid while enabling scalable training.

Such pipelines are particularly important for transformer-based image-to-graph models, where relational reasoning is learned implicitly from supervision statistics. Inconsistent topology across training samples can bias the model toward dataset-specific structural artifacts, reducing its ability to generalize. Conversely, enforcing topological consistency across heterogeneous datasets allows the model to focus on learning higher-level relational patterns, such as branching regularities and connectivity constraints, that are invariant across imaging conditions.

Common graph extraction tools, such as Voreen [9], provide deterministic and reproducible graph construction from volumetric data, but their output remains

sensitive to preprocessing choices and parameter settings. This emphasizes the need for careful control of graph extraction when aggregating data from multiple sources.

By coupling multi-target data aggregation with topology-preserving graph extraction, large-scale training becomes feasible without sacrificing structural coherence. This combination provides a foundation for studying scaling behavior and generalization in image-to-graph transformers, bridging the gap between dataset diversity and relational consistency.

2.4.3 Scaling and Zero-Shot Generalization

Beyond improving in-distribution performance, an important objective of large-scale models is zero-shot generalization. In medical imaging, this setting is particularly relevant, as retrieving annotations is expensive and retraining models on a specific dataset is time- and resource-consuming. New datasets often differ considerably in many aspects, such as resolution, contrast mechanisms, and anatomical variability, while annotated graph supervision remains unavailable [34, 33, 35].

Empirical evidence from both vision transformers and medical foundation models suggests that scaling training data diversity improves robustness to domain shift [36, 14]. However, for structured prediction tasks such as graph prediction, zero-shot performance depends not only on the amount of data used, but also on the diversity of structural patterns and distributions encountered during training [32, 33]. Models trained on limited or homogeneous graph distributions tend to encode fragile priors that fail under strong distribution shifts, even when visual features are well learned.

Multi-target scaling mitigates this issue by exposing the model to a wide spectrum of target samples. As the number of training targets increases, the model is increasingly forced to rely on relational cues that are consistent across domains, rather than dataset-specific correlations [28]. This effect is particularly evident when training distributions include both real and synthetic data, spanning different acquisition settings and noise characteristics [2, 28]. When training combines multiple targets with topology-preserving pipelines and well-matched source domains [2, 13], models can apply the same relational rules to new data. This makes it possible to extract graphs from unseen medical datasets without additional annotation, which is a key requirement for practical deployment.

Chapter 3

Methods

Figure 3.1 provides an overview of the complete methodological pipeline adopted in this work. The framework builds upon the RelationFormer architecture [13] and the cross-domain and cross-dimensions training setup of Berger et al. [2], while incorporating the proposed botanical source-domain pretraining and multi-target diversity strategy described in the following sections.

3.1 Base Architecture: Relationformer

In this work, the image-to-graph framework introduced by Berger et al. [2] is extended. This framework is itself based on the *Relationformer* architecture [13]. *Relationformer* is a transformer-based model that predicts graph-structured outputs directly from images in a single stage by jointly detecting nodes and their relations.

The core *Relationformer* architecture consists of a convolutional backbone for visual feature extraction followed by a transformer encoder-decoder dedicated to graph prediction. The modifications introduced by Berger et al. mainly adapt the training setup for cross-domain and cross-dimensional learning, while leaving the underlying architecture largely unchanged. In this work, I therefore keep the *Relationformer* architecture, loss formulation, and all original prediction heads unchanged. Figure fig. 3.2 shows the architecture.

Starting Setting Consider an image space $I \in \mathbb{R}^{D \times \#\text{ch}}$, where $D = \prod_{i=1}^{d} \dim[i]$ for a d-dimensional image and $\#ch$ denotes the number of channels. The model $\mathcal{F}$ predicts $\mathcal{F}(I) = \mathcal{G}$ for a given image I, where $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a graph composed of the set of vertices/objects $\mathcal{V}$ and edges/relations $\mathcal{E}$. Each vertex $v_i \in \mathcal{V}$ has a node/object location specified by a bounding box $v^i_{\text{box}} \in \mathbb{R}^{2 \times d}$ for 2D coordinates or $v^i_{\text{box}} \in \mathbb{R}^{3 \times d}$ for 3D ones, with an object label $v^i_{\text{cls}} \in \mathbb{Z}^C$. Similarly, each edge $e^{ij} \in \mathcal{E}$ has an edge or relation label $e^{ij}_{\text{rln}} \in \mathbb{Z}^L$, where C is the number of object

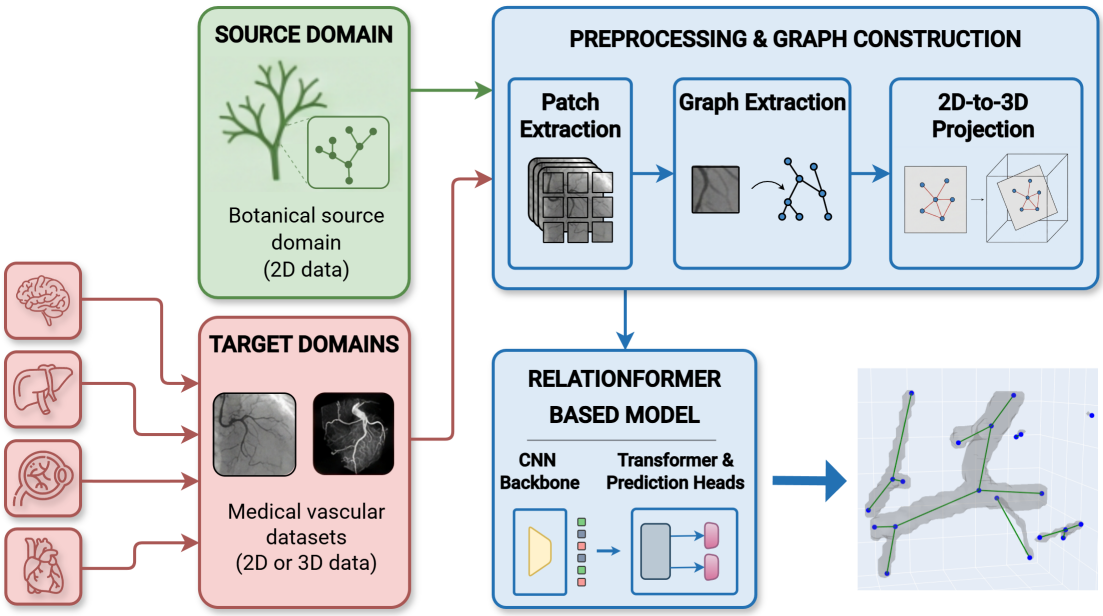

Figure 3.1: Overview of the proposed topology-aware transfer learning framework. A botanical source domain (2D data) and multiple vascular target domains (2D or 3D) are processed through a unified preprocessing and graph construction pipeline, including patch extraction, deterministic graph extraction, and 2D-to-3D projection. The resulting image-graph pairs are used to train a Relationformer-based model that jointly predicts nodes and relations, enabling structurally aligned cross-domain and cross-dimension learning.

classes and L is the number of relation types. In the graph extraction setup, only a single foreground node category is considered. The object classifier therefore predicts whether each object token corresponds to a node or to the background (no-object). For relations, the model performs binary classification for each pair of nodes, predicting either edge or no-edge. In the considered case, $\mathcal{G}$ is also undirected, while in the original *Relationformer* paper it can be either directed or undirected. The algorithmic complexity of predicting graph $\mathcal{G}$ depends on its size $|\mathcal{G}| = |\mathcal{V}| + |\mathcal{E}|$, which is of order $\mathcal{O}(N^2)$ for N nodes.

3.1.1 Object Prediction

Carion et al. [37] proposed DETR, a transformer-based object detector that formulates detection as a set prediction problem, using object queries and one-to-one matching instead of anchors or non-maximum suppression (NMS). Instead of using anchors or pixel-wise assignments, DETR works on the full image thanks to its

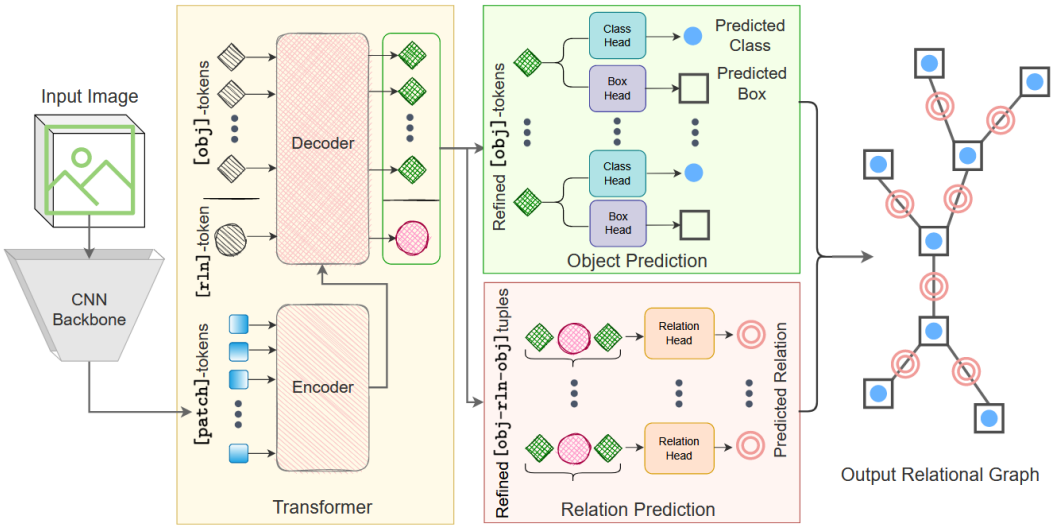

Figure 3.2: Relationformer architecture from Shit et al. [13], composed of a CNN backbone for feature extraction, a transformer to extract rich object tokens, and two heads for object and relation prediction, yielding the output relational graph.

encoder–decoder transformer architecture [38]. Given an input image I, a convolutional backbone [10] is used to extract low-resolution, high-level feature maps, where each spatial location encodes semantic information. Next, to restore spatial awareness, the features are flattened into a sequence and coupled with a sinusoidal positional encoding [39]. This encoding uniquely identifies each spatial position, allowing attention layers to reason about geometry and relative position.

The transformer's encoder consists of a stack of transformer layers with multi-head self-attention and feed-forward networks, which process the sequential features. Thanks to self-attention, each token can attend to all other tokens, allowing the model to capture long-range dependencies and global context. The output of the encoder is a sequence of context-enriched feature embeddings corresponding to the input image features. The encoder output is then passed to the decoder, which additionally takes as input a fixed number N of learnable object queries (`[obj]`-tokens) that are independent of the image content. These queries act as object hypotheses and interact with the encoded image features through cross-attention.

DETR outputs N predictions from N `[obj]`-tokens, but the number of ground truth objects varies per image, the predictions are unordered, and multiple queries might try to predict the same object. For this reason, DETR utilizes a direct Hungarian set-based assignment to perform one-to-one matching between the ground truth objects and the predictions from the N `[obj]`-tokens. The bipartite

matching assigns a unique predicted object from the N predictions to each ground truth object by searching for the lowest-cost assignment. The cost of matching a prediction p_i to a ground truth object g_i includes classification mismatch, bounding box distance (e.g., L1), and overlap quality (e.g., GIoU). Only matched predictions are considered valid, while the remaining ones are labeled as `background`. Subsequently, the box regression loss is computed only for valid predictions. For the classification loss, all predictions, including `background` objects, are considered.

Standard DETR relies on global attention, where each query attends to *all* spatial locations in the image feature map. For computer vision tasks, this approach is inefficient since it is computationally expensive and object evidence is usually localized in a small spatial region. For this reason, in *Relationformer*, deformable attention from deformable-DETR (def-DETR) [40] is used for faster convergence and improved computational efficiency. Following the concept of deformable convolutions, deformable attention enables the queries to attend to a small set of spatial features determined from learned offsets of the reference points.

Let us consider an image feature map $\boldsymbol{f}_I$ from the backbone, the q^{th} `[obj]`-token with associated features $\boldsymbol{f}_q$, and a reference point $\boldsymbol{x}_q$ indicating the spatial location of the query. For the m^{th} attention head, the k^{th} sampling offset $\Delta\boldsymbol{x}_{mqk}$ is computed around the reference point based on the query features $\boldsymbol{f}_q$. The sampled image features $\boldsymbol{f}_I(\boldsymbol{x}_q + \Delta\boldsymbol{x}_{mqk})$ are then projected through a head-specific linear layer $\boldsymbol{W}'_m$ and weighted by the attention coefficient $\boldsymbol{A}_{mqk}$, which is also predicted from the query features $\boldsymbol{f}_q$. This projection maps the image features into a query-compatible space, allowing each attention head to focus on different aspects of the features. Finally, another linear layer $\boldsymbol{W}_m$ merges the outputs of all heads. Formally, the deformable attention operation (DefAttn) for M heads and K sampling points is defined as:

$$\text{DefAttn}(f_q, x_q, f_I) = \sum_{m=1}^{M} W_m \left[\sum_{k=1}^{K} A_{mqk} \cdot W'_m \, f_I(x_q + \Delta x_{mqk}) \right] \tag{3.1}$$

3.1.2 Relation Prediction

Joint object–relation graph generation requires predicting both nodes and edges. Given N nodes, the number of possible relations is $O(N^2)$. With a naive transformer formulation, one could represent each object with a token and each possible relation with an additional token. This would require $O(N^2)$ relation tokens. Since the self-attention mechanism scales quadratically with the number of tokens, this would result in a computational cost of $O(N^4)$, which quickly becomes intractable even for a moderate number of nodes.

To overcome this combinatorially challenging formulation, *Relationformer* uses an inductive bias that exploits learned object representations to infer interactions.

After learning object features using N `[obj]`-tokens, these tokens are allowed to interact with each other, and a single additional token is used to summarize those interactions. This extra learnable token is called the `[rln]`-token. It can be interpreted as a query over object–object interactions, aggregating relevant information such as which objects matter, how objects co-occur, and the global relational context of the scene. In this process, related objects are incentivized to have strong correlation in embedding space, and unrelated objects are penalized to be dissimilar.

Pairwise relations are then classified by combining pairs of refined `[obj]`-tokens with the shared `[rln]`-token. Thus, instead of requiring $O(N^2)$ tokens, it needs only $N + 1$ tokens in total. These consist of N `[obj]`-tokens and one `[rln]`-token.

Using a dedicated `[rln]`-token, separate from the `[obj]`-tokens, allows the model to better capture interdependencies among relations while reducing the burden on `[obj]`-tokens, which specialize in object prediction. Since `[obj]`-tokens and the `[rln]`-token are concatenated, they can attend to each other and exchange global semantic information. In contrast, using a large number of `[rln]`-tokens would lead to a drastic increase in decoder complexity, which may result in computational intractability.

3.1.3 Architecture

The architecture of *Relationformer* can be seen in fig. 3.2. It consists of:

- **Backbone**: Given the input image I, a convolutional backbone [10] extracts features $\boldsymbol{f}_I \in \mathbb{R}^{D_f \times \#\text{emb}}$, where D_f is the spatial dimensions of the features and #emb denotes the embedding dimension. This feature dimension is then reduced to d_{emb} that represents the embedding dimension of the transformer.

- **Transformer**: *Relationformer* uses a transformer encoder-decoder architecture with deformable attention [40], which speeds up the training convergence of DETR.

- **Encoder**: the encoder remains unchanged from [40], using multi-scale deformable self-attention. **Relationformer** uses a different number of layers based on each task's requirement.

- **Decoder**: the decoder uses as input $N + 1$ tokens for the joint object-relation prediction task, where N is the number of `[obj]`-tokens, preceded by a single `[rln]`-token. The decoder also receives the contextualized image features from the encoder. In order to make the two inputs share information (without affecting computation time), deformable cross-attention between the two is used. The self-attention in the decoder remains unchanged. The `[obj]`-tokens and `[rln]`-tokens exchange information between them and with the image

features, gradually building a hierarchical object and relational semantics. In this case, `[obj]`-tokens learn to attend to specific spatial positions in the image, while the `[rln]`-tokens learns how objects (points) interact in the context of their semantic or global reasoning.

- **Object Detection Head**: the object detection head has two components. The first is a stack of fully connected layers, i.e. a multi-layer perceptron (MLP), that regresses object locations. The second is a single-layer classification module. For each `[obj]`-token o^i, the object detection head predicts an object class $\tilde{v}^i_{cls} = \boldsymbol{W}_{cls}(o^i)$ and an object location $\tilde{v}^i_{box} = \text{MLP}_{box}(o^i)$, $\tilde{v}^i_{box} \in [0,1]^{2\times d}$ in parallel, where d represents the image dimension, $\boldsymbol{W}_{cls}$ is the classification layer, and MLP_{box} is the box predictor MLP. The architecture uses normalized bounding box coordinates for scale-invariant prediction. In the considered case of spatio-structural graphs (where points represent positions in the image, for example joints or bifurcations), nodes are typically coordinates without a ground-truth bounding box. To make them compatible with the architecture, each point is treated as the center of a uniform bounding box.

- **Relation Prediction Head**: in parallel to the object detection head, the input of the relation head is the pair-wise `[obj]`-token and a shared `[rln]`-token. Those are processed as $\tilde{e}^{ij}_{rln} = \text{MLP}_{rln}([\, o^i;\, r;\, o^j\,])_{i\neq j}$, where r is the refined `[rln]`-token and MLP_{rln} is a three-layer fully-connected network headed by layer normalization [41]. Since in the considered case we're not interested in obtaining an ordered graph, the network is trained to learn object token *order* invariance as well.

Since the object and relation prediction heads operate in parallel, both are applied to the complete set of refined decoder tokens. The object prediction head processes all `[obj]`-tokens to estimate node existence, class labels, and spatial locations, while the relation prediction head jointly processes all ordered pairs of `[obj]`-tokens together with the shared `[rln]`-token to predict pairwise relations. During inference, node existence is determined solely by the object classification scores, and only relations whose endpoints correspond to predicted (non-background) nodes are retained in the final graph.

3.1.4 Relationformer Loss Function

For the object detection task, *Relationformer* employs a combination of multiple loss terms. In particular, two losses are used for bounding box prediction, together with a classification loss and a relation classification loss:

- **Box regression loss** $\mathcal{L}_{reg}$, implemented as an ℓ_1 regression loss on the predicted bounding box coordinates;

- **Box overlap loss** $\mathcal{L}_{gIoU}$, computed as the generalized intersection over union between predicted and ground-truth boxes;
- **Classification loss** $\mathcal{L}_{cls}$, defined as a cross-entropy loss over object classes, including the background class;
- **Relation loss** $\mathcal{L}_{rln}$, defined as a cross-entropy loss computed on a subset of object pairs, consisting of each valid relation and a small number of randomly sampled background relations, in order to avoid evaluating the loss over all possible object pairs.

$$\begin{aligned}\mathcal{L}_{\text{total}} = &\sum_{i=1}^{N} \mathbb{1}_{v^i_{\text{cls}} \neq \varnothing} \Big(\lambda_{\text{reg}} \mathcal{L}_{\text{reg}}(v^i_{\text{box}}, \tilde{v}^i_{\text{box}}) + \lambda_{\text{gIoU}} \mathcal{L}_{\text{gIoU}}(v^i_{\text{box}}, \tilde{v}^i_{\text{box}}) \Big) \\ &+ \lambda_{\text{cls}} \sum_{i=1}^{N} \mathcal{L}_{\text{cls}}(v^i_{\text{cls}}, \tilde{v}^i_{\text{cls}}) + \lambda_{\text{rln}} \sum_{\{i,j\} \in \mathcal{R}} \mathcal{L}_{\text{rln}}(e^{ij}_{\text{rln}}, \tilde{e}^{ij}_{\text{rln}})\end{aligned} \tag{3.2}$$

where λ_{reg}, λ_{gIoU}, λ_{cls} and λ_{rln} are loss-specific weights.

3.2 Berger's Framework and Setup

The Relationformer [13] architecture described in section 3.1 defines the core model used in this work. In this thesis, the network architecture, prediction heads, and base loss components remain unchanged. The same training framework proposed by Berger et al. [2] is also adopted and applied without architectural modifications.

Berger's framework does not alter the transformer architecture. On the other hand, it proposes changes at training level to address three limitations of image-to-graph learning:

1. Severe class imbalance in relation prediction.
2. Distribution shift between source and target domains.
3. Dimensional mismatch between 2D and 3D data.

These challenges become crucial in transfer learning scenarios where graph density, visual appearance and spatial dimensionality differ across datasets. A model trained with standard supervision techniques does not reliably generalize under such shifts.

To address these issues, Berger et al. [2] introduce:

- a regularized edge sampling loss to stabilize relation learning across domains with different graph statistics,

- a supervised domain adaptation framework that aligns both image-level and graph-level representations,
- a projection mechanism that enables joint training on 2D and 3D data.

In this work, all three components are retained as defined by Berger in the original setup.

3.2.1 Problem Setting and Notation

The base image-to-graph formulation and notation were introduced in section 3.1. In this section, I extend this setting for the transfer learning scenario considered by Berger et al. [2].

Two domains are assumed:

- a **source domain** with image–graph pairs $\left(I^S, \mathcal{G}^S\right)$
- a **target domain** with image–graph pairs $\left(I^T, \mathcal{G}^T\right)$

The prediction model $\mathcal{F}$ remains identical across domains, working with both of them jointly. It maps an input image to a graph prediction $\mathcal{F}(I) = \hat{\mathcal{G}}$. The architecture, object and relation queries, and prediction heads are shared between source and target data. No domain-specific parameters are introduced.

Domain Shift In the considered setting, source and target domains differ in three main aspects:

- **Image statistics**: texture, signal-to-noise ratio, contrast, or acquisition modality.
- **Graph statistics**: number of nodes, edge density, node degree, tortuosity, and structural regularity.
- **Dimensionality** (for 3D tasks): 2D source images versus 3D target volumes

Even if the main tasks remain the same, these differences alter the distribution of both visual features and relational structure. Standard supervised training is not sufficient to enforce alignment between these distributions, and specific techniques are therefore applied.

Training Objective During training, the model is exposed to both the source and target domains. The goal is to exploit the source domain to obtain a model with higher performance on the target set, optimizing it such that:

- it learns meaningful relational structure from the source domain,
- it adapts to the target domain despite distribution shifts,
- and it remains stable under different graph densities.

The total loss therefore combines:

- object-level detection losses,
- relation prediction losses
- and additional regularization terms for domain alignment

3.2.2 Regularized Edge Sampling Loss

One of the main differences between the source and target domains in the considered cross-domain TL setting is the node and edge distribution. The number of nodes, edge density, and average node degree can vary substantially across datasets, directly affecting the relation prediction task. Moreover, relation prediction is one of the most unstable components in the examined setting, and this instability arises from the combinatorial nature of edge prediction. For a graph with N predicted nodes, the number of possible unordered node pairs grows as $O(N^2)$. However, only a small fraction of these pairs corresponds to true edges, while all remaining pairs represent negative relations. As a result, the relation classification problem is heavily imbalanced. In the original Relationformer work [13], the relation loss $\mathcal{L}_{rln}$ was computed over a fixed number of ground-truth edges/relations. This number was arbitrarily chosen as a dataset-specific hyperparameter. While this strategy is sufficient for training and evaluating on a single domain, it does not generalize well to cross-domain scenarios with different graph statistics.

Edge Imbalance in Graph Prediction Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a ground-truth graph for one training sample, using the notation introduced in section 3.1. The set of **active (positive) edges** is

$$A = \mathcal{E}.$$

The set of **background (negative) edges** consists of all unordered node pairs that are not connected, therefore:

$$B = (\mathcal{V} \times \mathcal{V}) \setminus \mathcal{E}$$

Since $|\mathcal{V} \times \mathcal{V}| \sim N^2$ and $\mathcal{E} \ll N^2$, it follows that:

$$|B| \gg |A|$$

If the relation loss is computed over all pairs in $\mathcal{V} \times \mathcal{V}$, background edges dominate the gradient. The model minimizes the loss by over-predicting "no-edge" for most pairs, which leads to under-connected graphs.

To avoid this effect, previous work [13] samples a fixed number m of relations, and computes the cross-entropy loss only to this subset. The sampled relation set is:

$$R = A \cup \bar{B}$$

where $\bar{B} \subset B$ is a randomly chosen subset of background edges such that

$$|A| + |\bar{B}| \leq m$$

The relation loss is then

$$\mathcal{L}_{rln} = \sum_{\{g,h\} \in R} \mathcal{L}_{CE} \left(e^{gh}_{rln}, \tilde{e}^{gh}_{rln} \right) \tag{3.3}$$

This strategy introduces 2 problems:

1. The hyperparameter m is dataset-dependent.
2. The loss behaves differently when graph densities vary across domains.

If m is too small, background edges are under-represented and the model over-predicts edges, since it does not have enough negative supervision. If m is too large, background edges dominate and the model under-predicts connectivity. The sensitivity becomes crucial in cross-domain training, where the number of nodes and edges differs substantially between source and target datasets.

