POLITECNICO DI TORINO

MASTER's Degree in COMPUTER ENGINEERING

MASTER's Degree Thesis

Real-Time Defect Identification in Welding Processes through Thermal and Audio Signal Analysis

Supervisors

Prof. Raffaella SESANA

Prof. Luca SANTORO

Candidate

Taha KAMALISADEGHIAN

JULY 2026

Abstract

This thesis investigates real-time welding-defect identification through multimodal analysis of thermal and acoustic signals. The task is formulated as a hierarchical two-stage problem in which binary defect screening precedes multiclass defect diagnosis across 12 classes. Using the Intel Robotic Welding Multimodal Dataset (3841 runs), the study compares three offline fusion approaches—Hierarchical Ensemble, Backbone Fusion, and WeldFusionNet—under a session-disjoint grouped split protocol with all 12 classes represented in the test partition (train=2676, val=385, test=780), and introduces WeldFusionNet-RT, a real-time-native variant trained directly on 1-second sliding windows.

The experimental workflow combines synchronized feature extraction, offline evaluation, and recording-level replay with latency and throughput profiling. The Hierarchical Ensemble uses XGBoost and ExtraTrees with 417 handcrafted features and isotonic calibration. Backbone Fusion combines pretrained ResNet18 video embeddings with log-mel spectrogram audio embeddings via weighted probability fusion. WeldFusionNet is an end-to-end neural network with Conv1D and MobileNetV3 encoders, dual classification heads, and focal loss training. WeldFusionNet-RT shares the same architecture but is trained natively on 1-second windows to minimize inference latency.

Under cross-session evaluation on the 780-run test partition, WeldFusionNet leads binary defect screening with binary F1 of 0.9895 (ROC-AUC 0.9981, accuracy 0.9833); Backbone Fusion follows at binary F1 0.9384 and Hierarchical Ensemble at 0.9163. For 12-class multiclass diagnosis, Backbone Fusion attains the highest macro-F1 (0.8319), with WeldFusionNet at 0.8178 and Hierarchical Ensemble at 0.7459. A paired bootstrap on macro-F1 (1000 resamples, 95% CI [−0.026, +0.053]) and a McNemar test on the binary task ($p \approx 6 \times 10^{-12}$) confirm that WeldFusionNet is significantly stronger than Backbone Fusion on binary screening but statistically tied on multiclass diagnosis. The dominant residual failure across all three offline models is class 05 (Undercut), whose audio-thermal signature is closest to that of a good weld. Under recording-level real-time evaluation on a 240-run balanced subset (20 per class), WeldFusionNet processes each replay inference event in 72.2 ms mean (P95 83.5 ms) at binary F1 0.9886, Backbone Fusion in 127.5 ms (P95 141.3 ms) per event at 0.9698, and Hierarchical Ensemble in 151.7 ms (P95 170.0 ms) per event at 0.9559. WeldFusionNet-RT achieves strong window-level performance under a separate config-grouped evaluation protocol (binary F1 0.9946, macro-F1 0.9853; window-level, not directly comparable to recording-level metrics) with a per-window GPU latency of 3.2 ms (P95 3.8 ms), demonstrating the computational feasibility of sub-second streaming inference. Because its labels are inherited from recording-level annotations and its split differs from the session-disjoint offline protocol, these results are interpreted as continuous window-level scoring rather than verified defect-onset localization, and as a proof-of-concept rather than direct superiority over the recording-level offline models. The thesis concludes that multimodal welding-monitoring systems should be evaluated through both predictive quality and operational feasibility, and that deployment mode—streaming versus post-weld—determines model selection as much as offline accuracy rank.

ACKNOWLEDGMENTS

I owe my deepest gratitude to my supervisor, prof. Raffaella Sesana, whose guidance, rigor, and steady confidence in this work shaped it at every stage and pushed me to hold it to a higher standard.

My sincere thanks go to prof. Luca Santoro, CTO of Therness SRL, and to Pietro Paolo De Crescenzo, COO of Therness SRL, for their trust, their technical insight, and the opportunity to ground this research in a real industrial setting. Their support turned an academic idea into something with practical purpose.

To my family and friends: thank you for the patience, encouragement, and belief that carried me through. This achievement is as much yours as it is mine.

Taha Kamalisadeghian
Turin, July 2026

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AE	Acoustic emission
AUC	Area under the curve
AWS	American Welding Society
BF	Backbone Fusion
BN	Batch normalization
CI	Confidence interval
CNN	Convolutional neural network
CPU	Central processing unit
DCT	Discrete cosine transform
DL	Deep learning
ECE	Expected calibration error
FC	Fully connected
FPS	Frames per second
GMAW	Gas metal arc welding
GPU	Graphics processing unit
GTAW	Gas tungsten arc welding
HE	Hierarchical Ensemble
IR	Infrared
ISO	International Organization for Standardization
MAG	Metal active gas welding
MCE	Maximum calibration error
MFCC	Mel-frequency cepstral coefficient
ML	Machine learning
MLP	Multilayer perceptron
MTL	Multi-task learning
NDT	Non-destructive testing
P95	Ninety-fifth percentile
PR	Precision–recall
RMS	Root mean square
ROC	Receiver operating characteristic
ROC-AUC	ROC area under the curve
RT	Real-time
SHAP	SHapley Additive exPlanations
SNR	Signal-to-noise ratio
STFT	Short-time Fourier transform
TIG	Tungsten inert gas welding
WFN	WeldFusionNet
ZCR	Zero-crossing rate

Chapter 1

Introduction

1.1 Motivation and Industrial Context

Welding is a critical joining process in sectors such as automotive manufacturing, shipbuilding, heavy steel construction, and robotic fabrication [1, 2]. In these settings, weld quality directly affects structural integrity, repair cost, and production continuity. Conventional quality assurance remains important, but it is often applied after completion of the weld through visual inspection, destructive examination, or delayed non-destructive testing. Such procedures can verify final quality, yet they do not provide immediate feedback when process drift begins during welding. A defect that is detected only after completion may require rework, interrupt production schedules, and compromise batch consistency [1, 3].

For this reason, the shift from retrospective inspection to online quality monitoring has received increasing attention in welding automation research [4, 5]. Monitoring systems intended for industrial deployment are generally expected to satisfy constraints beyond classification accuracy, including response time compatible with process intervention, tolerance to sensor noise and process variability, and interpretability of outputs in relation to physically meaningful weld behavior.

1.1.1 Industrial Significance and Economic Cost of Defects

The economic consequences of undetected weld defects extend across the entire production chain. In automotive manufacturing, a single defective weld in a structural component can necessitate recall campaigns, costly disassembly, and requalification of production batches. In shipbuilding and offshore infrastructure, structural weld failures carry not only financial consequences but also safety and regulatory implications that may ground operations entirely. Estimates from the manufacturing sector suggest that weld-related rework and scrap account for a significant fraction of total production costs in high-volume fabrication environments [4, 1].

The economic argument for online monitoring is therefore straightforward in principle: if a defect can be identified at the moment it forms, rather than after the part has been completed, finished, and entered the supply chain, the cost of intervention drops substantially. A real-time system that raises an alarm while the torch is still active allows immediate parameter adjustment or abort, eliminating downstream rework. This opportunity is only

realizable, however, if the monitoring system is fast enough to act within the process window. A classification algorithm that takes several seconds to reach a verdict offers no advantage over post-weld inspection; a system capable of raising a defect warning within a fraction of a second of its formation is genuinely different from retrospective analysis.

The hardware barriers to online monitoring have receded considerably. Thermal cameras and synchronized microphones suitable for robotic cell integration are now commodity components; the unsolved problem is no longer sensor availability but algorithmic reliability under real process conditions and deployment feasibility under operational constraints.

1.1.2 Why This Problem is Hard

The difficulty is structural rather than merely technical. Weld defects arise from coupled interactions between heat input, torch geometry, shielding gas behavior, filler transfer dynamics, base material properties, and joint preparation quality, so no single measurable quantity reliably predicts defect formation—the same arc-sound pattern can have different causes depending on the welding configuration. Both sensing modalities compound this: thermal cameras record radiated heat from the pool, not defect geometry itself, and microphones record plasma dynamics and droplet transfer, not metallurgical state. This indirection means the relationship between sensor observation and defect label must be learned from data. The learning problem is further complicated by natural class imbalance—most production welds are sound by design, so defect examples are structurally rare—and by the varying observability of different defect types: gross porosity produces distinct signatures, while subtle lack-of-fusion may be acoustically and thermally indistinguishable from adjacent classes. Unlike standard classification benchmarks with fixed-length independent samples, welding runs vary in duration and defect evidence may be concentrated in a brief interval, requiring aggregation strategies that remain sensitive to localized anomalies. Finally, even a highly accurate model is not deployable if its inference latency exceeds the available decision window: the process moves at a fixed rate and the system must keep pace. These constraints—physical ambiguity, sensor indirection, class imbalance, temporal aggregation, and runtime feasibility—motivate the hierarchical two-stage design, multimodal sensing approach, and joint quality-and-latency evaluation protocol developed in this thesis.

1.2 Research Problem

This thesis addresses the problem of identifying welding defects in real time from thermal and acoustic observations while preserving both predictive quality and operational feasibility. The objective is not limited to offline classification accuracy. It explicitly includes deployment-oriented requirements such as latency, throughput, and stability at recording level. Throughout this thesis, “real-time” refers to a sliding-window replay protocol in which pre-recorded runs are processed as a stream of overlapping windows, each classified independently and aggregated to a recording-level decision. This protocol approximates operational monitoring conditions without requiring live sensor integration, and all latency and throughput figures are measured under this replay setting.

The classification task is formulated as a two-stage hierarchy. In the first stage, the system determines whether a recording represents a sound weld or a defective weld. In the second stage, recordings identified as defective are assigned to a defect category. This structure reflects the operational sequence commonly applied in process monitoring, in which rapid defect screening precedes finer diagnosis.

In the dataset version used in this work, the label set includes one normal class, coded as `00`, and eleven defect classes, coded as `01` to `11`. These codes are retained throughout the thesis because they map directly to the published dataset taxonomy and allow compact reporting of class-wise performance.

1.3 Research Objectives and Questions

The work is guided by three research questions:

1. Can thermal and acoustic features provide reliable binary defect detection at recording level?
2. Which multimodal fusion strategy provides the best tradeoff between multiclass quality and real-time execution constraints?
3. How should model selection be performed when offline accuracy and runtime behavior suggest different deployment choices?

These questions are addressed through three corresponding objectives: establishing whether multimodal thermal-acoustic fusion achieves reliable binary defect detection under a leakage-aware grouped split; comparing Hierarchical Ensemble, Backbone Fusion, and WeldFusionNet on multiclass discrimination quality and real-time execution under identical data conditions; and determining whether offline and runtime performance rankings agree, with deployment guidance for cases where they do not.

1.4 Scope and Contributions

The thesis makes four main contributions. First, it frames welding-defect monitoring as a joint sensing-and-decision problem in which thermal and acoustic information are evaluated under the same session-disjoint split protocol. Second, it provides a controlled comparison of three offline fusion approaches—Hierarchical Ensemble (XGBoost + ExtraTrees with handcrafted features), Backbone Fusion (pretrained ResNet18 + log-mel spectrogram embeddings), and WeldFusionNet (end-to-end Conv1D + MobileNetV3 neural network)—for both binary screening and multiclass diagnosis across 12 classes. Third, it introduces WeldFusionNet-RT, a real-time-native variant of WeldFusionNet trained directly on 1-second sliding windows. WeldFusionNet-RT achieves 3.2 ms per-window GPU latency and window-level macro-F1 of 0.9853 (window-level, not directly comparable to recording-level metrics) on its window-level test partition; this score is reported as a separate streaming-native experiment rather than as a head-to-head multiclass winner, because the window-level metric is not directly comparable to the recording-level metrics of the three offline models. The contribution is the demonstration

that native window-level training enables continuous sub-second window-level scoring during the weld without the train–inference distribution mismatch that affects offline models applied to short windows. The model performs window-level classification under recording-level supervision; true defect-onset localization would require fine-grained temporal labels and is identified as future work. Fourth, it examines whether model ranking is affected when real-time operating constraints are evaluated alongside offline classification quality, and discusses the implications for deployment-oriented model selection.

The thesis is limited to the available dataset snapshot, the selected thermal and acoustic modalities, and a recording-level formulation of online monitoring. It does not attempt to define universal industrial acceptance rules for every welding process. Instead, it investigates how multimodal evidence should be analyzed within one carefully controlled experimental setting. An important scope decision is to treat deployment constraints as first-class evaluation targets rather than secondary diagnostics, because a model that achieves the highest offline multiclass score may not satisfy deployment-level constraints if its runtime profile is incompatible with process-control requirements [6].

1.5 Thesis Structure

Chapter 2 covers the technical background on sensing modalities, multimodal fusion, and the machine learning methods employed. Chapter 3 surveys prior work and identifies the evaluation gaps this study addresses. Chapters 4 and 5 develop the methodology and experimental setup; Chapters 6 and 7 report and interpret the results. Chapter 8 concludes with priorities for future work; the appendix collects supplementary tables and confusion matrices.

Chapter 2

Background

2.1 Welding Processes, Defect Formation, and Quality Criteria

Arc-welding quality depends on a coupled interaction between heat input, torch motion, filler transfer, shielding conditions, and metallurgical response. Disturbance in any of these factors can alter weld-pool geometry, cooling behavior, and fusion quality, thereby generating imperfections that are observable either during welding or after solidification [1, 5]. Defect formation can therefore be understood as a process phenomenon, not merely a geometric outcome, since it arises from disturbances in the welding process itself: porosity may follow gas entrapment, lack of fusion may follow insufficient energy transfer or poor joint preparation, and crack-related failures may emerge from thermal stress, rapid cooling, or unfavorable local chemistry.

International standards and welding codes provide an important interpretive framework for this problem. ISO 6520-1 classifies geometric imperfections in fusion welding, ISO 5817 defines quality levels for fusion-welded joints according to severity and intended service conditions, and AWS D1.1/D1.1M provides a widely used structural welding code for steel applications [7, 3, 8]. For machine-learning studies, these standards matter because they clarify that class labels are not arbitrary identifiers: they correspond to quality-relevant failure modes that can affect fitness for service. A monitoring model is generally most credible when its predictions can be connected to meaningful defect mechanisms and to the operational consequences of those mechanisms.

From a sensing perspective, defect evolution is rarely instantaneous. A recording may begin with apparently stable behavior and later diverge because of shielding disturbance, arc-length variation, filler-transfer irregularity, or local thermal accumulation. This temporal evolution explains why recording-level interpretation is necessary. Isolated frames or short waveform fragments may capture only partial evidence, whereas run-level monitoring can represent how stability changes over time [1, 4].

2.2 Thermal Sensing for Process Monitoring

Thermal imaging offers a non-contact view of heat distribution around the weld pool, adjacent material, and cooling region. Even when absolute radiometric calibration is imperfect,

relative temperature structure can reveal local overheating, asymmetrical heat flow, unstable pool shape, or abrupt changes in cooling pattern [9]. In industrial settings this is of particular interest because thermal sensing captures spatial evidence that is difficult to infer from scalar process variables alone.

Thermal sensing is nevertheless constrained by emissivity variation, reflective surfaces, camera viewpoint, and environmental radiation. These effects can distort apparent intensity even when the underlying process is unchanged [9, 4]. For this reason, many monitoring approaches emphasize robust spatial or temporal descriptors rather than absolute temperature estimates. Accordingly, the primary value of thermal sensing may reside less in direct thermometry and more in stable characterization of heat patterns and their variation through time.

2.3 Acoustic Sensing for Process Monitoring

Arc sound is generated by plasma fluctuation, droplet transfer, metal interaction, and structural vibration, which means that it contains indirect information about process stability and energy transfer. Compared with imaging systems, microphones and acoustic sensors are inexpensive and easy to integrate, making them suitable for environments in which camera positioning or line-of-sight visibility is constrained [10, 11, 12].

Acoustic monitoring studies show that time-domain and time-frequency behavior can support penetration assessment, process-state recognition, and defect-related discrimination [10, 11, 12]. In practice, however, raw waveform analysis is seldom sufficient. Feature extraction is typically required to summarize energy distribution, spectral concentration, bandwidth, and transient behavior in a form suitable for classification. Features such as MFCCs, spectral centroid, spectral bandwidth, spectral rolloff, and zero-crossing rate are widely used because they transform complex audio patterns into a compact representation that remains informative for statistical learning [13, 14].

The main weakness of acoustic sensing is context sensitivity. Background machinery, reverberation, microphone position, and the mechanical layout of the welding cell can strongly affect observed sound [11, 12]. As a result, classification accuracy achieved under one set of acoustic conditions does not ensure comparable performance under different conditions. This limitation motivates the use of complementary sensing channels.

2.4 Multimodal Learning and Sensor Complementarity

Multimodal monitoring is motivated by the observation that thermal and acoustic channels encode different aspects of the same welding event. Thermal features reflect heat distribution and pool evolution, whereas acoustic features capture arc dynamics and process instability. A combined system may therefore compensate for limitations inherent to each individual sensing modality if the two channels are aligned and fused appropriately [15, 16].

Three broad fusion families are commonly distinguished in multimodal learning.

Early or feature-level fusion concatenates raw or extracted features from each modality into a single representation before classification. This approach is straightforward to

implement and allows a single classifier to learn joint patterns across modalities. However, it can be sensitive to differences in feature scale and dimensionality between modalities, and it does not model modality-specific structure before combination [15].

Intermediate or deep fusion processes each modality through a dedicated encoding branch before combining the learned embeddings at an internal representation layer. This allows the model to learn modality-specific abstractions while still capturing cross-modal interactions in the joint space. The main drawback is increased model complexity and the need for larger training sets to avoid overfitting, particularly when modality-specific branches have many parameters [15].

Late or decision-level fusion trains independent classifiers for each modality and combines their output probabilities or predictions. This preserves modality-specific decision streams, which can improve robustness when one modality is noisy or unavailable. However, late fusion may underuse fine-grained cross-modal interactions because the modalities are combined only at the decision stage [15].

Figure 2.1 illustrates where each paradigm combines the modalities and what that implies for representational capacity and robustness.

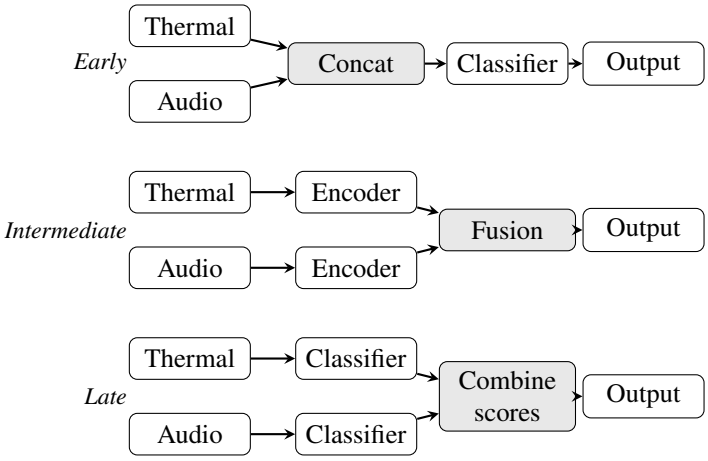

Figure 2.1: Multimodal fusion taxonomy. *Early fusion* concatenates raw features before classification. *Intermediate fusion* encodes each modality independently and combines learned embeddings. *Late fusion* trains separate classifiers and combines their output scores. The three approaches compared in this thesis—Hierarchical Ensemble, Backbone Fusion, and WeldFusionNet—correspond to early, late, and intermediate fusion respectively.

2.5 Class Imbalance and Evaluation Metrics

In supervised classification, class imbalance arises when certain categories are represented by substantially fewer training and evaluation samples than others. Standard accuracy can be misleading under such conditions because a classifier that predicts only the majority class may achieve high accuracy while failing entirely on minority categories [17].

To address this, evaluation metrics that account for class-level performance are preferred. Macro-averaged F1 treats all classes equally by computing the per-class F1 score and averaging without weighting by class frequency. Weighted-averaged F1 weights each class by its support,

giving more influence to frequent classes. In defect-monitoring applications, macro-F1 is often more informative because it penalizes models that ignore rare but safety-relevant defect categories [18].

Common mitigation strategies include class-weighted loss functions, oversampling of minority classes, and threshold calibration. In this thesis, class-weighted focal loss is used for deep fusion training, and macro-F1 serves as the primary multiclass evaluation metric to ensure that rare defect categories are not overlooked.

2.6 Real-Time Monitoring as a Design Requirement

Offline recognition quality is a necessary but insufficient criterion for industrial monitoring. A classifier that performs well on a static test set may still be unsuitable for online use if its inference latency exceeds the available decision window, throughput is inconsistent, or predictions degrade under mild sensing disturbances. Prior work on robotic and hybrid welding monitoring repeatedly emphasizes this integration of quality and responsiveness [4, 5].

Accordingly, real-time behavior is treated here as a design requirement on equal footing with classification quality. Latency, throughput, and robustness under replay are analyzed alongside accuracy and F1 measures—consistent with the broader systems argument that tail latency can be as consequential as mean performance when decisions must be produced under operational time constraints [6].

2.7 Gradient Boosted Trees and Random Forests

Two families of ensemble tree models are used in the Hierarchical Ensemble approach: gradient boosted trees (as implemented in XGBoost) and randomized trees (ExtraTrees, a variant of Random Forests). Understanding their theoretical basis clarifies the design choices made in Chapter 4.

2.7.1 Gradient Boosted Trees

Gradient boosting is an ensemble method that constructs a strong predictor by sequentially adding weak learners, each trained to correct the residual errors of the current ensemble [19]. Given a loss function $\mathcal{L}(y, F(x))$, the method initializes $F_0(x)$ with a constant prediction and iteratively updates:

$$F_m(x) = F_{m-1}(x) + \eta\, h_m(x),$$

where $h_m(x)$ is a regression tree fitted to the negative gradient (pseudo-residuals) of $\mathcal{L}$ with respect to the current prediction, and $\eta \in (0, 1]$ is a learning rate that controls step size. The key insight of Friedman's formulation is that fitting a tree to the gradient of the loss is equivalent to performing gradient descent in function space, where the domain is the set of real-valued functions rather than a fixed-dimensional parameter vector [19].

XGBoost extends this formulation with a regularized objective that penalizes both the squared leaf weights and the number of leaves per tree [20]. This regularization

substantially reduces overfitting on small-to-medium datasets and allows the use of deeper trees than the original gradient boosting formulation. Additional enhancements include column subsampling (analogous to Random Forest's feature bagging), row subsampling, and hardware-aware parallelization of split finding. These properties make XGBoost well-suited for high-dimensional handcrafted feature vectors where many features may be correlated or irrelevant.

2.7.2 Random Forests and ExtraTrees

Random Forests build an ensemble of independent decision trees, each trained on a bootstrap sample of the training data with a random subset of features considered at each split [21]. Ensemble variance is reduced by averaging predictions (for regression) or voting (for classification) across trees, while individual tree bias is controlled by the number of estimators. The combination of bootstrap sampling and feature randomization decorrelates the trees, making the ensemble more robust than any individual tree.

ExtraTrees (Extremely Randomized Trees) extends this idea by further randomizing split selection: rather than finding the locally optimal split threshold for each candidate feature, the algorithm draws the threshold uniformly at random from the observed feature range and selects the split with the best randomly evaluated criterion [22]. This additional randomization increases variance for individual trees but further reduces the variance of the ensemble, and makes the training procedure substantially faster because no sorting or exhaustive search is required at each node.

In the context of this thesis, XGBoost and ExtraTrees are combined in a weighted ensemble because they have complementary inductive biases: XGBoost is generally stronger on per-feature discrimination due to its additive gradient correction, while ExtraTrees introduces high diversity through aggressive randomization, which stabilizes predictions on minority classes.

2.8 Convolutional Neural Networks and Transfer Learning

2.8.1 Convolutional Neural Networks

Convolutional neural networks (CNNs) are a class of deep learning architectures designed to process data with spatial or temporal structure [23]. The fundamental operation is a discrete convolution:

$$(I * K)(u, v) = \sum_{m} \sum_{n} I(u - m, v - n) K(m, n),$$

where I is the input feature map, K is a learned filter (kernel), and the output is a feature map that captures local patterns matching the filter. By stacking convolutional layers with nonlinear activations, CNNs learn hierarchical representations: early layers detect low-level features (edges, textures), while deeper layers combine these into increasingly abstract representations (shapes, objects).