Ratio-Based Regularization Berger et al. [2] replace the fixed-size sampling with a ratio-based strategy. In this case, instead of fixing the total number of sampled relations, they fix the ratio between active and background edges:

$$r = \frac{|\hat{A}|}{|\hat{B}|}, \quad r \in (0,1] \tag{3.4}$$

The key idea is to maintain a consistent balance between positive and negative edges across all samples and domains.

Let A and B be the original sets of active and background edges for a batch. The method constructs **upsampled multisets** $\hat{A}$ and $\hat{B}$ such that:

- All original active edges are included
- One of the sets is duplicated until the ratio r is satisfied

- Only one of the two sets is upsampled. The other remains unchanged.

Formally, let $\hat{R} = \hat{A} \cup \hat{B}$. If the batch contains relatively few active edges with respect to background edges, elements from A are duplicated until $\hat{A}/\hat{B} = r$. If instead active edges dominate, background edges are duplicated. Since duplication is performed only when the ratio constraint is violated, at most one class is upsampled.

The relation loss becomes

$$\mathcal{L}_{Reslt} = \sum_{\{g,h\} \in \hat{R}} \mathcal{L}_{CE} \left(e^{gh}_{rln}, \tilde{e}^{gh}_{rln} \right) \tag{3.5}$$

This formulation ensures that:

- No positive edge is discarded.
- No background edge is systematically ignored
- The gradient contribution of each class remains stable.

Berger et al. [2] report that a default value of $r = 0.15$ performed consistently across datasets with different graph densities.

Behavior Across Domains The advantage of this formulation becomes evident when applied in cross-domain training. When considering Berger's setup with a sparse road network with low node degree and a dense vascular graph with high branching frequency, the optimal m value differs for the two datasets. On the other hand, when using ratio-based sampling, supervision remains invariant to graph density. The loss always enforces the same relative importance between positive and negative edges. This invariance is essential in Berger's framework, which may contain samples from both source and target domains.

In this thesis, the regularized edge sampling loss is adopted exactly as defined in the original framework. No modifications to the sampling ratio or duplication strategy are introduced. This choice ensures that relation supervision remains stable across experiments, enabling comparison with the original training setup.

3.2.3 Supervised Domain Adaptation

In the considered cross-domain transfer learning setting, differences between source and target datasets are not strictly limited to graph statistics. Visual characteristics of the images can also vary significantly. Differences may include background intensity distributions, contrast, noise patterns, spatial resolution, and acquisition modality. At a structural level, graphs may differ in curvature, branching behavior, and edge regularity.

When trained jointly on both domains, a model may learn domain-specific features instead of domain-invariant structural representations. This limits generalization and weakens transfer performance between the considered domains. To address this issue, Berger et al. [2] introduce a supervised domain adaptation framework for image-to-graph transformers. The objective is to align feature representations across domains while preserving task-invariant cues. The alignment is performed at both image-level and graph-level features, using adversarial training.

Image-Level Alignment The image-level domain classifier operates on features representations extracted by the convolutional backbone directly from the input image. The convolutional backbone outputs a feature map with spatial dimensions $H' \times W' \times C$ (or $H' \times W' \times D' \times C$ in the 3D case). Each spatial location (u, v) inside this feature map represents the receptive field, or patch, of the original image centered at that location, and is treated as an individual sample. A small domain classification network predicts whether the considered feature originates from the source or the target domain.

Let $D \in \{0,1\}$ denote the domain label, where 0 is assigned to the source domain and 1 to the target one. The image-level domain loss is defined as a binary cross-entropy:

$$\mathcal{L}_{img} = -\sum_{u,v} \left[D \log p_{u,v} + (1 - D) \log \left(1 - p_{u,v}\right) \right] \tag{3.6}$$

where $p_{u,v}$ is the predicted probability that patch (u, v) belongs to the target domain. This classifier tries to distinguish domains, however the backbone is trained adversarially to prevent this discrimination, thanks to GRL.

Graph-Level Alignment Aligning only low-level image features is not sufficient, even if visual appearance is aligned. Structural differences may still be present at higher representation levels, harming the learning process. To address this, Berger et al. [2] introduce a graph-level domain classifier.

The input to the classifier is the concatenated transformer output tokens $T \in \mathbb{R}^{(\#o+\#r)} \times d$, which represent both object and relation embeddings. $\#o$ and $\#r$ are the number of object and relation tokens, respectively, and d is the number of hidden channels per token. These tokens encode higher-level structural information about the predicted graph. The graph-level domain classifier, similarly to the image-level one, predicts whether the considered representations originate from the source or target domain. The corresponding loss is:

$$\mathcal{L}_{graph} = -\left[D \log T + (1 - D) \log \left(1 - T\right) \right] \tag{3.7}$$

Gradient Reversal Mechanism Both domain classifiers are preceded by Gradient Reversal Layer (GRL) [18]. During the forward pass, the GRL acts as an identity function and does not modify the features. During backpropagation, instead, it multiplies the gradient by a negative scalar $-\lambda$, where $\lambda > 0$. As a result:

- the domain classifier is trained in the standard way to minimize the domain classification loss, improving its ability to distinguish between source and target representations;
- the feature extractor receives the reversed gradient and is therefore optimized to maximize the domain classification loss, encouraging it to produce domain-invariant features.

Consistency Regularization In addition to image- and graph-level alignment, Berger et al. [2] add a consistency constraint between the two classifiers. This is necessary since the two classifiers work on different feature spaces and could disagree. While the GRL ensures only that each classifier is fooled independently, it does not ensure that both representations carry the same domain signal, so the image classifier could predict *"source"* while the graph classifier predicts *"target"*. To align the two classifiers' predictions, Berger et al. [2] apply consistency regularization [42]. The consistency loss penalizes the squared difference between:

- the average image-level domain prediction
- the graph-level domain prediction

Formally:

$$\mathcal{L}_{cst} = \left\| \frac{1}{|I|} \sum_{u,v} d_{img}\left(p_{u,v}\right) - d_{graph}\left(D\right) \right\|_2 \tag{3.8}$$

This encourages agreement between the two domain signals. To make them comparable, the image-level signal is averaged, since it is computed as the sum over all considered patches and is therefore on a different scale with respect to the graph-level signal.

Role in the Overall Framework The supervised domain adaptation component ensures that:

- Low-level features are aligned across domains.
- High-level structural embeddings are aligned across domains.
- The model does not rely on domain-specific cues and is not biased toward either the source or the target domain.

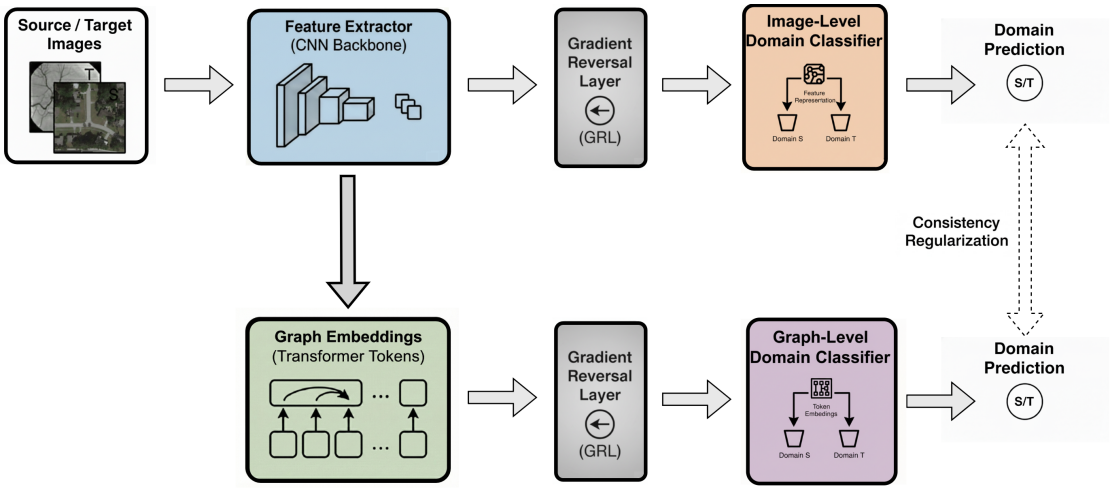

Figure 3.3: Supervised domain adaptation framework adopted from Berger et al. [2]. Image-level features extracted by the CNN backbone and graph-level embeddings produced by the transformer are each passed through a Gradient Reversal Layer (GRL) and corresponding domain classifiers. Adversarial training encourages domain-invariant representations at both visual and relational levels, while a consistency regularization term aligns image- and graph-level domain predictions.

Importantly, also in this case no architectural changes are introduced, and the framework is used as explained in Berger et al. [2] paper.

3.2.4 Combined Training Objective

As already mentioned, Berger's framework [2] does not modify the Relationformer [13] architecture. It modifies the optimization objective to make the learning process more stable in the context of cross-domain and cross-dimension transfer learning. The final loss formulation extends the original Relationformer one [13] reported in section 3.1, with additional regularization terms that address sparsity and domain shift.

The original components are briefly recalled, and their extensions are explained.

Object Detection Loss Object prediction follows the formulation introduced in section 3.1. After bipartite matching between predicted and ground-truth nodes using the Hungarian matching algorithm, three loss terms are computed:

- an L_1 regression loss $\mathcal{L}_{reg}$ on bounding box coordinates,
- a generalized IoU loss $\mathcal{L}_{gIoU}$,

- a classification loss $\mathcal{L}_{cls}$

The regression and IoU losses are evaluated only on matched predictions, while the classification loss is evaluated over all object queries. These components remain unchanged in Berger's framework. They ensure accurate node localization and classification [2].

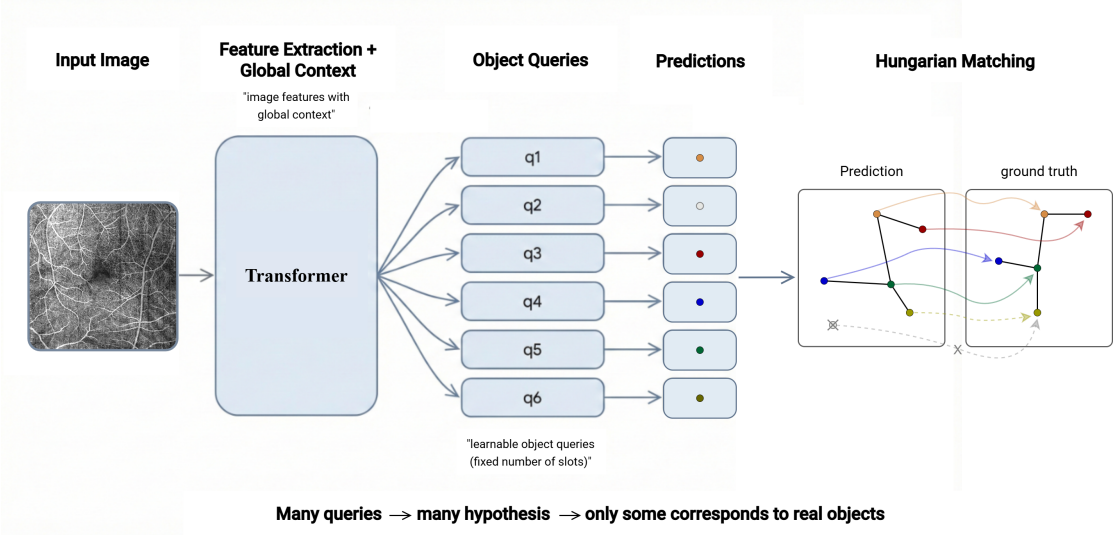

Figure 3.4: Image-to-graph transformer with set-based prediction and Hungarian matching. The input image is encoded into global feature representations, which are decoded using a fixed set of learnable object queries. Each query produces a node hypothesis with coordinates and class probabilities (node vs. no-object). Hungarian matching performs a global one-to-one assignment between predictions and ground-truth nodes by minimizing a joint classification and localization cost. Colored arrows indicate matched correspondences. The greyed node and crossed arrow illustrate an unmatched prediction, which is supervised as no-object during training and removed at inference.

Regularized Relation Loss The original Relationformer [13] architecture computes the relation loss over a fixed number of sampled edges. Berger replaces this strategy with the regularized edge sampling loss $\mathcal{L}_{Reslt}$ seen in section 3.2.2.

By regulating the class imbalance explicitly, this term prevents the relation loss from being dominated by background edges in sparse graphs. It also avoids the under-penalization of false positives in dense-graphs. As a result, the relation learning signal remains consistent across domains.

Domain Adaptation Losses To reduce distribution shift between source and target domains, Berger introduces adversarial alignment both at image-level and graph level through the losses $\mathcal{L}_{img}$ and $\mathcal{L}_{graph}$, as seen in section 3.2.3. In addition, the consistency loss $\mathcal{L}_{cst}$ is considered, minimizing the discrepancy between the image-level and graph-level domain predictions. This prevents the model from aligning only low-level features while ignoring structural representations, or biasing toward one of the two domains. The domain adaptation terms are active only during joint training on source and target data. They are not used during pure target-domain fine-tuning phase.

Final Objective The complete training objective combines all the components discussed:

$$\begin{aligned}\mathcal{L} = \sum_{i=1,[v^i_{cls}=1]}^{N} & \left[\lambda_{reg}\mathcal{L}_{reg}\left(v^i_{box}, \tilde{v}^i_{box}\right) + \lambda_{gIoU}\mathcal{L}_{gIoU}\left(v^i_{box}, \tilde{b}^i_{box}\right)\right] \\ & + \lambda_{cls}\sum_{i=1}^{N}\mathcal{L}_{cls}\left(v^i_{box}, \tilde{v}^i_{box}\right) \\ & + \lambda_{DA}\left(\mathcal{L}_{img} + \mathcal{L}_{graph} + \mathcal{L}_{cst}\right) \\ & + \lambda_{Reslt}\sum_{\{g,h\}\in\hat{R}}\mathcal{L}_{CE}\left(e^{gh}_{rln}, \tilde{e}^{gh}_{rln}\right)\end{aligned} \tag{3.9}$$

with λ_{reg}, λ_{gIoU}, λ_{cls}, λ_{Reslt}, and λ_{DA} as weights.

3.2.5 2D-to-3D Transfer Learning Framework

A further limitation of image-to-graph transfer learning is the scarcity of fully annotated 3D datasets. While large collections of labeled 2D images exist, as already pointed out, graph or segmentation annotations are expensive and difficult to obtain. This imbalance motivates the learning transfer from 2D source domains to 3D target domains.

Berger et al. [2] propose a framework that allows training a 3D image-to-graph transformer using 2D source data, without modifying the model architecture. The approach exploits a projection function that embeds 2D samples into a 3D volume compatible with the target domain.

Problem Setting The target model operates on 3D input volumes and predicts graphs embedded in three-dimensional space. Direct pretraining on 2D data is therefore impossible, as both the image dimensionality and the node coordinates differ.

Rather than adapting the model to load and use 2D images, or introducing dimension-specific components, Berger's framework adapts the input data. The same transformer is trained on both projected 2D samples and original 3D samples within the same optimization process.

Project Function Let I^s denote a source image and $\mathcal{G}^S = (\mathcal{V}, \mathcal{E})$ its corresponding graph. The projection function $\prod$ maps this pair to a 3D representation that matches the target input format. The projection is defined as a function

$$\prod : \left(I^S, \mathcal{G}^S\right) \longrightarrow (\bar{I}, \bar{\mathcal{G}})$$

and consists of three steps:

1. Resize I^s from $\left(H^{I^S} \times W^{I^S}\right)$ to the target domain's spatial patch size $\left(H^T \times W^T\right)$ by a linear downsampling operator $D : I^S \to I^{S'}$, where $D \in \mathbb{R}^{H^T W^T \times H^{I^S} \times W^{I^S}}$. G remains unchanged as normalized coordinates are used.

2. An empty 3D volume I is initialized as $I = \mathbf{0}^{H^T W^T D^T}$. The resized 2D image $I^{s'}$ is inserted at a fixed depth location $z_{1/2} = 0.5$ inside the empty volume I. At this point also the graph needs to be adapted to the new 3D formulation. Each node coordinate $v = (x, y) \in \mathcal{V}$ is ensured to have three dimensions by appending (or updating if the graph already had a trivial z coordinate) the depth coordinate, obtaining $v' = \left(x, y, z_{1/2}\right)$. The resulting graph is $\mathcal{G}' = (\mathcal{V}', \mathcal{E})$, where now $\mathcal{V}' \subset [0,1]^3$, effectively preserving graph connectivity. The final 3D volume containing the 2D image is represented as I'.

3. To expose the model to different spatial configurations, a three-dimensional rotation is applied. Let $R(\alpha, \beta, \gamma)$ be a rotation matrix parameterized by three angles. This rotation matrix is applied on the volume I' obtaining $\bar{I} = R(I')$, and to the graph nodes coordinates obtaining $V'' = \{Rv' |\ v' \in \mathcal{V}'\}$.

At this point the pair $\left(\bar{I}, \bar{\mathcal{G}}\right)$ is treated as a normal 3D sample.

A limitation of unconstrained continuous 3D rotations is that nodes initially located inside the projected volume may be mapped outside its spatial boundaries. The original Berger approach did not consider this possibility [2]. To address this issue, a new constrained angle-sampling strategy is adopted to enforce the geometric validity of the rotated graph.

Specifically, rotation parameters (α, β, γ) are sampled such that the rotated node coordinates remain within the normalized spatial domain. Formally, rotations are accepted only if the condition

$$R(\alpha, \beta, \gamma)v' \in [0,1]^3, \quad \forall v' \in \mathcal{V}'$$

is satisfied.

If such parameters are found, the rotation is applied jointly to the volumetric image and the node coordinates. If no valid rotation can be found after a fixed number of attempts, the sample is rotated using a discrete rotation multiple of 90°. This fallback guarantees validity, while preserving rotational augmentation. Independently of the applied rotation, a random reflection along each spatial axis is applied to the final configuration. Each axis-aligned flip along the x, y, and z direction is sampled independently and applied jointly to the volumetric image and the node coordinates.

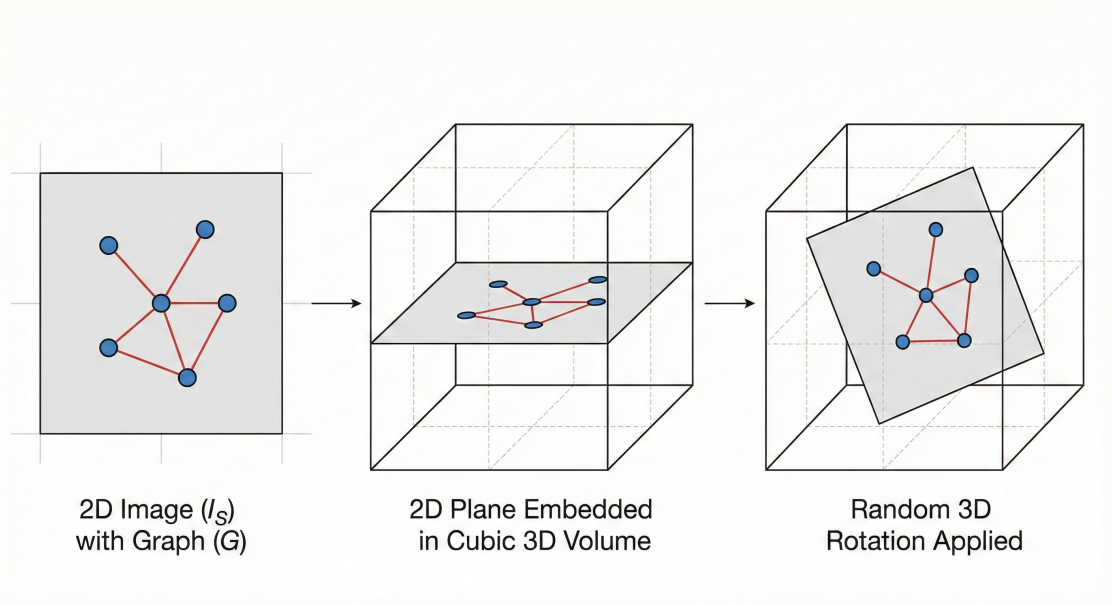

Figure 3.5: Schematic illustration of the 2D-to-3D projection function used for cross-dimension transfer learning in the considered image-to-graph framework. A 2D image with its associated graph is first resized and embedded as a planar slice within the center of an empty cubic 3D volume, while preserving the graph topology and relative node positions. The embedded image-graph pair is then subjected to a shared random 3D rotation, resulting in a rotated planar representation inside the volume. This simple projection enables the use of labeled 2D image-graph data for training 3D image-to-graph models without requiring handcrafted projections or model modifications, following the framework introduced by Berger et al [2].

Interaction With Transfer Learning Components The projection function does not try to replicate the appearance or statistics of the target domain. Differences in noise, contrast, or structural regularity are handled by the regularized edge sampling loss and the domain adaptation framework. Its only role is to make

2D data a usable auxiliary training source for the 3D model. By keeping the projection simple, the framework avoids handcrafted assumptions about the target data. The alignment between projected 2D and real 3D samples emerges through joint optimization.

Training Strategy During pretraining, projected 2D samples and native 3D samples are processed by the same model. All loss components described in section 3.2.4 are active. During fine-tuning, only real 3D data is used, and the domain adaptation losses are disabled. This strategy allows the model to reuse structural knowledge learned from 2D graphs while adapting to volumetric input.

Key Properties This framework has three relevant properties. First, it preserves the original architecture and prediction heads. Second, it does not require learning an explicit 2D-to-3D mapping. Third, it generalizes across domains, as the projection is independent of the target modality. As a result, 2D datasets can be used to support training in 3D settings where graph annotations are scarce.

3.3 Botanical Source Domain Selection

The considered framework relies on an auxiliary source domain to provide structured supervision for graph learning. The choice of this domain is guided not by visual similarity, but by the structural properties of the graphs it provides. Since the downstream task concerns vascular networks, an ideal source domain should reflect branching patterns governed by similar generative principles. In this work, plant branching systems are selected as the source domain, motivated by the shared biophysical constraints that shape both botanical and vascular transport networks.

3.3.1 Biological Transport Networks as Structural Priors

The influence of transport constraints on network morphology sets the ground to motivate the use of flow-derived data. Comparative studies have shown that networks formed by fluid flow in different physical contexts exhibit similar statistical properties. For example, Pelletier and Turcotte [43] analyze the geometry of river basins and leaf venation patterns, showing that both systems display comparable branching statistics and scaling relationships. These findings suggest that optimization of flow pathways induces universal geometric regularities across transport networks.

Moving from general transport networks, biological distribution systems, such as plant vasculature and animal circulatory networks, are organized to efficiently transport fluids inside the organism while minimizing energetic cost. A fundamental

principle describing this optimization is Murray's law [5, 26], which states that the radii of the parent and daughter branches at a bifurcation satisfy

$$r_0^\gamma = \sum_{i=1}^{N} r_i^\gamma \tag{3.10}$$

where r_0 is the radius of the parent branch, r_i are the radii of the child branches, and $\gamma \approx 3$ for laminar flow conditions. This relationship emerges from the minimization of total energy consumption, balancing viscous dissipation and the metabolic cost of maintaining fluid volume. Importantly, eq. (3.10) implies that biological transport branching systems are not arbitrary graphs, but structured and hierarchical networks where geometric and topological properties are strongly coupled.

Beyond local branching rules, the large-scale organization of biological transport networks is captured by the West-Brown-Enquist (WBE) model [6], which describes such systems as self-similar, space-filling fractal networks optimized for resource distribution. *Self-similar* implies that small branches resemble the pattern of larger ones, or that subtrees resemble the whole tree. This does not mean identical shapes; it means statistical invariance across scales. *Space-filling* means that these networks expand to reach all regions of the volume or surface that they serve. Under this model, the volumetric flow Q through a branch scales with its radius as:

$$Q \propto r^3 \tag{3.11}$$

reflecting the cubic dependence predicted by fluid dynamics. The WBE theory further predicts hierarchical organization and scale invariance, properties that can be found in connectivity patterns and branching statistics of both plant and animal vasculature systems. These characteristics directly influence graph-level statistics such as node degree distributions, path lengths, and branching angles, making biological networks particularly suitable as structural priors for vessel topology learning.

3.3.2 From Internal Vasculature to External Branching

Even though the downstream task focuses on vascular graphs, the source domain in this work consists of external tree branching structures rather than leaf venation. The validity of this choice is supported by the pipe model theory [27], which establishes a quantitative relationship between a plant's external branching architecture and its internal vasculature system. According to this model, the cross-sectional area of a branch is proportional to the total area of the vascular elements it contains

$$A_{\text{branch}} \propto A_{\text{conductive}} \tag{3.12}$$

indicating that the microscopic tree structure is a direct morphological expression of the underlying transport network. Consequently, the topology of branches

encodes the same functional constraints that govern internal fluid distribution, demonstrating the scale-invariance and fractal organization previously discussed in section 3.3.1. Learning from tree graphs therefore exposes the model to the geometric and relational patterns characteristic of biological transport systems, even when microscopic vessel annotations are unavailable.

3.3.3 Implications for Graph-Based Representation Learning

From a machine learning perspective, these theoretical results imply that plant branching networks provide a meaningful inductive bias for vascular graph prediction. First, the hierarchical organization implied by eq. (3.10) and eq. (3.11) naturally induces graphs with limited node degrees and recursive branching patterns, matching those observed in vascular datasets. Second, the space-filling property predicted by the WBE model leads to dense local connectivity and heterogeneous branch lengths, features difficult to capture using engineered networks such as urban road systems. Finally, the correspondence expressed in eq. (3.12) ensures that branch topology reflects the internal functional organization of the underlying vascular tissue, providing a biologically grounded proxy for transport networks.

By training on plant graphs, the model is exposed to curved segments, asymmetric bifurcations, hierarchical connectivity, and loop absence that are typical of vascular structures but largely absent in man-made networks. This exposure encourages learning relational features that are invariant to image appearance and instead capture fundamental structural regularities.

3.3.4 Role in the Transfer Learning Framework

Within the overall training pipeline, the botanical source domain provides dense supervision for both node detection and edge prediction. During the pretraining phase, the model learns how structural components interact in biological transport networks, based on the biological transport laws, effectively obtaining a prior over possible graph configurations. This prior reduces the reliance on large annotated medical datasets and improves the stability and topology learning when fine-tuning on limited vascular annotations.

The 2D and 3D settings are handled separately, as they present different challenges. These differences determine how the botanical source is used and how the model learns from it.

2D Setting In the 2D case, raw RGB images of plants paired with their graph annotations are used. This choice is motivated by the *Relationformer* [13] convolutional backbone, which is pretrained on ImageNet and can therefore process natural

image statistics. Although plant images differ from medical images, the backbone extracts low-level features that remain useful. Once the backbone extracts image features, the transformer encoder and decoder take over. The transformer does not need to learn basic visual patterns from scratch because of the pretrained backbone. Instead, it exploits the structured output space to learn and predict how to place nodes and how to define connectivity and relations between them. This division between the pretrained backbone, which extracts visual features, and the transformer, which predicts graph structure, makes the 2D training process efficient.

3D Setting In the 3D case, the domain gap is larger and the task becomes more difficult. Volumetric medical images differ from 2D images in several ways. First, these volumes do not contain color information in the same way as common natural images, showing mainly contrast between different tissues. Second, 3D model architectures are mainly used for medical applications (since 3D image volumes are typical of medical tasks) or other volumetric tasks. Because 3D data is rare outside medical contexts, there are no general-purpose 3D datasets like ImageNet [44]; consequently, the considered 3D backbone lacks pretraining.

To address this issue, the gap is bridged using a different approach. Static plant images are not used directly for 3D training; instead, plant data serves as a foundation for procedural synthesis, as described in section 4.3. By applying a fixed, rule-based process to image and graph data, a 3D model can be trained using 2D data in a controlled environment without requiring a pretrained backbone on natural RGB images.

3.3.5 Degree-2 Node Retention

Standard graph extraction pipelines typically simplify skeleton graphs by collapsing degree-2 nodes, replacing chains of intermediate nodes with straight-line segments between junctions. While this simplification can be effective for infrastructure networks such as road graphs [2], and beneficial for the training process in terms of complexity, it removes local geometric information that is characteristic of biological branching systems. In vascular and botanical transport networks, centerlines frequently exhibit tortuosity and gradual curvature that encode meaningful structural properties.

To preserve this geometric information, degree-2 nodes are retained during preprocessing rather than being contracted. Maintaining these intermediate nodes preserves the local curvature of branches and prevents the graph from degenerating into piecewise linear approximations between bifurcations. This representation more faithfully reflects the morphology of biological transport networks and provides richer relational supervision for the learning framework. In particular, the retained

nodes allow the model to capture curvature patterns and local geometric continuity, which are common in vascular anatomy and plant branching structures.

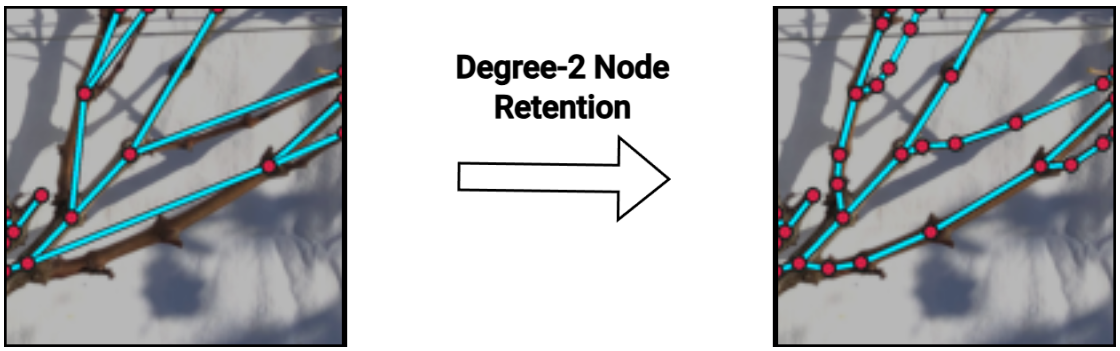

Figure 3.6: Illustration of degree-2 node retention. Intermediate nodes are preserved along vessel centerlines to maintain branch curvature and geometric continuity, providing a more faithful representation of the underlying vascular topology.

3.4 Multi-Target Diversity for Dataset Zero-Shot Learning

Building on the transfer framework described in previous sections, the model is evaluated to determine whether exposure to heterogeneous vascular data improves robustness to domain shifts and enables zero-shot inference on previously unseen datasets. A key limitation of existing approaches is their strong dependence on the training distribution: when applied to a new vascular domain, performance often degrades and retraining becomes necessary.

The multi-target strategy proposed in this section aims to reduce this dependency by encouraging the model to learn relational features that are shared across datasets rather than tied to a specific modality or acquisition protocol. While the core architecture and loss formulation remain unchanged, the training distribution is expanded to include multiple and diverse target domains, each contributing different structural statistics.

3.4.1 Problem Setting

Let $D_t^{(k)}$ denote a collection of K heterogeneous target datasets, each containing images and the corresponding graph annotations $(I, \mathcal{G})$ extracted through the preprocessing pipeline described in appendix 4. Unlike single-target training, where the model is optimized for one specific distribution, training samples here are drawn

from the union of all available target domains. This exposes the model to variability in acquisition modality, spatial resolution, and vascular density and morphology during optimization.

Datasets are first instantiated independently and then merged through dataset concatenation to form a unified training set. Let D_s be the source botanical domain, and

$$\mathcal{D}_t = \bigcup_{k=1}^{K} \mathcal{D}_t^{(k)}$$

the aggregated domain. The final training distribution is therefore defined as:

$$\mathcal{D} = \mathcal{D}_s \cup \mathcal{D}_t$$

Let n_k denote the number of available training patches for dataset $D_t^{(k)}$. In this collection, n_k varies strongly across datasets, as shown in table 5.2. Some domains provide several thousand patches, while others provide only a few hundred, and some provide only a handful. This imbalance arises from data availability and from the filtering steps described in appendix 4, where patches with too few or too many nodes and edges are discarded. While this filtering improves label quality, it reduces the usable size of some datasets. This imbalance has two main implications.

- First, it changes the effective training prior. If one dataset appears much more often in training, the model treats its graph statistics as the default. This can bias the node head toward a specific spacing between junctions and bias the relation head toward a specific edge length distribution.
- Second, it affects optimization stability. When small datasets appear rarely, the model sees them as outliers. The gradients from those domains become sparse and noisy. The model then fails to adapt to them, even if they contain useful structures.

For these reasons, the raw concatenated pool is not used directly as the sampling distribution. Instead, an effective target pool is constructed using a capped contribution per dataset.

3.4.2 Capped Per-Dataset Contribution

A maximum number of target patches per dataset is defined for the newly introduced auxiliary datasets, denoted by $N_{\max}$, while the reference target domain (syntheticMRI [25]) is left uncapped. For each dataset $D_t^{(k)}$ with n_k available training patches, the considered capped size is

$$\tilde{n}_k = \min(n_k, N_{\max}).$$

Then, $\tilde{n}_k$ patches are drawn from each dataset to form a balanced target pool

$$\tilde{\mathcal{D}}_t = \bigcup_{k=1}^{K} \tilde{\mathcal{D}}_t^{(k)}, \qquad |\tilde{\mathcal{D}}_t^{(k)}| = \tilde{n}_k.$$

For datasets where $n_k > N_{\max}$, sampling is performed without replacement and the subset is reshuffled at each epoch. This means that, across epochs, the model is exposed to many different patches from large datasets, but any single epoch is not dominated by them. For datasets where $n_k \leq N_{\max}$, all patches are included and only their order is shuffled.

This strategy creates a controlled mix of domains. While it does not fully equalize all datasets, it limits the influence of high-resource domains while retaining their variability.

3.4.3 Two Target-Scale Regimes

Two training configurations are defined to study the role of domain diversity and dataset scale. Both configurations rely on the capped contribution introduced above with $N_{\max}$ for auxiliary datasets. The syntheticMRI [25] dataset is treated separately as a baseline domain and is not subject to this cap, as it defines the reference training scale.

Regime 1: fixed-budget progressive diversification (∼4k). In the first configuration, the total number of target patches is kept constant at a fixed budget $B = 4000$. Training starts from a baseline distribution composed exclusively of syntheticMRI samples. Additional datasets are then introduced progressively.

For each newly added dataset $D_t^{(k)}$, up to $N_{\max}$ patches are included, and the number of syntheticMRI samples is reduced accordingly so that the total size of the target pool remains equal to B. This yields a sequence of training distributions with increasing domain diversity but constant overall scale.

Formally, letting $\mathcal{D}_{\text{syn}}$ denote the syntheticMRI subset and $\mathcal{D}_{\text{new}}$ the union of the added datasets,

$$|\hat{\mathcal{D}}_t^{(4k)}| = B, \qquad \hat{\mathcal{D}}_t^{(4k)} = \mathcal{D}_{\text{syn}}^{(B-|\mathcal{D}_{\text{new}}|)} \cup \mathcal{D}_{\text{new}}.$$

Regime 2: Progressive Scale Expansion In the second experiment, training also uses the syntheticMRI-only baseline as a starting point. Additional datasets are then introduced incrementally, each contributing at most N_{max} patches. In contrast to Regime 1, the number of syntheticMRI samples is not reduced, and the total size of the target pool increases as new domains are added.

The resulting target distribution can be expressed as

$$\hat{\mathcal{D}}_t^{(\mathrm{exp})} = \mathcal{D}_{\mathrm{syn}} \cup \bigcup_{k=1}^{K'} \tilde{\mathcal{D}}_t^{(k)}, \qquad |\tilde{\mathcal{D}}_t^{(k)}| \leq N_{\max}.$$

This configuration generates a sequence of training sets with both increasing domain diversity and increasing scale.

3.4.4 Sampling Within Training Batches

The implementation draws mini-batches from a mixture of source and target samples, following the transfer setup used in previous sections. The model heads and losses remain unchanged; only the selection and presentation of target samples are modified.

The following sampling logic is used.

1. Build the target pool for the chosen regime: either $\hat{\mathcal{D}}_t^{(4k)}$ or $\hat{\mathcal{D}}_t^{(\mathrm{full})}$.
2. Shuffle the pool at the beginning of each epoch.
3. Draw batches from the shuffled list, using the standard `PyTorch` dataset and `dataloader` pipeline.

In the 5k regime, the pool changes at each epoch because the fill samples are re-drawn and the capped subsets from large datasets are re-sampled. In the full regime, the pool is fixed but shuffled at each epoch.

Chapter 4

Datasets Construction and Preprocessing Pipeline

Some supporting components have been implemented that are not part of the main architecture, but were useful for constructing datasets, retrieving graphs, and visualizing them.

Output Format and Dataset Organization Each valid patch is saved along with its corresponding segmentation mask and graph annotation. Graphs can be exported in multiple formats, including a serialized Python dictionary, a JSON representation, or a VTK polyline format for visualization. When using the VTK format, node coordinates are normalized by the patch size, following the same convention used by downstream processing tools.

Patches are organized into training, validation, and test splits according to a predefined split file. Optional limits on the number of patches per image or per split can be applied to control dataset size. The same patch extraction procedure is used for all 2D datasets in this work, ensuring consistency across experiments. 3D datasets undergo a conceptually similar procedure, which produce the same training, validation an test patch sets for each dataset. Importantly, test, validation and training samples are divided before being patched, to avoid any for of data leakage between the three sets.

4.1 Ground-Truth Graph Construction Framework

A fundamental step to train the graph-extraction model was to obtain ground-truth graphs to supervise the training process. Some of the used datasets already provided

graph annotations [45, 46], while for all the others it was necessary to extract them. To this end, two main methods were tried that, even if similar, provided different outputs. Ultimately, only one of the two approaches was used, as it was faster and more widely adopted. The first was the Voreen tool [9, 7], while the second was implemented from scratch.

4.1.1 Voreen Graph Extraction Tool

Voreen (*Volume Rendering Engine*) is an open-source application development framework for the interactive visualization and analysis of multi-modal volumetric datasets [9]. It provides GPU-based volume rendering and data analysis techniques. The Voreen framework consists of a multi-platform `C++` library, which can be integrated into existing applications, and a Qt-based stand-alone application. Conceptually, Voreen is built around the idea of processors connected in a data-flow network. Each processor performs a single, well-defined task, such as loading a volume, computing a skeleton, extracting topology, or computing features, and data dependencies between processors are made explicit through typed input and output ports. This makes complex processing pipelines transparent, reconfigurable, and reproducible, which is particularly important for exploratory and research-oriented workflows [9]. Since its graph extraction process is deterministic, robust, and scalable, Voreen proves to be more suitable for ground-truth extraction than other graph generation algorithms.

From Binary Vessel Segmentation to Abstract Vessel Graphs In this thesis, Voreen was not used for rendering, but as a stage of the processing framework to extract vessel graphs from binary volumetric segmentations. The starting point of the pipeline is a foreground/background segmentation of the vascular structures in the 3D volume. The following graph extraction is entirely independent of the imaging modality and operates solely on binary data, which makes it broadly applicable across different acquisition techniques. The vessel graph extraction pipeline implemented in Voreen and used in this work follows the scalable and robust approach described by Drees et al. [7], which is included in Voreen starting from version 5.1. The pipeline is explicitly designed to handle arbitrarily large volumes, irregular vessel shapes, anisotropic voxel spacing, and non-tree-like network topologies, all of which are common in real biomedical datasets.

At a high level, the pipeline consists of four main stages that are evaluated iteratively:

- Skeletonization
- Topology extraction

- Voxel-branch assignment and feature extraction
- Iterative refinement (graph pruning)

These stages are executed repeatedly until a fixed point is reached, meaning that further refinement no longer changes the extracted graph.

Skeletonization The first step reduces the binary vessel segmentation to a one-voxel-thick centerline representation using a thinning algorithm. Voreen uses a modified version of the classical 3D thinning approach by Lee et al. [47], which relies solely on local 26-neighborhood information and therefore scales well to large volumetric datasets.

To make this step feasible for datasets that do not fit into main memory, the implementation explicitly tracks surface voxels of the object and processes them iteratively. During thinning, only surface voxels can be deleted, while interior voxels are preserved until surrounding voxels are removed. The algorithm identifies surface voxels, stores their positions on disk, and tests whether removing a given surface voxel would alter the topology of the foreground object. If the voxel is classified as topology-preserving, it is removed from the volume. The volume data itself is stored on disk and accessed via memory-mapped blocks, which keeps memory usage bounded while maintaining acceptable performance.

An important extension is support for anisotropic voxel spacing. Instead of thinning uniformly in all directions, the algorithm dynamically selects thinning directions based on real-world voxel spacing. This avoids biasing the skeleton toward axes with higher resolution and improves robustness for microscopy datasets, where axial resolution is typically much lower than in-plane resolution.

Topology Extraction and Proto-Vessel Graph Construction Once a voxel skeleton is available, the next step converts it into an explicit graph representation. Skeleton voxels are classified based on their number of neighbors into end points (degree-1 points), regular points (degree-2 points) or branch points (degree-N points, with $N > 2$). Connected components of end and branch voxels become graph nodes, while chains of regular voxels between nodes form a graph edge. Unlike earlier approaches, this step does not assume a tree topology, preserving loops and cyclic structures. Topology extraction scans the volume once and groups connected skeleton voxels using a connected-component labeling algorithm that works efficiently with data stored on disk. Rather than placing nodes at single voxel locations, their positions are defined as the average position of the corresponding voxel group, which leads to more stable node locations and fewer discretization artifacts. Degree-2 nodes get pruned, preserving nodes only at junctions and terminations.

The result of this step is a proto-vessel graph consisting of nodes, edges, and associated centerlines. At this stage, the graph still contains many small spurious branches caused by surface irregularities and discretization noise, especially in high-resolution datasets.

Voxel-Branch Assignment and Feature Extraction To compute meaningful geometric and morphological features for each vessel segment, the pipeline uses a mapping between foreground voxels and graph edges. A simple nearest-centerline assignment is not sufficient, because vessels with very different radii can be spatially close, leading to incorrect assignments. The implemented solution first assigns voxels using a Voronoi-style nearest-centerline mapping. This is followed by connected-component-based remapping step that identifies regions where voxels were incorrectly assigned to the wrong vessel segment. Remaining non-assigned regions are then flood-filled from neighboring labeled regions (by propagating the label of those neighboring regions).

This multi-stage assignment ensures that each voxel is consistently associated with the correct vessel segment, even in dense or irregular networks. Based on this mapping, a range of per-edge features is computed, including length, straightness, radius statistics, volume, cross-sectional area, and additional morphological descriptors. These features are aggregated in a single pass over the volume [7].

Iterative Refinement and Bulge-Size-Based Pruning A main contribution of the pipeline, and central reason for its robustness, is the iterative refinement step. Classical skeleton-based graph extraction methods tend to produce a rapidly increasing number of small, spurious branches as image resolution increases or surface noise becomes more pronounced. To address this, Voreen [9] introduces a dimensionless, scale-invariant pruning criterion called bulge size. Intuitively, the bulge size measures how far a branch extends from the parent vessel relative to the radii of the involved vessels. Branches that do not exceed a user-defined bulge size threshold are considered spurious and removed.

After pruning, the graph structure changes, which invalidates the previous voxel-branch assignment and feature extraction, therefore the pipeline recomputes skeletonization, topology extraction and feature extraction on the refined graph. This cycle is repeated until no further branches are removed. In practice, convergence is reached after a small number of iterations, even for very large datasets. Thanks to this iterative approach, resolution invariance can be achieved. Without refinement, increasing voxel resolution can increase the number of branches and the final graph would explode in terms of number of nodes and edges.

4.1.2 Centerline-Based Graph Generation

The main limitation of the Voreen graph extractor is that it discards degree-2 nodes, maintaining only junction and termination points. In particular, for datasets with highly tortuous vessels, this represents a limitation when the graph is intended to describe not only connectivity (which is the only possible analysis in that case) but also the vessel topology. To address this issue, a fully custom graph extraction pipeline was implemented, taking inspiration from the Voreen approach [7]. The main objective of this pipeline was to directly transform volumetric vessel segmentations into structured graph representations, while retaining full control over each intermediate processing step. Unlike black-box solutions, this design allowed for fine-grained experimentation with connectivity criteria, geometric constraints, and topological refinement strategies. The proposed pipeline was organized into two main stages: centerline extraction and graph construction with post-processing and refinement.

Centerline Extraction and Preprocessing As in Voreen [7], the first step consists of skeletonization followed by centerline extraction. Starting from the binary vessel segmentation, a morphological skeletonization operation was applied to extract a one-voxel-wide centerline approximating the medial axis of the vessel structures. This operation significantly reduced data complexity while preserving the global topology of the vascular network.

Skeletonization alone was not sufficient to capture all relevant vessel structures. In particular, very thin branches were sometimes excluded, resulting in isolated foreground regions that did not intersect the extracted skeleton. To address this issue, these regions were explicitly detected and, if located beyond a predefined distance threshold, identified as orphan regions. Orphan candidates were either connected to the main skeleton or discarded based on a combination of Euclidean distance, surface geodesic distance, and angular consistency. This decision logic was designed to distinguish true vessel endpoints from spurious bridges between unrelated branches, thus preventing the introduction of non-physiological connections.

Graph Construction from Centerline Representation Once the centerline points were extracted and filtered, the graph structure was constructed. Each centerline voxel was mapped to a graph node with an associated spatial position in world coordinates. Edges were created by connecting nodes corresponding to neighboring voxels according to the 26-connectivity of the skeleton, as done in Voreen [7], resulting in an initial graph encoding the voxel-level connectivity of the skeleton.

At this stage, the graph was still over-connected and contained an excessive number of nodes and edges. Therefore, several pruning and filtering steps were

applied to reduce the graph size while preserving topological consistency.

Topological Refinement and Pruning The refinement stage addressed both local and global inconsistencies. Triangular subgraphs arising from voxel discretization were identified and simplified by removing the longest edge in each triangle, effectively restoring a tree-like local structure. Subsequently, edges were evaluated based on their consistency with the underlying segmentation. For each edge, intensity values along the corresponding spatial path were sampled from the original volume, and edges traversing low-intensity regions were flagged as invalid. When necessary, surface geodesic distances were used as an additional criterion to discard edges deviating excessively from the vessel surface.

Very small connected components, with size below a minimum threshold, were removed, as they were typically associated with noise or segmentation artifacts rather than meaningful vessel structures. To increase geometric regularity, Laplacian smoothing was applied to node positions while preserving graph connectivity. Additionally, vessel radii were estimated for each node using the distance transform of the segmentation.

Finally, an iterative pruning procedure was applied to further simplify the graph topology. Degree-2 nodes forming near-collinear configurations (angle between adjacent edges above 165°) were progressively collapsed, and the incident edges merged. This operation reduced unnecessary intermediate nodes while preserving the overall vessel shape and connectivity, resulting in a more compact and interpretable representation.

Limitations Although the proposed pipeline enabled degree-2 nodes to be preserved, producing graphs that were visually and structurally more suitable for topology preservation beyond pure connectivity analysis, several factors motivated the choice of Voreen for ground-truth extraction.

The custom graph extraction pipeline required longer execution times per sample, and given the large number of samples requiring graph extraction, Voreen remained the more practical solution. In particular, Voreen is designed as a complete and integrated graph extraction tool, rather than a combination of independent scripts. While degree-2 node preservation could be beneficial for future applications, the proposed model needed to be compared with methods using Voreen-generated ground truth graphs, which primarily focus on connectivity estimation rather than detailed topological evaluation.

For these reasons, despite its flexibility and interpretability, the custom centerline-based pipeline was not selected for large-scale ground-truth graph generation. Nevertheless, its development provided valuable insights into the voxel-to-graph conversion process and informed several design choices adopted in the final preprocessing strategy.

4.1.3 Dataset-Specific Graph Post-Processing

To obtain a consistent representation across datasets and uniform data handling, all graphs were processed using the same post-extraction protocol. When graph annotations were not provided with the dataset, graphs were extracted using the Voreen pipeline [9, 7] with identical parameter settings across domains. Voreen outputs node and edge lists as two separate *.csv* files. Since image resolutions differ from dataset to dataset, node coordinates were normalized by the corresponding image size so that all graphs share a common coordinate system. The same normalization was applied to datasets that already provided graph annotations such as the plant one [46]. This step matters because the transformer predicts coordinates directly, and inconsistent scaling can look like label noise. To standardize also the graph format, graphs represented as node-edge csv pairs were converted to Visualization Toolkit (*.vtp*) files, which provide a uniform structure for networks and graphs, proving to be better suited than csv files.

No additional topological simplification or pruning was applied at this stage. The post-processing step is therefore limited to coordinate normalization and format conversion, ensuring that the original graph structure is preserved while enabling a unified processing pipeline.

4.2 Patch Extraction Framework

Patch extraction is a central component of the data construction pipeline in both the 2D and 3D settings. In particular, for 3D tasks it determines the effective supervisory signal available to the model in the absence of a pretrained backbone. Depending on the dataset, patch extraction is applied either after graph construction or directly on image-segmentation pairs prior to graph extraction. Unlike standard image patching, the procedure operates on structured annotations and preserves graph connectivity across patch boundaries.