Three properties make CNNs well-suited for the visual inputs in this thesis. First, weight sharing: the same filter is applied across all spatial positions, dramatically reducing the

number of parameters compared to fully connected networks and encoding the assumption that useful features are spatially invariant. Second, local receptive fields: each neuron sees only a small region of the input, matching the observation that informative patterns in thermal images are often localized. Third, hierarchical composition: learned features at each depth are composites of simpler features, allowing complex thermal patterns to be expressed as combinations of basic edge and gradient responses.

2.8.2 Residual Networks

Training very deep CNNs was historically hampered by the vanishing gradient problem, in which gradients diminish exponentially as they propagate through many layers, preventing effective parameter updates in early network layers [24]. Residual networks (ResNets) address this by introducing shortcut connections that add the input of a block directly to its output [25]:

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + \mathbf{x},$$

where $\mathcal{F}(\mathbf{x}, \{W_i\})$ is the residual mapping learned by the block and $\mathbf{x}$ is the identity shortcut. These connections allow gradients to flow directly through the shortcut path during backpropagation, effectively bypassing the nonlinear residual function when necessary. The result is that networks with hundreds of layers can be trained stably. ResNet18, used in the Backbone Fusion approach, has 18 weighted layers and approximately 11.7 million parameters [25].

2.8.3 Lightweight Networks: MobileNetV3

Standard CNNs are often computationally expensive for deployment on constrained hardware. MobileNetV3 addresses this by combining depthwise separable convolutions, inverted residuals, linear bottlenecks, and a hardware-aware neural architecture search to minimize latency while preserving accuracy [26]. A depthwise separable convolution factorizes a standard convolution into a depthwise convolution (applied separately to each input channel) and a pointwise convolution (a 1×1 convolution across channels), reducing computation by a factor of approximately $1/D_K^2$ where D_K is the kernel size. MobileNetV3-Small, used in WeldFusionNet, targets particularly constrained settings and has approximately 2.5 million parameters, of which a subset is trainable when the early feature layers are frozen.

2.8.4 Transfer Learning

Transfer learning is the practice of initializing a model with parameters learned on a source task and domain, then adapting it to a different target task or domain [27]. The motivation is that representations learned on large, well-annotated source datasets (such as ImageNet for vision tasks) capture general perceptual features—edges, textures, shapes—that are also informative in many target domains, even when the domains differ substantially in content [28].

Empirical studies show that transferability varies by network depth: early layers capture highly general features that transfer well across domains, while later layers become increasingly

task-specific and may require more adaptation for new tasks [28]. This motivates the common practice of freezing the early feature extraction layers (as in WeldFusionNet, where the first 50% of MobileNetV3 feature layers are frozen) while allowing the final layers and classification heads to be trained on the target data. This strategy is especially valuable when the target dataset is small, as it reduces the risk of catastrophic forgetting and overfitting on limited labeled examples.

In the context of thermal welding images, transfer from ImageNet is imperfect because the source domain (natural photographs) differs from the target domain (infrared thermal sequences). Nevertheless, low-level features such as edge detection, gradient responses, and local texture patterns are sufficiently domain-general to provide a useful starting point, and empirical results in industrial inspection domains consistently show that ImageNet-pretrained features outperform random initialization when labeled industrial data is limited [29, 30].

2.9 Audio Signal Processing and Feature Extraction Theory

2.9.1 Short-Time Fourier Transform

The Short-Time Fourier Transform (STFT) is the foundational operation for time-frequency analysis of non-stationary signals such as arc sound [31]. For a discrete-time signal $x[n]$, the STFT at time frame m and frequency bin k is:

$$X[m,k] = \sum_{n=0}^{N-1} x[n+mH]\, w[n]\, e^{-j2\pi kn/N},$$

where $w[n]$ is an analysis window of length N (typically a Hann or Hamming window), H is the hop length (number of samples between successive frames), and $e^{-j2\pi kn/N}$ is the discrete Fourier basis at frequency bin k. The power spectrogram is $|X[m,k]|^2$, which represents the energy at each time-frequency cell.

The window length N and hop length H trade off time and frequency resolution: a longer window gives finer frequency resolution but coarser time resolution, while a shorter window gives finer time resolution but coarser frequency resolution. For arc sound analysis at 16 kHz, typical settings use $N = 1024$ (64 ms window) and $H = 256$ or $H = 512$, yielding an acceptable compromise between the two resolutions.

2.9.2 Mel Filterbank and Log-Mel Spectrogram

The mel scale is a perceptually motivated frequency scale that compresses the high-frequency range of the spectrum, reflecting the roughly logarithmic relationship between perceived pitch and physical frequency in human hearing [13]. The conversion from Hertz to mel is:

$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right).$$

A mel filterbank consists of M overlapping triangular filters distributed uniformly on the mel scale. Each filter integrates the power spectrum within its frequency range, compressing the

STFT into a M-dimensional representation per frame. The log-mel spectrogram is obtained by applying the natural logarithm to the filterbank energies, further compressing the dynamic range and producing a representation that is more robust to amplitude scaling.

While the mel scale was originally designed to model human auditory perception, it has proven broadly useful for machine-learning-based audio classification because it discards fine-grained spectral detail in frequency ranges where the signal-to-noise ratio is often poor, while preserving perceptually salient structure in lower frequency bands. For arc sound analysis, the mel filterbank serves a similar role: it compresses the high-frequency noise content of the arc plasma while retaining spectral structure associated with droplet transfer rates and arc stability.

2.9.3 Mel-Frequency Cepstral Coefficients

Mel-frequency cepstral coefficients (MFCCs) are derived from the log-mel spectrogram by applying a discrete cosine transform (DCT) [13]:

$$c_k = \sum_{m=1}^{M} \log S_m \cos\left(\frac{\pi k (m - 0.5)}{M}\right), \quad k = 1, 2, \ldots, K,$$

where S_m is the energy in the m-th mel band and K is the number of cepstral coefficients retained (typically 12–20). The DCT decorrelates the mel band energies, producing a compact representation in which successive coefficients are approximately uncorrelated. The lower cepstral coefficients capture broad spectral shape (overall energy distribution), while higher coefficients capture finer spectral detail.

In practice, MFCCs are computed per-frame and then summarized at the run level using statistical aggregates (mean, standard deviation, percentiles). This run-level aggregation is appropriate when the classification target is at the recording level rather than at the frame level, as is the case in this thesis.

2.10 Neural Network Training, Optimization, and Regularization

Training deep neural networks requires managing the interplay between model capacity, optimization dynamics, and regularization. The three approaches compared in this thesis rely on several well-established training techniques whose theoretical properties are relevant to understanding the experimental design in Chapters 4 and 5.

2.10.1 Batch Normalization

Batch normalization (BN) is a technique that normalizes the activations within each mini-batch to have zero mean and unit variance, then applies a learned affine transformation [32]. For a mini-batch of activations $\{x_1, \ldots, x_B\}$, the operation is:

$$\hat{x}_i = \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \varepsilon}}, \quad y_i = \gamma \hat{x}_i + \beta,$$

where $\mu_{\mathcal{B}}$ and $\sigma^2_{\mathcal{B}}$ are the mini-batch mean and variance, ε is a small constant for numerical stability, and γ, β are learnable scale and shift parameters.

Batch normalization has two primary effects on training dynamics. First, it reduces internal covariate shift: as parameters in early layers change, the distribution of inputs to later layers shifts accordingly, making optimization harder. By re-centering and re-scaling activations at each layer, BN reduces this effect and allows higher learning rates. Second, BN acts as a mild regularizer by injecting noise through the batch statistics, reducing the need for dropout in some architectures [32]. In the welding classification context, BN is applied after each convolutional layer in WeldFusionNet and within the MLP heads of Backbone Fusion to stabilize training on the relatively small dataset.

2.10.2 Dropout Regularization

Dropout is a stochastic regularization technique that randomly sets a fraction p of unit activations to zero during each forward pass in training, effectively training an ensemble of thinned sub-networks [33]. Most modern deep-learning frameworks (including PyTorch, used in this thesis) implement *inverted dropout*: surviving activations are rescaled by $1/(1-p)$ during training so that the expected activation magnitude is preserved between training and inference, and dropout is simply disabled at inference with no additional output scaling. The regularization effect arises from the forced redundancy: a network trained with dropout cannot rely on any single activation and must learn distributed representations. This is particularly valuable when the number of training examples is modest relative to the model capacity, as is the case for the minority defect classes in the welding dataset.

In WeldFusionNet, dropout with $p = 0.5$ is applied to the fully connected layers of the classification head. This choice reflects the higher regularization need for the final layers, which are the most prone to memorizing training examples, compared to the convolutional feature extraction layers.

2.10.3 Adam Optimizer and Learning Rate Scheduling

The Adam optimizer is an adaptive gradient method that maintains per-parameter running estimates of the first and second gradient moments, applying bias-correction to compensate for their zero initialization [34]. The key property for this thesis is per-parameter adaptivity: because different branches of the multi-modal architectures (thermal encoder, audio encoder, fusion head) may produce gradients of very different magnitudes, a uniform learning rate would under-update some components and over-update others. Adam's per-parameter step sizes naturally balance these magnitudes without manual tuning of per-layer rates.

In WeldFusionNet, Adam is combined with cosine annealing scheduling, which decays the learning rate smoothly from a maximum to a minimum over training [35]. This allows aggressive parameter updates early in training followed by fine-grained convergence at the end, without requiring manual decay decisions.

2.10.4 Focal Loss for Class-Imbalanced Learning

Standard cross-entropy loss treats all training examples equally, which causes the gradient to be dominated by the frequent majority class under imbalance. Focal loss [36] corrects this by modulating the per-example loss with a factor that down-weights easy, correctly classified examples:

$$\mathcal{L}_{\text{focal}}(p_t) = -\alpha_t(1 - p_t)^{\gamma} \log(p_t),$$

where p_t is the model's probability for the ground-truth class, $\gamma \geq 0$ is the focusing parameter (set to $\gamma = 2.0$ in WeldFusionNet), and α_t is a class-frequency weighting term. The $(1 - p_t)^{\gamma}$ factor approaches zero for high-confidence correct predictions and approaches one for hard or misclassified examples, concentrating gradient signal on the minority defect classes that are most informative for learning.

2.10.5 Multi-Task Learning

Multi-task learning (MTL) trains a shared model on multiple related objectives simultaneously, using the signal from each task to regularize the others [37]. In WeldFusionNet, the combined training objective is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{multiclass}} + \lambda \mathcal{L}_{\text{binary}},$$

where $\mathcal{L}_{\text{multiclass}}$ is the 12-class focal loss, $\mathcal{L}_{\text{binary}}$ is the binary defect detection loss, and λ is a weighting coefficient. The binary auxiliary head reinforces the model's ability to distinguish normal from defective runs, which is the higher-level structure in the label hierarchy. By jointly optimizing both objectives, the shared backbone is encouraged to learn representations that are useful for both discrimination levels, reducing the risk that the 12-class head overfits to the multiclass structure at the expense of the broad defect-detection signal.

2.11 Probability Calibration in Classification

All fusion approaches compared in this thesis produce raw probability outputs that may be systematically over- or under-confident. Post-hoc calibration corrects this before the two-stage decision gate (Section 4.6), because the binary screening threshold and the confidence-based triage logic described in Chapter 7 depend on probabilities that are well-aligned with empirical accuracy. Two methods are used: isotonic regression for the tree-based Hierarchical Ensemble and temperature scaling for the neural network approaches.

A classifier is said to be calibrated if its predicted probability $\hat{p}(y = c \mid x)$ reflects the true fraction of samples with label c among all samples assigned that probability [38]. Formally, a perfectly calibrated classifier satisfies:

$$P(y = c \mid \hat{p}(y = c \mid x) = p) = p, \quad \forall\, p \in [0, 1].$$

In practice, modern neural networks and gradient-boosted models are often miscalibrated: neural networks tend to be overconfident (the predicted probability of the top class exceeds the actual fraction of correct predictions), while some ensemble methods exhibit underconfidence

in certain class ranges [39, 38].

2.11.1 Isotonic Regression Calibration

Isotonic regression calibration is a non-parametric post-hoc method that maps raw predicted probabilities to calibrated probabilities by fitting a monotone step function to validation-set outcomes [39]. Given validation pairs $(\hat{p}_i, y_i)$, isotonic regression finds the non-decreasing function f minimizing $\sum_i (f(\hat{p}_i) - y_i)^2$. The resulting mapping is applied to test-set predictions. Isotonic regression can correct any systematic miscalibration that is monotone with respect to the original score, but it requires a sufficiently large validation set to avoid overfitting the step function.

2.11.2 Temperature Scaling

Temperature scaling is a parametric calibration method that divides the logits of a neural network by a learned scalar temperature $T > 0$ before the softmax operation [38]:

$$\hat{p}_{\text{cal}}(y = c \mid x) = \frac{\exp(z_c/T)}{\sum_{c'} \exp(z_{c'}/T)},$$

where z_c is the logit for class c. When $T > 1$, the softmax distribution is softened (predictions become less confident); when $T < 1$, it is sharpened. The temperature is optimized by minimizing the negative log-likelihood on the validation set, keeping all other model parameters fixed. Temperature scaling preserves the ranking of classes (argmax is unchanged) and requires only a single parameter, making it easy to apply and resistant to overfitting. It has been shown to be among the most consistently effective post-hoc calibration methods for deep classifiers [38].

Chapter Summary

This chapter established the technical foundations for the thesis. It reviewed welding processes, the physical formation of the defect classes studied here, and the quality criteria used to judge them; the complementary thermal and acoustic sensing modalities and the rationale for combining them through multimodal fusion; the class-imbalance problem and the evaluation metrics (macro-F1, per-class F1) it motivates; and real-time inference as a first-class design requirement. It then covered the machine-learning components used by the model families—gradient-boosted trees and random forests, convolutional neural networks and transfer learning, audio signal-processing features, neural-network training and regularization, and probability calibration. These foundations underpin the model designs in Chapter 4 and the evaluation protocol applied throughout.

Chapter 3

Related Work

3.1 Thermal and Vision-Based Monitoring

Thermal and vision-oriented monitoring approaches are well-represented in the welding monitoring literature because they provide direct access to weld-pool geometry, heat distribution, and visible process irregularity. Earlier studies emphasized process observation and rule-based interpretation, whereas later work introduced statistical classifiers and data-driven models for defect-related inference [1, 9]. The main advantage of this family is the richness of spatial information: images can capture local concentration of heat, asymmetry, and transient changes that are difficult to reconstruct from scalar process signals.

The main weakness of thermal-only monitoring is sensitivity to acquisition conditions. Camera angle, emissivity variation, smoke, reflections, and surface condition can alter observed intensity even when the physical process remains similar [9]. Reviews of robotic and hybrid welding monitoring therefore emphasize that sensor placement and calibration strategy are critical determinants of external validity [2, 5]. In other words, thermal sensing provides relatively rich spatial information about the weld process, but its transfer behavior across different acquisition conditions is not guaranteed.

Among specific thermal monitoring studies, Zhang et al. investigated real-time weld quality monitoring using infrared sensing and demonstrated that thermal features can capture pool geometry changes associated with penetration variation [1]. Alfaro and Franco reported on infrared thermography applied to GTAW, showing that relative temperature distribution provides useful information even when absolute calibration is imperfect [9]. More recent reviews by Tessema et al. and Curiel et al. survey a broader range of imaging modalities for robotic and hybrid welding, confirming the widespread use of thermal and visual data while also highlighting persistent challenges in sensor placement and environmental robustness [5, 2].

A common limitation across these studies is that evaluation is typically performed under controlled laboratory conditions with limited dataset diversity. Few thermal monitoring studies report performance under varying ambient conditions, sensor viewpoints, or welding process parameters. This makes it difficult to assess the external validity of reported accuracy figures.

The application of deep learning to thermal and visual welding monitoring has accelerated

in recent years. Convolutional neural networks have been applied to pool image sequences for penetration prediction, bead geometry estimation, and defect classification. The underlying motivation is that CNNs can learn spatially discriminative representations directly from raw image data, bypassing the need for hand-engineered geometric descriptors [23, 24]. Surveys of visual-based defect detection in industrial settings confirm that CNN-based approaches consistently outperform classical feature-engineering methods when sufficiently large labeled datasets are available, but highlight that data scarcity remains the primary barrier to adoption in specialized manufacturing contexts [29].

3.2 Audio-Only Monitoring Approaches

Audio-driven approaches model welding quality through waveform and time-frequency signatures of the arc process. Existing studies show that acoustic features can capture information relevant to penetration state, process stability, and defect-related behavior [10, 11, 12]. Because microphones are inexpensive and easy to integrate, the modality is suitable for scalable monitoring and for environments in which optical sensing is difficult to maintain.

The limitation of audio-only approaches is that they are highly context dependent. Background machinery, structural resonance, shielding-gas flow, and microphone placement can all influence the observed signal. As a result, careful preprocessing and well-chosen feature design are important, and transfer across cells or process variants remains difficult [11, 12]. These studies suggest that acoustic sensing is valuable, but rarely sufficient in isolation for robust industrial deployment.

Cui et al. used arc sound signals for weld penetration assessment and showed that spectral features can distinguish penetration states under controlled GMAW conditions [10]. Jiao et al. extended this line of work to GMAW arc sound quality monitoring, reporting that time-frequency features combined with machine learning classifiers can separate process states with moderate accuracy [11]. Griffin et al. investigated acoustic emission signals for MAG welding defect detection, demonstrating that acoustic features capture information about energy release events associated with defect formation [12].

These studies collectively show that acoustic monitoring is feasible, but several methodological concerns limit the strength of the evidence. Most studies evaluate on a single welding setup with limited defect diversity. Cross-environment validation is rarely reported, and the sensitivity of acoustic features to microphone placement and ambient noise is acknowledged but not systematically quantified.

A recurring finding in the audio-based welding literature is the importance of frequency band selection. Arc sound spectra contain both low-frequency components associated with power supply switching and droplet transfer events and high-frequency components associated with arc plasma instability. The relative importance of these bands varies by welding mode (spray transfer, short-circuit transfer, globular transfer), which limits the generalizability of feature sets tuned to one process variant. This reinforces the case for learning-based representations that can adapt the effective frequency weighting to the target task, rather than relying on fixed feature definitions.

3.3 Multimodal Monitoring in Welding Diagnostics

The motivation for multimodal fusion is well established: sensing channels that fail under different conditions can compensate for one another when combined appropriately [15]. In welding diagnostics, multimodal work has combined acoustic, visual, thermal, and electrical signals to improve monitoring robustness. For example, Wu et al. fused visual and acoustic information for penetration monitoring in variable-polarity plasma arc welding (VPPAW) and demonstrated that cross-modal integration can improve discriminative performance relative to single-sensor analysis [16]. More recent industrial monitoring studies similarly argue for sensor combination when stable quality control is required in the presence of process variability [4, 5].

At the same time, multimodal welding literature remains methodologically uneven. Some studies emphasize process-state identification, some focus on a narrow defect subset, and many report only one sensor pairing or one model family. This heterogeneity makes it difficult to infer whether reported gains arise from the sensing combination itself, from model capacity, or from dataset-specific evaluation conditions.

Beyond sensor-level combination, several recent studies have applied deep learning architectures to welding inspection. Bacioiu et al. demonstrated that a convolutional vision pipeline can automate defect classification in TIG welding from camera images with competitive accuracy relative to manual inspection [40]. Stemmer et al. extended multimodal DL to the unsupervised setting, combining audio and video streams on a large robotic welding dataset without requiring class labels during training [41]. Broader surveys of visual industrial inspection confirm that CNN-based architectures outperform classical feature-engineering methods when sufficient labeled data are available, but highlight data scarcity as a persistent barrier in specialized manufacturing settings [29]. Despite this progress, systematic comparisons between fusion strategies under matched experimental conditions remain uncommon.

In adjacent manufacturing domains, multimodal sensor fusion has been studied for tasks such as tool wear monitoring in machining, surface quality prediction in additive manufacturing, and process state recognition in forming operations [30, 42]. These studies reinforce the general finding that sensor combination can improve monitoring robustness, but they also show that the benefit of fusion depends on the quality of synchronization, the complementarity of modalities, and the evaluation protocol used. Zhao et al. provide a comprehensive survey of deep learning applications to machine health monitoring, demonstrating that CNN-based architectures applied to vibration spectra and acoustic emission signals achieve state-of-the-art results in fault diagnosis tasks while also highlighting the importance of domain adaptation when source and target operating conditions differ [30]. This observation is directly relevant to welding monitoring, where sensor conditions can change between production campaigns or welding configurations.

The work by Stemmer et al. on unsupervised audio-video welding defect detection is among the most recent multimodal contributions and demonstrates potential advantages of richer multimodal observation [41]. However, the unsupervised formulation differs substantially from the supervised classification task addressed in the present thesis, limiting

direct comparison.

Table 3.1: Summary of selected prior studies on welding and industrial defect monitoring. Scale uses consistent size bands: Small (<100 samples), Medium (100–500), Large (>500). Best Metric is reported in the study's own terms; NR = not reported. Split Strategy: Grouped = group-disjoint partitioning; Random = i.i.d. split; NR = not reported; Survey = no experimental split.

Reference	Year	Modalities	Fusion	Defect Types	Scale	Best Metric	Split Strategy	RT?
Zhang et al. [1]	2008	Thermal	Single	Multiple	Medium	Accuracy (NR)	NR	Yes
Alfaro & Franco [9]	2010	IR thermal	Single	Penetration	Small	Qualitative	NR	No
Wu et al. [16]	2017	Visual + audio	Late fusion	Penetration	Small	Accuracy	NR	No
Bacioiu et al. [40]	2019	Vision	Single	TIG defects	Small	Accuracy	Random	No
Hamzeh et al. [4]	2020	Multi-sensor	Feature fusion	Process states	Medium	Quality index	NR	No
Cui et al. [10]	2020	Audio	Single	Penetration	Small	Accuracy	NR	No
Zhao et al. [30]	2019	Vibration / AE	Single	Machinery faults	Survey	Accuracy	Survey	No
Czimmermann et al. [29]	2020	Vision	Single	General mfg.	Survey	Survey	Survey	No
Jiao et al. [11]	2023	Arc sound	Single	Process states	Medium	F1 / Accuracy	NR	No
Griffin et al. [12]	2023	Acoustic emis.	Single	MAG defects	Small	Detection rate	NR	No
Stemmer et al. [41]	2024	Audio + video	Unsupervised	Multiple	Large (>4000)	Anomaly score	Grouped	No
Present thesis	2026	Thermal + audio	Multi-family	11 defect types	Large (3841)	Macro-F1 + latency	Grouped	Yes

Table 3.1 summarizes selected prior studies discussed in this chapter. The comparison highlights that few prior works evaluate multiple fusion strategies under matched conditions, and real-time evaluation remains uncommon. The inclusion of adjacent manufacturing monitoring work (Zhao et al., Czimmermann et al.) shows that deep learning has demonstrated strong performance in related industrial inspection tasks, providing methodological precedent for the approaches used here.

Another pattern visible in the table is metric heterogeneity. Some studies report qualitative outcomes, some emphasize classification accuracy, and only a minority discuss class-balanced metrics or runtime behavior. This complicates direct comparison across papers and reinforces the need for a common evaluation framework when assessing competing fusion strategies.

3.4 Deep Learning for Industrial Inspection: Broader Context

The welding monitoring literature exists within a larger body of work on deep learning for industrial visual and sensor inspection [29, 30]. This broader context is relevant because it establishes the generalizability of methods and reveals patterns that transcend any single application domain.

Surveys of visual defect detection across manufacturing domains consistently identify several recurring themes [29]. First, data scarcity: labeled defect examples are rare in industrial production by design, because most production output should be defect-free. This creates fundamental class imbalance that must be addressed through specialized loss functions, oversampling, or data augmentation. Second, domain shift: a model trained on one production configuration may underperform on a nominally similar configuration due to changes in lighting, sensor wear, or material batch. Third, the gap between benchmark

performance and deployed performance: models evaluated on curated laboratory datasets often show degraded results when deployed in real production environments where additional variability and noise are present.

Deep learning architectures applied to industrial inspection have evolved from direct classification of raw images to more sophisticated pipelines incorporating attention mechanisms, self-supervised pretraining, and few-shot learning [29]. However, the majority of published work evaluates in controlled settings with clean, well-calibrated sensors and balanced class distributions. The present thesis deliberately operates under less idealized conditions by using an industrial dataset collected in a robotic welding cell with naturally occurring class imbalance and the noise characteristics of real arc welding.

Machine health monitoring in rotating machinery and manufacturing provides additional relevant methodological precedent [30]. Convolutional networks applied to vibration spectrograms and acoustic emission signals have demonstrated strong diagnostic performance for bearing faults, gear faults, and tool wear, confirming that time-frequency representations combined with deep feature learning are broadly applicable to process monitoring tasks. The key difference in the welding context is the combination of two qualitatively different sensing modalities (thermal imaging and arc sound), which requires explicit fusion design rather than treating the problem as single-modality classification.

3.5 Evaluation Practices and Data Partitioning in Industrial ML

A methodological concern that cuts across the reviewed literature is the quality of evaluation practices, particularly data partitioning and leakage control. In many published welding monitoring studies, the split strategy is either unreported or described only as a ratio (for example, 80% training and 20% test) without specification of whether samples from the same recording session, workpiece batch, or experimental configuration appear in both partitions [5, 4].

This omission matters because industrial datasets often contain correlated samples: multiple recordings from the same workpiece, the same operator, or the same session share systematic characteristics that are not present in a different session. A model that "learns" these session-specific characteristics achieves inflated test accuracy on within-session splits but may perform substantially worse when deployed on recordings from new sessions. This is sometimes called session leakage or temporal leakage, and its severity depends on how strongly session identity correlates with the target label.

Proper leakage control requires that split boundaries respect the correlation structure of the data. In practice this means grouping recordings by session or configuration identity and assigning all recordings from a given group to exactly one partition. This grouped split strategy reduces the risk that the model's success reflects memorization of session-specific patterns rather than genuine defect discrimination. Among the studies reviewed, only Stemmer et al. [41] describe their data partitioning strategy in sufficient detail to assess leakage risk. The present thesis adopts a grouped split policy as a methodological requirement rather than an optional enhancement.

A related issue is the choice of evaluation metric. Standard accuracy is misleading in

the presence of class imbalance, as noted in Section 2.5. Many industrial inspection studies report only overall accuracy or the accuracy on the majority class, which can make a model that ignores rare defect types appear successful. Macro-averaged F1, which treats each class equally, and per-class F1, which reports discriminability for individual defect types, are more informative metrics for assessing monitoring system quality across the full defect taxonomy.

3.6 Evaluation Gaps in the Existing Literature

The reviewed literature leaves three methodological gaps directly relevant to this work. Most studies evaluate a single modality or fusion configuration without controlled cross-strategy comparison, so the effect of fusion choice on the binary-screening/multiclass-diagnosis tradeoff remains empirically unresolved: Cui et al. [10] and Jiao et al. [11] evaluate acoustic monitoring in isolation, while Wu et al. [16] demonstrate a multimodal benefit but do not compare early, intermediate, and late fusion strategies under a controlled common split. Data partitioning is routinely underreported—Table 3.1 makes this concrete: only Stemmer et al. and the present study document a grouped partitioning policy—leaving it unclear whether reported gains reflect genuine generalization or session-level leakage. The third gap is runtime: online welding supervision requires timely and stable inference [4, 6], yet quantitative latency and throughput measurements are absent from most reviewed studies, including those that discuss real-time requirements in principle [1]. Taken together, the reviewed studies suggest that thermal, acoustic, and multimodal approaches can each contribute to defect detection, but offer limited guidance on choosing among fusion strategies when recognition quality and real-time feasibility must be evaluated jointly.

3.7 Position of the Present Thesis

This thesis sits between application-driven welding monitoring studies and general multimodal learning research: it is not a process-engineering report, nor a general fusion architecture study. Its contribution is a controlled comparison of three fusion families on a common dataset, under a grouped split that enforces cross-configuration generalization, with a shared evaluation protocol covering both predictive quality and runtime feasibility.

What this framing makes explicit is a distinction that prior work tends to collapse: binary defect screening and multiclass defect diagnosis are different problems with different operational consequences, and offline recognition quality and recording-level feasibility are not the same criterion. Evaluating them separately—and jointly—is what allows the central question to be answered directly: whether the best offline multiclass model is also the right choice for real-time deployment.

Chapter Summary

The literature confirms that thermal, acoustic, and multimodal signals each carry defect-relevant information, and that deep learning has demonstrated strong performance in related manufacturing inspection tasks. What it does not provide is a controlled comparison of fusion

strategies under matched experimental conditions with explicit runtime evaluation—the gap that motivates the methodology developed in the chapters that follow.

Chapter 4

Methodology

4.1 Methodological Overview

This chapter defines the experimental methodology used to study real-time welding-defect monitoring from thermal and acoustic observations. The design follows three constraints emphasized in the literature: defect evidence is multimodal and process dependent, measurements are noisy and condition sensitive, and practical value depends on latency as well as classification quality [1, 9, 10, 11, 6].

The method is structured into five connected stages:

1. multimodal run-level data structuring,
2. leakage-aware split construction,
3. modality-specific feature extraction,
4. hierarchical binary-plus-multiclass learning,
5. recording-level real-time aggregation with runtime profiling.

This structure is intentional: the same data definition and split policy are shared across all model families, so differences in results can be attributed to fusion and learning choices rather than to differences in data handling.

4.2 Formal Problem Definition

Let the dataset be

$$\mathcal{D} = \{(r_i, x_i^{(a)}, x_i^{(t)}, y_i, g_i)\}_{i=1}^{N},$$

where r_i is the run identifier, $x_i^{(a)}$ is the audio stream, $x_i^{(t)}$ is the thermal/video stream, y_i is the class label, and g_i is the grouped split key. The dataset also distributes a process-sensor stream per run (pressure, CO_2 flow, feed rate, weld current, wire consumed, voltage), but the four models in this study (Hierarchical Ensemble, Backbone Fusion, WeldFusionNet, and WeldFusionNet-RT) all consume only the audio and thermal/video modalities. The process-sensor stream is retained as an extension path for future multimodal studies and is

not used as a model input in the reported results. This keeps the comparison aligned with the conventional audio–thermal weld-monitoring literature.

The class set is

$$C = \{00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11\},$$

where 00 denotes non-defect and the remaining codes denote defect categories.

Each run is mapped to a fused representation z_i and processed by a hierarchical predictor composed of:

$$f_b(z_i) \in \{0, 1\}, \tag{4.1}$$

$$f_m(z_i) \in C. \tag{4.2}$$

Here, f_b is a defect gate (normal vs defect), and f_m is a multiclass model used only when defect evidence is present.

The final decision rule is

$$\hat{y}_i = \begin{cases} 00, & \text{if } f_b(z_i) = 0, \\ \arg\max\limits_{c \in C \setminus \{00\}} p(c \mid z_i), & \text{if } f_b(z_i) = 1. \end{cases}$$

This gate-and-refine formulation is intended to approximate a common process-control workflow: first isolate potentially defective runs, then apply finer diagnosis only where needed.

4.3 Data Structuring and Leakage Control

The run is the primary analytical unit. All modalities are indexed by the same run identifier and merged only through explicit joins, with missing-modality flags preserved. This prevents silent row loss and keeps modality availability interpretable in downstream analyses.

To reduce leakage risk, splitting is performed on grouped identities (g_i) rather than on individual files. Concretely, recordings sharing a session or configuration context are assigned to the same partition. This preserves the integrity of the evaluation by preventing near-duplicate experimental conditions from appearing in both training and test sets.

4.3.1 Streamed Inference Timeline

For real-time replay, each run is converted into overlapping windows

$$\mathcal{W}_i = \{w_{i,1}, w_{i,2}, \ldots, w_{i,T_i}\},$$

with window duration Δ_w and stride Δ_s.

Window sizes and strides are specified in video frames rather than seconds, because the implementation operates directly on video-frame indices read from each recording's AVI container; the corresponding wall-clock duration depends on the native frame rate of

the recording (nominally 30 FPS for the dataset, with the model pipeline resampling to 25 Hz where necessary). For the Hierarchical Ensemble and Backbone Fusion, the replay protocol uses a 50-frame chunk (approximately 1.7 s at 30 FPS, or 2 s at 25 FPS) advanced by a 125-frame stride. For WeldFusionNet, the replay protocol uses a 250-frame chunk over which audio and 8 video frames are uniformly sampled, advanced by the same 125-frame stride. The abstract values $\Delta_w = 1.0\,\text{s}$ and $\Delta_s = 0.25\,\text{s}$ in the equation above describe a deployment-oriented design budget (one inference event approximately every 0.25 s); the frame-indexed replay implementation used in this thesis emits inference events at the wider 125-frame stride and therefore produces fewer events per recording than that abstract budget would suggest.

This windowing protocol applies differently across model families. For the Hierarchical Ensemble, Backbone Fusion, and WeldFusionNet, windowing is an *inference-only* operation: all three models are trained on full-run feature vectors, embeddings, or temporally-resampled tensors extracted from the complete recording, and the sliding window is applied only when replaying a recording in the real-time evaluation protocol. For WeldFusionNet-RT, windowing is part of *both training and inference*: the model is trained natively on 1-second input chunks (Section 4.5.4) with a 0.5 s stride, so the window size directly determines the input format seen during optimization.

4.4 Feature Representation Design

The feature design uses descriptors selected for analytical transparency and computational tractability, with preference for representations that remain stable under moderate noise. Feature-based representations remain a standard strategy in time-series classification when compact summaries and efficient inference are required [43, 44].

4.4.1 Audio Features

Arc sound carries information about short-term energy fluctuation, spectral spread, and process instability; related welding studies use waveform and time-frequency characteristics for penetration and defect monitoring [10, 11, 12]. Following this line, audio is resampled to $f_s = 16\,\text{kHz}$ (mono), and frame-level descriptors are summarized at run level.

Let $y[n]$ denote waveform samples over an analysis segment of length N_s:

Signal magnitude statistics

$$\mu_{|y|} = \frac{1}{N_s}\sum_{n=1}^{N_s}|y[n]|, \qquad \sigma_{|y|} = \sqrt{\frac{1}{N_s}\sum_{n=1}^{N_s}(|y[n]| - \mu_{|y|})^2}$$

These terms capture global amplitude level and variability.

Root-mean-square energy

$$\text{RMS}(m) = \sqrt{\frac{1}{L}\sum_{n=0}^{L-1} y_m[n]^2}$$

RMS approximates short-time energy, useful for characterizing arc-power fluctuations.

Zero-crossing rate

$$\mathrm{ZCR}(m) = \frac{1}{2L} \sum_{n=1}^{L} |\operatorname{sgn}(y_m[n]) - \operatorname{sgn}(y_m[n-1])|$$

ZCR is used as a compact proxy for high-frequency/noisy behavior.

Spectral centroid

$$C(m) = \frac{\sum_k f_k |X_m[k]|}{\sum_k |X_m[k]|}$$

Centroid indicates where spectral mass is concentrated.

Spectral bandwidth

$$B(m) = \sqrt{\frac{\sum_k (f_k - C(m))^2 |X_m[k]|}{\sum_k |X_m[k]|}}$$

Bandwidth quantifies dispersion around the centroid.

Spectral rolloff

$$\sum_{k=0}^{k_r} |X_m[k]| = 0.85 \sum_k |X_m[k]|$$

Rolloff tracks the upper-frequency distribution boundary. The 0.85 cumulative energy threshold follows the default convention used in standard audio analysis libraries [14].

MFCCs

MFCC coefficients are computed per frame and aggregated by mean/std over time, consistent with established cepstral representations [13, 14].

Each MFCC coefficient is summarized at run level by seven statistics (mean, standard deviation, minimum, maximum, median, inter-quartile range, range). Spectral centroid, bandwidth, rolloff, and flatness are retained with mean and standard deviation each, to better distinguish structured arc harmonics from broadband or noisy regimes. Spectral contrast (7 bands) and chroma (12 pitch classes) are also retained with mean and standard deviation per band, and a harmonic–percussive source separation produces three additional scalar descriptors (mean harmonic energy, mean percussive energy, and their ratio) as a compact transient/steady-state discriminator.

4.4.2 Thermal/Video Features

Thermal imaging provides spatial evidence of heat concentration and heat-flow dynamics around the weld pool; prior work shows that these cues are informative but sensitive to emissivity and acquisition conditions [9, 1]. The baseline therefore uses robust distributional and temporal descriptors rather than absolute radiometric assumptions.

Frames are sampled every s_t frames (default $s_t = 12$ offline, $s_t = 2$ real-time). For grayscale frame $I_t(u, v)$:

Frame intensity statistics

$$\mu_t = \frac{1}{|\Omega|} \sum_{(u,v)\in\Omega} I_t(u,v), \qquad \sigma_t = \sqrt{\frac{1}{|\Omega|} \sum_{(u,v)\in\Omega} (I_t(u,v) - \mu_t)^2}$$

μ_t and σ_t summarize global heat level and spread.

Hot-region ratio

$$\rho_t = \frac{1}{|\Omega|} \sum_{(u,v)\in\Omega} \mathbb{I}\left(I_t(u,v) \geq P_{95}(I_t)\right)$$

This term measures concentration of high-intensity pixels relative to each frame distribution.

Inter-frame motion descriptor

$$\delta_t = \frac{1}{|\Omega|} \sum_{(u,v)\in\Omega} |I_t(u,v) - I_{t-1}(u,v)|$$

δ_t captures temporal variation in thermal structure.

To enrich spatial characterization, four additional frame-level quantities are computed: histogram entropy, mean gradient magnitude from Sobel filters, histogram spread based on the normalized $P_{90} - P_{10}$ intensity range, and edge density from Canny responses. These descriptors capture thermal disorder, boundary sharpness, dynamic range, and edge concentration without assuming calibrated temperature values.

For each run, mean/std/min/max over $\{\mu_t\}$, $\{\sigma_t\}$, $\{\rho_t\}$, $\{\delta_t\}$, entropy, gradient magnitude, histogram spread, and edge density are retained.

4.4.3 Process-Sensor Features

When auxiliary process channels are present, each numeric signal $s_j(t)$ is summarized as

$$\phi_j = \{\text{mean}(s_j), \text{std}(s_j), \min(s_j), \max(s_j)\}.$$

Process-sensor descriptors are not consumed by any of the four models in this thesis. All models (Hierarchical Ensemble, Backbone Fusion, WeldFusionNet, WeldFusionNet-RT) operate on audio and thermal/video evidence only. The handcrafted process-sensor features described above are retained in the on-disk feature inventory as an extension path for broader multimodal studies but were not part of any reported model input.

4.4.4 Feature Inventory

The merged representation used by the Hierarchical Ensemble contains 417 handcrafted features. Table 4.1 shows how each group reaches its count. The complete per-feature dictionary is reported in Appendix A.3.

4.5 Fusion Model Families and Learning Procedure

The study compares three fusion families under identical split conditions. Figure 4.1 provides a high-level overview of the complete pipeline shared by all approaches.

Table 4.1: Feature group breakdown for the 417-dimensional Hierarchical Ensemble input vector.

Group	Count	Derivation
Audio (195 total)		
MFCCs	140	20 coefficients × 7 summary stats (mean, std, min, max, median, IQR, range)
Spectral centroid, bandwidth, rolloff, flatness	8	4 descriptors × (mean, std)
Spectral contrast	14	7 frequency bands × (mean, std)
Chroma	24	12 pitch classes × (mean, std)
RMS, ZCR	4	2 descriptors × (mean, std)
Onset rate, tempo	2	scalar summaries (run-level)
Harmonic energy, percussive energy, H/P ratio	3	scalar summaries from HPSS decomposition
Thermal (214 total)		
Intensity stats (μ_t, σ_t)	8	2 descriptors × 4 stats (mean, std, min, max)
Hot-region ratio (ρ_t)	4	1 descriptor × 4 stats
Inter-frame motion (δ_t)	4	1 descriptor × 4 stats
Entropy, gradient, spread, edge density	32	4 descriptors × 8 stats (mean, std, min, max, p10, p90, skew, IQR)
Pool geometry (area, bbox, aspect ratio, centroid, symmetry, gradient, spatter)	166	7 descriptors × extended aggregation (mean, std, median, IQR, range, slope, segment stats, differencing)
Cross-modal interactions (8 total)		
Ratio terms	8	e.g. $\rho_{\text{hot}}/\text{RMS}$, $\nabla_{\text{thermal}}/B$, and 6 further combinations
Total	**417**	

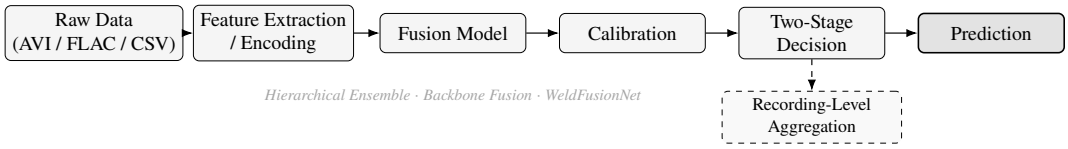

Figure 4.1: System pipeline overview: raw multimodal data flows through modality-specific feature extraction or encoding, a fusion model, post-hoc calibration, and a two-stage decision process with recording-level aggregation.

All three fusion families share the same two-stage decision logic, illustrated in Figure 4.2. Stage 1 screens for the presence of any defect (binary); stage 2 is invoked only when stage 1 flags a defect and diagnoses the specific defect class.

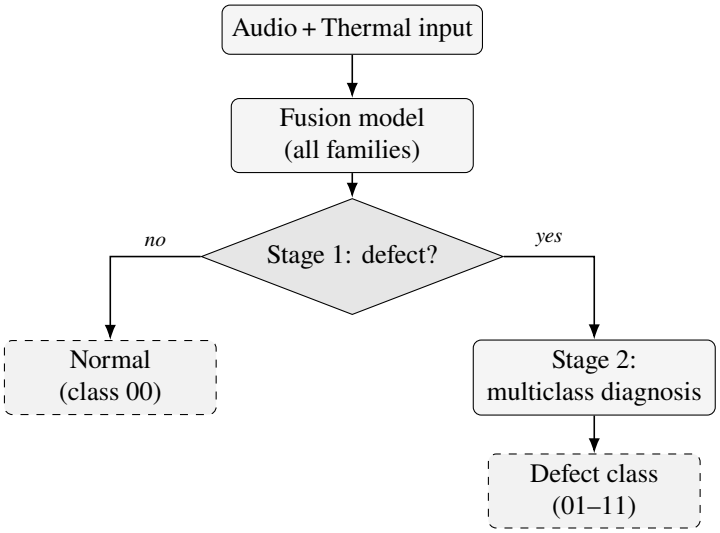

Figure 4.2: Two-stage decision structure shared by all three fusion families. Stage 1 performs binary defect screening; only recordings flagged as defective are passed to Stage 2 for multiclass diagnosis.

4.5.1 Hierarchical Ensemble

The Hierarchical Ensemble treats defect classification as a two-stage decision problem: first screen for the presence of any defect, then diagnose its type. This decomposition is motivated by the observation that the binary decision (good vs. defect) and the multiclass decision (which defect) have different class priors and benefit from separately tuned classifiers.

Feature representation. The input is a concatenated handcrafted feature vector

$$z_i = [z_i^{(a)} \| z_i^{(t)} \| z_i^{(\times)}] \in \mathbb{R}^{417},$$

where $z_i^{(a)} \in \mathbb{R}^{195}$ contains audio spectral descriptors (20 MFCCs with 7 summary statistics each; spectral centroid, bandwidth, rolloff, and flatness with mean and std; spectral contrast over 7 bands with mean and std; 12 chroma coefficients with mean and std; RMS and ZCR with mean and std; onset rate, tempo, and harmonic–percussive energies and ratio), $z_i^{(t)} \in \mathbb{R}^{214}$ contains thermal pool geometry descriptors (per-frame weld pool area, bounding box, aspect ratio, centroid, left-right symmetry, Sobel-based thermal gradient, and spatter count, each aggregated with mean, std, median, IQR, range, temporal slope, segment statistics, and differencing features), and $z_i^{(\times)} \in \mathbb{R}^8$ contains cross-modal interaction terms (e.g., $\rho_{\text{hot}}/\text{RMS}$, $\nabla_{\text{thermal}}/B$).

Stage 1: Binary screening. A weighted ensemble of gradient-boosted trees (XGBoost) and randomized trees (ExtraTrees) predicts the binary defect probability:

$$\hat{p}_i^{\text{bin}} = w_{\text{xgb}}\, \hat{p}_{\text{xgb}}(z_i) + w_{\text{et}}\, \hat{p}_{\text{et}}(z_i),$$

with $w_{\text{xgb}} = 0.6$, $w_{\text{et}} = 0.4$. XGBoost uses 700 estimators, max depth 5, learning rate 0.035, subsample 0.85, and automatic `scale_pos_weight` for class imbalance. ExtraTrees uses 900 estimators with `balanced_subsample` class weighting [22, 20]. All numerical hyperparameters (estimator counts, learning rate, subsampling ratio, and ensemble weights) were selected by grid search evaluated on the validation macro-F1; the selection procedure is described in Section 4.12.1. Isotonic regression calibrates $\hat{p}_i^{\text{bin}}$ on validation predictions to correct for miscalibrated probabilities [39].

Stage 2: Defect-type diagnosis. A second ensemble of the same structure is trained *only on defect samples* (i.e., runs where $y_i \neq 00$), performing 11-class classification over $C \setminus \{00\}$. Training on defect-only data allows the type classifier to focus discriminative capacity on inter-defect boundaries rather than the dominant good-vs-defect distinction.

Stage 3: Specialist reclassification (optional). After training, the validation confusion matrix is inspected. If any class pair (c_a, c_b) exhibits a confusion rate exceeding 30%, a one-vs-rest binary classifier is trained for the confused class and its output is blended into the type probabilities:

$$p'(c_a \mid z_i) = (1 - \alpha)\, p(c_a \mid z_i) + \alpha\, p_{\text{spec}}(c_a \mid z_i),$$

with blending weight $\alpha = 0.35$. If no pair exceeds the threshold, this stage is skipped.

Threshold selection. The binary threshold τ_b^* is chosen to maximize defect F1 on the validation set via grid search over $[0.20, 0.80]$ with step 0.01. The same grid is shared across all model families (see Section 4.6.1) so that threshold calibration is comparable.

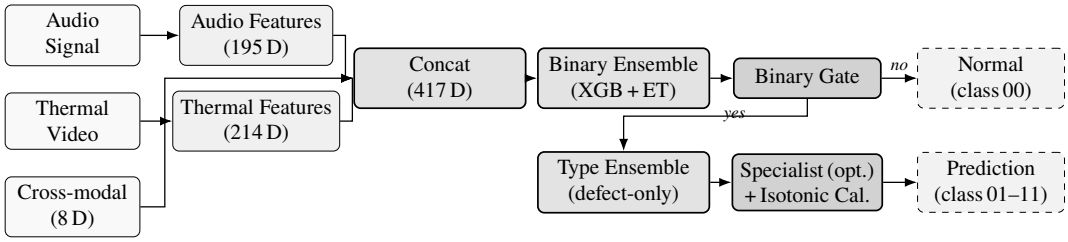

Figure 4.3: Hierarchical Ensemble architecture: handcrafted audio, thermal, and cross-modal features are concatenated into a 417-dimensional vector. A binary ensemble (XGBoost + ExtraTrees) screens for defects; positive cases are passed to a type ensemble trained on defect-only samples, with optional specialist reclassification and isotonic calibration.

4.5.2 Backbone Fusion

Backbone Fusion leverages a pretrained deep network as a fixed visual feature extractor, replacing handcrafted thermal descriptors with learned representations. The motivation is that models pretrained on large-scale image datasets (ImageNet) capture general perceptual features that transfer to downstream visual tasks with limited domain-specific data [25, 24]. For the audio modality, no welding-specific pretrained acoustic backbone is available, so the audio path uses a log-mel spectrogram projected to a fixed-dimensional embedding via a seeded random projection (described below).