Input Setup The source-domain construction strategy introduced by Berger et al. [2] is followed and adapted to the plant source domain. The procedure works on paired image, segmentation and graph annotations, and produces fixed-size training patches. Let an annotated sample (2D for simplicity) be defined as (I, M, G) where:

- $I \in \mathbb{R}^{H \times W \times 3}$ is an RGB image.
- $M \in \{0,1\}^{H \times W}$ is a binary segmentation mask.
- $G = (\mathcal{V}, \mathcal{E})$ is a graph represented by its set of nodes $\mathcal{V}$ and edges $\mathcal{E}$.

4.2.1 Botanical and 2D Datasets

For botanical and 2D datasets, graphs are first extracted or provided and patches are subsequently generated from the image-graph pairs.

Image Rescaling Before patch extraction, image and graph coordinates are optionally rescaled by a factor $s > 0$. This is done to capture more image content inside each patch, avoid patches with scarce foreground and graph information, and provide better supervision to the model. If the input image has very large resolution and dimensions, extracting 128^2 patches would lead to one to a few branches per image, which would not help the model learn complex structures and patterns. Let the resized image dimensions be:

$$H' = \lfloor H/s \rfloor, \quad W' = \lfloor W/s \rfloor$$

Graph node coordinates are rescaled accordingly:

$$(x_i', y_i') = \left(\frac{W'}{W} x_i, \frac{H'}{H} y_i \right) \tag{4.1}$$

This step ensures a consistent working resolution while preserving graph geometry.

Sliding Window Patch Extraction Squared patches of fixed size $S \times S$ using a sliding-window scheme with stride:

$$\Delta = \lfloor S\,(1 - \alpha) \rfloor \tag{4.2}$$

Where $\alpha \in [0, 1)$ denotes the overlap ratio between consecutive patches. Each patch is defined by its top-left corner (x_0, y_0) and corresponds to the spatial region:

$$\Omega = [x_0, x_0 + S) \times [y_0, y_0 + S) \tag{4.3}$$

The image and segmentation patches are given by:

$$\begin{aligned} I_\Omega &= I\,[y_0 : y_0 + S, x_0 : x_0 + S] \\ M_\Omega &= M\,[y_0 : y_0 + S, x_0 : x_0 + S] \end{aligned} \tag{4.4}$$

Graph Cropping The graph G is cropped to the patch domain Ω to form the local patch graph:

$$G_\Omega = (\mathcal{V}_\Omega, \mathcal{E}_\Omega)\,. \tag{4.5}$$

Graph cropping is less direct than cropping an image or a segmentation mask because edges can cross the patch boundary. The graph G_Ω is therefore constructed by selecting nodes inside the patch and clipping edges to Ω.

Nodes inside the patch are selected as:

$$\mathcal{V}_\Omega = \{v_i \in \mathcal{V} \,|\, (x_i, y_i) \in \Omega\}\,, \tag{4.6}$$

and are mapped to patch-local coordinates as:

$$\left(x_i^\Omega, y_i^\Omega\right) = (x_i - x_0,\; y_i - y_0)\,. \tag{4.7}$$

For edges, three cases are considered:

- **Fully inside edges:** if both endpoints lie in Ω, the edge is retained.
- **Partially intersecting edges:** if exactly one endpoint lies in Ω, the edge is clipped at the patch boundary and a boundary node is introduced at the intersection point.
- **Crossing edges:** if both endpoints lie outside Ω but the segment intersects the patch, two boundary nodes are introduced and connected.

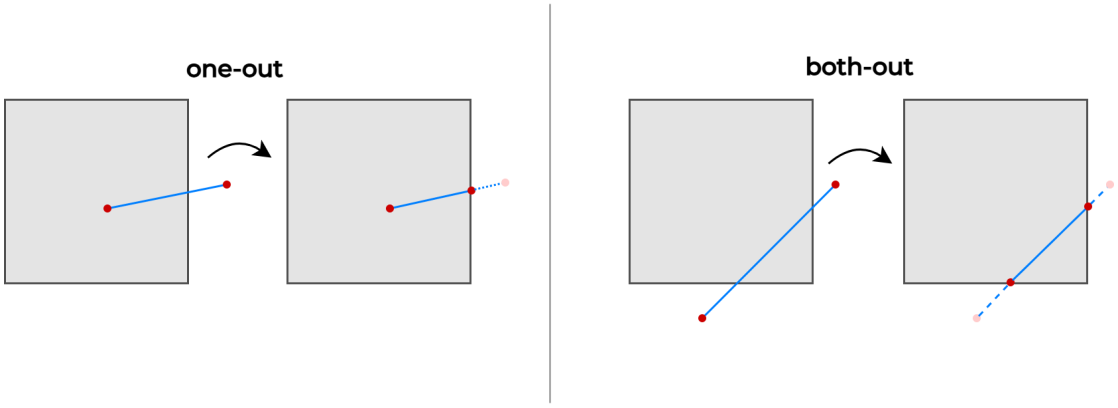

Figure 4.1: Graph cropping with edge clipping to the borders. On the left, the case where one of the two termination points falls outside of the patch; in this case one node is added in the intersection point and the edge updated with the new node. On the right, the case where both termination points fall outside the patch but the edge still intersects the patch; in this case two nodes are added and the edge updated with the two new terminations.

Edge clipping is performed using the Liang–Barsky method [48]. Given an edge with endpoints $v_0, v_1 \in \mathbb{R}^2$, the segment is represented in parametric form as:

$$v(t) = v_0 + t\,(v_1 - v_0), \qquad t \in [0, 1]. \tag{4.8}$$

The interval of parameter values for which $v(t)$ satisfies the rectangular constraints of the patch is then determined, yielding a bounded interval

$$[t_{\text{in}}, t_{\text{out}}] \subseteq [0,1]. \tag{4.9}$$

If this interval is empty, the segment does not intersect Ω.

When the interval is non-empty, the intersection points with the patch boundary are given by:

$$c_{\text{in}} = v(t_{\text{in}}), \qquad c_{\text{out}} = v(t_{\text{out}}). \tag{4.10}$$

These points correspond to the entering and leaving points of the segment with respect to Ω. Depending on whether the original endpoints lie inside or outside Ω, one or both intersection points are added as boundary nodes and connected to preserve local connectivity.

All retained and newly introduced nodes are expressed in patch-local coordinates using eq. (4.7). To prevent duplicate boundary nodes caused by numerical precision, intersection points with nearly identical coordinates are merged using a fixed tolerance. During graph cropping, node coordinates are represented in three dimensions by assigning a zero value to the depth coordinate, allowing the geometric operations of the original framework to be used. This process ensures that all edges in G_Ω remain fully inside the patch while preserving the original connectivity.

4.2.2 3D Vessels Patch Extraction

This section describes the procedure used to obtain patches from the vessel datasets used. The considered datasets were chosen from the vesselFM [28] D_{real} set and were preprocessed in the same way as in vesselFM before graph extraction. The extraction process is deterministic and identical across datasets to ensure consistent topology statistics. For these vascular 3D datasets, patch extraction is performed before graph generation. This ordering differs from the pipeline used for source data (botanical or roads) because vessel volumes are typically large and highly sparse. Extracting graphs directly on full-resolution volumes would be computationally expensive and would produce graphs with a wide and uncontrolled range of sizes. By first restricting the spatial support through patch extraction, the graph generation step operates on localized regions with bounded complexity.

The followed pipeline was:

1. Convert each dataset to a common folder and file format.
2. Extract fixed-size patches from *images and segmentations only* (no graphs at this stage, since no dataset provides graph annotations).
3. Apply data augmentation.

4. Run Voreen on each patch mask to extract a graph (same procedure as section 4.1.1).

Step 1: Dataset Conversion Each raw dataset uses a different naming convention and storage format. Using the VesselFM [28] preprocessing setup, all datasets were converted to a uniform structure with a consistent folder organization. The conversion script is a dispatcher that selects the correct dataset-specific converter based on the input folder name and writes the converted volumes to the chosen output path. This step ensures that all later stages can use the same file discovery logic and readers. At this stage, each dataset sample is defined by a pair of:

- $I \in \mathbb{R}^{H \times W \times D \times 3}$ raw image.
- $M \in 0, 1^{H \times W \times D}$ binary segmentation mask.

both with original resolutions.

Step 2: Patch Sampling Strategy Vessel structures are sparse, therefore uniform random sampling would produce many empty patches. To reduce this imbalance, cropping is guided by the mask foreground.

Let

$$\mathcal{F} = \{(x, y, z) \,|M\,(x, y, z) = 1\}$$

be the set of vessel voxels where there is foreground signal in the mask M. The smallest axis-aligned bounding box containing $\mathcal{F}$ (the total amount of foreground voxels) is computed as

$$\boldsymbol{b} = [z_{min}, z_{max}) \times [y_{min}, y_{max}) \times [x_{min}, x_{max}) \tag{4.11}$$

The image and mask are first cropped to this region, ensuring that subsequent random crops are centered on relevant anatomy. This operation reduces the probability of empty samples while preserving the full vessel information and extent inside the case.

Cubic patches of fixed size $S \times S \times S$ are extracted using a random spatial crop, padding the volumes if necessary so that every dimension is at least S.

$$\begin{aligned} I_\Omega &= I\,[z_0 : z_0 + S, y_0 : y_0 + S, x_0 : x_0 + S] \\ M_\Omega &= M\,[z_0 : z_0 + S, y_0 : y_0 + S, x_0 : x_0 + S] \end{aligned} \tag{4.12}$$

where the start index (z_0, y_0, x_0) is sampled uniformly inside the valid range defined by $\mathcal{F}$. The used padding is reflective padding, which means values outside the image boundary are created by mirroring the existing intensities at the border,

rather than filling with a fixed value. Formally, given the original image I, if padding is required along an axis, the padded region is

$$I_{pad}(i) = I(\phi(i)) \tag{4.13}$$

where ϕ maps indices outside the domain back inside by reflection.

Step 3: Data Augmentation Pipeline After cropping, patches undergo a sequence of stochastic spatial transformations, specifically:

- axis-wise flips
- 3D rotations
- elastic deformations
- zoom operations

These transforms are applied jointly to image and mask to preserve alignment. Let T_θ denote the composition of these random transforms parameterized by θ. The final training sample is:

$$(\tilde{I}, \tilde{M}) = T_\theta\left(P\left(I, M\right)\right) \tag{4.14}$$

Intensity values are then normalized to [0,1] using either min-max scaling or percentile scaling, depending on dataset configuration.

Step 4: Graph Extraction After obtaining the patches for the entire dataset, I extract the graphs using the exact process explained in section 4.1.1, always using the same parameter settings.

Handling Large `tiff` Files For very large microscopy volumes stored as `tiff` stacks, loading the full volume in memory is not feasible. Patches are therefore sampled by reading only the required sub-volume from disk, ensuring constant memory usage while preserving the same spatial sampling distribution as in-memory extraction.

4.2.3 Patch Filtering and Constraints

Not all extracted patches are retained. A set of filtering rules is applied to remove samples that do not provide meaningful supervision or that are unsuitable for training. The same filtering strategy is applied uniformly to all dataset patches used, including botanical and 2D data as well as 3D datasets. These constraints reduce dataset imbalance and ensure that the final training set contains patches that are both informative and computationally manageable.

Supervision validity. Patches with an empty graph, i.e., containing no nodes or edges after cropping, are discarded since they do not provide a learning signal for the relational task.

Foreground support. Patches with fewer than a minimum number of foreground pixels in the segmentation mask are removed. This avoids retaining samples where the graph is present but the visual context is too limited to be informative.

Complexity control. Patches whose number of nodes or edges falls outside a predefined range are excluded. The lower bound removes nearly empty samples, while the upper bound prevents overly complex patches that would lead to excessive memory usage.

4.3 Botanical Data Preparation Pipeline

Two processing steps were required to use the 2D botanical dataset as a source domain. Segmentations were first reconstructed from graph annotations, and the resulting data were then transformed to obtain inputs compatible with the 3D network.

4.3.1 Graph-Derived Segmentation Reconstruction

Unlike the road network dataset [45], the plant dataset [46] was not provided with segmentations. Since segmentations are required for procedural input synthesis for the 3D model (described in the following section), a segmentation reconstruction script was implemented that starts from the graph and generates a binary segmentation mask.

Let an image I have spatial size (H, W) and an associated graph $G = (\mathcal{V}, \mathcal{E})$. An initial binary mask $M^{(0)} \in \{0, 255\}^{H \times W}$ is constructed by drawing each edge $e_{ij} = (v_i, v_j)$ using the parametric representation:

$$e_{ij}(t) = v_i + t(v_j - v_i), \quad t \in [0,1]. \tag{4.15}$$

Pixels that lie within a fixed width $w = 3$ of the segment are then marked, so that the full graph is obtained as the union of all rendered edges. For each pixel p in the image:

$$M^{(0)}(p) = \begin{cases} 255, & \text{if } \exists\, e_{ij} \in \mathcal{E} \text{ such that } \operatorname{dist}(p, e_{ij}) \leq \frac{w}{2}, \\ 0, & \text{otherwise.} \end{cases} \tag{4.16}$$

Thus, if the considered pixel lies sufficiently close to an edge (dist $\leq w/2$), it is marked as foreground (value 255); otherwise, it is assigned to the background

(value 0). The implementation also allows the generation of thicker segmentations using the same approach described in section 4.3.2.

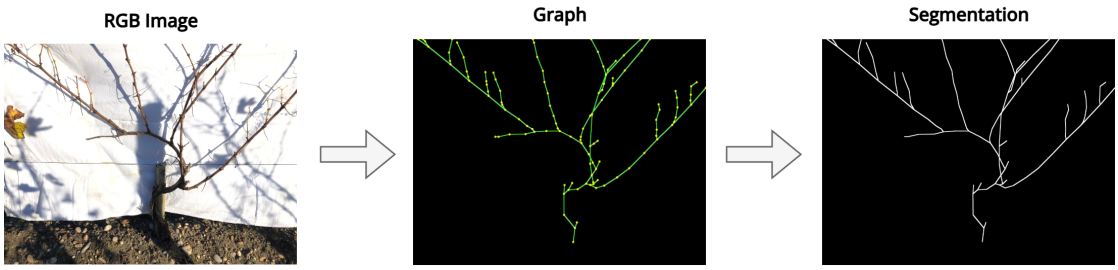

Figure 4.2: Botanical data preparation pipeline. A raw RGB plant image is first associated with its annotated branching graph, which encodes node positions and connectivity. The graph structure is then rasterized to reconstruct a corresponding segmentation mask, enabling successive generation of the 3D input images for supervision, while preserving the underlying topology.

4.3.2 Procedural Generation of 3D Input Images

As mentioned in the section introduction, for the 3D setting input images were not taken from raw plant photographs. Instead, image-like inputs were procedurally generated starting from the corresponding segmentations, following the same cross-dimensional transfer learning strategy used by Berger et al. [2] for road-network data and adapting it to plant data. The procedure was deterministic and applied uniformly across all splits. The resulting inputs were two-dimensional images with a single channel.

Let a segmentation patch be $M_\Omega \in \{0,1\}^{H\times W}$, where Ω represents the patch domain and foreground pixels correspond to plant structures. First the segmentation values are normalized to the unit interval:

$$M^{(0)} = \frac{1}{255} M_\Omega \tag{4.17}$$

To obtain a dense representation with smooth tubular structures, a sequence of two-dimensional convolution operations is applied using a fixed kernel $K_{\text{ones}} = \mathbb{1}^{3\times 3}$. The convolution is applied iteratively as:

$$X^{(k+1)} = K_{\text{ones}} * X^{(k)}, \quad k = 0, \ldots, N-1 \tag{4.18}$$

Where $*$ represents the convolution operation, with unit stride and zero padding. In practice 4 convolution steps are applied. After the final iteration, values are clipped

to the interval $[0, 1]$ and rescaled to standard image intensity range, producing the final image as:

$$I_\Omega = 255 \cdot X \tag{4.19}$$

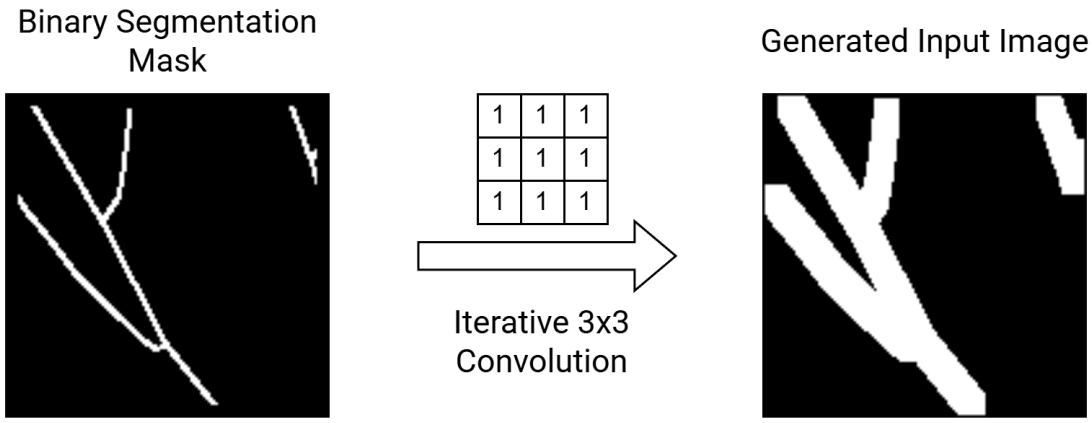

Figure 4.3: Procedural generation of input images from binary segmentations. The original mask is iteratively convolved with a 3×3 kernel to produce thicker and smoother tubular structures used as model inputs.

The final output image $I_\Omega \in \mathbb{R}^{H \times W}$ constitutes the input image used for training the 3D model. This operation thickens the original binary structures from the plant segmentations and ignores RGB content, which is not useful for a non-pretrained 3D model. It also produces smooth intensity variations around the foreground to emulate medical scans while preserving the original topology encoded in the mask.

Overall, this approach serves two purposes. First, it reduces the risk of producing patches with very sparse or nearly empty foreground content. Second, it provides a smoother and more robust signal when patches are extracted around graph structures, especially in cases where annotations are thin or partially fragmented. The resulting thickened mask is used as the patch input, while still preserving the original vessel topology. The botanical source increases the amount of structured training data available for image-to-graph learning. Second, it provides a semantic prior that aligns with vascular topology rather than image appearance. This strategy focuses supervision on graph structure, which is the core objective of the task. It allows the model to learn biologically meaningful connectivity patterns without requiring additional medical annotations.

Chapter 5

Experiments and Results

This chapter defines the protocol used to evaluate the proposed approach. It describes the datasets, preprocessing pipeline, training configuration, and evaluation procedures applied in all experiments. The focus is on implementation details and controlled comparisons to ensure a fair assessment of the methodological contributions.

The underlying Relationformer-based framework and loss design introduced by Berger et al. [2] are kept untouched. This ensures that observed performance differences arise from the training strategy and data choices rather than from architectural modifications.

Two main questions are investigated:

- **Semantic prior**: Do plant branching structures provide a better structural prior than road networks when transferring to vessel graphs?
- **Zero-shot generalization**: Does training on increasingly varied target data improve performance on completely unseen datasets without additional fine-tuning?

The scope of this chapter is to explicitly define the evaluated questions, the components that are held constant (architecture and loss), and the factors that vary (source domain and training scale), ensuring that the experimental setup is fully reproducible.

5.1 Datasets

Source Datasets The source datasets used are:

- **20cities** [45] (road networks) as the conventional non-biological source used in prior work [2].

- **PlantsGuyot3d2** [46] (plant branching graphs) as the proposed semantic-aligned source.

Dataset	Dimension	Spatial Size	# Train	# Val	# Test
20 U.S. Cities	2D	128×128	$99.2k$	$24.8k$	$25k$
plants-Guyot3d2	2D	128×128	$25.9k$	$6.4k$	$6.7k$

Table 5.1: Source dataset summary. The reported number of training, validation and test samples are the total number of samples available.

The selected source datasets were jointly used with the target datasets during pretraining to enable the model to learn more general graph representations and to compensate for the limited availability of annotated vessel data.

Target Datasets The vessel target data comprised both 2D and 3D datasets, depending on the final task considered. The 2D datasets included 128×128 patches from the OctaSynth dataset [49].

For the 3D setting, syntheticMRI [25] was used as the default target dataset. For the multi-target setup, the same dataset protocol introduced in vesselFM was followed. In particular, the curated real data collection D_{real} was partially employed. All datasets that were publicly available and belonged to D_{real} were used with the exact preprocessing and patch extraction procedures described in the original work. This included dataset-specific operations such as resampling, cropping, intensity clipping, and mask post-processing, ensuring that the data adhered to consistent vascular imaging characteristics. The full list of target datasets used was reported in table 5.2. All considered 3D patches had a size of $64 \times 64 \times 64$.

By adopting these datasets without modification, the effect of the proposed methodological contributions is isolated from differences in data curation.

5.2 Data Preprocessing and Graph Construction

All datasets were preprocessed so that at the end of the process they had the exact same format and organization: an input image/volume, a segmentation mask, and a ground-truth graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where:

- Each node is represented by its x, y (and z if 3D) normalized coordinates and an id.
- Each edge is represented by the pair of ids representing the connected nodes.

	Dataset	Tissue Type	Imaging Modality	# Patches	Pre processing
Evaluation	SMILE-UHURA [50]	human brain	MRA	500	-
	OCTA [51]	mouse brain	OCTA	100	-
	TubeTK [52]	human brain	MRA	1168	r, mp
	tUbeNet [53]	mouse liver	HREM MRI	236	-
		mouse brain	two-photon mic.	429	-
	TopCoW [54]	human brain	CTA	1146	r
		human brain	MRA	762	r
	VesSAP [55]	mouse brain	light-sheet mic. (EB)	113	ic
		mouse brain	light-sheet mic. (WBA)	113	ic
	HR-Kidney [56]	mouse kidney	X-ray	8625	mp
	3D-IRCADb-01 [57]	human liver	CT	547	c, mp
	IXI [58]	human brain	MRA	2070	-
	VesselExpress [59]	mouse brain	light-sheet mic.	6429	-
		mouse heart	light-sheet mic.	10	-
		mouse bladder	light-sheet mic.	200	-
	MiniVess [60]	mouse brain	two-photon mic.	285	-
	HiP-CT [61]	human kidney	CT	424	-

r: resampled; c: cropped; mp: mask post-processed (e.g., smoothing or conversion from multi-class to binary); ic: intensities clipped.

Table 5.2: Overview of the target vascular datasets used in training and evaluation. For each dataset, the tissue type, imaging modality, number of available patches, and applied preprocessing operations are reported. Preprocessing abbreviations are defined below the table.

This uniform representation allows the training loop to handle multiple datasets without redundancy.

Graph Extraction From Segmentation Masks When graph annotations were not provided, as in the case of all the vessel target datasets listed in appendix 4, graph annotations were generated using the Voreen pipeline [7], following exactly the procedure described in section 4.1.1.

For all datasets, the bulge size parameter was set to 3, as suggested in [15], while all other parameters were kept at their default values. No dataset-specific adjustments were introduced.

Patch Extraction and Normalization All datasets were standardized using the patch extraction procedure described in appendix 4. Inputs were cropped into fixed-size patches:

- 128×128 for 2D images,
- $64 \times 64 \times 64$ for 3D volumes.

Graph node coordinates were normalized to the patch coordinate system, yielding a [0,1] range along each spatial axis. Patch filtering was applied following section 4.2.3, retaining samples with a number of nodes and edges in the [6,70] range and at least 50 foreground pixels. These settings were kept identical across all datasets.

Edge Handling at Patch Borders Patches may cut vessels at the borders. This issue arises both for datasets with existing graphs (20Cities [45], Plants-Guyot3d2 [46]) and for datasets where graphs are extracted from full images (OctaSynth [49]). Border effects are handled using the procedure described in section 4.2. For anisotropic 3D volumes, reflective padding is applied before patch extraction, as detailed in section 4.2.2. When graph annotations are available, graph edges are clipped to the patch borders to preserve local connectivity. The same procedure is applied to all relevant datasets without introducing dataset-specific modifications.

5.3 Implementation Details

Model Implementation All experiments used the Relationformer architecture [13], with the training framework described in section 3.1. The model was implemented in PyTorch, and the original codebase was extended to support cross-domain and cross-dimension transfer learning in the botanical and multi-target setups.