Video embedding extraction. For each run, $K = 12$ frames are sampled uniformly from the thermal video. Each frame is resized to 224×224 and normalized with ImageNet statistics. A ResNet18 backbone [25], pretrained on ImageNet with the final fully connected layer removed, maps each frame to a 512-dimensional vector:

$$e_k = \phi_{\text{ResNet18}}(\text{frame}_k) \in \mathbb{R}^{512}, \quad k = 1, \ldots, K.$$

The frame-level embeddings are aggregated by concatenating their mean and standard deviation across the K frames:

$$h_i^{(v)} = [\text{mean}(e_1, \ldots, e_K) \| \text{std}(e_1, \ldots, e_K)] \in \mathbb{R}^{1024}.$$

Audio embedding extraction. The audio waveform is converted to a log-mel spectrogram with 128 mel bands at 16 kHz. The spectrogram is mean-pooled over the time axis and projected to a 1024-dimensional space via a fixed random projection matrix (seeded for reproducibility):

$$h_i^{(a)} = W_{\text{proj}} \cdot \text{mean}_t\big(\text{MelSpec}(x_i^{(a)})\big),$$
$$h_i^{(a)} \in \mathbb{R}^{1024}.$$

Modality-specific classifiers. Each embedding is independently classified by a multilayer perceptron (MLP):

$$p_a(c \mid x_i^{(a)}) = \text{MLP}_a(h_i^{(a)}) : \ 1024 \to 256 \to 64 \to |C|, \tag{4.3}$$
$$p_v(c \mid x_i^{(t)}) = \text{MLP}_v(h_i^{(v)}) : \ 1024 \to 512 \to 256 \to 64 \to |C|, \tag{4.4}$$

where each hidden layer uses BatchNorm, ReLU, and dropout ($p = 0.3$ for the first layer and $p = 0.2$ for subsequent layers). Both classifiers are trained with AdamW (lr=10^{-3}, weight decay=10^{-4}) and class-weighted cross-entropy loss. Optimization uses ReduceLROnPlateau scheduling (patience=5, factor=0.5) and early stopping on validation macro-F1 (patience=10).

Weighted probability fusion. The modality posteriors are combined by a convex mixture:

$$p(c \mid x_i) = \alpha \, p_a(c \mid x_i^{(a)}) + (1 - \alpha) \, p_v(c \mid x_i^{(t)}),$$

where α is determined by grid search over $[0, 1]$ with step 0.05, maximizing the composite score $S = 0.6 \cdot \mathrm{F1}_{\mathrm{bin}} + 0.4 \cdot \mathrm{F1}_{\mathrm{macro}}$ on the validation set. The selected weight is $\alpha = 0.25$, indicating a preference for the video modality.

Temperature scaling. After fusion, a single scalar temperature T is learned to minimize negative log-likelihood on the validation set [38]:

$$p_{\mathrm{cal}}(c \mid x_i) = \mathrm{softmax}\left(\frac{\log p(c \mid x_i)}{T}\right).$$

This corrects overconfident predictions without changing the ranking of classes.

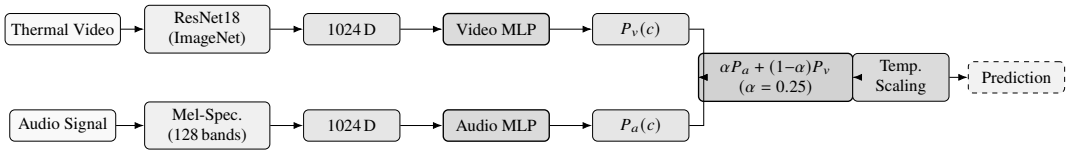

Figure 4.4: Backbone Fusion architecture: thermal video frames are encoded by a pretrained ResNet18 and audio by log-mel spectrogram projection, each producing 1024-dimensional embeddings. Modality-specific MLP classifiers produce posterior probabilities that are combined by weighted fusion ($\alpha = 0.25$ audio, 0.75 video) and calibrated via temperature scaling.

4.5.3 WeldFusionNet

WeldFusionNet is an end-to-end neural network that jointly learns modality-specific encoders and a shared fusion representation, eliminating the need for handcrafted feature engineering. Unlike the previous two approaches, WeldFusionNet operates on temporally aligned raw signals rather than precomputed summary statistics, allowing the model to discover task-relevant temporal patterns directly from data.

Input preparation. Each run is converted into a synchronized fixed-shape multimodal tensor over two modalities: audio and thermal video. The audio stream is represented as per-frame spectral features $(T, C_a) = (25, 44)$ where the 25 timesteps are uniformly distributed over the trimmed recording duration and the 44 channels comprise 13 MFCCs, their first- and second-order time derivatives (13 + 13), plus RMS, spectral centroid, spectral bandwidth, zero-crossing rate, and spectral rolloff. The video stream provides $F = 8$ uniformly sampled frames at 224×224 resolution with ImageNet normalization. Z-score normalization (fit on training set only) is applied to the audio input; the video input is normalized with ImageNet statistics. WeldFusionNet therefore observes the entire recording in a single forward pass through two encoder branches; WeldFusionNet-RT (Section 4.5.4) reuses the same two-encoder architecture but operates instead on 1-second windows of 25 timesteps at 25 Hz.

Modality encoders. The audio stream is processed by a 1D convolutional encoder consisting of two successive convolutional blocks followed by adaptive average pooling and a fully

connected projection layer. Each convolutional block applies a 1D convolution (with 64 output channels), batch normalization [32], and a ReLU nonlinearity. The first block uses a kernel of width 5 to capture medium-range temporal patterns; the second uses a kernel of width 3 for finer-grained refinement. Adaptive average pooling then reduces the temporal dimension to a single vector, and a fully connected layer projects the result to a 128-dimensional audio embedding $h^{(a)} \in \mathbb{R}^{128}$.

The video encoder uses MobileNetV3-Small [26] pretrained on ImageNet, with the first 50% of feature layers frozen. Each of the $F = 8$ frames is processed independently to produce a 576-dimensional feature vector. A learned temporal attention mechanism weights the frame contributions:

$$\alpha_f = \frac{\exp(w^\top e_f)}{\sum_{f'=1}^{F} \exp(w^\top e_{f'})}, \qquad h^{(v)} = \mathrm{FC}\left(\sum_{f=1}^{F} \alpha_f \, e_f\right) \in \mathbb{R}^{256},$$

where $w \in \mathbb{R}^{576}$ is a learned attention weight vector and e_f is the MobileNetV3 output for frame f.

Gated fusion and classification heads. The audio and video embeddings are passed through learned modality gates (sigmoid-weighted scalar per modality) and concatenated into a 384-dimensional joint representation, which is passed through a fusion MLP with hidden layers totaling 256 units. Each hidden layer applies batch normalization, ReLU activation, and dropout (rate 0.5) [32, 33]. This produces a 256-dimensional fused representation h_i that encodes combined evidence from the two modalities. Two classification heads share this representation:

$$\hat{y}_{\text{type}} = \mathrm{FC}_{64\to 12}(h_i), \tag{4.5}$$

$$\hat{y}_{\text{bin}} = \sigma(\mathrm{FC}_{64\to 1}(h_i)), \tag{4.6}$$

yielding 12-class logits and a binary defect probability. The total parameter count is approximately 1.08M, of which roughly 0.80M are trainable; the frozen MobileNetV3 layers account for the remainder. The architecture also exposes an optional auxiliary thermal-statistics encoder branch (34-dimensional handcrafted thermal features projected to a 64-dimensional embedding), but this branch is disabled in the production checkpoint reported in Chapter 6; the production WeldFusionNet consumes only the audio and thermal-video modalities.

Multi-task focal loss. The training objective combines two losses with task-specific weights:

$$\mathcal{L} = \lambda_{\text{type}} \, \mathcal{L}_{\text{focal}} + \lambda_{\text{bin}} \, \mathcal{L}_{\text{BCE}},$$

with $\lambda_{\text{type}} = 0.7$ and $\lambda_{\text{bin}} = 0.3$. The focal loss [36] down-weights well-classified examples to focus learning on hard cases:

$$\mathcal{L}_{\text{focal}} = -\sum_{i=1}^{N} w_{y_i}(1 - p_i)^{\gamma} \log p_i,$$

where $p_i = p(y_i \mid h_i)$ is the predicted probability of the true class, w_{y_i} is the inverse-frequency class weight, and $\gamma = 2.0$ is the focusing parameter. When $\gamma = 0$, this reduces to standard cross-entropy; with $\gamma = 2$, a well-classified example with $p_i = 0.9$ contributes approximately $(1 - 0.9)^2 = 0.01$ times the per-example loss of standard cross-entropy, down-weighting easy examples by roughly two orders of magnitude. The binary head uses standard BCE with logits.

Training procedure. Optimization uses AdamW with cosine annealing [35]:

$$\eta_t = \eta_{\min} + \frac{1}{2}(\eta_{\max} - \eta_{\min})\left(1 + \cos\frac{t\pi}{T_{\max}}\right).$$

This subsection defines the production architecture and the abstract training objective. The exact optimizer settings, schedule lengths, augmentation policy, label smoothing, dropout rate, early-stopping criterion, and batch size used to fit the Chapter 6 checkpoint are reported as the canonical configuration in Section 4.12.3 and summarized in Table 5.4; numbers stated elsewhere in this section are descriptive shorthand and may differ from the production checkpoint settings.

Post-hoc calibration. After training, three sequential calibration stages are applied on the validation set: (i) temperature scaling to correct overconfident softmax outputs, (ii) class-prior rebalancing to adjust for training-deployment distribution mismatch by rescaling $p(c) \propto p_{\text{deploy}}(c)/p_{\text{train}}(c)$ and renormalizing, where p_{train} is the empirical training-set class frequency and p_{deploy} is a fixed prior estimated from the same training partition (it is *not* fitted on the test-label distribution; no test-set leakage is introduced), and (iii) confidence-gated reclassification, where if a specific class pair exhibits a confusion rate above 30% on the validation set, low-confidence predictions of one class are reassigned to the other.

4.5.4 WeldFusionNet-RT

WeldFusionNet-RT shares the identical encoder and fusion architecture as WeldFusionNet (Conv1D audio encoder, MobileNetV3-Small video encoder, gated fusion MLP, dual classification heads). The key difference is in the training regime: rather than training on full recordings and applying a sliding window only at inference time, WeldFusionNet-RT is trained *natively* on 1-second windows extracted with a 0.5-second stride. This reduces the full-recording-versus-window input mismatch that affects WeldFusionNet when its full-run-trained representation is applied to short windows at runtime. Other forms of distribution shift (cell-to-cell variation, drift in process parameters, weak recording-level supervision propagated to window labels) remain.

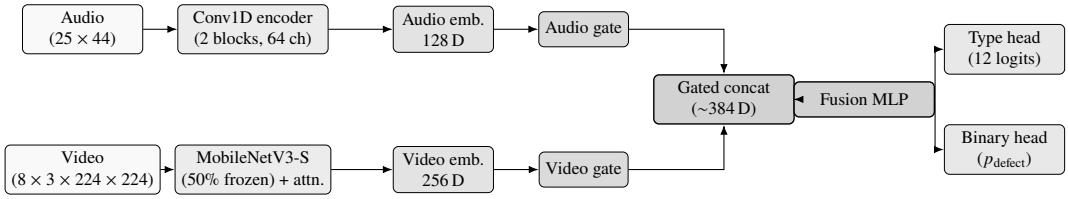

Figure 4.5: Production WeldFusionNet architecture (also used by WeldFusionNet-RT). The two models share the encoder, gated-fusion, and head structure but differ in training unit, split protocol, and evaluation unit (Section 4.5.4). The audio sequence (25 timesteps × 44 channels: 13 MFCC + 13 ΔMFCC + 13 Δ^2MFCC + RMS + spectral centroid + bandwidth + ZCR + rolloff) is encoded by a Conv1D module to a 128-dimensional embedding. Video (8 frames at 224 × 224) is encoded by MobileNetV3-Small with temporal attention to a 256-dimensional embedding. Per-modality gates apply learned sigmoid-weighted scalars before concatenation; the fused representation passes through a fusion MLP to dual classification heads (12-class type, 1-class binary defect). No process-sensor input is consumed.

Window-level training. The 1-second window matches the inference unit at deployment. Each window yields a $(25 \times C_a)$ audio tensor and $F = 8$ uniformly sampled video frames; the encoder and fusion architecture are identical to those of WeldFusionNet. The same multi-task focal loss is used, with $\lambda_{\text{type}} = 0.7$, $\lambda_{\text{bin}} = 0.3$, and $\gamma = 2.0$. Training ran for 43 of a 50-epoch budget using AdamW (lr=3×10^{-4}, weight decay=10^{-4}), 3-epoch warm-up, and cosine annealing; early stopping on the composite score $S = 0.6 \cdot \text{F1}_{\text{bin}} + 0.4 \cdot \text{F1}_{\text{macro}}$ (patience 10) halted training before the full budget. Batch size is 32 and dropout is 0.3; data augmentation includes additive Gaussian noise on the audio ($\sigma = 0.01$), pitch shift (±2 semitones, p=0.3), and brightness jitter on the video frames.

Evaluation protocol. WeldFusionNet-RT is evaluated on a config_id grouped split (not the strict cross-session split used for the three offline models). The test partition contains 751 runs, which are segmented into 64,221 windows. Each window is classified independently; no majority vote or recording-level aggregation is applied. Latency is measured as single-window GPU inference time at batch size 1 over 200 warm-started forward passes.

Scope limitation. Because WeldFusionNet-RT uses a different split criterion (config_id groups rather than session identity), its quality metrics are not directly comparable to the cross-session results in Chapter 6. It is included in the real-time comparison table (Table 6.4) as a representative of the latency benefit achievable when the model is trained for the streaming inference unit; cross-session generalization of this variant is identified as future work.

4.6 Two-Stage Calibration and Decision Logic

4.6.1 Binary Threshold Tuning

Given validation defect probabilities $\hat{p}_i$, the binary threshold is selected by

$$\tau_b^* = \arg \max_{\tau \in \{0.20, 0.21, \ldots, 0.80\}} \text{F1}_{\text{defect}}(\tau).$$

The same search grid is used for all model families to keep threshold calibration comparable.

4.6.2 Recording-Level Gate in Real-Time Evaluation

Let $\hat{p}_{i,t}$ denote event-level defect probability for run i at event t. The aggregated defect evidence is

$$\bar{p}_i = \frac{1}{T_i} \sum_{t=1}^{T_i} \hat{p}_{i,t}.$$

The recording-level binary decision is

$$\hat{b}_i = \mathbb{I}(\bar{p}_i \geq \tau_r),$$

with $\tau_r = 0.70$ in the current replay outputs: a recording-level defect alarm fires only when the mean event-level defect probability exceeds 0.70. This is a conservative choice, and the results are sensitive to it; a sweep over τ_r is left to future work.

If $\hat{b}_i = 0$, output is forced to `00`. Otherwise, the final defect class is selected by maximum mean posterior over $C \setminus \{00\}$.

4.7 Real-Time Runtime Profiling

Each emitted event stores measured inference latency

$$\ell_{i,t} = t_{i,t}^{\text{end}} - t_{i,t}^{\text{start}}.$$

Run-level summaries are

$$\ell_i^{(50)} = P_{50}(\{\ell_{i,t}\}), \quad \ell_i^{(95)} = P_{95}(\{\ell_{i,t}\}), \quad \bar{\ell}_i = \text{mean}(\{\ell_{i,t}\}).$$

Throughput is estimated as

$$\text{TPS}_i = \frac{1000}{\bar{\ell}_i}$$

(events per second).

Both central tendency (mean/P50) and tail behavior (P95) are reported, since tail latency is critical in interactive and monitoring systems [6].

4.8 Deployment Gate Formulation

The thesis uses a provisional deployment gate on recording-level metrics aligned with the 4 events/s design target ($\Delta_s = 0.25$ s scheduling interval implied by Section 4.3.1):

$$\mathrm{F1}_{\text{binary defect}} \geq 0.50 \tag{4.7}$$

$$\mathrm{Acc}_{\text{binary}} \geq 0.50 \tag{4.8}$$

$$\mathrm{MacroF1}_{\text{multiclass}} \geq 0.10 \tag{4.9}$$

$$\bar{\ell}_{\text{mean-of-means}} \leq 250\,\text{ms} \tag{4.10}$$

$$\ell^{(95)}_{\text{mean-of-means}} \leq 250\,\text{ms} \tag{4.11}$$

These thresholds serve as minimum viability checks rather than deployment-readiness criteria. The multiclass macro-F1 threshold of 0.10 is intentionally permissive: for a 12-class problem, uniform random prediction yields approximately 0.083, so the threshold filters only models that perform at or below chance level. The binary thresholds of 0.50 are intentionally permissive lower bounds: because the dataset is defect-majority (approximately four out of five runs are defective), a trivial always-defect predictor would reach binary F1 close to 0.89, so a 0.50 cut-off filters only models well below trivial baselines. Stronger binary-screening criteria — specificity, false-positive rate, false-negative rate, balanced accuracy, and missed-defect counts — are reported in Section 6.10 (Tables 6.12 and 6.13) and govern operational interpretation. The latency bounds reflect the design budget of a 4-events/s monitoring cycle ($\Delta_s = 250$ ms). Throughput is treated as a processing-capacity figure $1000/\bar{\ell}$ derived from event latency and is reported alongside latency in Chapter 6 (Table 6.4); it is not a separate gating criterion, because under a fixed event stride the actual output rate is determined by the stride, not by processing capacity. These values are not derived from industrial standards and should be interpreted as comparative filters within the present experimental setting, not as evidence of operational readiness.

4.9 Decision-Flow Summary

At recording level, each run is divided into overlapping windows. Each window's modality-specific evidence is fused and converted into class probabilities, and these event-level probabilities are aggregated across the recording. A recording is labeled non-defective when its aggregated defect evidence stays below the recording-level threshold; otherwise the final label is the defect class with the strongest mean posterior. Latency and throughput are measured alongside, so predictive quality and temporal feasibility can be read together.

4.10 Statistical Evaluation Protocol

Point estimates on a single held-out test partition do not, by themselves, support claims that one model is better than another. To accompany the headline metrics with uncertainty bounds and pairwise significance tests, three complementary procedures are applied to the predictions of every model on the same canonical test partition (§6.8):

1. **Bootstrap confidence intervals.** A common set of $B = 1000$ paired resample indices is drawn with `numpy.random.default_rng(seed=2026)`. Each resample produces a multinomial sample of the test partition with replacement, and macro F1, binary $\mathrm{F1}_{\mathrm{defect}}$, accuracy, and weighted F1 are recomputed per resample per model. The 2.5–97.5 percentile of the resampled distribution provides a 95% confidence interval. Using the same indices across models permits paired interpretation downstream.

2. **McNemar's test on the binary defect task.** For every model pair, a 2×2 contingency table is built from per-sample correctness on the binary defect label. Exact binomial McNemar is used when $b + c < 25$ and the asymptotic χ^2 approximation otherwise (STATSMODELS `mcnemar`).

3. **Paired bootstrap on the macro-F1 difference.** Using the same indices as procedure (1), per-resample Δ macro F1 values are computed for every model pair. The 2.5–97.5 percentile yields a paired 95% CI; an interval excluding zero is interpreted as a significant difference at $\alpha \approx 0.05$.

Bootstrap reproducibility is guaranteed by the fixed `rng=2026` seed; the full $1000 \times 4 \times 3$ raw resample matrix is persisted to `results/posthoc/bootstrap_ci.json` so that figures can be regenerated without rerunning the analysis.

In addition, calibration of each model's probability outputs is evaluated via Expected Calibration Error (ECE), Brier score, and Maximum Calibration Error (MCE) on 15 equal-width bins of the top-class confidence, separately for the multiclass and binary heads (§6.9). Five sanity baselines (majority class, stratified random, multinomial logistic regression on 1196 audio+thermal+video-embedding features, multinomial logistic regression on the 172 handcrafted audio+thermal features only, and k-nearest-neighbors on the 1196-feature input) are also trained on the canonical split and reported to bound how much the production architectures contribute beyond what the input features alone make achievable (§6.10).

4.11 Design Choices and Alternatives Considered

The three fusion approaches occupy different points in the design space of multimodal classification. This section gives the rationale for each and the alternatives that were excluded.

4.11.1 Why Three Fusion Families?

Comparing three distinct approaches, rather than committing to one, isolates the effect of the modeling assumption itself: each makes a different assumption about where predictive information resides in the multimodal signal.

The Hierarchical Ensemble is classical machine learning: handcrafted features feeding tree classifiers. It is interpretable—every prediction traces back to observable quantities such as spectral statistics and thermal pool descriptors—and cheap to train. Its limits are that hand-designed features may miss discriminative signal, and that per-window feature extraction makes real-time inference expensive.

Backbone Fusion takes the transfer-learning route: pretrained networks act as fixed feature extractors and only the classification heads are trained, avoiding end-to-end optimization on the relatively small welding dataset [28, 27]. It rests on one assumption—that ImageNet features transfer to thermal welding imagery—which is unproven in general but reliable for the low- and mid-level visual features that dominate here.

WeldFusionNet goes fully end-to-end: temporal encoders through fusion layer are optimized jointly, letting the model fit feature extraction directly to the welding data. The cost is overfitting risk, sharpest on the minority defect classes, countered with focal loss [36], multi-task learning, and regularization [33, 32].

4.11.2 Alternatives Not Included

Four alternatives were excluded, each for a specific reason.

Attention-based and transformer architectures. The short windows used here (1 s at 25 Hz, only 25 timesteps) give long-range attention little to exploit, and the extra capacity would need far more data than is available to train stably.

Recurrent architectures (LSTM, GRU). Their sequential computation is hard to parallelize and raises inference latency, and 1D CNNs with appropriate kernel sizes match them on fixed-length sequences while training faster [24].

Unsupervised and semi-supervised pretraining. Contrastive or masked-autoencoder pretraining on unlabeled recordings would confound the comparison: result differences would then reflect the pretraining stage rather than the fusion strategy under study.

Single-modality baselines. This study compares fusion strategies, not single-modality against multimodal. Audio-only and thermal-only baselines would add context but expand scope beyond the three research questions of Chapter 1; modality contribution is instead probed by the robustness analysis (Section 6.3.1), which removes individual sensing channels.

4.11.3 Feature Dimensionality and the 417-Feature Representation

The 417-dimensional feature vector was built iteratively, not fixed in advance. The starting set followed the audio feature-extraction literature [13, 14] and the thermal spatial-descriptor literature [9, 1]. Exploratory analysis on the training set then pruned descriptors with near-zero variance or near-perfect correlation to an existing feature, and kept those with class-discriminative patterns.

The eight cross-modal terms were added after ratio combinations of thermal hot-region intensity and acoustic energy proved to separate certain defect classes that neither modality separated alone. The reasoning is physical: high thermal concentration with low acoustic energy can signal a different failure mode than both channels elevated together. Precomputing these ratios lets the tree ensemble use cross-modal structure without having to discover it.

4.12 Training Configuration and Reproducibility

This section gives the training settings needed to reproduce each model family, complementing the architecture descriptions above.

4.12.1 Hierarchical Ensemble Training

The Hierarchical Ensemble requires no gradient-based training. Model fitting proceeds in four steps. First, the 417-dimensional feature matrix is constructed from the training split and z-score normalized using statistics computed from the training data only (mean and standard deviation per feature). Normalization statistics are persisted and applied to the validation and test splits without re-estimation, preventing leakage of test distribution information. Second, XGBoost is fit with early stopping on the validation set binary log-loss (patience of 50 rounds), which prevents overfitting to the training partition. Third, ExtraTrees is fit on the same normalized training data without early stopping (tree depth and number of estimators are fixed hyperparameters). Fourth, isotonic regression calibration is fit on the validation set predictions. This four-step procedure is fully deterministic given a fixed random seed and requires no GPU access.