For 2D experiments, feature extraction relied on a ResNet-101 backbone [10] pretrained on ImageNet [44] combined with sine positional encodings and four feature levels. Images were provided as three-channel RGB inputs and were fed

directly to the pretrained backbone without modifying the first convolution layer. The transformer used a hidden dimension of 512 with 8 attention heads and 3 encoder and decoder layers. The model employed 80 object tokens and 5 relation tokens, and deformable attention was used in both encoder and decoder.

For 3D experiments, inputs were single-channel volumetric patches. This included both medical vessel data and botanical/road datasets, where images were converted to grayscale and projected into 3D volumes during preprocessing. Feature extraction was performed with a squeeze-and-excitation residual encoder [62] trained from scratch, using an embedding dimension of 256 and a patch size of $4 \times 4 \times 4$. The deformable transformer used a hidden dimension of 552, 6 attention heads, and 4 encoder and decoder layers. In this setting, the model used 120 object tokens and 2 relation tokens. For the multi-target experiments, the maximum number of samples for the auxiliary datasets was set to $N_{max} = 300$, which represented a compromise between ensuring sufficient intra-dataset variability and preventing large datasets from dominating the training distribution. The choice was heuristic, and no dedicated ablation on this parameter was performed in this work.

Across all experiments, the architectural configuration remained fixed within each dimensional regime, and only the input dimensionality and data pipeline differed. This ensured that performance differences arose from the training setup and data composition rather than from architectural changes.

Training Configuration All models were trained using the AdamW optimizer. The optimizer used two parameter groups: one group contained the domain discriminator parameters, while the second group contained all remaining network parameters. This setup allowed the discriminator to use a dedicated learning rate while keeping shared weight decay across the model.

The learning rate followed a warmup phase followed by a decay schedule during training. For 2D experiments, models were trained using a step decay schedule implemented with `StepLR`, where the learning rate was reduced once at epoch `LR_DROP`. For 3D experiments instead, a `LambdaLR` scheduler was used, which applied a linear warmup phase followed by polynomial decay with exponent 0.9.

All experiments ran for a fixed number of epochs. The initial training phase, in which the model was pretrained jointly on source and target domains, was performed for 50 epochs. A subsequent training stage using only the target dataset was then run for 100 epochs. Models were evaluated using the final training checkpoint. No model selection based on validation performance was performed.

For domain adaptation, adversarial training was used with a gradually increasing gradient reversal strength. This schedule limited the influence of the domain loss at the beginning of training and increased it as the model converged.

For relation supervision, the same ratio-based edge sampling value suggested by Berger et al. [2] was used, with edge upsampling so that all positive edges were

Parameter	**2D Training**	**3D Training**
Optimizer	AdamW	AdamW
Initial learning rate	2×10^{-4}	7×10^{-5}
Backbone learning rate	2×10^{-5}	7×10^{-5}
Weight decay	1×10^{-3}	1×10^{-4}
Scheduler	StepLR	polynomial decay (LambdaLR)
Epochs	50	50
Batch size	32	32
Input patch size	128×128	$64 \times 64 \times 64$
Gradient clipping	0.1	5.0
Edge sampling ratio r	0.15	0.15
Max sampled edges	9999	9999
Edge upsampling	Yes	Yes
alpha (experiment 1)	1.0	0.6
alpha (experiment 2)	-	0.6
alpha (experiment 3)	-	1.0
Loss weight λ_{bbox}	2.0	3.0
Loss weight λ_{class}	3.0	4.0
Loss weight λ_{card}	1.0	0.8
Loss weight λ_{node}	5.0	2.0
Loss weight λ_{edge}	5.0	6.0
Loss weight λ_{domain}	1.0	0.1
Domain adversarial training	Enabled	Enabled
Consistency regularization	Enabled	Enabled
Mixed precision	Enabled	Enabled

Table 5.3: Training hyperparameters used for 2D and 3D experiments.

always included in the loss.

The complete set of hyperparameters was reported in table 5.3.

Batch Normalization Calibration During 3D evaluation it was observed that predictions differed significantly depending on whether the model was executed in `train()` or `eval()` mode. This behavior arises because Batch Normalization layers rely on running statistics that may not reflect the distribution of the target dataset after transfer learning. To stabilize inference, the Batch Normalization statistics are recalibrated before testing. The model is temporarily set to training mode and a forward pass is performed over the validation set without gradient updates. This

step updates the running mean and variance of the Batch Normalization layers while leaving the model weights unchanged. The model is then switched back to evaluation mode to produce test predictions. Test samples are never used during calibration to avoid data leakage.

5.4 Evaluation Metrics

Predicted vascular graphs are evaluated using a set of metrics that measure both detection performance and structural similarity between the predicted graph and the ground-truth graph. These metrics assess whether the model correctly identifies graph components (nodes and edges) and whether the resulting graph preserves the topology and geometry of the vascular network. The evaluation protocol follows the methodology used by Berger et al. [2].

5.4.1 Object Detection Metrics

For both node and edge detection metrics, mean average precision (mAP) and mean average recall (mAR) are employed [63]. To compute the intersection over union (IoU) for node detections, a fixed-size bounding box is placed around each predicted and ground-truth node. This converts the node locations into small spatial objects that allow overlap evaluation.

For edge detections, bounding boxes are constructed around the line segments connecting the corresponding node pairs. The bounding box spans the spatial extent of the edge between its two endpoints. To avoid degenerate boxes for nearly axis-aligned edges, a minimum spatial size m is enforced along each dimension. If the difference between the coordinates of the edge endpoints along one dimension is smaller than m, the bounding box size along that dimension is set to m.

A predicted node or edge is considered a correct detection when the IoU between the predicted and ground-truth bounding boxes exceeds a predefined threshold. Based on these matches, true positives, false positives, and false negatives are determined and used to compute precision and recall. The mAP corresponds to the area under the precision-recall curve obtained by varying the detection confidence threshold, while mAR measures the average recall across thresholds. Following the COCO evaluation protocol, the mean AP and mean AR are computed by averaging the results across IoU thresholds ranging from 0.5 to 0.95.

Mean Average Precision (Node/Edge mAP) mAP evaluates how precisely the model predicts node or edge locations. Precision measures the fraction of predicted nodes or edges that correspond to real ones among all predictions. Higher mAP indicates more accurate detections and fewer false positives.

Node Mean Average Recall (Node/Edge mAR) mAR measures how many ground-truth nodes or edges the model successfully detects. Recall corresponds to the fraction of ground-truth nodes or edges that are matched by predictions. Higher mAR indicates that the model captures a larger portion of the true graph structure.

Overall, when $mAR \gg mAP$, many ground-truth elements are detected but with numerous false positives. Conversely, when $mAP \gg mAR$, predictions are accurate but many ground-truth elements are missed.

5.4.2 Topological Metrics

The TOPO score [64] evaluates whether the predicted graph preserves the topological structure of the vascular network. It measures whether the predicted graph reproduces the same connectivity relations between nodes as in the ground-truth graph, ensuring that branches and junctions are connected in the correct way rather than only detecting individual node and edge displacements.

The metric is computed using two values:

- **TOPO precision**: measures the fraction of predicted connections that correspond to valid topological connections in the ground-truth graph.
- **TOPO recall**: measures the fraction of ground-truth connections that appear in the predicted graph.

High TOPO precision indicates few incorrect connections, while high TOPO recall points out that the predicted graph retains most real vessel branches. This metric is widely used in road-network and graph extraction tasks since it evaluates structural correctness rather than only local detection accuracy.

The same implementation and parameters used by Biagioni et al. [64] were used. The original formulation of the TOPO metric is defined for planar graphs and is therefore commonly applied to two-dimensional datasets. For this reason, TOPO scores are reported only for 2D experiments in this work.

5.4.3 Graph Distance Metric

The Street Mover Distance (SMD) [65] measures the structural difference between two graphs by comparing their geometric representations. This metric approximates the Wasserstein distance between the predicted graph and the ground-truth graph. In practice, both graphs are converted into sets of equidistant points sampled along their edges, which represent the network structure. The metric then computes the minimum transport cost required to transform one point distribution into the other by moving these elements in space. The cost reflects the geometric displacement between the sampled points of the two graphs.

A lower SMD indicates that the predicted graph closely matches the ground-truth graph in terms of geometric alignment. Unlike the detection metrics presented above, SMD evaluates the geometric similarity between the predicted and ground-truth graphs as a whole. It is therefore sensitive to global spatial differences such as displaced vessel segments or overall misalignment of the vascular network.

However, it is important to note that SMD measures geometric transport cost rather than semantic graph quality. During evaluation, it was observed that predictions containing correct branches and an overall better topology, but slightly shifted from the ground truth, could yield a higher SMD than topologically poorer predictions that lie closer to the ground-truth graph. This behavior arises from the way SMD is computed. The transport cost depends primarily on the spatial distance between sampled points. As a result, missing structures may incur a lower cost than displaced ones, since point matching is performed based on the shortest distance between elements.

Furthermore, SMD does not explicitly evaluate graph connectivity. Disconnected components, broken branches, or incorrect edge relationships may therefore produce relatively low SMD values if their geometry lies close to the ground-truth graph. In some cases, predictions containing many spurious nodes or edges may achieve a lower SMD than predictions with fewer but more semantically correct structures.

For this reason, SMD should be interpreted together with detection and topology metrics. While SMD provides a useful measure of global geometric similarity, it should not be used as the sole reference metric to assess graph quality.

5.5 Experiments

This section describes the experiments performed to study how structural pretraining, domain diversity, and dataset scale affect vascular graph extraction. The goal is to isolate the contribution of each factor under controlled training conditions. All experiments use the same model architecture and optimization settings described earlier, so differences in performance reflect only changes in data composition and training regimes.

5.5.1 Structural Source-Domain Comparison

The first experiment evaluates the effect of structural pretraining and examines whether structural alignment between the source and target domains influences transfer learning performance. Auxiliary source datasets provide additional graph supervision, while differing in their underlying structural organization. Two main source domains are considered: infrastructure networks and botanical branching networks. Both provide large collections of annotated graphs, while differing in topology, branching hierarchy, and growth constraints.

In this setup, the model is first trained jointly on the source domain and the target vascular dataset. After this first joint phase, training continues using only vascular data while keeping all model parameters trainable. This procedure follows the transfer learning protocol introduced by Berger et al. [2], and enables the adaptation of relational features learned from the source domain to the vascular prediction task.

The comparison between the two source domains is conducted under identical training conditions. For road networks, both a large configuration (99.2k source patches) and a size-matched configuration (25.9k source patches) are evaluated. The large road configuration reproduces the training scale used in [2] and serves as a high-capacity reference. The size-matched configuration ensures that observed differences between roads and plants cannot be attributed to dataset scale. In addition, variants of the preprocessing pipeline that retain intermediate degree-2 nodes in the source graphs are evaluated. This retention preserves local curvature information and tests whether finer geometric supervision improves relational modeling.

Both source sets are also compared against a baseline model trained without the structural pretraining phase, using only vascular target data and a random weight initialization [66], standard practice when no pretraining is available. The experiment is performed on both 2D and 3D targets. Results are measured using graph similarity metrics and detection accuracy to capture both geometric and topological consistency.

5.5.2 Sample Efficiency Analysis

To evaluate how structural pretraining affects annotation requirements, a controlled sample-efficiency experiment is conducted. Models pretrained on each structural source domain are fine-tuned on progressively smaller subsets of the target vascular dataset. The validation and test splits remain fixed across all configurations to ensure consistent and comparable evaluation.

Subsampling is performed at volume level to preserve spatial coherence and graph statistics of each sample. Training is conducted using annotation fractions of 100%, 50% and 30% of the available target data. All other training parameters, including learning rates, batch size, and optimization schedule, remain unchanged. This ensures that performance differences arise solely from the amount of annotated data and the structural prior used during pretraining.

Both plant-pretrained and road-pretrained models are evaluated under these reduced supervision settings. This allows a direct comparison of how structural alignment influences label efficiency. Performance is measured using graph similarity metrics and detection accuracy on the held-out test set.

5.5.3 Multi-Target Training for Generalization

Beyond single-source pretraining and different structural-prior effects, the last experiment studies how combining multiple vascular datasets affects robustness to domain shifts. In this analysis, cumulative training sets that progressively include additional anatomical regions and imaging modalities are constructed. Each configuration increases the diversity of vascular structures seen during training while keeping the training procedure unchanged. This experiment evaluates whether exposure to heterogeneous data improves generalization to unseen datasets. It also tests whether diversity alone can provide benefits even when the total number of training samples remains fixed.

Fixed-Scale Multi-Domain Training In the fixed-scale regime, the total number of target patches is kept constant. The baseline configuration consists of syntheticMRI [25] samples only. As additional vascular datasets are introduced, up to N_{max} patches are drawn from each auxiliary domain. To maintain a fixed total budget, the number of syntheticMRI patches is reduced accordingly.

This design isolates the effect of domain diversity. Because the total number of training samples does not change, any performance difference can be attributed to the impact of exposing the model to heterogeneous vessel structures rather than increasing the dataset size.

Cumulative configurations (E1-E4) are defined to progressively increase anatomical and modality diversity, as summarized in table 5.4. Each configuration adds new datasets while preserving the same total number of training samples.

Scale-Expanded Multi-Domain Training In the scale-expanded regime, the syntheticMRI [25] subset remains unchanged when new datasets are introduced. In this case, each auxiliary domain also contributes up to N_{max} patches, and the total number of training samples increases as domains are added.

This regime measures the combined effect of domain diversity and increased data volume. It reflects a practical scenario where new annotated datasets become available without replacing existing data. As in the fixed-scale setting, configurations are cumulative with respect to the previous one. Each step increases the number of training domains and expands the overall dataset size.

Dataset Zero-Shot Generalization These three configurations (baseline, fixed-scale regime, and scale-expanded regime) are evaluated using dataset zero-shot transfer. In this setting, the model is tested on previously unseen vascular datasets without additional fine-tuning. Zero-shot evaluation measures how well relational representations generalize across different imaging modalities and tissues. It removes

Dataset	E1	E2	E3	E4
syntheticMRI	✓	✓	✓	✓
TopCoW-ct	✓	✓	✓	✓
TopCoW-mra	✓	✓	✓	✓
HiP-CT	✓	✓	✓	✓
MiniVess	✓	✓	✓	✓
VesselExpress-Bladder		✓	✓	✓
VesselExpress-Heart		✓	✓	✓
tubenet-MRI		✓	✓	✓
tubenet-2photon			✓	✓
DeepVess			✓	✓
3diracdb1			✓	✓
TubeTK			✓	✓
VesselExpress-Brain				✓
IXI (train-val)				✓
HRKidney				✓

Table 5.4: Datasets included in each cumulative multi-domain training configuration (E1-E4). Checkmarks indicate the datasets used in the corresponding experiment. Configurations are cumulative, with each subsequent setting adding new datasets to the previous ones.

the possibility of adapting to the target distribution and therefore provides a direct assessment of robustness.

By comparing models trained under different regimes, it is assessed whether the observed improvements arise primarily from domain diversity or from increased dataset size. This comparison allows us to determine which strategy provides stronger cross-domain dataset generalization.

5.6 Results

This section presents the results obtained from the experiments described in section 5.5. Three aspects of the training strategy are evaluated: the choice of structural source domain, the effect of structural pretraining on sample efficiency, and the impact of multi-domain training on generalization.

For each experiment, quantitative metrics are reported and the resulting training configurations are compared. These results illustrate how structural alignment and domain diversity influence learning behavior in image-to-graph transformers.

5.6.1 Structural Source-Domain Comparison

Structural Pretraining vs Baseline The first observation arises from the comparison of all pretrained models with the baseline configuration trained without structural pretraining. Results are reported for both the 2D and 3D target datasets.

On the 2D dataset, structural pretraining improves performance relative to the baseline across most evaluation metrics. The large road-pretrained configuration (99.2k source samples) improves all detection and topology metrics, while SMD remains unchanged. The size-matched road configuration (25.9k) shows mixed behavior, with small improvements on some metrics and slight decreases on others. In contrast, plant-based pretraining produces consistent gains across all reported metrics, including node detection, edge prediction, and topology measures.

A similar trend appears in the 3D experiment. All pretrained configurations outperform the baseline across the available metrics. Both road-based and plant-based pretraining improve node and edge prediction metrics and reduce SMD compared to the model trained without structural supervision.

Plants vs Roads Using table 5.5, plant-based and road-based structural pretraining can be compared under matched dataset size (25.9k source samples). Across both datasets, plant-based pretraining produces higher node detection, edge prediction, and topology metrics than road-based pretraining. This improvement appears consistently across the evaluated metrics, indicating that the benefit is not restricted to a single component of the graph prediction task.

On the 2D dataset, the differences between the two structural priors are substantial. Node mAP increases from 0.251 for road pretraining to 0.516 for plant pretraining, while node mAR increases from 0.352 to 0.581. The largest improvements appear in edge prediction metrics. Edge mAP increases from 0.166 to 0.378, corresponding to a relative improvement of more than 120%, while edge recall increases from 0.245 to 0.463. Improvements are also visible in topology metrics, where TOPO precision and recall increase from 0.754 and 0.682 to 0.828 and 0.749 respectively. These results show that plant-based pretraining improves both local detection accuracy and the global structural consistency of the predicted graphs.

A similar trend appears in the 3D experiments. When comparing size-matched configurations, plant-pretrained models achieve higher node and edge prediction metrics than road-pretrained models. For example, node mAP increases from 0.260 to 0.396 and edge mAP from 0.091 to 0.167 (gain of 84%). The magnitude of the improvement is smaller than in the 2D case, but the overall trend remains consistent.

Plant-based pretraining also approaches or surpasses the performance of the large road configuration (99.2k samples), despite using fewer source samples. In the 2D experiment, the plant-pretrained model clearly exceeds the large road

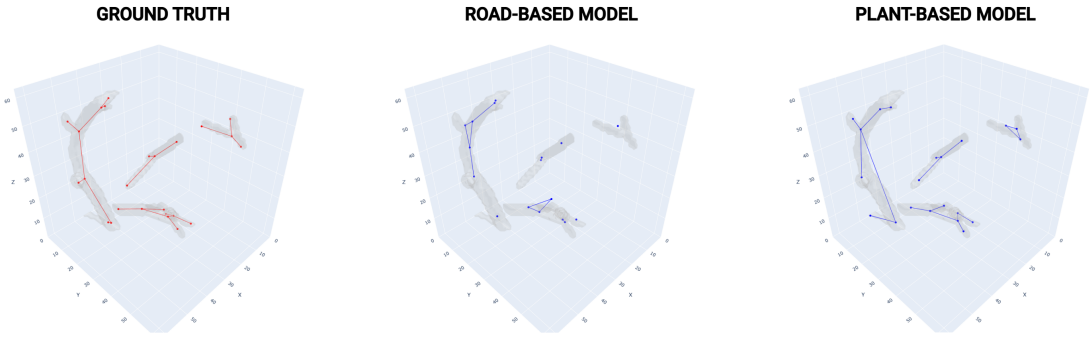

Figure 5.1: Qualitative comparison of graph predictions for a representative 3D patch from syntheticMRI dataset [25]. From left to right: ground-truth graph, prediction from the road-pretrained model, and prediction from the plant-pretrained model. The plant-based model better preserves the underlying branching topology.

configuration across all metrics. For example, node mAP increases from 0.307 to 0.516, while edge mAP increases from 0.205 to 0.378. Similar improvements appear in the remaining metrics.

In the 3D experiment, the performance of the plant-pretrained model remains comparable to the large road configuration. For instance, node mAP reaches 0.396 compared to 0.406 for the large road model, while edge mAP reaches 0.167 compared to 0.183. Although slightly lower, the results remain close despite the substantially smaller source dataset.

These observations indicate that structural alignment between the source and target domains can provide a stronger transfer signal than dataset scale alone.

Degree-2 Nodes Retention The final comparison evaluates the impact of retaining intermediate degree-2 nodes in the source graphs during preprocessing. In the standard graph extraction pipeline, degree-2 nodes are removed to produce a more compact representation of the vascular tree. In this experiment, these nodes are preserved in the structural source datasets to retain local geometric information along vessel segments. Table 5.5 reports the results obtained when degree-2 nodes are kept during structural pretraining. Introducing degree-2 nodes improves performance for both structural source domains, with the effect particularly visible in the 2D experiment.

When testing the OctaSynth dataset [49], retaining intermediate nodes leads to higher node detection, edge prediction, and topology metrics compared to configurations where degree-2 nodes are removed, producing predicted graphs that more closely match the target representation.

For road-based pretraining on the 2D dataset, retaining degree-2 nodes leads to clear improvements in node and edge prediction metrics, together with an

Source	Num. Source Samples	SMD↓	Node mAP↑	Node mAR↑	Edge mAP↑	Edge mAR↑	TOPO Prec.↑	TOPO Rec.↑
2D Target: OctaSynth								
No pretrain	-	0.005	0.248	0.353	0.152	0.256	0.719	0.720
roads	99.2k	0.005	0.307	0.398	0.205	0.304	0.773	0.752
roads	25.9k	0.006	0.251	0.352	0.166	0.245	0.754	0.682
plants	25.9k	0.003	0.516	0.581	0.378	0.463	0.828	0.749
roads+2-degree	25.9k	0.002	0.476	0.552	0.369	0.472	0.792	0.830
plants+2-degree	25.9k	**0.0006**	**0.629**	**0.692**	**0.553**	**0.630**	**0.843**	**0.884**
3D Target: syntheticMRI								
No pretrain	-	0.021	0.194	0.301	0.017	0.059	-	-
roads	99.2k	**0.014**	0.406	0.489	0.183	0.257	-	-
roads	25.9k	0.021	0.260	0.322	0.091	0.141	-	-
plants	25.9k	0.017	0.396	0.469	0.167	0.239	-	-
roads+2-degree	25.9k	0.015	0.449	0.520	0.180	0.253	-	-
plants+2-degree	25.9k	0.015	**0.462**	**0.530**	**0.202**	**0.282**	-	-

Table 5.5: Comparison of plant and road structural priors on 2D (OctaSynth [49]) and 3D (syntheticMRI [25]) targets. Rows labeled "+2-degree" retain degree-2 nodes in the source graphs during preprocessing. Gray rows denote models without pretraining.

evident reduction in SMD. Plant-based pretraining benefits even more from the degree-2 representation, showing further gains in node and edge metrics as well as improvements in topology measures and structural similarity.

A similar improvement appears in the 3D experiment on syntheticMRI [25]. Although the ground-truth graphs extracted with Voreen [9, 7] do not contain explicit degree-2 nodes, models pretrained with degree-2 nodes still achieve higher node and edge prediction metrics. For example, road-based pretraining improves node mAP from 0.260 to 0.449 and edge mAP from 0.091 to 0.180, while plant-based pretraining increases node mAP from 0.396 to 0.462 and edge mAP from 0.167 to 0.202.

Overall, preserving intermediate nodes during structural pretraining improves both detection and relational prediction metrics across the two structural source domains and the two target datasets.

5.6.2 Sample Efficiency Analysis

Figure 5.2 reports edge mAR on the IXI test set [58] for plant- and road-pretrained models at 30%, 50%, and 100% of the available training annotations.

At all annotation budgets, the plant-pretrained model attains higher edge mAR than the road-pretrained model. Performance increases with larger amounts of labeled data for both initializations. The difference between the two curves is larger at lower annotation budgets and decreases as the full training set is used, while remaining present at 100%.

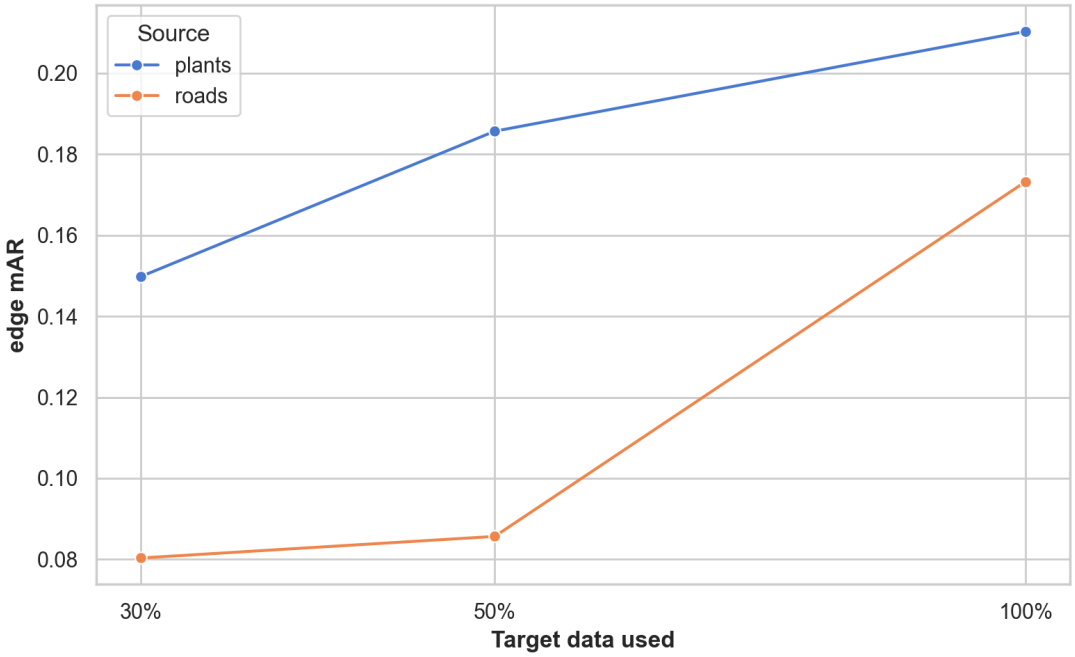

Figure 5.2: Edge mean Average Recall (mAR) on the IXI test set [58] for plant- and road-pretrained models evaluated using 30%, 50%, and 100% of the available training annotations.