Hyperparameters for XGBoost and ExtraTrees were selected by grid search over candidate values evaluated on the validation macro-F1. For XGBoost, the search covered estimator counts in $\{300, 500, 700, 900\}$, max depth in $\{3, 5, 7\}$, learning rate in $\{0.01, 0.035, 0.1\}$, and subsample in $\{0.7, 0.85, 1.0\}$. For ExtraTrees, the search covered estimator counts in $\{500, 700, 900\}$. The ensemble weights w_{xgb} and w_{et} were swept over $\{0.3/0.7, 0.4/0.6, 0.5/0.5, 0.6/0.4, 0.7/0.3\}$ using the same composite validation score ($S = 0.6 \cdot \text{F1}_{\text{bin}} + 0.4 \cdot \text{F1}_{\text{macro}}$) used for threshold selection. Because the feature dimension (417) is large relative to the training size (2676), a moderate tree depth (max depth 5 for XGBoost) was consistently favored by the search to prevent overfitting to rare class-feature combinations.

4.12.2 Backbone Fusion Training

The MLP classifiers in Backbone Fusion are trained in two phases. In the first phase, the ResNet18 and audio embedding extraction are run in inference-only mode to generate the embedding matrices for all training and validation examples. These embeddings are stored to avoid recomputing them at each training step. In the second phase, the MLP heads are trained on the stored embeddings using AdamW with weight decay 10^{-4}, class-weighted cross-entropy, and ReduceLROnPlateau scheduling (factor 0.5, patience 5 epochs). Early stopping is applied based on validation macro-F1 with patience 10 epochs. The best checkpoint is restored before evaluation.

Because the backbone weights are frozen throughout, the Backbone Fusion training procedure is fast and memory-efficient: the entire forward pass through the backbone runs once per epoch at most, and the MLP training operates on fixed-size embedding vectors. This also means that the temperature scaling calibration step must be performed on backbone-derived embeddings rather than on raw modality inputs.

4.12.3 WeldFusionNet Training

WeldFusionNet is trained end-to-end on all trainable parameters simultaneously. The production checkpoint reported in Chapter 6 uses the regularized training configuration in `src/configs/fusionnet_regularized.yaml`:

- **Optimizer:** AdamW with initial learning rate 3×10^{-4} and weight decay 5×10^{-4},
- **Schedule:** CosineAnnealingLR with $T_{\max} = 60$ epochs, 3-epoch linear warm-up, and $\eta_{\min} = 10^{-6}$,
- **Loss:** focal loss ($\gamma = 2.0$) with label smoothing 0.10 on the 12-class head plus binary cross-entropy on the binary head, combined as $\mathcal{L} = 0.7 \cdot \mathcal{L}_{\text{focal}} + 0.3 \cdot \mathcal{L}_{\text{binary}}$,
- **Batch size:** 32 (with automatic fallback to 8 when GPU memory is below 5 GB),
- **Max epochs:** 60, with early stopping on the validation composite score $S = 0.6 \cdot \text{F1}_{\text{bin}} + 0.4 \cdot \text{F1}_{\text{macro}}$ (patience 12),
- **Dropout:** 0.5; **label smoothing:** 0.10; **augmentation:** additive Gaussian noise ($\sigma = 0.03$) on the audio tensor,
- **MobileNetV3 frozen layers:** first 50% of feature layers,
- **Video frames per recording:** $F = 8$, uniformly sampled across the run.

Audio is rendered as a fixed-shape (25×44) tensor per recording (Section 4.5.3 input preparation), so no variable-length padding is needed at the batch level; thermal video is rendered as eight uniformly spaced frames per recording.

The multi-task learning setup requires that both classification heads see every training example, which simplifies the data pipeline compared to architectures that use separate data loaders for each head. The binary label for each run is derived from the 12-class label: runs with label 00 are mapped to binary class 0 (normal) and all other runs to binary class 1 (defect).

Chapter Summary

This chapter set out the methodology: run-level data definition with leakage control, multi-modal feature construction, and four fusion models—the three offline approaches (Hierarchical Ensemble, Backbone Fusion, WeldFusionNet) plus the real-time-native WeldFusionNet-RT—all evaluated under matched or comparable splits, hierarchical calibration, and latency-aware metrics. The design-choices section gave the rationale for each architecture, the excluded alternatives, and the construction of the 417-feature representation; the training-configuration section gave the settings needed to reproduce each model. The next chapters apply this methodology to measure predictive behavior and real-time feasibility on the current dataset snapshot.

Chapter 5

Experiments

5.1 Dataset Source and Scope

The experiments are conducted on the Intel Robotic Welding Multimodal Dataset, published by Intel Labs and associated with prior work on welding-defect analysis from audio-visual signals [45, 41]. The dataset was selected for this thesis because it combines synchronized sensing modalities, a diverse defect taxonomy, and annotations at a scale sufficient for binary screening evaluation and partial multiclass analysis. Its suitability for comprehensive multiclass evaluation is constrained by the uneven representation of certain defect categories in the test partition, as discussed below.

The published dataset description reports more than 4,000 annotated welding samples and multimodal streams (video, audio, sensor signals, and post-weld imagery) [45, 41]. This breadth supports a multi-model comparison under a unified protocol, although the uneven distribution of test samples across defect categories limits the reliability of class-level performance estimates.

Label provenance. The Intel Robotic Welding Multimodal Dataset card reports that the annotations represent the *intended* defect category for each weld, induced by configuring the welding process parameters, rather than expert-validated post-weld inspection of every recording. A single weld may therefore carry residual labeling noise or contain more than one defect type. All supervised models in this thesis learn the annotated/intended defect category, not necessarily an expert-confirmed physical defect for every individual run. This affects the absolute interpretation of class boundaries (especially for visually subtle defects such as class 05 undercut) but does not affect the comparative ranking among models trained and evaluated under the same labels.

Dataset license and use. The dataset is gated access under the Intel Research Use License Agreement and is restricted to research use. The deployment recommendations developed in Chapter 7 are framed as research-use modeling guidance; productionizing the trained checkpoints against this dataset would require separate licensing.

Frame-rate handling. The dataset card lists video at a nominal frame rate of approximately 30 FPS. This thesis resamples synchronized video to 25 Hz (25 FPS) to align with the audio frame-decimation pipeline and to keep the per-window timestep count fixed at $T = 25$ across modalities (Section 4.5.3).

Table 5.1: Summary of the Intel Robotic Welding Multimodal Dataset used in this thesis.

Property	Value
Dataset name	Intel Robotic Welding Multimodal Dataset
Provider	Intel Labs (Hugging Face listing)
Public link	Hugging Face dataset page
Access mode	Public dataset card with gated download access
Reported scale	More than 4,000 annotated welding samples
Modalities per sample	Video, synchronized audio, welding-sensor time series, post-weld images
Archive size (reported)	Approximately 39.9 GB compressed archive
Application scope	Welding defect detection and classification in industrial settings

5.2 Local Snapshot Used in This Thesis

All numerical results in this manuscript correspond to a single frozen local snapshot. The snapshot contains $N = 3841$ runs grouped under 226 session/configuration identities, with complete availability of the two primary channels used in this work: audio and thermal video. The auxiliary process-sensor stream is also present on disk for every retained run and could be used in future multimodal studies, but it is not consumed by any of the four models reported here. The published dataset card lists 236 session sub-directories; the local count of 226 reflects the modality-availability filtering described in the next paragraph (sessions with no modality-complete run were dropped from the grouping pool).

The difference between the published figure (>4,000 samples) and the snapshot count (3,841) reflects the filtering applied during data preparation: only runs for which all three primary modalities—video, audio, and sensor CSV—were confirmed present on disk were retained. Runs with at least one missing or unreadable file were excluded to prevent silent missing-data errors during feature extraction. The 159+ excluded entries represent either partially downloaded archive segments (the dataset is distributed as a gated 39.9 GB archive) or recordings for which one modality stream was absent in the source collection. This filtering was applied before any model training or split construction, so the 3,841-run count is the operative dataset throughout this thesis.

The dataset contains $N = 3841$ runs, split using session-disjoint partitioning (train=2676, val=385, test=780). The realized proportions are approximately 70/10/20 (69.7/10.0/20.3), and the grouping constraint ensures that no welding session appears in more than one partition. All 12 classes (00–11) appear in the test set.

Table 5.2: Observed split inventory for the local snapshot used in experiments.

Partition	Runs	Groups	Notes
Train	2676	157	Session-disjoint
Validation	385	23	Session-disjoint
Test	780	46	Session-disjoint
Total manifest	3841	226	Snapshot fingerprint locked

5.3 Label Taxonomy and Distribution

The active code set in the snapshot is `00-11`, where `00` denotes normal welds and the remaining codes denote defect categories. Table 5.3 reports the operational mapping and split-level counts.

Table 5.3: Label-code taxonomy and split-level run counts for the current snapshot.

Label Code	Label Name	Defect/Normal	Train Runs	Val Runs	Test Runs	Total Runs
00	good_weld	normal	588	72	159	819
01	excessive_penetration	defect	201	40	80	321
02	burn_through	defect	239	21	60	320
03	porosity	defect	240	40	60	340
04	porosity_with_excessive_penetration	defect	321	60	99	480
05	undercut	defect	120	20	20	160
06	overlap	defect	102	18	40	160
07	lack_of_fusion	defect	240	20	60	320
08	excessive_convexity	defect	81	19	20	120
09	spatter	defect	200	40	80	320
10	warping	defect	210	30	80	320
11	crater_cracks	defect	134	5	22	161

Figure 5.1 shows the class imbalance observed in the full snapshot, and Figure 5.2 confirms that the grouped split preserves broad code coverage while keeping the test partition fixed.

With the session-disjoint grouped split, all 12 classes are represented in the test partition. The grouping constraint (no session shared between partitions) is combined with class-aware sampling, which ensures coverage of every class across train, validation, and test sets and enables per-class evaluation across the full defect taxonomy. Although class imbalance remains (some defect categories have fewer runs than others), no class is absent from the test set.

5.4 Defect Class Characterization

Each of the eleven defect categories produces a distinct combination of thermal, acoustic, and process-sensor signatures that reflects its physical formation mechanism. Figure 5.3 illustrates the canonical multimodal profile for class 00 (Good Weld), which serves as the reference baseline: a symmetric, moderate-intensity thermal pool; a rhythmic audio waveform with flat RMS envelope; a spectrogram dominated by stable low-frequency energy; and all six sensor channels remaining within nominal operating ranges. Classes at the other extreme,

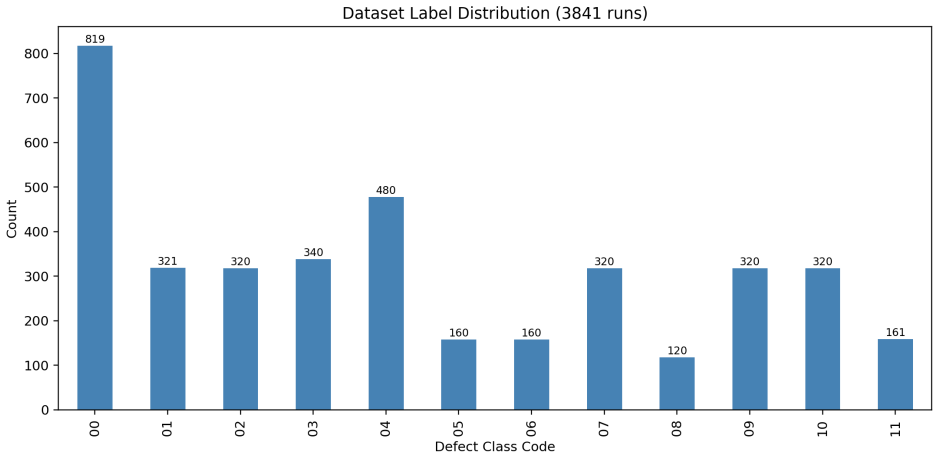

Figure 5.1: Run-count distribution by label code in the local snapshot (3841 runs).

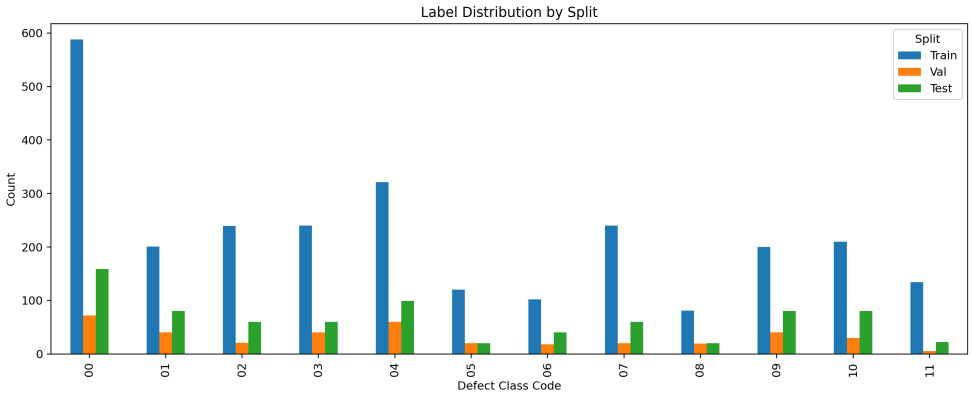

Figure 5.2: Train/validation/test label distribution for the grouped split protocol.

such as class 02 (Burn Through) and class 11 (Crater Cracks), exhibit abrupt transient events that are visible across modalities or—in the case of crater cracks—emerge only at the final arc-off instant, making the defect largely invisible in midpoint-sampled thermal features. The full per-class profiles, together with brief physical explanations for all twelve classes, are provided in Appendix A.8.

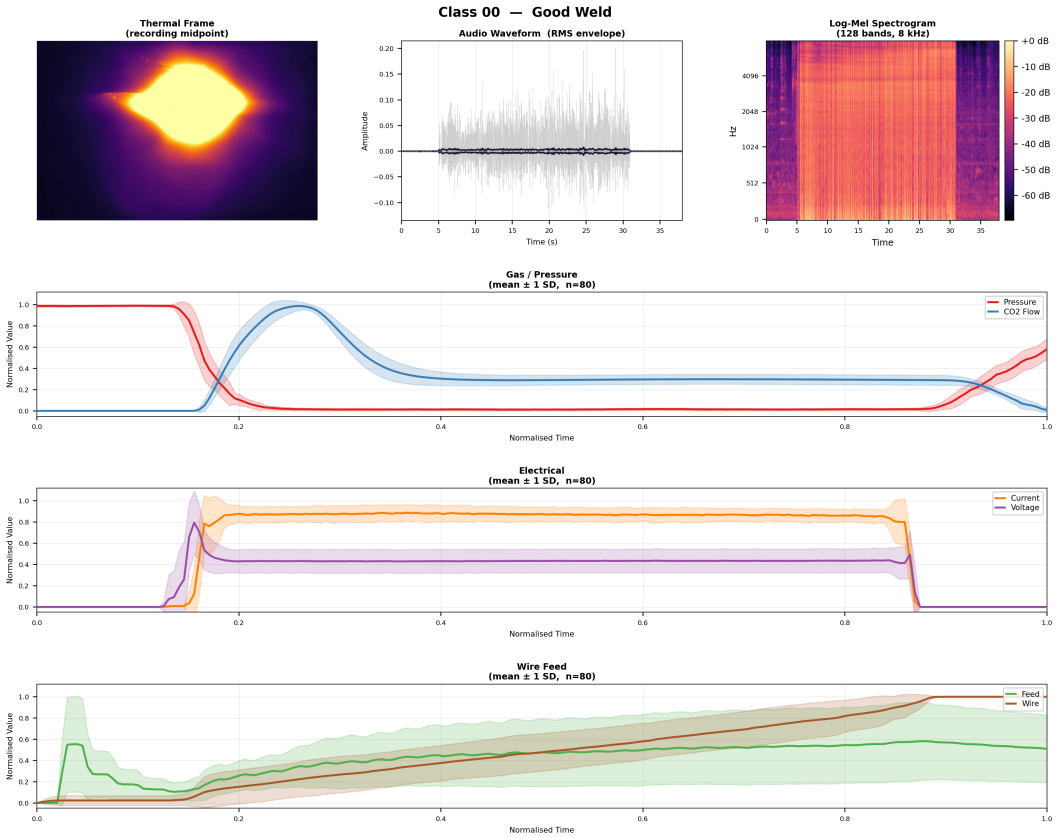

Figure 5.3: Multimodal profile for class 00 (Good Weld), used as the reference baseline. Top row (left to right): thermal frame at the recording midpoint (Inferno colormap), audio waveform with RMS envelope overlay, and log-mel spectrogram (128 bands, 8 kHz). Lower panels: class-averaged sensor traces (mean ± 1 SD) grouped by Gas/Pressure, Electrical, and Wire Feed channels.

5.5 Modality-Level Data Inspection

In addition to quantitative metrics, representative samples are inspected qualitatively to verify that each modality carries interpretable process information.

5.5.1 Thermal/Video Snapshot Examples

Figure 5.4 provides a side-by-side still-frame contrast between a representative normal weld and a Spatter run (class 09) at the frame of maximum scatter activity. Spatter is used as the reference defect throughout this section because it produces unambiguous deviations across all three modalities simultaneously: ejected molten particles visible in the thermal stream; impulsive acoustic bursts in the audio; and elevated gas pressure instability in the

sensor channels. Classes with more subtle signatures (Undercut, Warping, Crater Cracks) are discussed qualitatively in Appendix A.8.

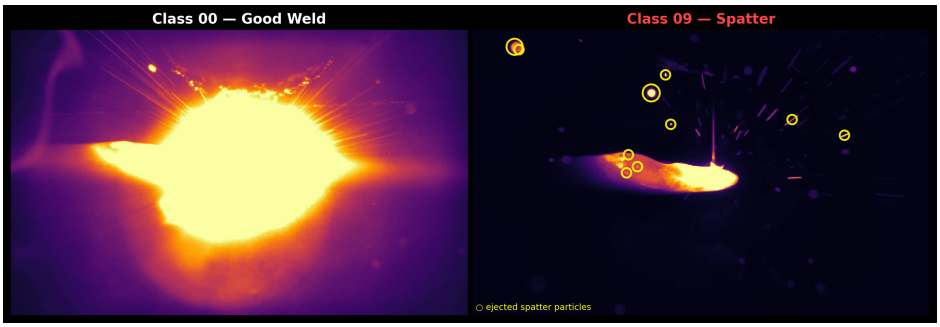

Figure 5.4: Side-by-side thermal frame comparison: normal weld (left, class 00) and Spatter (right, class 09) at the frame of maximum spatter activity. The normal pool is bright and stable. The Spatter frame shows molten droplets automatically detected outside the main weld pool via connected-component analysis on the 98.5th-percentile thermal threshold (circled in yellow) – these ejected particles cool rapidly and create the characteristic crackling audio and pressure instability captured by the other modalities.

Figure 5.5 extends this view to full sequences, making the weld-pool evolution and transient thermal events visible over normalized weld progress.

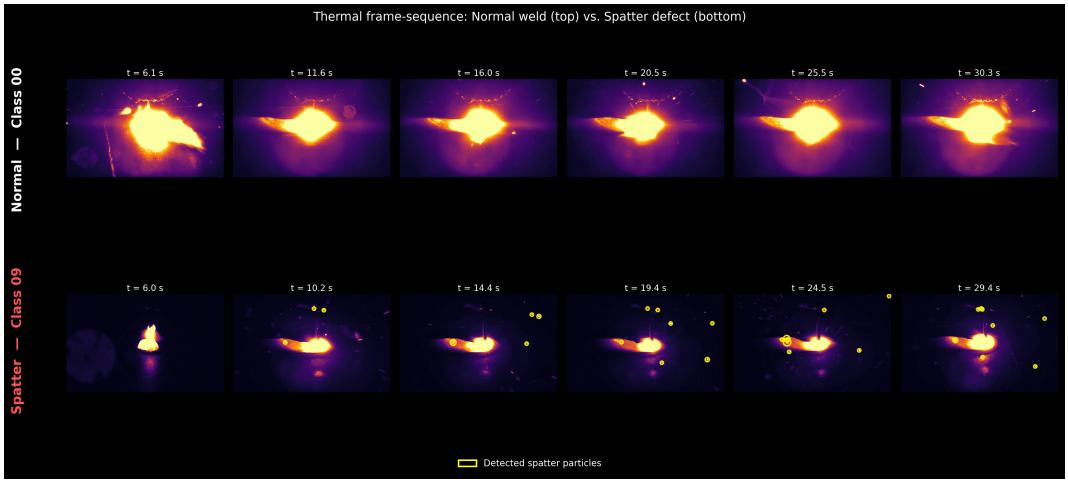

Figure 5.5: Thermal frame-sequence comparison restricted to the active-arc portion of each recording: normal run (top, class 00) and Spatter run (bottom, class 09) at six matched timestamps. The normal pool remains compact and stable across all six frames. The Spatter row shows recurring ejected hot particles (circled in yellow, detected automatically via connected-component analysis outside the main pool) appearing in every active-arc frame – the intermittent thermal signature that drives the impulsive acoustic bursts in the same recordings.

Figure 5.6 condenses the full frame sequence into two scalar time series per recording. The top row plots *mean frame intensity*: the average pixel brightness across the entire thermal image at each moment, which rises when the arc is active and falls when it terminates. The bottom row plots the *hot-pixel ratio*: the fraction of pixels that exceed a fixed high-temperature threshold, indicating how much of the frame is at extreme temperature at each instant. Both quantities are plotted against normalized weld progress (0 = start, 1 = end of recording),

so the arc-active phase appears as the central plateau bracketed by near-zero values at the recording margins.

The normal weld (blue, left) shows a stable plateau with modest, low-amplitude oscillations throughout the active arc phase. The Spatter run (red, right; class 09) shows sharp, isolated spikes in both mean intensity and hot-pixel ratio during the active phase — each spike corresponds to a molten droplet being ejected and briefly illuminating a wider area before cooling. This impulsive, non-stationary pattern is the hallmark of the spatter defect and is absent in the normal run. Both series are from individual recordings; the near-zero margins at each end reflect the pre-arc and post-arc periods.

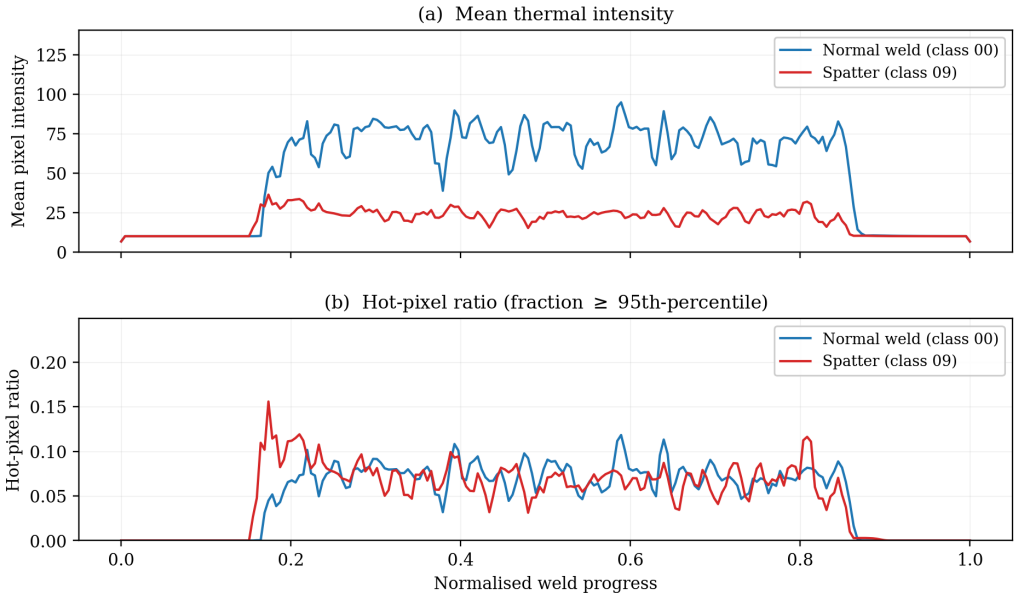

Figure 5.6: Thermal profile diagnostics with normal weld (blue) and Spatter run (red, class 09) overlaid for direct comparison. **Panel (a)**: mean pixel intensity per frame – the normal weld holds a high, stable plateau (~70–90) while the Spatter run operates at much lower mean intensity (~20–30) but is noticeably noisier. **Panel (b)**: hot-pixel ratio (fraction of pixels above the 95th-percentile threshold) – both runs share a similar baseline, but the Spatter run shows visibly larger frame-to-frame fluctuations corresponding to ejected hot particles.

Figure 5.7 presents representative thermal frames across all 12 classes (00–11), illustrating the visual diversity of welding conditions captured in the dataset. These frames are extracted from the middle of each recording and rendered with the Inferno colormap to highlight thermal intensity variations.

5.5.2 Audio Snapshot Examples

Figure 5.8 shows the single-run log-mel spectrogram for a representative normal (class 00) recording, providing a baseline reference for the time-frequency representation that Backbone Fusion uses as its audio input.

Figure 5.9 presents a multi-view audio diagnostic comparison using matched windows from curated normal and defect recordings. The waveform, RMS envelope, and zero-crossing trend are shown together to reveal both transient and steady-state differences.

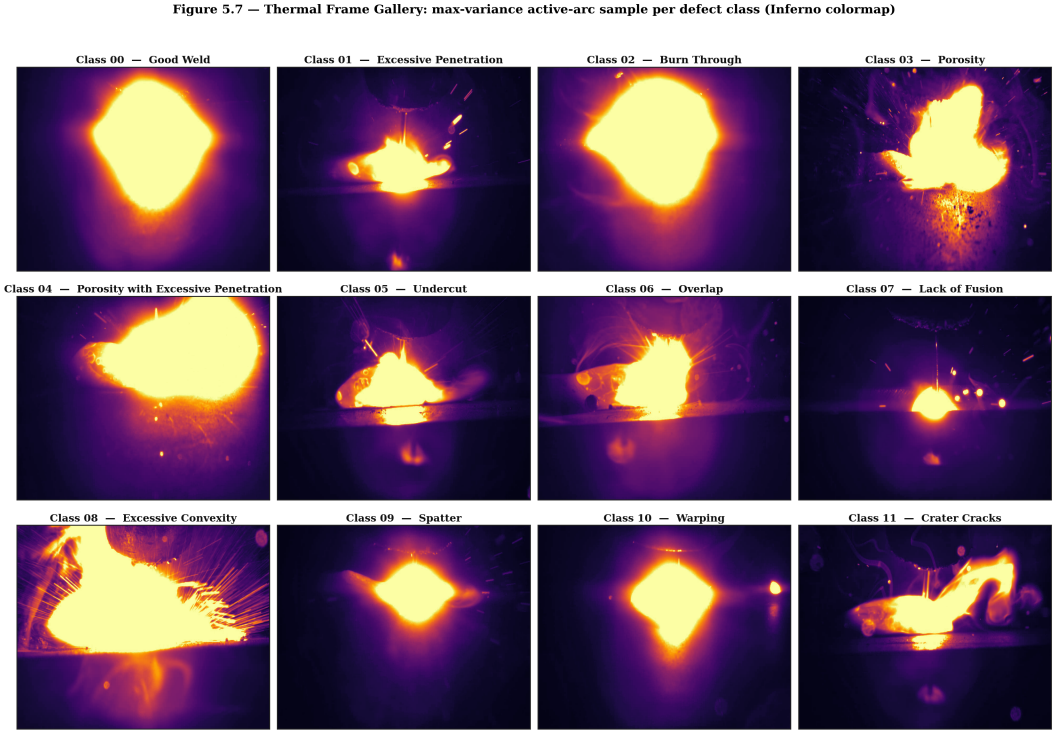

Figure 5.7: Thermal frame gallery showing one representative example per class (00–11). Colormap: Inferno. Each frame is selected as the maximum-spatial-variance frame within the active-arc portion of the recording – variance is high when the frame contains both bright pool and scattered bright/dark features, which captures the visual signature of the defect (ejected particles, asymmetric geometry, scattered hot spots) rather than just a uniform bright pool. Visual differences in pool geometry, heat distribution, and spatter patterns are evident across defect types.

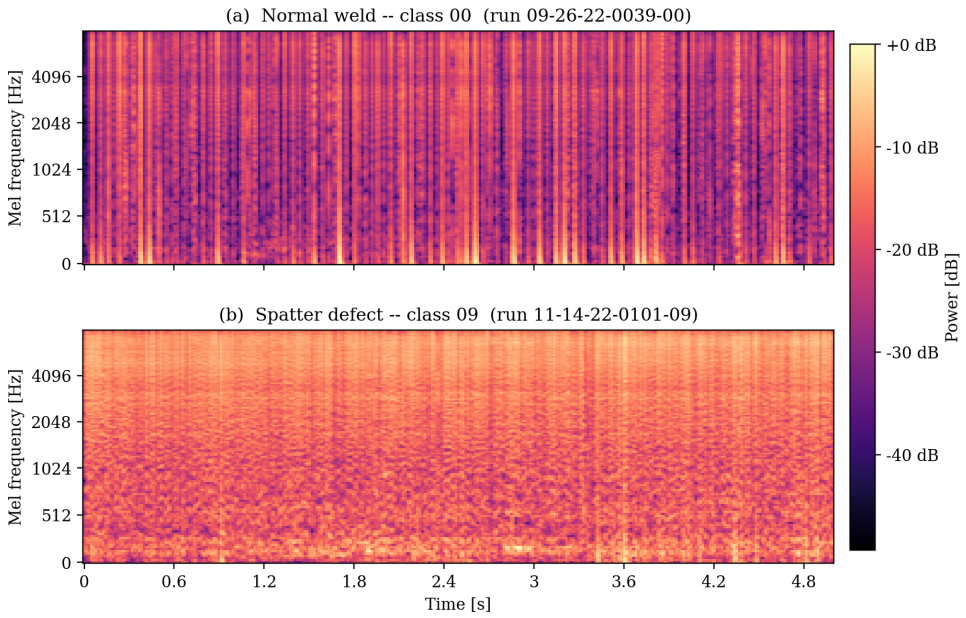

Figure 5.8: Log-mel spectrogram comparison (128 mel bands, $f_{\max} = 8\,\text{kHz}$, shared dB scale). **Panel (a)**: representative normal weld (class 00) – regular impulsive striations from the steady arc dominate the time–frequency plane. **Panel (b)**: Spatter run (class 09) – the rhythmic structure dissolves into diffuse broadband energy with smeared high-frequency content, the spectral signature of the crackling acoustic bursts caused by ejected molten droplets.

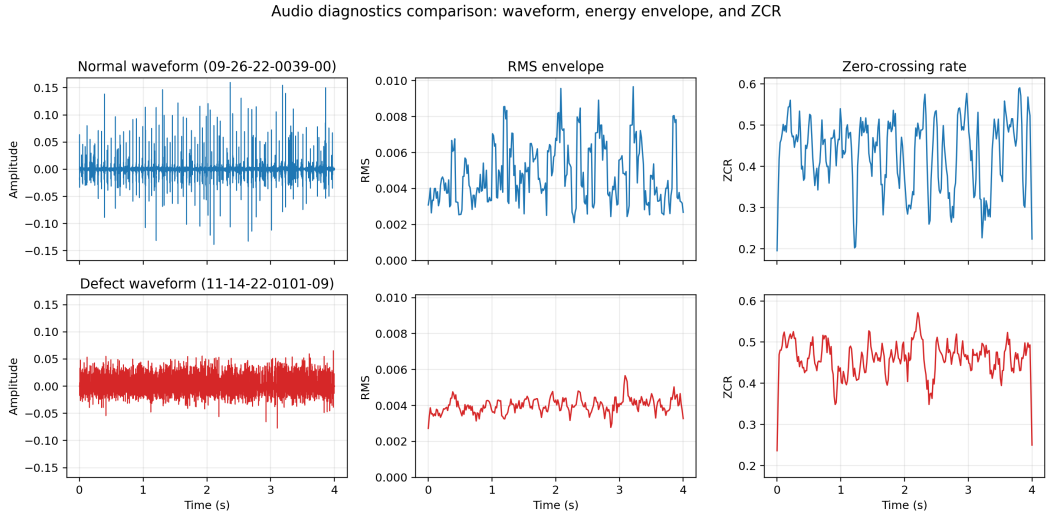

Figure 5.9: Audio diagnostics comparison for representative normal (top row) and Spatter (class 09, bottom row) recordings. Each row shows the time-domain waveform, the short-time RMS envelope, and the zero-crossing-rate (ZCR) trend over the recording, with shared y-axes across rows. The two rows have similar peak amplitudes and overall envelope levels; the diagnostic signature of the Spatter recording is the presence of impulsive ZCR excursions and intermittent transients that depart from the smoother RMS profile of the normal run, rather than a uniformly higher amplitude.

5.5.3 Process-Sensor Snapshot Example

Figure 5.10 illustrates representative process-sensor trajectories from a normal sample, included here for completeness because the dataset distributes these six channels (pressure, CO_2 flow, feed rate, weld current, wire consumed, voltage) alongside the audio and thermal-video streams. None of the four models in this thesis consume the process-sensor stream as a model input; the figure serves as a qualitative visualisation of the auxiliary data available in the dataset for future multimodal studies.

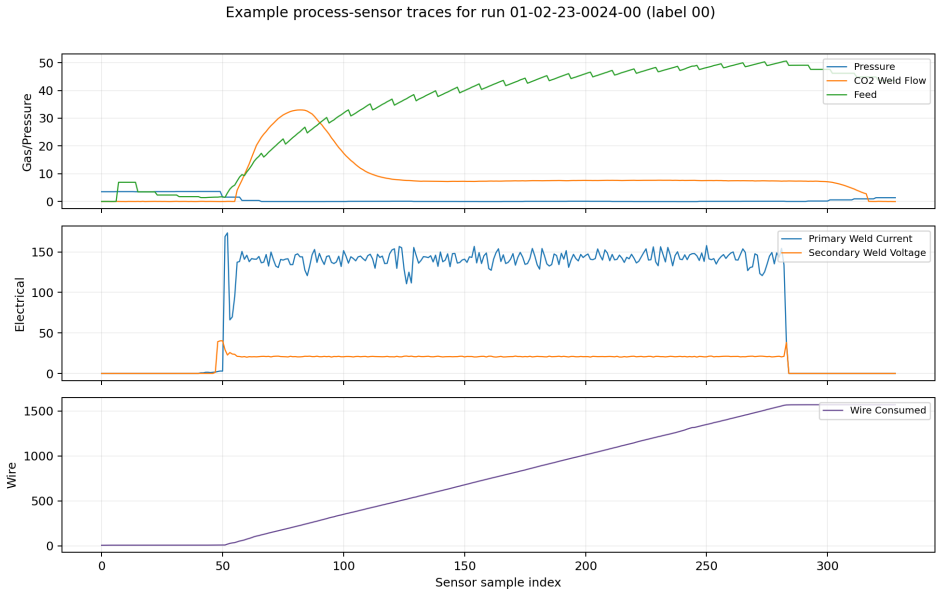

Figure 5.10: Example process-sensor traces for a normal weld sample.

5.6 Split Policy and Comparison Conditions

To reduce leakage, splitting is performed in grouped identity space rather than at file level. Let G denote grouped process identities; then train, validation, and test groups are pairwise disjoint:

$$G_{\text{train}} \cap G_{\text{val}} = \varnothing, \quad G_{\text{train}} \cap G_{\text{test}} = \varnothing, \quad G_{\text{val}} \cap G_{\text{test}} = \varnothing.$$

The target ratio is approximately 70% train, 10% validation, and 20% test. Because the split must assign whole sessions to a single partition to maintain group disjointness, the actual resulting split is train=2676, val=385, test=780 runs (approximately 70/10/20), with all 12 classes represented in every partition. All model families are evaluated on the same partitions so that differences in outcomes can be attributed to model behavior rather than resampling effects.

5.7 Preprocessing and Feature-Extraction Setup

The baseline preprocessing choices are held constant across experiments: audio at 16 kHz, thermal frame sub-sampling for offline and real-time contexts, and sliding-window replay

with 1.0 s window and 0.25 s stride.

The Hierarchical Ensemble uses a merged feature table of 417 handcrafted descriptors (214 thermal + 195 audio + 8 cross-modal), consistent with the feature design formalized in Chapter 4. Backbone Fusion and WeldFusionNet operate on raw or lightly preprocessed modality inputs rather than on the handcrafted feature table.

5.8 Compared Baselines and Training Settings

Three offline fusion approaches and one real-time-native variant are compared under the same two-stage decision logic:

- **Hierarchical Ensemble**: XGBoost + ExtraTrees hierarchical classification with 417 handcrafted features (214 thermal pool geometry, 195 audio spectral, and 8 cross-modal interactions), plus isotonic calibration.
- **Backbone Fusion**: modality embeddings from pretrained ResNet18 (video) and log-mel spectrogram processing (audio), followed by MLP classifiers. Final probabilities use weighted fusion ($\alpha = 0.25$ audio, 0.75 video) and temperature scaling.
- **WeldFusionNet**: end-to-end neural network (≈1.08M parameters) with a Conv1D audio encoder, a MobileNetV3 video encoder, gated fusion with dual heads (12-class + binary), focal loss ($\gamma = 2.0$), multi-task learning, and CosineAnnealingLR scheduling.
- **WeldFusionNet-RT**: same architecture as WeldFusionNet but trained natively on 1-second sliding windows (0.5 s stride). Evaluated on 64,221 windows from a config_id grouped split (751 test runs); latency measured at batch size 1 over 200 warm-started GPU forward passes.

Table 5.4 summarizes the principal training settings used for these approaches.

Table 5.4: Model-family hyperparameter overview used in the comparative experiments.

Setting	Hierarchical Ensemble	WeldFusionNet	Backbone Fusion
Primary classifier	RF/ET/GBM candidate set	Residual MLP fusion net	LogReg/GBM per modality
Key capacity	n_estimators=400	hidden_dim=n/a, dropout=n/a	C grid=[0.1, 0.25, 0.5, 1.0, 2.0, 4.0, 10.0]
Training epochs	n/a (tree fit)	n/a	n/a (candidate fit)
Batch size	n/a	n/a	n/a
Learning rate	n/a	n/a	n/a
Weight decay	n/a	n/a	n/a
Optimization	latency-regularized val selection	focal loss + cosine LR	weight grid + latency-regularized selection
Binary/MC threshold search	yes	yes	yes

5.9 Learning-Curve Diagnostics

To characterize data efficiency and training stability, two complementary learning-curve analyses are included. First, data-size curves are generated by retraining all three approaches on stratified training subsets (10%, 20%, 40%, 60%, 80%, 100%) while keeping validation and test partitions fixed. Second, an epoch-wise WeldFusionNet regularization sweep is reported, comparing the production WeldFusionNet configuration against three alternative regularization settings (reg-A, reg-B, reg-C) on per-epoch validation macro-F1.

For the neural models in the data-size analysis, a fixed bounded schedule is used to keep retraining tractable (Backbone Fusion: 12 epochs, WeldFusionNet: 6 epochs), while all preprocessing, split policy, and evaluation metrics remain unchanged. These truncated schedules are substantially shorter than the full training runs (Backbone Fusion: up to 100 epochs with early stopping; WeldFusionNet: up to 60 epochs with early stopping). As a consequence, the neural models at each data-size fraction may not be fully converged, which could understate their achievable performance at small data sizes and compress the apparent gap between model families. The Hierarchical Ensemble is unaffected by this limitation because tree fitting converges deterministically in a single pass. The learning curves should therefore be interpreted as *trend indicators* of data efficiency rather than as definitive assessments of converged model performance at each fraction. Confidence intervals are not reported, as each point represents a single retraining run.

For the WeldFusionNet regularization sweep, the same model architecture, optimizer family, and training/validation/test partitions are used across all four configurations; only dropout, weight decay, label smoothing, and augmentation intensity vary. The exact configuration deltas are summarized in Appendix A.5. Selection between candidate configurations is governed by validation macro-F1; the held-out test results in Chapter 6 remain the authoritative model-comparison statement. These curves are reported as model-selection diagnostics, not as a competing test claim.

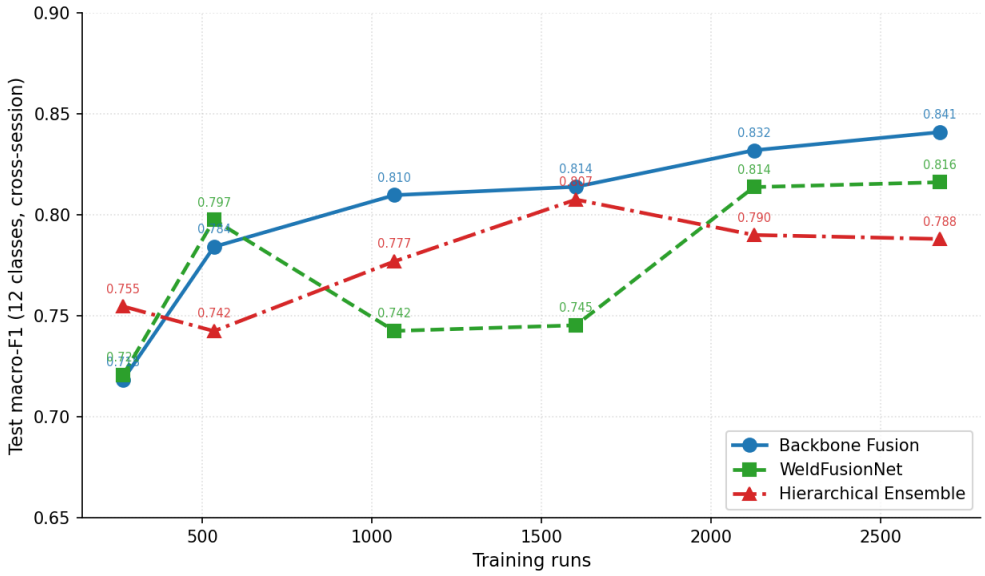

Figure 5.11: Data-size learning curve using test macro-F1 across training fractions for the three model families.

5.10 Evaluation Metrics

Evaluation is stage-aware and deployment-aware.

For stage-1 binary screening (defect vs non-defect), the main indicators are defect

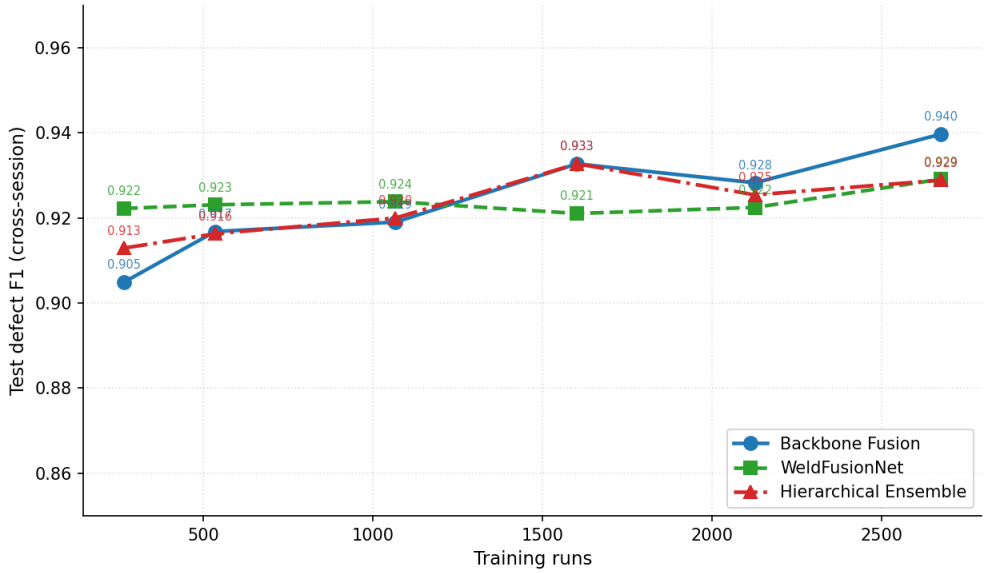

Figure 5.12: Data-size learning curve using test defect F1 across training fractions for the three model families.

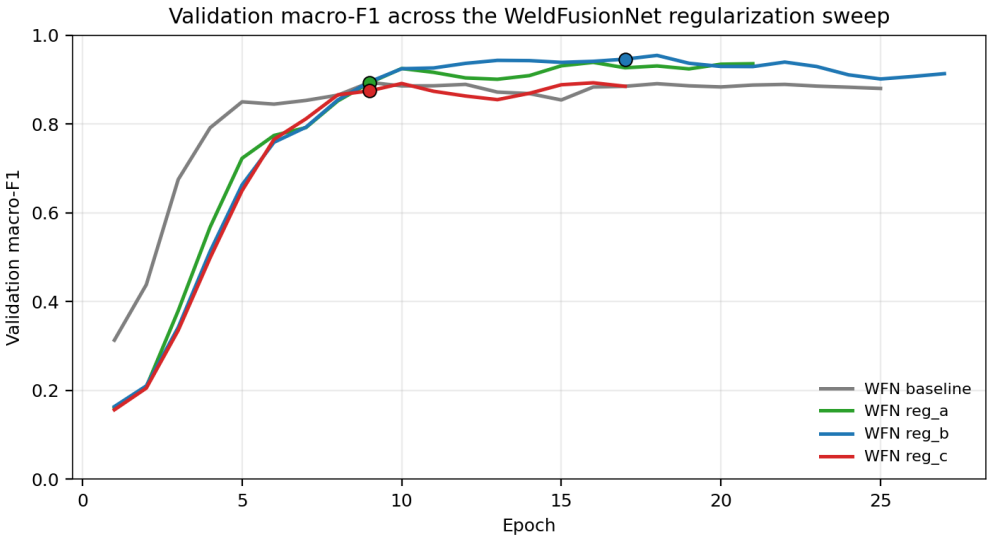

Figure 5.13: WeldFusionNet regularization sweep: smoothed validation macro-F1 per epoch for the production baseline configuration and three alternative regularization settings (reg-A: dropout 0.40, wd 3×10^{-4}; reg-B: dropout 0.45, wd 5×10^{-4}, thermal jitter; reg-C: dropout 0.50, wd 7×10^{-4}, medium augmentation). Marked dots indicate the validation-best (selected) checkpoint epoch for each variant. The regularized variants plateau slightly above the baseline on validation, but the corresponding differences on the held-out test partition are small (within roughly 0.03 macro-F1); the production checkpoint used in Chapter 6 is therefore the baseline configuration. The sweep is reported as model-selection transparency only, not as a competing test claim.

precision, defect recall, defect F1, binary accuracy, and ROC-AUC (where available):

$$\text{Precision}_d = \frac{TP}{TP + FP}, \tag{5.1}$$

$$\text{Recall}_d = \frac{TP}{TP + FN}, \tag{5.2}$$

$$\text{F1}_d = \frac{2\ \text{Precision}_d\,\text{Recall}_d}{\text{Precision}_d + \text{Recall}_d}. \tag{5.3}$$

For the final 12-class output of the two-stage decision policy (which includes class 00 alongside the eleven defect classes), macro-F1 and weighted-F1 are primary:

$$\text{MacroF1} = \frac{1}{|C|} \sum_{c \in C} \text{F1}_c, \tag{5.4}$$

$$\text{WeightedF1} = \sum_{c \in C} \frac{n_c}{N} \text{F1}_c\,. \tag{5.5}$$

Per-class F1 and confusion behavior are analyzed to identify concentrated error modes.

For real-time suitability, latency and throughput are measured at recording level:

$$\text{LatencyMean} = \frac{1}{T} \sum_{t=1}^{T} \ell_t, \tag{5.6}$$

$$\text{Throughput} = \frac{1000}{\text{LatencyMean}}, \tag{5.7}$$

where ℓ_t is event-level inference latency in milliseconds.

5.11 Offline and Real-Time Evaluation Protocol

Each model family is evaluated in two complementary contexts.

Offline evaluation measures standard test-set predictive quality from stored models. Real-time evaluation replays full runs with sliding windows, logs event-level outputs, and aggregates them to recording-level decisions. This two-context protocol is essential because a model that performs well offline may still fail deployment criteria once timing behavior is considered.

Figure 5.14 illustrates the conceptual sliding-window streaming schedule used to define the deployment-oriented latency budget: a 1-second window advanced at a 0.25-second stride implies up to four inference events per second. The implemented replay protocol used for the experiments reported in Chapter 6 is the frame-indexed protocol summarized in Table 5.5 below; Figure 5.14 should be read as the deployment-design schedule, not as the exact frame-indexed cadence of the reported replay.