5.6.3 Multi-Target Training for Generalization

Fixed-Scale Multi-Domain Training Table 5.6 reports zero-shot generalization results on the unseen SMILE-UHURA [50] and OCTA [51] datasets when progressively increasing the training domain diversity while keeping the total number of patches constant.

Across both datasets, the syntheticMRI baseline shows detection performance close to zero. On SMILE-UHURA, node mAP is 0.006 and edge mAP is 1×10^{-4}. On OCTA, node mAP is 0.004 and edge mAP is 1×10^{-4}, indicating that training on a single domain does not generalize to unseen vascular data.

Introducing additional domains in E1 leads to a marked increase in detection metrics. On SMILE-UHURA, node mAP rises from 0.006 to 0.065 and edge mAP

from 1×10^{-4} to 0.016. On OCTA, node mAP increases from 0.004 to 0.066 and edge mAP to 0.011. This initial step accounts for the largest absolute improvement across configurations.

Further increases in domain diversity continue to improve performance. On SMILE-UHURA, node mAP increases from 0.065 (E1) to 0.110 (E4), while edge mAP increases from 0.016 to 0.028. Edge recall rises from 0.024 to 0.061. Structural similarity also improves, with SMD decreasing from 0.062 in the baseline to 0.034 in E4.

A similar progression appears on OCTA. Node mAP increases from 0.004 in the baseline to 0.126 in E4, and node recall from 0.015 to 0.163. Edge mAP increases from near-zero values to 0.028. SMD decreases from 0.040 in the baseline to 0.030 in E4, with the lowest value observed in E2 (0.030).

Overall, increasing domain diversity under a fixed training budget leads to progressive improvements in node detection, edge prediction, and structural similarity across both unseen datasets. The largest relative gains are observed in the edge prediction metrics, where both edge mAP and edge mAR increase by a larger margin compared to the corresponding node metrics.

Test	**Target**	**SMD** ↓	**node mAP** ↑	**node mAR** ↑	**edge mAP** ↑	**edge mAR** ↑
SMILE-UHURA	syntheticMRI	0.062	0.006	0.024	1e-4	2e-4
	E1	0.063	0.065	0.091	0.016	0.024
	E2	0.047	0.089	0.131	0.023	0.042
	E3	0.041	0.098	0.142	0.023	0.046
	E4	**0.034**	**0.110**	**0.164**	**0.028**	**0.061**
OCTA	syntheticMRI	0.040	0.004	0.015	1e-4	2e-4
	E1	0.045	0.066	0.097	0.011	0.021
	E2	0.030	0.100	0.160	0.015	0.038
	E3	0.031	0.112	0.163	0.013	0.0325
	E4	**0.030**	**0.126**	**0.163**	**0.028**	**0.061**

Table 5.6: Zero-shot evaluation on the unseen SMILE-UHURA [50] and OCTA [51] datasets under fixed-scale multi-domain training. Results are reported for increasing training domain diversity (syntheticMRI [25], E1-E4, as reported in table 5.4) while keeping the total number of training patches constant. Performance improves consistently as additional domains are introduced, with E4 achieving the best results on both datasets, particularly in node and edge detection metrics.

Scale-Expanded Multi-Domain Training Table 5.7 reports zero-shot generalization results when additional domains are introduced without reducing the number of syntheticMRI patches [25], leading to a progressive increase in total training data.

Across both unseen datasets, introducing additional domains in E1 leads to a substantial increase in all detection metrics and a reduction in SMD. Node mAP increases from 0.007 to 0.071 on SMILE-UHURA [50] and from 0.004 to 0.068 on OCTA [51], while edge mAP rises from near-zero values to 0.019 and 0.016, respectively. At the same time, SMD decreases from 0.061 to 0.049 on SMILE-UHURA and remains stable at 0.037 on OCTA.

Target	**Test**	**SMD** ↓	**node mAP** ↑	**node mAR** ↑	**edge mAP** ↑	**edge mAR** ↑
SMILE-UHURA	syntheticMRI	0.062	0.006	0.024	1e-4	2e-4
	E1	0.049	0.071	0.105	0.019	0.037
	E2	0.039	0.088	0.132	0.024	0.050
	E3	0.040	0.090	0.139	0.026	0.050
	E4	**0.033**	**0.115**	**0.172**	**0.040**	**0.069**
OCTA	syntheticMRI	0.040	0.004	0.015	1e-4	2e-4
	E1	0.037	0.068	0.101	0.016	0.029
	E2	0.028	0.088	0.148	0.013	0.041
	E3	0.028	0.116	0.187	0.021	0.055
	E4	**0.024**	**0.127**	**0.200**	**0.026**	**0.067**

Table 5.7: Zero-shot evaluation on the unseen SMILE-UHURA [50] and OCTA [51] datasets under scale-expanded multi-domain training. Results are reported for progressively increasing training domain diversity (syntheticMRI [25], E1-E4, as reported in table 5.4) while retaining all syntheticMRI patches, leading to an increase in total training data. Performance improves consistently as additional domains are introduced, with E4 achieving the best results on both datasets.

Further configurations continue to improve performance. In E2, node mAP increases to 0.088 on SMILE-UHURA and 0.088 on OCTA, while edge mAP reaches 0.024 and 0.013, respectively. SMD decreases to 0.039 on SMILE-UHURA and 0.028 on OCTA. E3 maintains similar structural similarity while further improving detection metrics, particularly on OCTA, where node mAP increases to 0.116 and node recall to 0.187.

The best performance is obtained in E4 for both datasets. Node mAP reaches 0.115 on SMILE-UHURA and 0.127 on OCTA, while edge mAP increases to 0.040

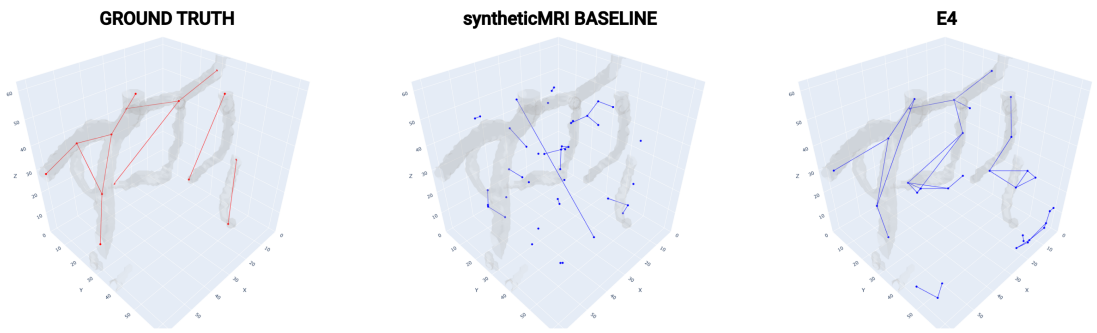

Figure 5.3: Qualitative comparison of graph predictions on a representative sample from an unseen dataset. From left to right: ground truth, syntheticMRI baseline, and the E4 configuration obtained under the scale-expanded multi-domain training regime. The baseline is close to random predictions, while E4 model more closely matches the ground-truth topology, with improved node detection and branch connectivity.

and 0.026, respectively. Edge recall reaches 0.069 on SMILE-UHURA and 0.067 on OCTA. SMD decreases to 0.033 and 0.024, representing the lowest structural distance values across configurations.

Overall, the scale-expanded regime produces progressive improvements in node and edge detection metrics together with reductions in structural distance, with E4 achieving the highest performance across both unseen datasets. Also in this scale-expanded regime, the largest gains were observed in the edge metrics.

Performance Trends Across Configurations Figure 5.4 visualizes the progressive performance changes across cumulative configurations (E1-E4) for both fixed-scale and scale-expanded regimes. Across both OCTA [51] and SMILE-UHURA [25], node mAP and edge mAP increase consistently with increasing domain diversity, while SMD exhibits a decreasing trend (since lower SMD values correspond to higher performance).

Under fixed-scale training, the largest performance increase occurs between the single-domain baseline and E1, followed by incremental improvements from E2 to E4. A similar overall progression is observed under the scale-expanded regime, with generally higher absolute performance values compared to fixed-scale training.

In both regimes, improvements are more pronounced for edge detection metrics than for node detection metrics. This pattern is consistent across both unseen datasets, and already noticeable from the tables 5.6 and 5.7.

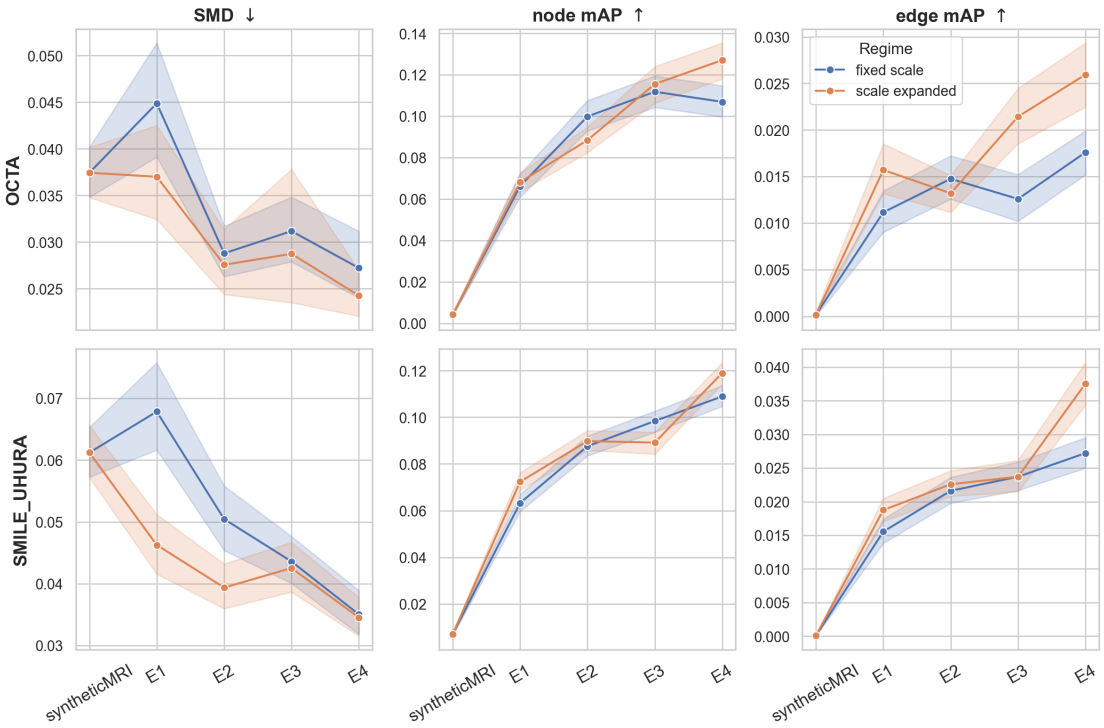

Figure 5.4: Performance across cumulative training configurations (syntheticMRI [25], E1-E4, as reported in table 5.4) for fixed-scale and scale-expanded multi-domain regimes. The top row reports results on OCTA [51] and the bottom row on SMILE-UHURA [50]. Columns show Structural Matching Distance (SMD), node mAP, and edge mAP. Shaded regions indicate variability across samples.

Statistical Validation To determine whether the observed differences between the single-domain baseline and the final multi-domain configuration (E4) reflect consistent performance improvements across test samples, paired t-tests were conducted for each metric and dataset. Figure 5.5 reports the results for the fixed-scale multi-domain training regime. An analogous analysis for the scale-expanded regime yielded consistent statistical significance across all metrics and is reported in appendix A.2.

Across both OCTA [51] and SMILE-UHURA [50], E4 significantly outperforms the baseline for all evaluated metrics ($p < 1e-4$). The reported p-values indicate that the probability of observing differences of this magnitude under the null hypothesis of no performance difference is orders of magnitude below the significance threshold of 0.05%.

For detection metrics (node mAP, node mAR, edge mAP, edge mAR), the mean differences are positive, indicating higher average performance under E4. The

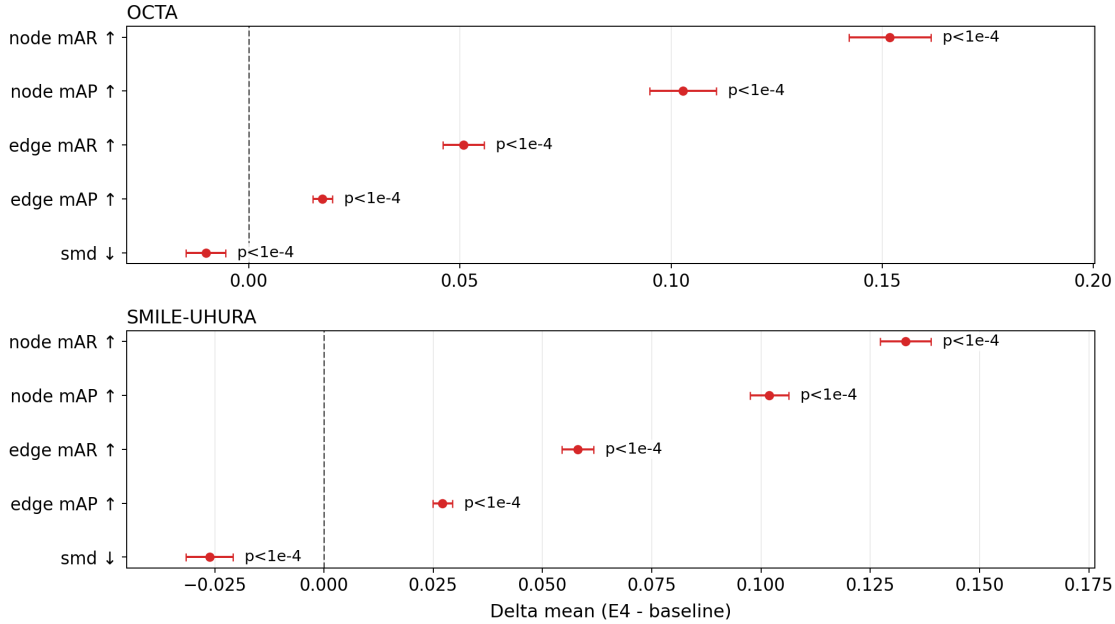

Figure 5.5: Paired t-test results comparing the single-domain baseline (syntheticMRI [25]) and the E4 configuration under the fixed-scale multi-domain training regime. Results are shown for OCTA [51] and SMILE-UHURA [50] across SMD, node mAP, node mAR, edge mAP, and edge mAR. The plotted values represent the mean paired difference, with error bars indicating the corresponding 95% confidence intervals.

corresponding 95% confidence intervals do not cross zero, which implies that the estimated improvement remains consistently above zero within the uncertainty bounds of the sample.

For SMD, the mean difference is negative, reflecting a reduction in structural distance under E4. The associated confidence intervals also remain strictly below zero, indicating a statistically reliable decrease in structural discrepancy.

In absolute terms, node-level metrics exhibit larger increases between baseline and E4. Edge-level metrics show substantial relative gains due to near-zero baseline performance, but the absolute improvements remain smaller than those observed for node metrics.

SMD exhibits consistent but comparatively smaller absolute reductions. Detailed test statistics and confidence intervals are reported in appendix A.2.

Distributional Analysis Figure 5.6 illustrates the distribution of performance metrics for the single-domain baseline and the E4 configuration on SMILE-UHURA [50] under the fixed-scale training regime. Comparable distributional patterns are

observed for OCTA [51] under the same regime, as well as for the scale-expanded setting (Appendix A.2).

Compared to the baseline, E4 exhibits a pronounced upward shift in edge and node detection metrics, with minimal overlap between distributions. The baseline values for edge metrics are concentrated near zero, while E4 yields higher values across test samples, in agreement with table 5.6. For SMD, the distribution under E4 shifts downward, indicating lower structural discrepancy. The reduction in overlap between the two configurations suggests that improvements are not driven by isolated outliers but occur systematically across samples.

The violin plots highlight differences in the spread of the metric distributions. For edge mAP and edge mAR, the baseline results are concentrated near very small values with little variability, while the E4 distributions cover a broader range of higher scores. A similar pattern appears for node metrics, where E4 shows higher medians and a wider spread across samples.

The density patterns also differ. Baseline values cluster near the lower bounds of the metrics, whereas E4 concentrates density at higher scores, with interquartile ranges shifted upward for detection metrics and downward for SMD.

Overall, the distributions show a clear separation between the two configurations across the evaluated metrics, with limited overlap in the central density regions.

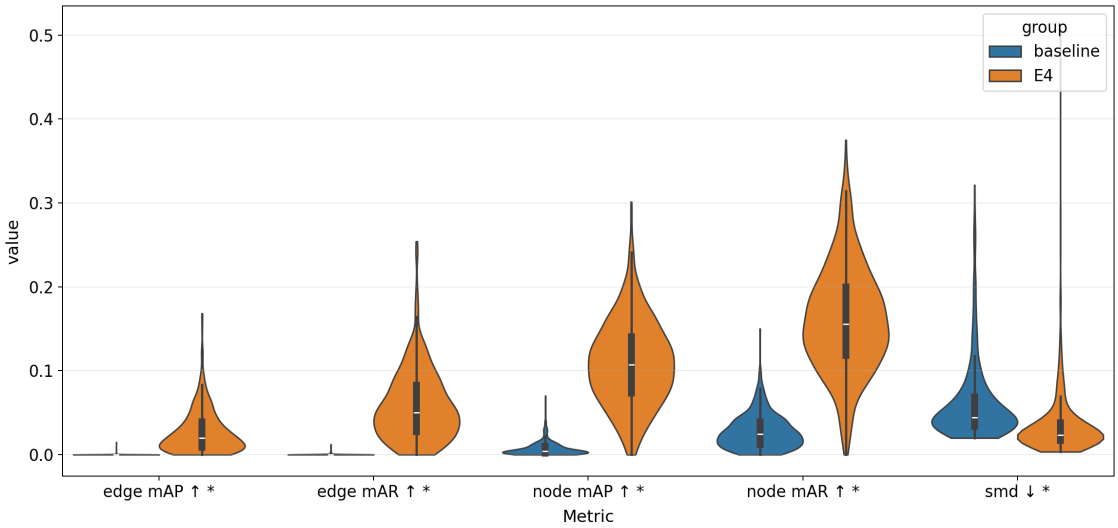

Figure 5.6: Violin plots showing the distribution of performance metrics on SMILE-UHURA [50] under the fixed-scale multi-domain training regime. Results are reported for the single-domain baseline (syntheticMRI [25]) and the E4 configuration across edge mAP, edge mAR, node mAP, node mAR, and SMD. Each violin represents the distribution across test samples.

Chapter 6

Discussion

The results indicate that structural congruence between source and target domains plays a central role in cross-domain image-to-graph pretraining. Previous approaches often focused on data scale as the main driver of transferability. The experiments presented here suggest that alignment between generative structural principles has a stronger impact. Botanical transport networks, shaped by biological growth constraints, encode branching patterns that resemble vascular organization more closely than infrastructure-based networks. This structural alignment is associated with improved graph reconstruction, reduced dependence on annotated vascular data, and stronger generalization to unseen datasets.

6.1 Summary of Contributions and Main Findings

This work examined whether structural alignment between source and target domains improves cross-domain pretraining for vascular graph extraction. The central hypothesis was that an auxiliary dataset with branching patterns closer to vascular anatomy would provide a more suitable inductive bias than infrastructure-based networks. The experiments confirm this hypothesis. Pretraining on a non-medical source domain improves performance, but the structural properties of that source domain strongly influence the final outcome.

By selecting a botanical dataset as the pretraining source, the model achieved higher graph prediction accuracy across both 2D and 3D vascular targets. The improvement appears consistently in quantitative metrics and in the qualitative structure of the predicted graphs. The gains are particularly clear in 3D settings, where relational reasoning and topological consistency are more demanding. These results indicate that the model benefits from structural priors that reflect biological branching processes rather than man-made layouts.

The study also shows that graph representation choices affect learning dynamics. Retaining degree-2 nodes in the source domain during preprocessing leads to further improvements, suggesting that preserving local geometric continuity supports more stable relational learning.

Beyond source-domain selection, the experiments highlight two additional findings. First, structurally aligned pretraining reduces the dependence on large annotated vascular datasets. Second, increasing diversity and scale across multiple vascular targets improves robustness to unseen domains. Together, these results suggest that both structural congruence and controlled multi-domain training play a central role in building a more generalizable vessel graph extractor.

6.2 Structural Congruence as an Inductive Bias

In medical imaging, large annotated datasets are rare, especially for structured tasks such as vessel graph extraction. Graph annotations require expert validation and careful preprocessing, which limits their availability. In this setting, cross-domain and cross-dimension transfer learning can play a crucial role. It allows the model to learn structural regularities from auxiliary domains and reuse them in the target task.

Pretraining Effectiveness This work combines vascular target datasets with auxiliary source datasets during a joint pretraining phase. The model is exposed to both domains before fine-tuning on vascular data alone. As shown in table 5.5, pretraining improves performance across all configurations considered. The gains appear with different source domains, different dataset sizes, and different graph representations. These results confirm that the model benefits from structural supervision beyond the target domain alone.

The findings support the claim by Berger et al. [2] that physical transport networks share transferable graph properties. Even when visual appearance differs, branching patterns and connectivity constraints exhibit common structure. The model can exploit these shared properties during training.

Cross-dimension transfer further strengthens this observation. By using 2D source data to support 3D vascular graph extraction, the model achieves clear improvements over training without auxiliary supervision. The gains are particularly evident in edge detection metrics, which directly reflect relational reasoning. This result is important in the medical context, where many imaging modalities produce volumetric data. The 2D-to-3D projection strategy requires only limited preprocessing, as described in sections 3.2.5 and 4.3.2. The procedure is simple and does not modify the model architecture. In light of the consistent performance gains, the added effort is minimal compared to the improvement obtained, making

the approach both practical and justified.

Plants vs Roads After establishing that pretraining is beneficial, the next question concerns the choice of the source domain. The central hypothesis of this work is that structural similarity between source and target domains influences transfer quality. Road networks, as used by Berger et al. [2], follow design principles shaped by traffic flow, urban planning, and economic constraints. Vascular systems, in contrast, emerge from biological growth processes governed by flow efficiency, transport cost, and spatial coverage. These differences suggest that the underlying topology of road networks may not fully reflect vascular organization.

To test this hypothesis, a botanical branching dataset was used as an alternative source domain. When comparing plants and roads under identical training configurations with 25.9k source samples, botanical pretraining consistently outperforms road-based pretraining in both 2D and 3D tasks. The improvements are visible across all metrics and are especially strong for edge detection. Since edge prediction captures connectivity and branching structure, these gains indicate that the relational prior learned from plants aligns more closely with vascular topology.

A further comparison considers botanical pretraining with 25.9k samples against road pretraining with 99.2k samples, matching the source dataset size originally used by Berger et al. [2]. This allows a direct comparison with their configuration under equivalent conditions. In 2D settings, the botanical source achieves higher performance despite using fewer samples. In 3D settings, performance matches or approaches the larger road configuration. These results show that structural alignment can compensate for reduced data scale. When the source domain reflects similar generative constraints, the model requires fewer examples to learn effective relational patterns. This has practical implications for resource usage and training time.

Degree-2 Retention Improves Relational Modeling An additional factor influencing performance is the representation of graph structure during preprocessing. In table 5.5, retaining degree-2 nodes in the source graphs leads to further improvements. Degree-2 nodes lie along branches and preserve local geometric continuity. Removing them simplifies the graph into segments between junctions and endpoints, which discards curvature information.

By retaining these nodes, the model receives supervision not only on global connectivity but also on local branch structure. This richer representation appears to support more stable relational learning. The improvement holds even when the target vascular annotations do not include degree-2 nodes. This suggests that exposure to finer-grained structural detail during pretraining helps the model form a more consistent internal representation of connectivity.