Latency measurement sample sizes. Due to the computational cost of full-run real-time replay, latency statistics are collected on a subset of the test recordings rather than all 780. For all three offline-trained approaches, latency is measured on 20 randomly sampled test recordings per class (240 total). These sample sizes produce point estimates of mean and P95 latency sufficient for comparing the approaches, but they are not large enough to characterize latency distributions with statistical precision; see Section 6.8 of the results chapter.

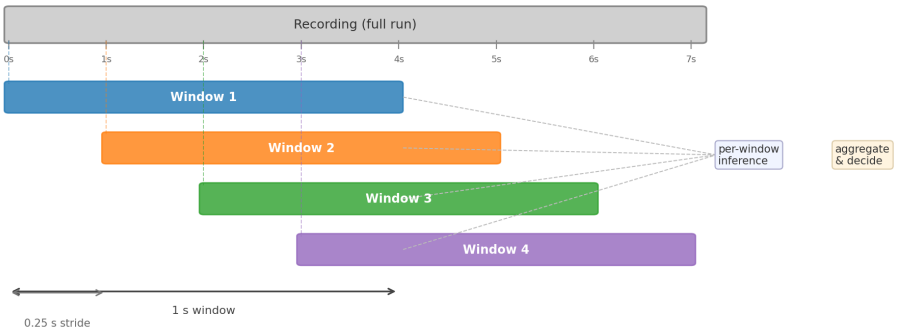

Figure 5.14: Conceptual streaming schedule used to define the 250 ms design budget: a 1-second window advancing at a 0.25-second stride yields four inference events per second. The implemented frame-indexed replay protocol used for the reported experiments differs in stride and context length (see Table 5.5); this figure illustrates the deployment-oriented schedule, not the exact replay cadence used in Chapter 6.

Timing protocol summary. To avoid confusion between context length, event stride, and the model's internal input representation, Table 5.5 summarizes the three quantities separately for every model.

Table 5.5: Per-event timing breakdown for the four models. The context length is the slice of video frames that feeds one inference event; the event stride is the gap between successive events during streaming replay; the internal input representation is the fixed-shape tensor the network actually consumes. Wall-clock duration of the context depends on the recording's native frame rate (nominally 30 FPS, resampled to 25 Hz in the pipeline).

Model	Context per event	Event stride	Internal input representation
Hierarchical Ensemble	50-frame video context	125-frame stride during replay	417 handcrafted features extracted from the context
Backbone Fusion	50-frame video context	125-frame stride during replay	ResNet18 video embeddings + log-mel audio embedding from the context
WeldFusionNet	250-frame video context	125-frame stride during replay	(25×44) audio tensor + 8 uniformly sampled video frames (1-second-equivalent internal input)
WeldFusionNet-RT	1-second window	0.5 s window stride	(25×44) audio tensor + 8 uniformly sampled video frames

Concretely: "50-frame chunk" refers to a slice of 50 video frames (roughly 1.7–2 s of wall-clock time, depending on native FPS), not to a 1-second window; and "250-frame context" for WeldFusionNet refers to the video frames within which 8 frames are uniformly sampled and a 25-timestep audio tensor is built, not to a 10-second model input. Whenever "1-second window" appears in this thesis, it refers to the model's internal input representation, not to the streaming context length.

Glossary for the real-time replay terminology. To prevent any remaining confusion, the following terms have distinct meanings in this thesis:

Concept	Meaning in this thesis
250 ms budget	Deployment-oriented design latency target (4 events/s scheduling interval); *not* the exact frame stride used in replay.
50-frame context	Implemented replay video-frame context for Hierarchical Ensemble and Backbone Fusion.
250-frame context	Implemented replay video-frame context for WeldFusionNet; reduced internally to a fixed (25×44) audio tensor and 8 uniformly sampled video frames.
1-second-equivalent input	WeldFusionNet's fixed internal representation; refers to the model's input shape, not to the raw video context length.
WeldFusionNet-RT window	Actual 1 s window with 0.5 s stride; evaluated under a separate config-grouped split (Section 4.5.4).
Table 6.4 macro-F1	Recording-level replay diagnostic under *mixed* multiclass aggregation (event-aggregated for WeldFusionNet; post-run for Hierarchical Ensemble and Backbone Fusion). Not a uniform event-level multiclass streaming metric.

5.12 Robustness and Stress Protocols

Beyond nominal test replay, robustness for the production WeldFusionNet checkpoint is evaluated through controlled perturbation experiments on a matched 20-run subset of test recordings, stratified across seven classes (codes 00, 01, 03, 04, 05, 06, 08); the same 20 runs are used for clean, SNR=20 dB, SNR=10 dB, and missing-audio conditions so that deltas isolate perturbation effects rather than population differences. An earlier late-fusion two-stage model variant ran a separate missing-modality experiment on a 30-run subset; its quantitative results are excluded from this thesis and summarized only as a provenance note in §A.4. They are not used to support claims about the production WeldFusionNet.

Noise perturbation. Additive Gaussian white noise is injected into the audio signal at two signal-to-noise ratio levels: 20 dB (moderate noise) and 10 dB (heavy noise). The thermal-video stream is left unperturbed. This protocol tests the sensitivity of audio-dependent features and fusion models to acoustic contamination.

Missing-modality condition. A single missing-audio condition is evaluated on the matched 20-run subset: the audio input to WeldFusionNet is zeroed to simulate microphone failure while the thermal-video stream is left intact. Because WeldFusionNet does not consume the process-sensor stream, no separate missing-sensor condition applies to it; an earlier late-fusion two-stage variant did consume an auxiliary sensor stream, but its quantitative missing-modality results are excluded here and only acknowledged as a provenance note in §A.4.

Matched-run comparison. For WeldFusionNet, the clean, noise-20 dB, noise-10 dB, and missing-audio conditions are all evaluated on the same 20-run subset, enabling paired comparison of metrics across conditions. The delta between perturbed and clean performance on this matched set is reported alongside absolute scores. The 20 runs are stratified across seven classes (codes 00, 01, 03, 04, 05, 06, 08), with exact per-class counts reported in the appendix; the term "matched" here refers to the cross-condition pairing, not to exact class balance.

Auxiliary cross-session split. An additional grouped split is constructed to probe

distribution-shift sensitivity from a different partition cut. This auxiliary split uses the dataset's native directory structure as the partitioning criterion: recordings from the `test_data` source directory are assigned to the test partition, while recordings from the `good` and `defect` directories form the training and validation partitions (split 87/13 within that pool). This produces a substantially different partition composition from the main session-disjoint split: the auxiliary test set contains 103 runs (vs. 780 in the main split) drawn exclusively from the `test_data` directory, while the auxiliary training set contains 3,258 runs. The group disjointness property is preserved: no configuration identity (`config_id` group) appears in both the auxiliary training and auxiliary test partitions. Models are retrained from scratch on the auxiliary training partition and evaluated on the auxiliary test partition, providing a complementary perspective on whether performance estimates are stable across different cross-session cuts.

5.13 Class Imbalance and Handling Strategy

The Intel Robotic Welding Multimodal Dataset exhibits substantial class imbalance at the run level. The dominant class (code 00, normal welds) constitutes the single largest category, while some defect codes are represented by as few as a few dozen runs. This imbalance has implications for model training, evaluation metric selection, and the interpretation of reported results.

5.13.1 Imbalance Characterization

For the purpose of this study, class imbalance is characterized along three dimensions. First, the *majority-to-minority ratio* captures the relative frequency between the most and least represented defect classes. When this ratio exceeds 10:1, standard training objectives are likely to be dominated by the frequent classes. Second, the *test-set support per class* determines the precision of per-class F1 estimates: a class with only 10 test examples has a potential F1 variation of 10 percentage points per misclassification. Third, the *binary imbalance ratio* (normal vs. any defect) determines whether the binary screening stage is imbalance-prone; in this dataset, the fraction of normal runs is approximately one-fifth of the total, which means the binary stage is mildly inverted—defective runs outnumber normal runs.

5.13.2 Mitigation Strategies Employed

Three complementary strategies are applied to reduce the negative effects of class imbalance on training and evaluation.

Stratified splitting. The grouped split is stratified so that each class is proportionally represented in each partition. Without stratification, random group assignment could result in some minority classes being entirely absent from the validation or test partition. Stratification guarantees that every class appears in every partition, albeit with proportionally fewer examples for rare classes.

Class-weighted losses. The MLP heads in Backbone Fusion are trained with class-weighted cross-entropy in which per-class weights are derived from inverse training-set class frequency. The end-to-end WeldFusionNet and WeldFusionNet-RT instead use focal loss ($\gamma = 2$) on the 12-class head, with class-frequency weights α_t likewise derived from inverse training-set class frequency. Both choices re-weight rare-class contributions to the training objective, but focal loss additionally down-weights well-classified examples to focus learning on hard cases.

Macro-F1 as primary metric. By averaging per-class F1 without weighting by class frequency, macro-F1 ensures that rare defect classes have equal influence on the aggregate performance score as frequent classes. A model that achieves 0.99 F1 on all frequent classes but 0.50 F1 on a rare class will report a lower macro-F1 than a model that achieves more balanced per-class performance. This metric choice directly reflects the monitoring goal: in a safety-critical welding context, every defect type matters regardless of its frequency in the training data.

5.14 Hardware and Software Environment

All experiments were executed on a single workstation running a 64-bit Linux operating system with an Intel CPU and a compatible NVIDIA GPU. The deep learning models (WeldFusionNet, Backbone Fusion neural heads) were trained and evaluated with GPU acceleration using CUDA. The tree-based models (Hierarchical Ensemble) were trained on CPU only, as tree induction does not benefit from GPU parallelism.

The software stack used throughout the experiments consists of Python 3.9, PyTorch for neural network training and inference, scikit-learn for tree-based models and probability calibration, and librosa for audio feature extraction [14]. All preprocessing operations (resampling, windowing, STFT, mel filterbank) were executed using librosa and NumPy. Results were recorded using structured logging with per-run latency and prediction timestamps.

A single fixed random seed (`seed = 42`) was used for all stochastic components: dataset splitting, model weight initialization, the random projection matrix in Backbone Fusion's audio embedding, and dropout mask generation during evaluation. This seed is set via a unified `set_seed()` utility that applies it to Python's `random` module, NumPy, and PyTorch (both CPU and CUDA) at the start of each training script. This ensures that comparisons across runs are deterministic given the same input data, though it does not produce multiple independent estimates of variance.

5.15 Reproducibility Controls

Reproducibility in machine learning experiments requires that the full chain from raw data to reported metrics can be re-executed and produces the same results. In this thesis, reproducibility is enforced through four controls.

Fixed snapshot. All models are trained and evaluated on a single frozen local snapshot of the dataset. The snapshot version is identified by its run count (3841 runs), split distribution

(train=2676, val=385, test=780), and the grouped identity count (226 groups). Any experiment producing different run counts on the same dataset indicates a preprocessing discrepancy.

Fixed split. The grouped stratified split is constructed once and stored as a partition file. All model families read the same partition assignment, ensuring that train/validation/test membership is identical across all experiments. This is a stronger reproducibility guarantee than re-running the split with the same seed, because it eliminates the risk of split code changes between experiments.

Deterministic preprocessing. Feature extraction and preprocessing operations are implemented as pure functions with no state beyond their inputs, and all random operations (where applicable) use fixed seeds. This ensures that the feature matrix seen by each model is identical across reruns, removing a common source of evaluation variance.

Logged inference latencies. Real-time evaluation logs event-level start and end timestamps for each window processed, enabling post-hoc verification that reported latencies correspond to actual execution on the evaluation hardware. The logged timestamps are retained with the evaluation outputs to allow comparison against hardware-specific performance targets if the evaluation platform changes.

5.16 Protocol Consistency

The experiment design follows a fixed sequence of snapshot selection, grouped split construction, feature extraction, model training, and evaluation. All reported comparisons are derived from the same frozen snapshot and the same partitioning logic. This consistency allows differences in reported performance to be interpreted as differences in model behavior rather than differences in data handling.

The evaluation derives point estimates from a single held-out partition. To accompany these point estimates with uncertainty bounds, Section 6.8 of the results chapter reports paired bootstrap confidence intervals on every headline metric (1000 resamples, seeded `rng=2026`), McNemar tests on the binary defect task, and a paired bootstrap on multiclass macro-F1 differences. Per-class F1 estimates for classes with few test samples remain sensitive to the specific composition of the test set; repeated re-splitting across independent partition draws is identified as a direction for future work.

Chapter Summary

This chapter established the experimental basis of the thesis: dataset scope (3841 runs, 12 classes), session-disjoint split protocol with all classes represented in the test partition (780 runs), grouped leakage control, three offline fusion approaches (Hierarchical Ensemble, Backbone Fusion, WeldFusionNet) plus the real-time-native WeldFusionNet-RT variant, stage-aware metrics, and real-time-oriented evaluation. It characterized the class imbalance present in the dataset and described the three-pronged mitigation strategy: stratified splitting, class-weighted focal loss, and macro-F1 as the primary evaluation metric. The hardware and software environment was documented alongside four reproducibility controls: fixed snapshot, fixed split, deterministic preprocessing, and logged inference timestamps. These

controls ensure that the quantitative evidence presented in Chapter 6 is traceable to specific data, model, and evaluation decisions rather than to incidental experimental choices.

Chapter 6

Results

6.1 Offline Test Performance

6.1.1 Stage-1 Binary Defect Detection

Table 6.1 compares binary defect screening (`00` vs non-`00`) on the held-out test set of 780 runs drawn from welding sessions disjoint from those used for training.

Table 6.1: Offline test binary performance comparison (`00` vs non-`00`).

Model	Binary F1	Precision	Recall	ROC-AUC	Accuracy
Hierarchical Ensemble	0.9163	0.9246	0.9082	0.9253	0.8679
Backbone Fusion	0.9384	0.9317	0.9452	0.9155	0.9013
WeldFusionNet	0.9895	0.9887	0.9903	0.9981	0.9833

WeldFusionNet leads binary defect screening with a binary defect F1 of 0.9895, followed by Backbone Fusion (0.9384) and Hierarchical Ensemble (0.9163). WeldFusionNet also achieves the highest binary precision (0.9887) and recall (0.9903), with binary accuracy of 0.9833. Backbone Fusion achieves the second-highest binary recall (0.9452) and accuracy (0.9013). Hierarchical Ensemble obtains the lowest binary accuracy (0.8679). ROC-AUC values separate the models clearly: 0.9981 (WeldFusionNet), 0.9253 (Hierarchical Ensemble), and 0.9155 (Backbone Fusion). The substantial AUC gap between WeldFusionNet and the two tree-based approaches indicates that the end-to-end neural architecture produces a more discriminative decision function across the full operating range, not only at the calibrated threshold.

6.1.2 Final 12-Class Classification

Table 6.2 reports aggregate metrics over the final 12-class output of the two-stage decision policy, where class 00 (good weld) is included alongside the eleven defect classes. The metrics therefore measure full-taxonomy 12-class performance after the stage-1 gate, not defect-only stage-2 performance.

Backbone Fusion achieves the highest offline macro-F1 (0.8319), followed by WeldFusionNet (0.8178) and Hierarchical Ensemble (0.7459). Accuracy follows a different ordering,

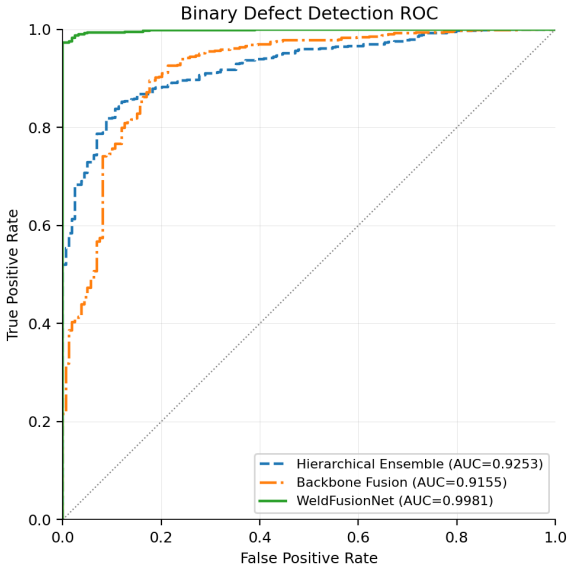

Figure 6.1: Binary defect detection ROC curves for all three models on the offline test set. AUC values confirm that WeldFusionNet provides the strongest discrimination between normal and defective welds: AUC = 0.9981 (WeldFusionNet), 0.9253 (Hierarchical Ensemble), and 0.9155 (Backbone Fusion).

Table 6.2: Offline final 12-class test performance comparison.

Model	Macro F1	Weighted F1	Accuracy
Hierarchical Ensemble	0.7459	0.7551	0.7744
Backbone Fusion	0.8319	0.8340	0.8372
WeldFusionNet	0.8178	0.8775	0.8872

with WeldFusionNet at 0.8872, Backbone Fusion at 0.8372, and Hierarchical Ensemble at 0.7744; weighted F1 follows accuracy (0.8775, 0.8340, 0.7551). WeldFusionNet's lead on accuracy and weighted F1, together with Backbone Fusion's lead on macro-F1, reflects a structural difference between the two models: WeldFusionNet is stronger on the frequent classes, whereas Backbone Fusion distributes its errors more evenly across the taxonomy. Hierarchical Ensemble trails on every multiclass aggregate, with a macro-F1 gap of 0.086 below Backbone Fusion. The macro-F1 ordering indicates that Backbone Fusion distributes errors more uniformly across the 12-class taxonomy than the handcrafted ensemble or the end-to-end network on this dataset, while Hierarchical Ensemble concentrates its errors in a small set of difficult minority classes (notably class 05).

Table 6.3: Offline final 12-class per-class F1 by model.

Code	Class	HE	BF	WFN
00	Good Weld	0.687	0.751	0.959
01	Excess. Penetration	0.421	0.821	0.672
02	Burn Through	0.675	0.694	0.769
03	Porosity	0.966	0.957	0.947
04	Porosity w/ Excess. Pen.	0.980	0.975	0.969
05	Undercut	0.000	0.519	0.091
06	Overlap	0.800	1.000	1.000
07	Lack of Fusion	0.857	0.892	0.902
08	Excess. Convexity	1.000	1.000	0.930
09	Spatter	0.987	1.000	0.988
10	Warping	0.577	0.615	0.876
11	Crater Cracks	1.000	0.759	0.710
	Macro F1	0.746	0.832	0.818

All 12 classes are represented in the 780-run test partition. Table 6.3 shows per-class F1 across the full defect taxonomy. Per-class leadership is split: WeldFusionNet leads on class 00 (good weld, 0.959), class 02 (burn through, 0.769), class 07 (lack of fusion, 0.902), and class 10 (warping, 0.876); Backbone Fusion leads on class 01 (excessive penetration, 0.821), class 05 (undercut, 0.519), class 06 (overlap, 1.000, tied with WeldFusionNet), and class 09 (spatter, 1.000); Hierarchical Ensemble leads on class 11 (crater cracks, 1.000) and class 08 (excessive convexity, tied at 1.000). All three models exceed 0.94 on classes 03 (porosity), 04 (porosity with excessive penetration), and 09 (spatter). Class 05 (undercut) is the consistent failure mode across approaches: Hierarchical Ensemble collapses to F1=0.000, WeldFusionNet reaches only 0.091, and Backbone Fusion attains 0.519, which is still the weakest among its defect classes. Class 10 (warping) is also depressed for Hierarchical Ensemble (0.577) and Backbone Fusion (0.615) compared with WeldFusionNet (0.876).

6.1.3 Offline Confusion-Matrix Gallery

The following representative figure provides class-wise error structure for the strongest offline model on accuracy and binary detection (WeldFusionNet). Confusion matrices for the

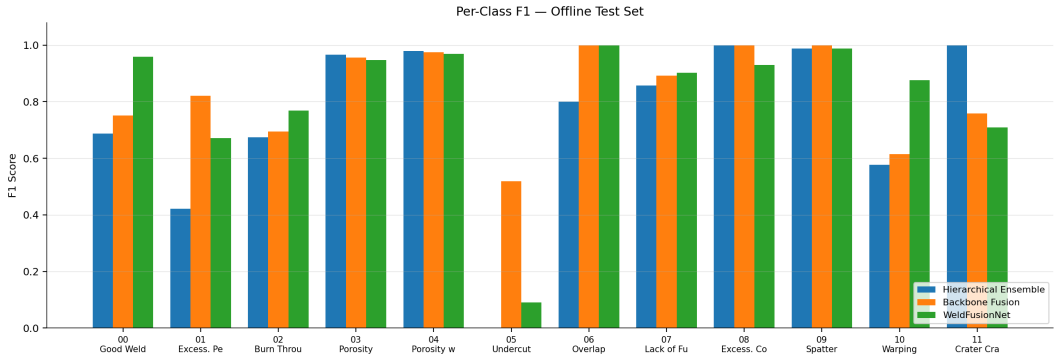

Figure 6.2: Per-class F1 for all three models on the offline final 12-class test set (780 runs). Class 05 (undercut) is the principal failure mode across approaches, and class 10 (warping) is the secondary one. WeldFusionNet leads on the head classes (good weld, lack of fusion, warping); Backbone Fusion distributes performance more evenly and leads on macro-F1 by partially recovering classes 01 and 05.

remaining models are provided in the Appendix.

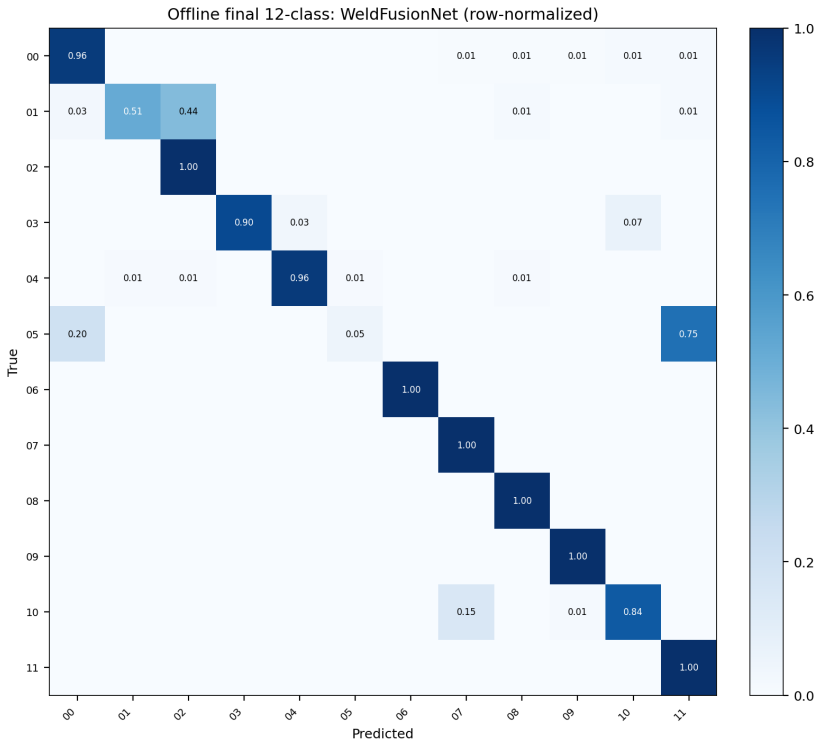

Figure 6.3: Offline final 12-class confusion matrix: WeldFusionNet (row-normalized).

The matrix confirms that error mass is concentrated in a small subset of challenging classes rather than uniformly distributed. WeldFusionNet's principal confusions are class 05 (undercut) mass-mapped to class 11 (crater cracks), and class 01 (excessive penetration) mapped to class 02 (burn through), consistent with the per-class F1 analysis. The top-confusion pairs in Table A.5 (Appendix) confirm the same overall pattern. Class 05 (undercut) collapses entirely for Hierarchical Ensemble, which routes all 20 undercut runs to class 06 (overlap), and is the dominant failure for Backbone Fusion (13 of 20 to class 11, crater cracks) and

WeldFusionNet (15 of 20 to class 11). The excessive-penetration → burn-through confusion (01→02) is the second large failure mode and is shared by all three models (37 runs for Hierarchical Ensemble, 35 for WeldFusionNet, and 12 for Backbone Fusion).