The benefit of degree-2 retention appears for both road and plant sources. However, botanical pretraining with degree-2 retention remains superior across all metrics and across both 2D and 3D tasks. This reinforces the broader conclusion that structural congruence between source and target domains plays a decisive role in shaping the learned inductive bias.

6.3 Sample Efficiency and the Annotation Tax

Figure 5.2 reports performance on the held-out IXI [58] test set under varying annotation budgets. Plant-pretrained models consistently outperform road-pretrained models across all levels of target supervision. The performance gap is most pronounced under limited supervision. With only 30% of the IXI [58] training data, the plant-pretrained model approaches the performance achieved by the road-pretrained model trained on the full dataset. This indicates that the structural prior learned from botanical networks compensates for missing vascular annotations.

These results extend the observations discussed in section 6.2. A biologically aligned source domain reduces the amount of source data required to achieve strong performance. Here, the experiments show that it also reduces the amount of target data needed. In other words, structural congruence improves both transfer efficiency and label efficiency.

The behavior across annotation levels provides further insight. As the amount of labeled vascular data increases from 30% to 100%, performance improves for both pretraining strategies. However, the relative advantage of botanical pretraining remains stable. The gap narrows slightly at higher annotation budgets, but it does not disappear. This pattern suggests that when supervision is scarce, the inductive bias dominates learning. As more annotated vascular data becomes available, the model relies more on direct task supervision, yet the structural prior continues to shape relational reasoning.

From a practical perspective, this finding directly addresses the annotation tax in vascular graph extraction. Graph supervision is costly and time-consuming to obtain. If structurally aligned pretraining allows competitive performance with a fraction of the annotated volumes, it reduces the burden on manual annotation and lowers the barrier to deploying graph-based vascular analysis. These results therefore show that structural congruence is not only a theoretical advantage, but also a concrete tool for improving sample efficiency in medical graph learning.

6.4 Multi-Target Scaling and Zero-Shot Generalization

When using the framework of Berger et al. [2] in a single-target setting, each new dataset requires retraining the model from scratch, including both pretraining and fine-tuning phases. This limits practical deployment, since vascular datasets often differ in acquisition protocol, anatomical region, and resolution. Even though botanical pretraining reduces the amount of required data and training time, the model remains tied to a specific target distribution.

To address this limitation, this work explores a multi-target training strategy with the goal of moving toward a foundation-like vessel graph extractor. Instead of optimizing the model for a single vascular dataset, the training pool progressively incorporates heterogeneous targets. The results show that increasing diversity in terms of tissues, organisms, imaging modalities, and vessel densities improves generalization to unseen domains.

Diversity as a Driver of Structural Invariance In the fixed-scale regime, the total number of training samples is kept constant at 4k while progressively increasing the number of target domains (E1 $\rightarrow$ E4). This design isolates the effect of structural diversity from data volume. As shown in Table 5.6, increasing diversity alone leads to consistent improvements on both SMILE-UHURA [50] and OCTA [51]. The steady reduction in SMD and the increase in node and edge metrics indicate that exposure to heterogeneous vascular structures encourages the model to learn representations that are less tied to a single dataset distribution.

When the model is trained only on syntheticMRI [25], zero-shot performance is poor on both unseen datasets. Edge detection metrics are close to zero, indicating that the model fails to recover meaningful connectivity. Node detection metrics are slightly higher, but this is largely due to unstable predictions with many false positives, as observed qualitatively in fig. 5.3. The model learns dataset-specific statistics and struggles to transfer to different structural distributions.

As additional datasets are introduced (E1 to E4), performance improves steadily across all metrics. This trend holds for both SMILE-UHURA [50] and OCTA [51]. In both cases, the Street Mover Distance decreases and node and edge detection metrics increase in a consistent manner. The improvement in edge metrics is particularly relevant, since correct edge prediction reflects successful relational reasoning and preservation of connectivity.

These results suggest that structural diversity forces the model to learn invariant graph properties rather than dataset-specific patterns. Exposure to variations in branching density, curvature, topology, and imaging characteristics reduces overfitting to a single structural distribution. Even though the total number of

training samples remains constant, the broader range of structural configurations improves robustness.

Graph extraction is particularly sensitive to dataset-specific biases. The node prediction head learns where to place junctions and endpoints, while the relation head learns which node pairs should be connected. Both components can overfit when the training distribution is narrow or dominated by a single domain. Multi-target diversity mitigates this effect because different datasets challenge different aspects of the pipeline. Some datasets contain dense capillary beds with many short edges, whereas others exhibit sparse tree-like structures with long segments and fewer junctions. Certain imaging modalities present vessels with sharp boundaries, while others show blurred structures due to partial volume effects. When exposed to this variation during training, the model must learn features that remain useful across heterogeneous conditions. This encourages reliance on relational cues and global consistency rather than local appearance statistics.

The capped sampling policy further strengthens this effect. By limiting the contribution of each domain, it ensures that structurally distinct datasets remain visible during training. Without such balancing, dominant domains would shape the learned representation disproportionately, reducing the benefit of structural diversity.

An interesting observation is that improvements appear more stable for node detection on OCTA [51], while SMILE-UHURA [50] shows a stronger relative gain in edge recall as diversity increases. This indicates that different unseen domains may benefit in different ways from structural diversity, but the overall trend remains consistent: diversity alone, even without increasing dataset size, improves zero-shot generalization.

These findings support the idea that diversity, rather than scale alone, is a primary driver of cross-dataset robustness in graph prediction tasks.

Scale as a Stabilizer of Multi-Domain Learning In the scale-expanded regime, additional target domains are introduced without reducing the number of samples per dataset. Diversity and data volume therefore increase together. Results in Table 5.7 show further improvements on both unseen datasets compared to the fixed-scale setting. Performance gains become more stable and more pronounced, especially for edge metrics. This suggests that diversity shapes invariant structural representations, while increased scale consolidates them and reduces variance in relational predictions.

Compared to the fixed-scale setting, performance improves further across both unseen datasets. For SMILE-UHURA [50], the progression from syntheticMRI [25] to E4 shows a steady reduction in SMD and consistent gains in node and edge metrics. Edge mAP and edge mAR increase markedly, indicating improved recovery of connectivity. For OCTA [51], the same trend appears. SMD decreases

consistently, while both node and edge detection metrics improve at each stage, with the strongest results observed at E4.

Unlike the fixed-scale regime, where improvements reflect structural diversity alone, the scale-expanded setup combines diversity with increased data volume. The results indicate that diversity remains the primary driver of generalization, while additional scale stabilizes the learning process. Performance trends become more consistent across configurations, with fewer fluctuations between intermediate stages. This effect is particularly visible in the edge metrics, which reflect connectivity reconstruction under distribution shift. These patterns are also evident in fig. 5.4.

An additional observation concerns stability. In the scale-expanded regime, performance gains are more monotonic than in the fixed-scale case. For example, small fluctuations observed between E2 and E3 under fixed scale are reduced when additional samples are available. This indicates that increased data volume stabilizes relational learning while diversity shapes invariance.

These findings confirm that structural diversity is essential for zero-shot robustness, and that additional scale strengthens this effect. Together, they support the view that a foundation-like vessel graph extractor requires exposure to heterogeneous vascular distributions, with sufficient data to consolidate invariant relational patterns.

Statistical Validation of Performance Differences To assess whether the observed improvements reflect systematic effects rather than random variation, Welch's t-tests were performed comparing the single-domain baseline and the E4 configuration. Across both OCTA [51] and SMILE-UHURA [50], all evaluated metrics show statistically significant differences at $\alpha = 0.05$. The p-values are several orders of magnitude below conventional thresholds ($p < 10^{-10}$ for OCTA and $p < 10^{-23}$ for SMILE-UHURA), indicating that the probability of observing such mean differences under equal-performance assumptions is negligible.

Given the sample sizes ($n = 100$ for OCTA and $n = 499$ for SMILE-UHURA), the tests provide stable estimates of the mean differences. The improvements are not confined to a single metric but appear consistently across node detection, edge detection, and structural similarity (SMD). This consistency across independent performance indicators strengthens the conclusion that the effect is robust. In particular, the large differences in node and edge recall suggest that the multi-domain configuration improves the model's ability to recover complete and connected graph structures, rather than merely increasing the number of detected elements.

Taken together, the statistical analysis supports a clear conclusion: the transition from single-domain training to the E4 multi-domain configuration leads to reliable and repeatable performance gains. The improvements are unlikely to be artifacts of sampling variability and instead reflect a structural change in model behavior under increased diversity and scale.

Systematic Distributional Shifts Across Test Samples Beyond mean comparisons, the distributional analysis provides further insight into how performance changes across individual test samples. As shown in the violin plots in fig. 5.6, the E4 configuration shifts the entire distribution of performance metrics relative to the baseline. For edge metrics in particular, the baseline model produces values concentrated near zero, indicating frequent failure to reconstruct connectivity. This pattern suggests that the single-domain model often fails at a structural level when evaluated under domain shift.

In contrast, the E4 distributions are consistently shifted upward for node and edge metrics and downward for SMD. The improvement is not limited to a small number of high-performing cases. Instead, the bulk of the distribution moves toward better performance, and the lower tail improves as well. This indicates that multi-domain training reduces catastrophic failures and stabilizes predictions across heterogeneous samples.

The reduction in SMD variance further suggests improved structural consistency. Lower and more concentrated SMD values imply that predicted graphs more closely match the global topology of the ground truth across most test volumes. The model does not simply perform better on average; it behaves more reliably across cases with varying vessel density and topology.

These distributional patterns support the statistical findings. They show that the improvement is systematic, affects the majority of samples, and reduces instability under distribution shift. This strengthens the conclusion that multi-target training improves both accuracy and robustness in zero-shot vascular graph extraction.

6.5 Limitations

The proposed framework presents several limitations that affect different components of the pipeline, including optimization, representation, supervision design, and conceptual scope.

A first limitation concerns the adversarial domain adaptation module. Berger et al. [2] note that the weighting parameter α in the gradient reversal layer requires careful tuning for each target dataset. In a multi-target setting with heterogeneous vascular domains, identifying a single optimal α becomes challenging. In this work, the value of α was kept fixed within each experimental setting to ensure consistent comparisons between configurations. However, only a limited set of α values was explored due to computational constraints, and the selected values may therefore not correspond to the optimal balance between task supervision and domain alignment for every dataset. As the results indicate that performance can be sensitive to the choice of α, some domains may operate under suboptimal hyperparameter settings. This sensitivity to hyperparameter selection may limit

scalability when extending the framework to larger and more heterogeneous training pools.

A second limitation relates to edge supervision. As the number of nodes increases, the number of possible negative edges grows quadratically. In the current configuration, all candidate edges are used for supervision in order to maximize relational signal. While this provides strong supervision, it introduces a large imbalance between positive and negative samples and increases computational cost. Alternative strategies such as hard negative sampling could reduce complexity and focus learning on informative relations. However, preliminary observations indicate that reducing the number of supervised edges degrades performance under the current loss formulation. This suggests that the model relies heavily on dense relational supervision, particularly in datasets with high node density. A more principled strategy for edge selection that preserves performance while improving efficiency remains an open problem.

A related limitation concerns computational scalability. Relation-based prediction scales poorly with increasing node count, since pairwise relations grow quadratically. In dense vascular volumes, this may limit inference speed and memory efficiency. Extending the framework toward whole-organ or higher-resolution volumes may therefore require architectural modifications or sparse relation modeling.

Another limitation arises from the graph construction pipeline. Ground-truth graphs are extracted using the Voreen tool, which removes degree-2 nodes during skeleton processing. This simplifies graph structure by collapsing continuous vessel segments into edges between junctions and endpoints. While this representation reduces graph complexity, it discards curvature information and local geometric detail. In datasets with high vessel tortuosity, this preprocessing step may remove relevant structural characteristics from the supervision signal. More generally, ground-truth graphs depend on the chosen extraction pipeline, meaning that the supervision signal reflects design choices rather than an intrinsic anatomical representation.

An additional limitation concerns the relationship between segmentation and graph supervision. The proposed model is segmentation-free in the sense that it does not require vessel segmentations as input, neither during training nor at inference time. This design reduces dependency on intermediate pixel-level predictions and avoids error propagation from segmentation to graph extraction. However, the construction of ground-truth graphs relies on segmentation masks processed through Voreen or similar extraction tools. In practice, this means that segmentation annotations remain necessary to generate training supervision. Without segmentation masks, ground-truth graphs cannot be derived using the current pipeline. As a result, the framework does not eliminate the need for segmentation data in the annotation process; it only removes segmentation as an

explicit modeling step. This dependency limits applicability in scenarios where segmentation annotations are unavailable.

The 2D-to-3D transfer strategy introduces further assumptions. Although the projection method is simple and effective, the rendered pseudo-volumetric representations do not contain true 3D anatomical continuity. The model may therefore learn priors from projected structures that approximate, but do not fully reproduce, volumetric branching behavior. This limits the strength of the cross-dimensional inductive bias.

From an evaluation perspective, the chosen metrics focus on detection accuracy and structural similarity. While node and edge mAP/mAR and SMD capture important aspects of graph quality, they do not directly assess physiological plausibility or downstream clinical utility. Improvements in these metrics do not necessarily translate into improved performance in tasks such as flow simulation, disease classification, or surgical planning. The absence of task-specific validation limits the clinical interpretation of the results.

Taken together, these limitations indicate that while structural congruence and multi-domain training improve transfer learning for vascular graph extraction, the framework remains sensitive to preprocessing choices, supervision density, and optimization parameters. Performance depends on graph construction pipelines, dense relational supervision, and careful balancing of adversarial alignment. Moreover, computational scalability and reliance on segmentation-derived ground-truth graphs constrain broader applicability. Further work is required to improve efficiency, reduce dependency on preprocessing artifacts, and evaluate clinical utility in downstream tasks.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

This thesis investigated transfer learning strategies for vascular graph extraction under limited annotation availability. The central hypothesis was that structural alignment between source and target domains is more important than dataset scale alone when learning relational representations.

The results support this hypothesis. Structural congruence between source and target domains plays a central role in cross-domain pretraining for vascular graph extraction. Pretraining on botanically aligned branching networks consistently yields stronger and more sample-efficient performance than infrastructure-based supervision, even when using fewer source samples. These findings show that biologically grounded priors transfer more effectively than priors derived from man-made networks.

In addition, the thesis demonstrated that graph representation choices influence relational learning. Retaining degree-2 nodes and preserving local curvature improves supervision consistency and enhances connectivity prediction. A unified and topology-consistent graph construction pipeline proved essential for stable multi-target training. Without consistent preprocessing, structural inconsistencies across datasets degrade generalization.

The multi-target training strategy further improved robustness. Exposing the model to heterogeneous vascular datasets encouraged it to rely on invariant structural patterns rather than dataset-specific artifacts. This diversity-driven training improved zero-shot generalization to previously unseen domains, particularly when combined with structurally aligned pretraining.

Overall, the results highlight a broader principle: for structured prediction tasks,

aligning inductive biases with the generative structure of the target domain is more effective than increasing pretraining scale without structural consideration. Biological pretraining, topology-aware supervision, and multi-target diversity provide a principled path toward data-efficient and generalizable vascular graph extraction.

More importantly, these findings move vascular graph extraction toward foundation-like models that do not require retraining from scratch for each new test dataset. By learning transferable structural priors across heterogeneous domains, such models can adapt through limited fine-tuning and generalize more reliably to unseen data. This shift is essential for scalable and clinically viable graph-based vascular analysis.

7.2 Future Work

While the proposed framework improves structural transfer and generalization, several technical challenges remain open.

Topology-Preserving Graph Representations This work showed that retaining degree-2 nodes and preserving local curvature improves relational supervision and structural consistency. However, current graph representations remain a discretized approximation of continuous vascular geometry. Future research could investigate richer topology-preserving representations that encode curvature, local orientation, or geometric continuity more explicitly. Such representations may provide stronger supervision signals and improve the model's ability to reconstruct fine-scale vascular structure without increasing annotation requirements.

Scalability with Respect to Graph Size The number of nodes and edges remains a limiting factor for transformer-based image-to-graph models. Relation prediction scales quadratically with the number of nodes, and the transformer decoder operates with a fixed number of object and relation tokens. As graph density increases, training becomes slower and prediction more complex. This constraint limits the applicability of the framework to very large vascular networks or high-resolution volumetric data.

Future work should explore strategies to address this scalability issue. Possible directions include hierarchical graph prediction, adaptive token allocation, node clustering schemes, or sparse attention mechanisms tailored to relational reasoning. Reducing the effective complexity of relation modeling would enable the extraction of larger vascular graphs without sacrificing structural fidelity.

Hard-Negative Edge Sampling Although regularized edge sampling mitigates class imbalance, many negative edges remain trivial and are repeatedly sampled

during training. These easy negatives contribute little to learning and may dilute the supervision signal. Future work could investigate hard-negative mining strategies that prioritize ambiguous or structurally plausible non-edges. Skewing the sampling distribution toward more informative negative examples may improve relational discrimination and reduce convergence time.

Declaration on the Use of Artificial Intelligence Tools

Artificial intelligence tools were used during the preparation of this thesis to assist with grammar refinement, and language editing of the written text. AI-based image generation tools were also used to generate graphical elements and to provide inspiration for illustrative figures, which were subsequently adapted and finalized by the author.

Appendix A

Additional Experimental Results

This appendix reports additional qualitative and statistical results that complement the experiments presented in Chapter 5. These include visual comparisons between source-domain pretraining strategies, statistical significance tests for the multi-domain experiments, and metric distributions for the scale-expanded training regimes.

A.1 Qualitative Comparison of Pretraining Domains

Figure A.1 shows additional qualitative comparisons between models pretrained on road networks and those pretrained on botanical branching structures. Across multiple samples, plant-based pretraining tends to produce predictions that better preserve branching topology and connectivity.

A.2 Statistical Significance Analysis

Tables A.1 and A.2 report the results of Welch's t-tests comparing the single-domain baseline and the E4 configuration for the OCTA and SMILE-UHURA datasets. The tests evaluate whether the observed performance improvements are statistically significant.

Metric	n_A	n_B	Mean$_A$	Mean$_B$	Δ(B-A)	t	p	Sig
edge mAP ↑	100	100	0.00015	0.02595	0.02579	-14.91	3.30e-27	yes
edge mAR ↑	100	100	0.00015	0.06679	0.06665	-22.72	3.85e-41	yes
node mAP ↑	100	100	0.00414	0.12711	0.12297	-27.47	1.58e-48	yes
node mAR ↑	100	100	0.01753	0.20032	0.18279	-36.23	1.83e-63	yes
SMD ↓	100	100	0.03743	0.02426	-0.01317	6.77	1.43e-10	yes

Table A.1: Welch t-test results comparing the baseline and E4 configurations on the OCTA [51] dataset ($\alpha = 0.05$).

Metric	n_A	n_B	Mean$_A$	Mean$_B$	Δ(B-A)	t	p	Sig
edge mAP ↑	499	499	8.47e-05	0.03755	0.03746	-23.48	1.15e-82	yes
edge mAR ↑	499	499	0.00023	0.06768	0.06745	-29.83	5.84e-113	yes
node mAP ↑	499	499	0.00703	0.11892	0.11190	-47.43	2.00e-191	yes
node mAR ↑	499	499	0.02747	0.17445	0.14699	-46.33	2.55e-198	yes
SMD ↓	499	499	0.06121	0.03452	-0.02669	10.21	2.86e-23	yes

Table A.2: Welch t-test results comparing the baseline and E4 configurations on the SMILE-UHURA [50] dataset ($\alpha = 0.05$).

A.3 Metric Distributions for Scale-Expanded Training

Figures A.3, and A.2 present violin plots showing the distribution of evaluation metrics across samples for different training regimes. These plots complement the aggregate statistics reported in Chapter 5 by illustrating variability across the test sets.

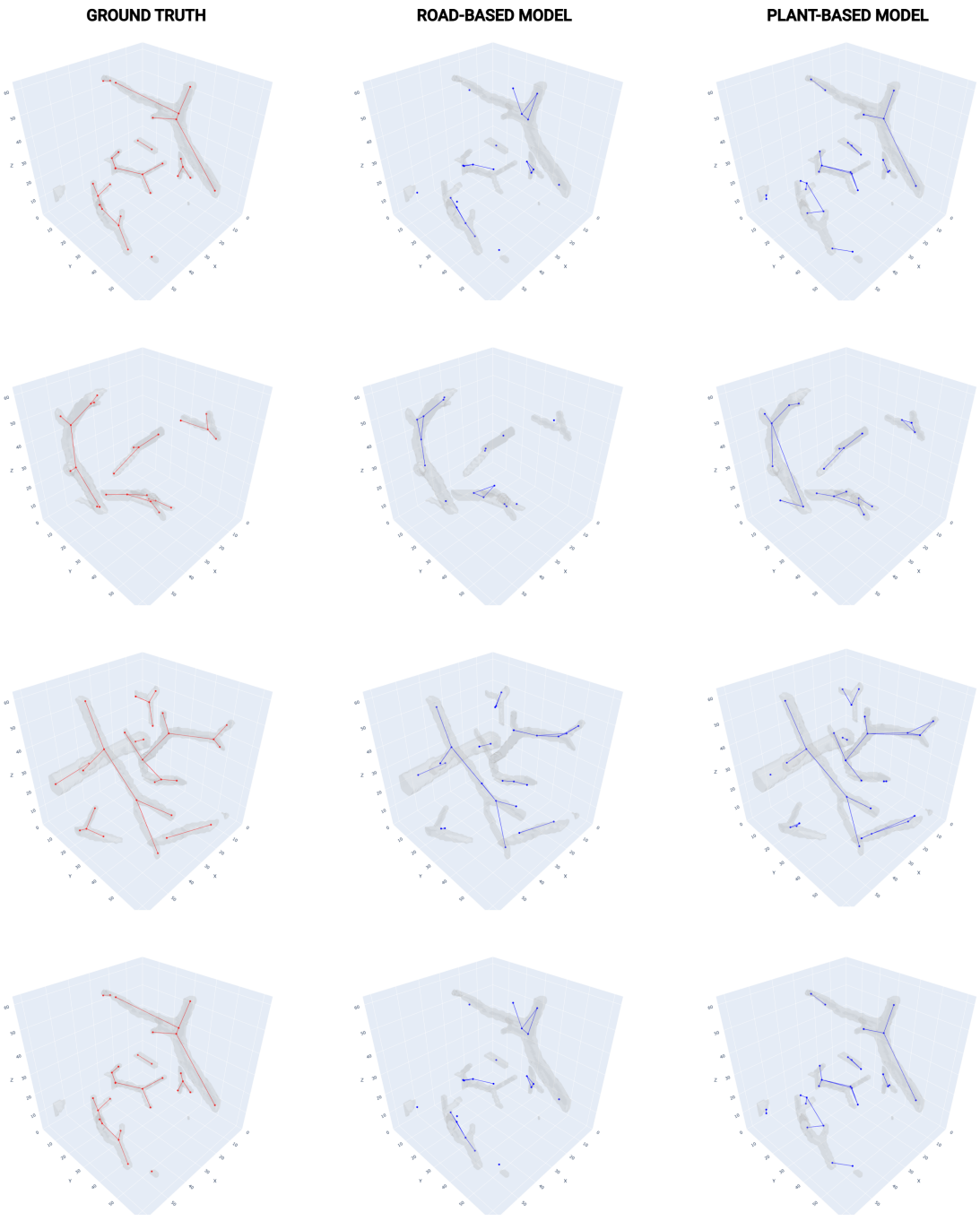

Figure A.1: Qualitative comparison of predictions from the road-pretrained and plant-pretrained models across multiple samples. The plant-based model generally produces graph structures that more closely follow the underlying branching topology.

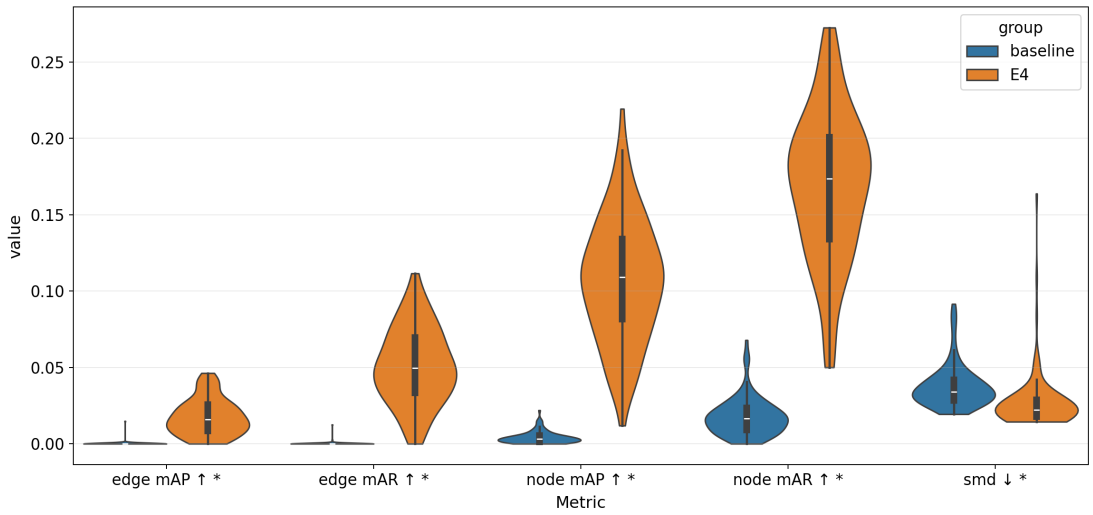

Figure A.2: Distribution of performance metrics on the OCTA [51] dataset under the diversity-expanded training regime.

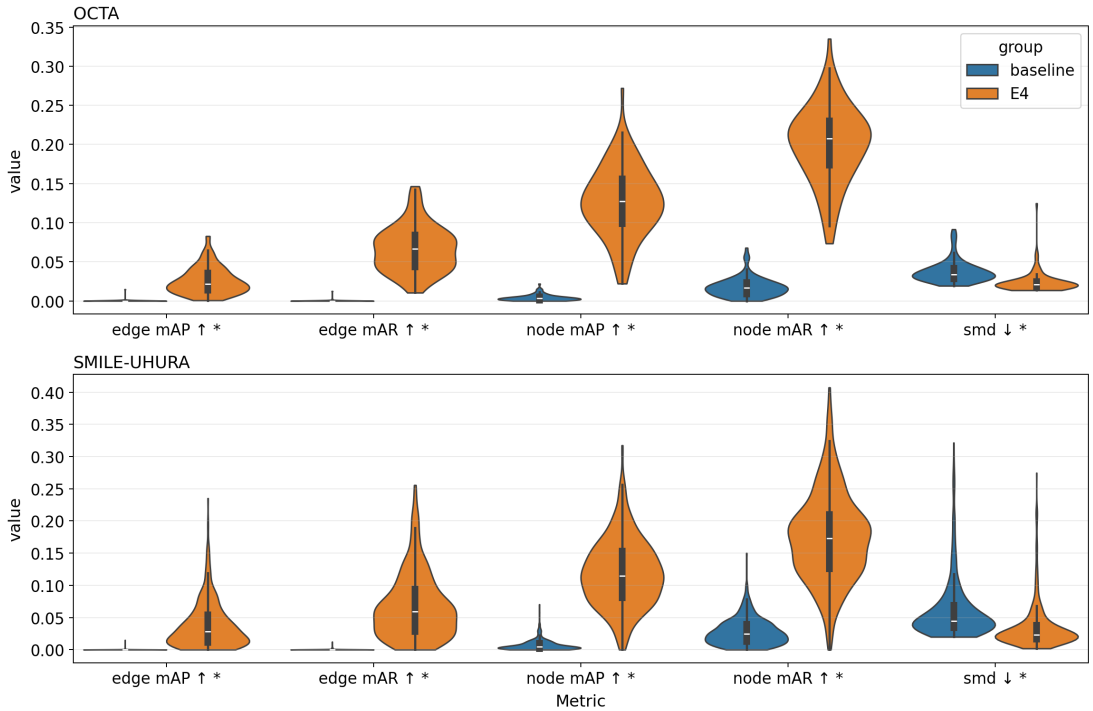

Figure A.3: Distribution of performance metrics on the OCTA [51] and SMILE-UHURA [50] dataset under the scale-expanded training regime.

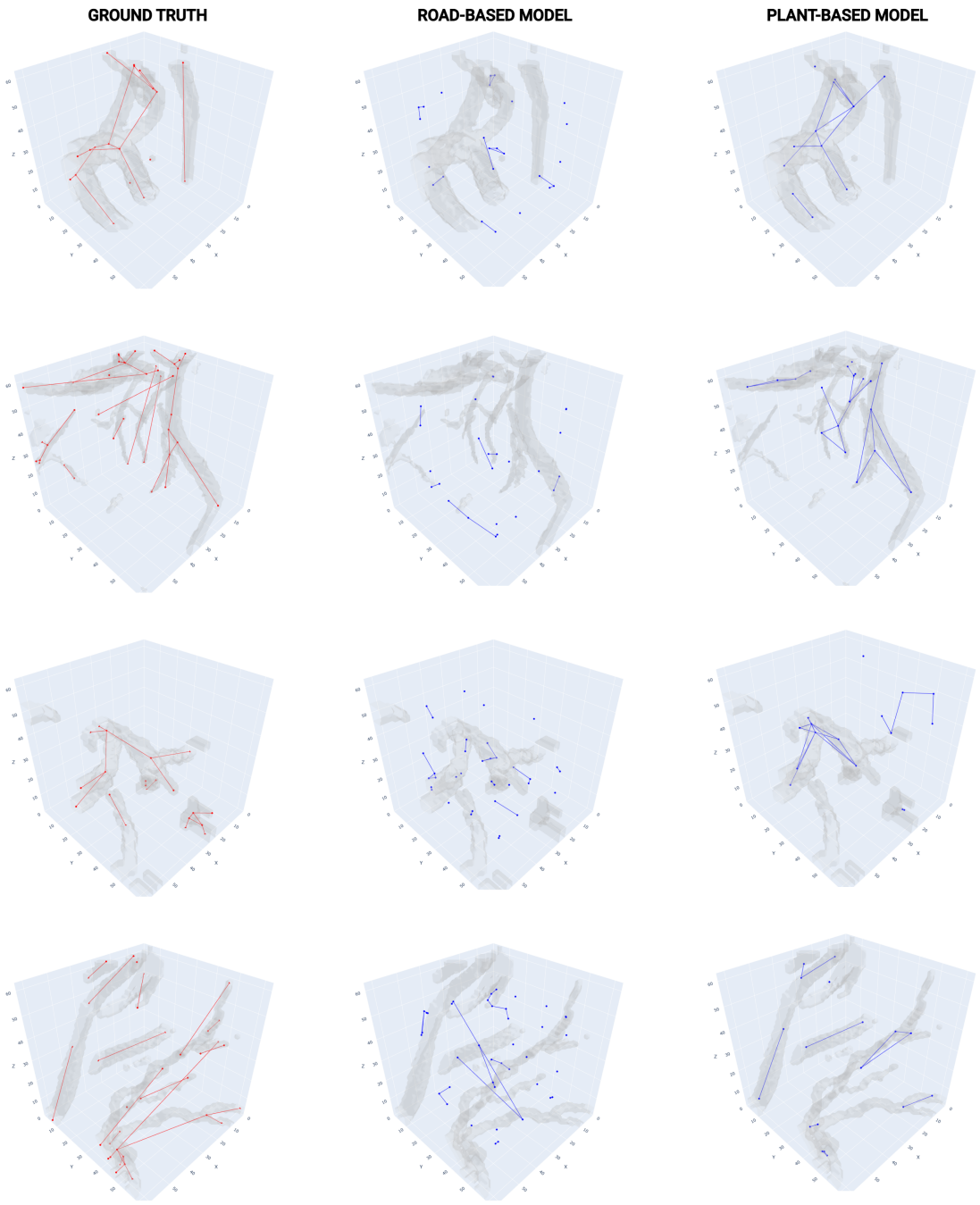

Figure A.4: Qualitative comparison between the syntheticMRI baseline and the E4 scale-expanded multi-domain configuration on SMILE-UHURA [50]. From left to right: ground truth, baseline prediction, and E4 prediction.

Bibliography

[1] Miguel O Bernabeu et al. «Computer simulations reveal complex distribution of haemodynamic forces in a mouse retina model of angiogenesis». In: *Journal of The Royal Society Interface* 11.99 (2014) (cit. on p. 1).

[2] Alexander H Berger, Laurin Lux, Suprosanna Shit, Ivan Ezhov, Georgios Kaissis, Martin J Menten, Daniel Rueckert, and Johannes C Paetzold. «Cross-Domain and Cross-Dimension Learning for Image-to-Graph Transformers». In: *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2025, pp. 64–74 (cit. on pp. 1, 5–7, 9–15, 21, 22, 24–32, 36, 47, 54, 56, 60, 62, 65, 79, 80, 82, 85).

[3] Shaoting Zhang and Dimitris Metaxas. «On the challenges and perspectives of foundation models for medical image analysis». In: *Medical image analysis* 91 (2024), p. 102996 (cit. on p. 1).

[4] Keyon Vafa, Peter G Chang, Ashesh Rambachan, and Sendhil Mullainathan. «What has a foundation model found? using inductive bias to probe for world models». In: *arXiv preprint arXiv:2507.06952* (2025) (cit. on p. 2).

[5] Cecil D Murray. «The physiological principle of minimum work: I. The vascular system and the cost of blood volume». In: *Proceedings of the National Academy of Sciences* 12.3 (1926), pp. 207–214 (cit. on pp. 2, 12, 34).

[6] Geoffrey B West, James H Brown, and Brian J Enquist. «A general model for the origin of allometric scaling laws in biology». In: *Science* 276.5309 (1997), pp. 122–126 (cit. on pp. 2, 12, 34).

[7] Dominik Drees, Aaron Scherzinger, René Hägerling, Friedemann Kiefer, and Xiaoyi Jiang. «Scalable robust graph and feature extraction for arbitrary vessel networks in large volumetric datasets». In: *BMC bioinformatics* 22.1 (2021), p. 346 (cit. on pp. 3, 42, 44, 45, 47, 59, 70).

[8] Weixing Wang, Nan Yang, Yi Zhang, Fengping Wang, Ting Cao, and Patrik Eklund. «A Review of Road Extraction from Remote Sensing Images». In: *Journal of Traffic and Transportation Engineering (English Edition)* 3 (May 2016). DOI: 10.1016/j.jtte.2016.05.005 (cit. on p. 3).

[9] Jennis Meyer-Spradow, Timo Ropinski, Jörg Mensmann, and Klaus Hinrichs. «Voreen: A rapid-prototyping environment for ray-casting-based volume visualizations». In: *IEEE Computer Graphics and Applications* 29.6 (2009), pp. 6–13 (cit. on pp. 3, 13, 42, 44, 47, 70).

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. «Deep residual learning for image recognition». In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778 (cit. on pp. 4, 6, 17, 19, 59).

[11] James Batten, Matthew Sinclair, Ben Glocker, and Michiel Schaap. *Image To Tree with Recursive Prompting*. 2023. arXiv: 2301.00447 [cs.CV]. URL: https://arxiv.org/abs/2301.00447 (cit. on p. 4).

[12] Sepideh Almasi, Xiaoyin Xu, Ayal Ben-Zvi, Baptiste Lacoste, Chenghua Gu, and Eric L. Miller. «A Novel Method for Identifying a Graph-Based Representation of 3-D Microvascular Networks from Fluorescence Microscopy Image Stacks». In: *Medical Image Analysis* 20.1 (2015), pp. 208–223. ISSN: 1361-8415. DOI: 10.1016/j.media.2014.11.007 (cit. on pp. 4, 13).

[13] Suprosanna Shit et al. *Relationformer: A Unified Framework for Image-to-Graph Generation*. 2022. arXiv: 2203.10202 [cs.CV]. URL: https://arxiv.org/abs/2203.10202 (cit. on pp. 5, 10, 12–15, 17, 21, 23, 24, 28, 29, 35, 59).

[14] Chinmay Prabhakar, Suprosanna Shit, Johannes C. Paetzold, Ivan Ezhov, Rajat Koner, Hongwei Li, Florian Sebastian Kofler, and Bjoern Menze. «Vesselformer: Towards Complete 3D Vessel Graph Generation from Images». In: *Medical Imaging with Deep Learning*. Ed. by Ipek Oguz et al. Vol. 227. Proceedings of Machine Learning Research. PMLR, Oct. 2024, pp. 320–331. URL: https://proceedings.mlr.press/v227/prabhakar24a.html (cit. on pp. 5, 14).

[15] Johannes C Paetzold et al. «Whole brain vessel graphs: A dataset and benchmark for graph learning and neuroscience». In: *Thirty-fifth conference on neural information processing systems datasets and benchmarks track (Round 2)*. 2021 (cit. on pp. 5, 6, 59).

[16] Sinno Jialin Pan and Qiang Yang. «A survey on transfer learning». In: *IEEE Transactions on knowledge and data engineering* 22.10 (2009), pp. 1345–1359 (cit. on p. 6).

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. «Imagenet classification with deep convolutional neural networks». In: *Advances in neural information processing systems* 25 (2012) (cit. on p. 6).

[18] Yaroslav Ganin and Victor Lempitsky. «Unsupervised domain adaptation by backpropagation». In: *International conference on machine learning*. PMLR. 2015, pp. 1180–1189 (cit. on pp. 6, 7, 27).

[19] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. «Learning to adapt structured output space for semantic segmentation». In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7472–7481 (cit. on p. 7).

[20] Xiaoke Shen and Ioannis Stamos. «Simcrosstrans: A simple cross-modality transfer learning for object detection with convnets or vision transformers». In: *arXiv preprint arXiv:2203.10456* (2022) (cit. on p. 9).

[21] Yueh-Cheng Liu, Yu-Kai Huang, Hung-Yueh Chiang, Hung-Ting Su, Zhe-Yu Liu, Chin-Tang Chen, Ching-Yu Tseng, and Winston H Hsu. «Learning from 2d: Contrastive pixel-to-point knowledge transfer for 3d pretraining». In: *arXiv preprint arXiv:2104.04687* (2021) (cit. on p. 9).

[22] Yutong Xie, Jianpeng Zhang, Yong Xia, and Qi Wu. «Unimiss: Universal medical self-supervised learning via breaking dimensionality barrier». In: *European Conference on Computer Vision*. Springer. 2022, pp. 558–575 (cit. on p. 9).

[23] Yi Wang, Zhiwen Fan, Tianlong Chen, Hehe Fan, and Zhangyang Wang. «Can we solve 3D vision tasks starting from a 2D vision transformer?» In: *arXiv preprint arXiv:2209.07026* (2022) (cit. on p. 9).

[24] Mihail Ivilinov Todorov, Johannes Christian Paetzold, Oliver Schoppe, and Giles Tetteh. «Suprosanna Shit, Velizar Efremov, Katalin Todorov-Völgyi, Marco Düring, Martin Dichgans, Marie Piraud, et al». In: *Machine learning analysis of whole mouse brain vasculature. Nature methods* 17.4 (2020), pp. 442–449 (cit. on p. 9).

[25] Matthias Schneider, Johannes Reichold, Bruno Weber, Gábor Székely, and Sven Hirsch. «Tissue metabolism driven arterial tree generation». In: *Medical image analysis* 16.7 (2012), pp. 1397–1414 (cit. on pp. 9, 38, 39, 57, 66, 69, 70, 72–77, 82, 83).

[26] Katherine A McCulloh and John S Sperry. «Murray's law and the vascular architecture of plants». In: *Ecology and biomechanics. Boca Raton, FL, USA: Taylor and Francis* (2006), pp. 105–120 (cit. on pp. 12, 34).

[27] K Shinozaki. «A quantitative analysis of plant form-the pipe model theory.» In: *I & II. Jpn. J. Ecol.* 14 (1964), pp. 133–139 (cit. on pp. 12, 34).

[28] Bastian Wittmann, Yannick Wattenberg, Tamaz Amiranashvili, Suprosanna Shit, and Bjoern Menze. «vesselFM: A Foundation Model for Universal 3D Blood Vessel Segmentation». In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, pp. 20874–20884 (cit. on pp. 12–14, 50, 51).

[29] Fabian Isensee, Paul F Jaeger, Simon AA Kohl, Jens Petersen, and Klaus H Maier-Hein. «nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation». In: *Nature methods* 18.2 (2021), pp. 203–211 (cit. on p. 12).

[30] Connor Shorten and Taghi M Khoshgoftaar. «A survey on image data augmentation for deep learning». In: *Journal of big data* 6.1 (2019), pp. 1–48 (cit. on p. 12).

[31] Luis Perez and Jason Wang. «The effectiveness of data augmentation in image classification using deep learning». In: *arXiv preprint arXiv:1712.04621* (2017) (cit. on p. 12).

[32] Antonio Torralba and Alexei A Efros. «Unbiased look at dataset bias». In: *CVPR 2011*. IEEE. 2011, pp. 1521–1528 (cit. on pp. 12, 14).

[33] John R Zech, Marcus A Badgeley, Manway Liu, Anthony B Costa, Joseph J Titano, and Eric Karl Oermann. «Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study». In: *PLoS medicine* 15.11 (2018), e1002683 (cit. on pp. 12, 14).

[34] Haoliang Li, YuFei Wang, Renjie Wan, Shiqi Wang, Tie-Qiang Li, and Alex Kot. «Domain generalization for medical imaging classification with linear-dependency regularization». In: *Advances in neural information processing systems* 33 (2020), pp. 3118–3129 (cit. on p. 14).

[35] Christopher J Kelly, Alan Karthikesalingam, Mustafa Suleyman, Greg Corrado, and Dominic King. «Key challenges for delivering clinical impact with artificial intelligence». In: *BMC medicine* 17.1 (2019), p. 195 (cit. on p. 14).

[36] Jun Ma, Yuting He, Feifei Li, Lin Han, Chenyu You, and Bo Wang. «Segment anything in medical images». In: *Nature Communications* 15.1 (2024), p. 654 (cit. on p. 14).

[37] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. «End-to-end object detection with transformers». In: *European conference on computer vision*. Springer. 2020, pp. 213–229 (cit. on p. 16).

[38] Vaswani Ashish. «Attention is all you need». In: *Advances in neural information processing systems* 30 (2017), p. I (cit. on p. 17).

[39] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. «Attention augmented convolutional networks». In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 3286–3295 (cit. on p. 17).

[40] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. «Deformable detr: Deformable transformers for end-to-end object detection». In: *arXiv preprint arXiv:2010.04159* (2020) (cit. on pp. 18, 19).

[41] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. «Layer normalization». In: *arXiv preprint arXiv:1607.06450* (2016) (cit. on p. 20).

[42] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. «Domain adaptive faster r-cnn for object detection in the wild». In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 3339–3348 (cit. on p. 27).

[43] Jon D Pelletier and Donald L Turcotte. «Shapes of river networks and leaves: are they statistically similar?» In: *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences* 355.1394 (2000), pp. 307–311 (cit. on p. 33).

[44] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. «Imagenet: A large-scale hierarchical image database». In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255 (cit. on pp. 36, 59).

[45] Songtao He, Favyen Bastani, Satvat Jagwani, Mohammad Alizadeh, Hari Balakrishnan, Sanjay Chawla, Mohamed M Elshrif, Samuel Madden, and Mohammad Amin Sadeghi. «Sat2graph: Road graph extraction through graph-tensor encoding». In: *European Conference on Computer Vision*. Springer. 2020, pp. 51–67 (cit. on pp. 42, 53, 56, 59).

[46] Theophile Gentilhomme, Michael Villamizar, Jerome Corre, and Jean-Marc Odobez. «Towards smart pruning: ViNet, a deep-learning approach for grapevine structure estimation». In: *Computers and Electronics in Agriculture* 207 (2023), p. 107736 (cit. on pp. 42, 47, 53, 57, 59).

[47] Ta-Chih Lee, Rangasami L Kashyap, and Chong-Nam Chu. «Building skeleton models via 3-D medial surface axis thinning algorithms». In: *CVGIP: graphical models and image processing* 56.6 (1994), pp. 462–478 (cit. on p. 43).

[48] You-Dong Liang and Brian A. Barsky. «A new concept and method for line clipping». In: *ACM Transactions on Graphics (TOG)* 3.1 (1984), pp. 1–22 (cit. on p. 49).

[49] Martin J Menten, Johannes C Paetzold, Alina Dima, Bjoern H Menze, Benjamin Knier, and Daniel Rueckert. «Physiology-based simulation of the retinal vasculature enables annotation-free segmentation of OCT angiographs». In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2022, pp. 330–340 (cit. on pp. 57, 59, 69, 70).

[50] Soumick Chatterjee et al. «SMILE-UHURA Challenge–Small Vessel Segmentation at Mesoscopic Scale from Ultra-High Resolution 7T Magnetic Resonance Angiograms». In: *arXiv preprint arXiv:2411.09593* (2024) (cit. on pp. 58, 71–73, 75–77, 82–84, 93, 95, 96).

[51] Lukas Glandorf et al. «Bessel beam optical coherence microscopy enables multiscale assessment of cerebrovascular network morphology and function». In: *Light: Science & Applications* 13.1 (2024), p. 307 (cit. on pp. 58, 71–77, 82–84, 93, 95).

[52] Elizabeth Bullitt, Donglin Zeng, Guido Gerig, Stephen Aylward, Sarang Joshi, J Keith Smith, Weili Lin, and Matthew G Ewend. «Vessel tortuosity and brain tumor malignancy: a blinded study1». In: *Academic radiology* 12.10 (2005), pp. 1232–1240 (cit. on p. 58).

[53] Natalie A Holroyd, Zhongwang Li, Claire Walsh, Emmeline Brown, Rebecca J Shipley, and Simon Walker-Samuel. «tUbeNet: a generalizable deep learning tool for 3D vessel segmentation». In: *Biology Methods and Protocols* 10.1 (2025), bpaf087 (cit. on p. 58).

[54] Kaiyuan Yang et al. «Benchmarking the cow with the topcow challenge: Topology-aware anatomical segmentation of the circle of willis for cta and mra». In: *ArXiv* (2025), arXiv–2312 (cit. on p. 58).

[55] Mihail Ivilinov Todorov et al. «Machine learning analysis of whole mouse brain vasculature». In: *Nature methods* 17.4 (2020), pp. 442–449 (cit. on p. 58).

[56] Willy Kuo, Diego Rossinelli, Georg Schulz, Roland H Wenger, Simone Hieber, Bert Müller, and Vartan Kurtcuoglu. «Terabyte-scale supervised 3D training and benchmarking dataset of the mouse kidney». In: *Scientific data* 10.1 (2023), p. 510 (cit. on p. 58).

[57] Luc Soler, Alexandre Hostettler, Vincent Agnus, Arnaud Charnoz, Jean-Baptiste Fasquel, Johan Moreau, Anne-Blandine Osswald, Mourad Bouhadjar, and Jacques Marescaux. «3d image reconstruction for comparison of algorithm database». In: *URL: https://www. ircad. fr/research/data-sets/liver-segmentation-3d-ircadb-01* 13 (2010), p. 4 (cit. on p. 58).

[58] Brain Development Organization. *IXI Dataset*. n.d. URL: http://brain-development.org/ixi-dataset/ (visited on 03/11/2026) (cit. on pp. 58, 70, 71, 81).

[59] Mohammad Haft-Javaherian, Linjing Fang, Victorine Muse, Chris B Schaffer, Nozomi Nishimura, and Mert R Sabuncu. «Deep convolutional neural networks for segmenting 3D in vivo multiphoton images of vasculature in Alzheimer disease mouse models». In: *PloS one* 14.3 (2019), e0213539 (cit. on p. 58).

[60] Charissa Poon, Petteri Teikari, Muhammad Febrian Rachmadi, Henrik Skibbe, and Kullervo Hynynen. «A dataset of rodent cerebrovasculature from in vivo multiphoton fluorescence microscopy imaging». In: *Scientific Data* 10.1 (2023), p. 141 (cit. on p. 58).

[61] Ekin Yagis et al. «Deep learning for vascular segmentation and applications in phase contrast tomography imaging». In: *arXiv preprint arXiv:2311.13319* (2023) (cit. on p. 58).

[62] Lu Meng, Lijun Zhou, Wentao Zhang, and Keli Xie. «Robust epileptic seizure prediction: A 3D-SERESNet framework for patient-specific and multi-patient generalization». In: *iScience* 28.12 (2025) (cit. on p. 60).

[63] Rafael Padilla, Sergio L Netto, and Eduardo AB Da Silva. «A survey on performance metrics for object-detection algorithms». In: *2020 international conference on systems, signals and image processing (IWSSIP)*. IEEE. 2020, pp. 237–242 (cit. on p. 62).

[64] James Biagioni and Jakob Eriksson. «Inferring road maps from global positioning system traces: Survey and comparative evaluation». In: *Transportation research record* 2291.1 (2012), pp. 61–71 (cit. on p. 63).

[65] Davide Belli and Thomas Kipf. «Image-conditioned graph generation for road network extraction». In: *arXiv preprint arXiv:1910.14388* (2019) (cit. on p. 63).

[66] Xavier Glorot and Yoshua Bengio. «Understanding the difficulty of training deep feedforward neural networks». In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings. 2010, pp. 249–256 (cit. on p. 65).