6.2 Recording-Level Real-Time Performance

Real-time evaluation simulates streaming inference over the held-out test set. For the three offline-trained models (Hierarchical Ensemble, Backbone Fusion, WeldFusionNet), 240 recordings are sampled (20 per class) and processed in a streaming fashion. The replay protocol uses the frame-indexed implementation summarized in Table 5.5: a fixed *context length* (50 video frames for Hierarchical Ensemble and Backbone Fusion; 250 video frames for WeldFusionNet) is advanced by a fixed *event stride* of 125 video frames over each recording, and one inference event is emitted per advance. At the dataset's native frame rate the 125-frame stride corresponds to several seconds of wall-clock time; the abstract per-second event-rate target referenced in Section 4.3.1 ($\Delta_w = 1.0\,\text{s}$, $\Delta_s = 0.25\,\text{s}$) describes a deployment-oriented design budget rather than the exact frame-indexed cadence used in this replay, which produces a smaller number of inference events per recording. Regardless of context length, the model's internal input is a fixed-shape tensor: 417 handcrafted features for Hierarchical Ensemble, ResNet18 video plus audio embeddings for Backbone Fusion, and a (25×44) audio tensor plus 8 uniformly sampled video frames for WeldFusionNet.

Per-event defect probabilities are averaged across the recording and compared against the recording-level threshold $\tau_r = 0.70$ to produce the recording-level binary decision (Section 4.6). The multiclass aggregation differs by model: WeldFusionNet averages per-event class posteriors across the recording and reports the arg max over defect classes, so its real-time multiclass output is event-aggregated. Hierarchical Ensemble and Backbone Fusion instead compute the multiclass label from a full-recording feature extraction performed once at the end of the stream, so their real-time multiclass output is a post-run diagnosis attached to the same replay (binary gating is still event-aggregated for both). The multiclass macro-F1 values reported in Table 6.4 should therefore be interpreted under this mixed aggregation regime rather than as a pure event-level streaming output for the two tree-based and pretrained-backbone models.

WeldFusionNet-RT is evaluated under a different protocol that matches its training regime: the full WeldFusionNet-RT test partition is segmented into 1-second windows with 0.5-second stride, yielding 64,221 windows. Each window is classified independently in a single forward pass; no aggregation across windows is applied. Latency is measured as single-window GPU inference time (batch size 1) over warm-started runs. Table 6.4 summarizes quality and runtime indicators. The table is partitioned into two protocol blocks: the three offline-trained models evaluated at the recording level on the 240-run session-disjoint subset, and WeldFusionNet-RT as a streaming-native proof-of-concept evaluated at the window level on its own configuration-grouped split. Window-level WeldFusionNet-RT scores are therefore marked as window-level and not directly comparable to recording-level metrics; the partition is preserved in every subsequent discussion of this table.

Under the streaming protocol, WeldFusionNet achieves the highest real-time binary defect

Table 6.4: Real-time comparison: offline models on 240 recordings sampled at 20 per class from the 780-run test set (50-frame video context for Hierarchical Ensemble and Backbone Fusion; 250-frame video context with internal 1-second-equivalent input for WeldFusionNet; mean-probability threshold $\tau_r = 0.70$ for binary aggregation). WeldFusionNet-RT is reported on 64,221 windows from its configuration-grouped test partition (1 s window, 0.5 s stride, single-window GPU forward pass at batch size 1 over 200 warm-started runs; window-level, not directly comparable to recording-level metrics); it is included in the same table only for compactness.

Model	Binary F1	Macro-F1	Accuracy	Lat. Mean (ms)	Lat. P95 (ms)
Recording-level replay (240-run subset, session-disjoint split)					
Hierarchical Ensemble	0.9559	0.7475	0.7792	151.7	170.0
Backbone Fusion	0.9698	0.8436	0.8500	127.5	141.3
WeldFusionNet	0.9886	0.8016	0.8375	72.2	83.5
Window-level proof-of-concept (different protocol — not directly comparable)					
WeldFusionNet-RT †	0.9946	0.9853	0.9812	3.2	3.8

† 64,221 windows from a `config_id`-grouped test split (751 runs);
single-window GPU forward pass, batch size 1, over 200 warm-started runs.

F1 of 0.9886 among the three offline models, followed by Backbone Fusion (0.9698) and Hierarchical Ensemble (0.9559). Real-time multiclass macro-F1 reorders: Backbone Fusion leads at 0.8436, ahead of WeldFusionNet (0.8016) and Hierarchical Ensemble (0.7475). The multiclass macro-F1 ordering in the replay setting matches the offline ordering, indicating that the replay protocol and the per-model multiclass aggregation step preserve the relative model ranking. Because multiclass aggregation differs across models (event-aggregated for WeldFusionNet; post-run full-recording diagnosis for Hierarchical Ensemble and Backbone Fusion), these values should be interpreted as recording-level replay diagnostics rather than as a uniform event-level multiclass streaming metric. Backbone Fusion's lead on multiclass macro-F1 under both protocols reflects more uniform per-class behavior across the 12-class taxonomy.

Under its separate window-level protocol, WeldFusionNet-RT, trained natively on 1-second windows, reports binary F1 of 0.9946, multiclass macro-F1 of 0.9853, and accuracy of 0.9812 (window-level, not directly comparable to recording-level metrics) on 64,221 test windows. These numbers describe behavior on the streaming-native protocol and *cannot* be read as a head-to-head improvement over the three offline models, which are evaluated at the recording level on a different split. Two factors inflate the gap relative to WeldFusionNet's chunked recording-level evaluation (macro-F1 0.8016): native window-level training, whose representations are calibrated to the 1-second temporal context rather than adapted from offline features, and the much larger evaluation pool (64,221 windows versus 240 sampled recordings) over a more permissive configuration grouping rather than a strict session-disjoint split. Windows sampled at 0.5-second stride from the same recording are also temporally correlated and not statistically independent. WeldFusionNet-RT is therefore reported here as a streaming-native proof-of-concept, not as a competitor in the recording-level ranking.

The models differ substantially in runtime cost. WeldFusionNet processes one inference event in a mean of 72.2 ms (P95: 83.5 ms). Backbone Fusion processes one event in 127.5 ms (P95: 141.3 ms), and Hierarchical Ensemble in 151.7 ms (P95: 170.0 ms). The relevant

budget here is set by the desired event stride at deployment time, not by the context length used per event; under a 250 ms scheduling target (equivalent to a 0.25 s event stride) all three offline-trained models complete inference within budget. WeldFusionNet-RT achieves the lowest per-window inference latency by a large margin: 3.2 ms mean (P95: 3.8 ms), reported under its own single-window protocol. All four configurations therefore satisfy their respective real-time targets, making latency a discriminating but not disqualifying factor in model selection.

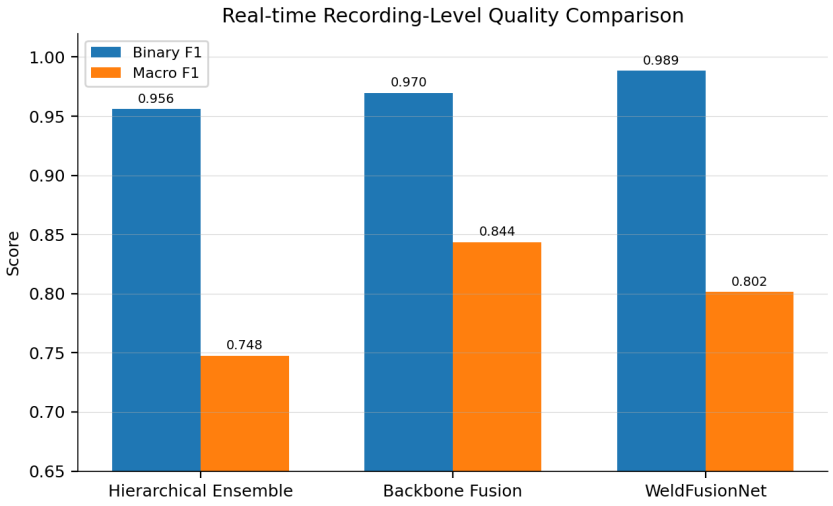

Figure 6.4: Real-time recording-level quality comparison for the three offline models (binary defect F1 and multiclass macro-F1). WeldFusionNet-RT uses a different window-level protocol and is not shown here; see Table 6.4.

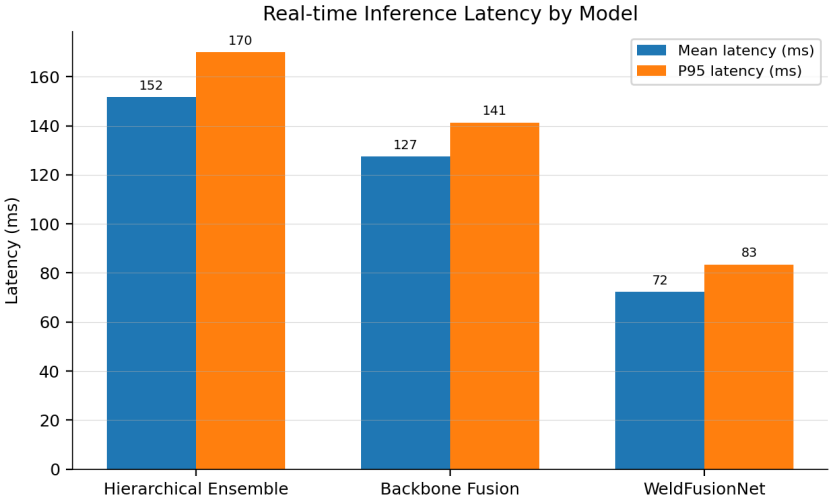

Figure 6.5: Real-time mean and P95 inference latency for the three offline models (Hierarchical Ensemble: 151.7 ms mean, 170.0 ms P95; Backbone Fusion: 127.5 ms, 141.3 ms; WeldFusionNet: 72.2 ms, 83.5 ms). WeldFusionNet-RT (mean 3.2 ms, P95 3.8 ms) is omitted from the chart as its scale would compress the other bars; see Table 6.4.

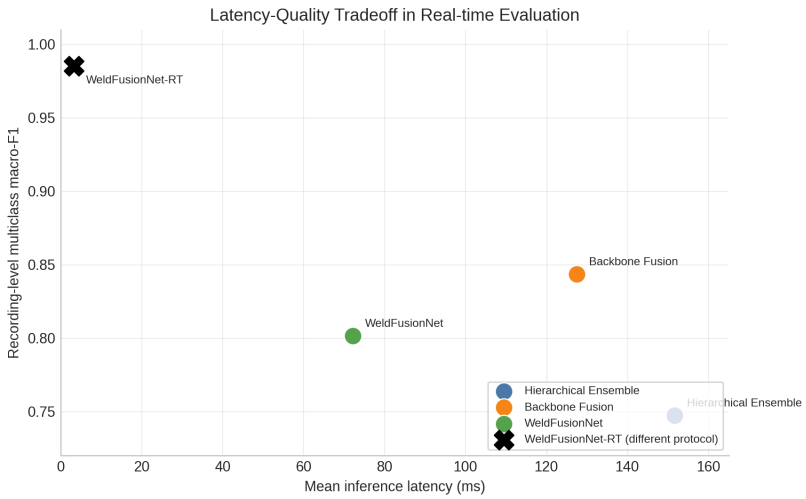

Figure 6.6: Latency-quality tradeoff in the real-time evaluation. The three circular markers are the directly comparable offline-trained models evaluated on the same 240-run recording-level replay subset; lower latency and higher macro-F1 is preferable. The black `X` marks WeldFusionNet-RT for reference only: it is evaluated under a different protocol (window-level, not directly comparable to recording-level metrics).

6.3 Robustness Results

6.3.1 Noise Stress (Matched Subset)

Audio-noise robustness is evaluated as a paired comparison: the same 20-run subset (stratified across codes 00, 01, 03, 04, 05, 06, 08 with exact per-class counts of 3/3/3/2/3/3/3) is replayed three times: once with the original audio, once with additive white Gaussian noise injected at SNR=20 dB, and once at SNR=10 dB. All other inputs (thermal video, run identity) are held identical across the three conditions, so the deltas between rows in Table 6.5 isolate the effect of audio contamination from any population shift.

Table 6.5: Matched-pair audio-noise robustness for WeldFusionNet on the same 20-run subset under clean and noise-injected conditions. Deltas are taken against the matched-clean baseline. Mean latency is measured under the robustness replay harness on this 20-run subset and is not comparable with the per-event latencies of the timing protocol reported earlier in this chapter.

Condition	Binary F1	Macro-F1	Δ Macro-F1	Mean latency (ms)
clean (matched r20)	0.8485	0.5500	+0.0000	114.4
noise SNR=20 dB (r20)	0.8824	0.4716	−0.0784	104.5
noise SNR=10 dB (r20)	0.9143	0.5056	−0.0444	104.4
missing audio (matched r20)	0.9189	0.2619	−0.2881	77.6

The matched-pair design yields modest macro-F1 degradation under noise ($\Delta = -0.078$ at SNR=20 dB and $\Delta = -0.044$ at SNR=10 dB), while binary defect F1 rises under noise on this 20-run subset (from 0.849 clean to 0.882 at SNR=20 dB and 0.914 at SNR=10 dB) because most of the subset is defective and the additional input noise pushes ambiguous-class predictions toward the defect side of the binary threshold. The macro-F1 effect is non-

monotonic: the heavier 10 dB condition degrades macro-F1 less than the 20 dB condition. With only 20 matched runs, a small number of per-run prediction flips moves macro-F1 in steps large enough to account for this reversal, so the two deltas should be read as similar-magnitude degradations rather than as a dose–response curve. The asymmetry between the binary and multiclass effect is consistent with the WeldFusionNet binary head being substantially better calibrated than its multiclass head (§6.9). The previously reported internal noise robustness comparison was constructed against a non-matched 30-run clean baseline drawn from a different class composition and therefore conflated population shift with the noise effect; that comparison is superseded by the matched results reported here.

6.3.2 Missing-Audio Stress (Matched Subset)

A missing-audio condition is also evaluated on the same 20-run subset by zeroing the audio input to the WeldFusionNet encoder while leaving the thermal-video stream intact. Macro-F1 drops from 0.550 clean to 0.262 under audio-zeroing ($\Delta = -0.288$), while binary defect F1 rises from 0.849 to 0.919, continuing the defect-ward drift already observed under additive noise on this defect-majority subset. This pattern indicates that the audio modality contributes substantial non-redundant signal for multiclass discrimination, whereas the binary screening decision remains strong without audio, consistent with the thermal-video stream carrying the dominant binary-screening signal; the upward shift in binary F1 nevertheless shows that audio removal does perturb the binary head's scores on this subset rather than leaving them unchanged. A symmetric missing-thermal-video condition was not evaluated and is identified as future work. Because the production WeldFusionNet does not consume the auxiliary process-sensor stream, no missing-sensor condition applies; an earlier late-fusion variant that did consume an auxiliary sensor stream is summarized as a provenance note in the appendix (§A.4), with quantitative legacy results excluded.

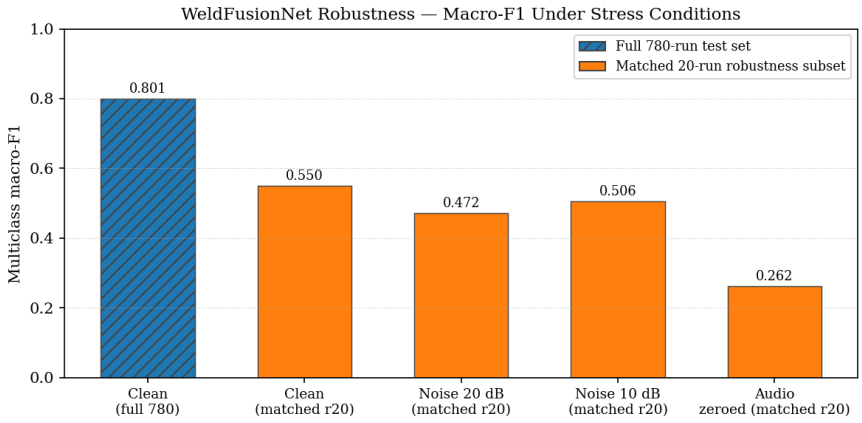

Figure 6.7: WeldFusionNet robustness summary: multiclass macro-F1 under clean, noise-stressed, and audio-zeroed conditions. The hatched "Clean (full 780)" bar is the full-test baseline shown for context only and is not population-matched to the other bars. The four solid bars all use the same matched 20-run subset (clean, SNR=20 dB noise, SNR=10 dB noise, audio zeroed), enabling direct delta interpretation.

These stress runs characterize the sensitivity of WeldFusionNet to acoustic contamination

and to complete audio loss, and identify the robustness gaps that remain despite the high nominal accuracy on the clean test partition.

6.3.3 Cross-Session Sensitivity Check

The primary results in Tables 6.1 through 6.4 are themselves obtained under a session-disjoint split: no welding session appears in both the training and test partitions. The main offline and real-time numbers are therefore already cross-session estimates. An additional grouped split constructed from the dataset's native directory structure provides a secondary sensitivity check on a different partition cut, and is summarized in Table A.4 in the Appendix. Across both cuts, binary defect detection remains strong (binary F1 in the 0.91–0.99 range), confirming that the screening stage is robust to the choice of grouping criterion. The two splits disagree, however, on multiclass macro-F1 leadership: Backbone Fusion leads under the main session-disjoint split, while Hierarchical Ensemble leads under the auxiliary configuration-grouped split. This dependence on the grouping criterion is a substantive sensitivity finding, since it implies that macro-F1 rankings on this dataset should not be treated as definitive on the basis of any single split; it is presented in the appendix table rather than reinterpreted as a competing main result.

6.4 Deployment-Oriented Model Selection

The three offline-trained approaches achieve binary defect F1 above 0.91 in both the offline and chunked-real-time protocols, and WeldFusionNet-RT exceeds 0.99 under its separate protocol (window-level, not directly comparable to recording-level metrics). Multimodal sensor fusion therefore provides a workable signal for weld quality diagnosis under cross-session conditions across the 12-class taxonomy. Multiclass macro-F1 spreads more widely (0.7459 to 0.8319 offline; 0.7475 to 0.8436 in chunked real-time; 0.9853 for WeldFusionNet-RT under its separate window-level protocol), reflecting that fine-grained discrimination is more sensitive to architectural and protocol choices than binary screening is.

Within the conditions evaluated in this thesis, WeldFusionNet is the strongest candidate for post-weld or end-of-recording quality decisions whenever binary defect detection is the primary criterion. It leads on binary defect F1 in both the offline (0.9895) and chunked-real-time (0.9886) protocols, on accuracy (0.8872 offline), on weighted F1 (0.8775 offline), and on ROC-AUC (0.9981). Its mean real-time inference latency of 72.2 ms (P95: 83.5 ms) is well within the 250 ms streaming budget, so the model is *real-time-feasible under replay* on the cross-session test partition. This is a feasibility result, not evidence of production-pipeline readiness in a live industrial setting: live deployment would require the additional engineering described in §8.1 (sensor synchronization, I/O latency, operator feedback, drift monitoring) and re-validation on the target line.

WeldFusionNet-RT applies only to continuous sub-second window-level scoring under recording-level supervision. It shares the WeldFusionNet architecture, is trained natively on 1-second sliding windows, and emits one window-level classification every 500 ms (the window stride) at a per-window inference latency of 3.2 ms. Its window-level macro-F1 of 0.9853

(window-level, not directly comparable to recording-level metrics) is not evidence of verified defect-onset localization, because the window labels are inherited from recording-level annotations. True defect-onset localization would require fine-grained temporal supervision and is identified as future work.

When multiclass macro-F1 is instead the primary criterion, Backbone Fusion becomes the stronger choice on point estimates, although the gap to WeldFusionNet is not statistically significant (§6.8.3). It leads on offline macro-F1 (0.8319) and chunked-real-time macro-F1 (0.8436), processes each 50-frame chunk in a mean of 127.5 ms (within the 250 ms budget), and remains competitive on binary detection (offline 0.9384, real-time 0.9698). The validation-fold macro-F1 of 1.0000 reported by Backbone Fusion and Hierarchical Ensemble during training reflects hyperparameter-selection bias and is not a generalization estimate (§6.8); the held-out test-session metrics govern the comparison.

Hierarchical Ensemble trails the two other offline models on every multiclass metric (offline macro-F1 0.7459; real-time macro-F1 0.7475) but retains a workable binary detection capability (offline 0.9163, real-time 0.9559). Its handcrafted feature pipeline is auditable and interpretable, which may be preferred in regulated industrial settings. The gap to the neural and pretrained-backbone approaches suggests that, for this dataset under cross-session conditions, the handcrafted-statistic representation does not encode the most discriminating patterns for the harder classes.

6.5 Physical Interpretation of Error Patterns

The quantitative results in Sections 6.1 and 6.2 identify class 05 (undercut), class 01 (excessive penetration), and class 10 (warping) as challenging defect categories under cross-session evaluation, with class 05 the dominant failure mode across all three offline-trained approaches. This section examines the physical reasons for these error patterns.

6.5.1 Sources of Confusion for Undercut (Class 05)

Undercut is a groove-shaped erosion of the parent material along the weld toe, formed when the molten pool fails to fully fuse with the surrounding base metal. From a process-monitoring perspective, undercut is observationally subtle: the defect is a geometric feature of the solidified bead rather than a transient process anomaly, and its formation conditions overlap with those of crater cracks (which form at the run terminus) and with marginal fusion conditions. The thermal camera observes pool temperature from above, but the toe-region erosion that defines undercut is partially out of the camera's preferred view and produces only subtle thermal signatures during active welding. Arc sound during the active phase does not distinguish undercut-prone runs from sound runs reliably, since the pool may behave normally for most of the weld. The top-confusion analysis in Table A.5 confirms this overlap, with an asymmetric pattern across model families that is quantified below. Even WeldFusionNet, trained end-to-end on the joint audio and thermal-video signal, recovers class 05 only partially (F1 = 0.091).

The per-class confusion neighborhoods (Table 6.6) quantify this overlap. Hierarchical

Ensemble routes 100% (20/20) of class-05 test runs to class 06 (overlap), collapsing the undercut category entirely; both Backbone Fusion and WeldFusionNet instead push 13–15 of 20 class-05 runs to class 11 (crater cracks), matching the *toe-region terminal artifact* hypothesis. The corresponding precision–recall curves (Figure 6.8) show that no model achieves both high precision and high recall on class 05: Backbone Fusion attains perfect precision (1.000) at very low recall (0.350); WeldFusionNet has a steep curve collapsing near-zero recall. Figure 6.9 shows the corresponding confidence distributions: the $p_{\text{class 05}}$ assigned to true-05 samples is barely separated from the distribution assigned to other classes, indicating that the decision boundary cannot be sharpened by threshold tuning alone.

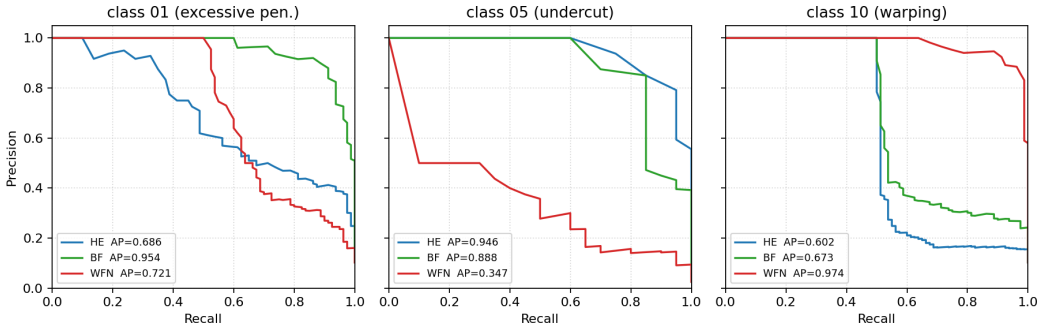

Figure 6.8: Per-class precision–recall curves on the test split for the three hardest classes (01 excessive penetration, 05 undercut, 10 warping). Class 05 is the only class where no model approaches the top-right corner.

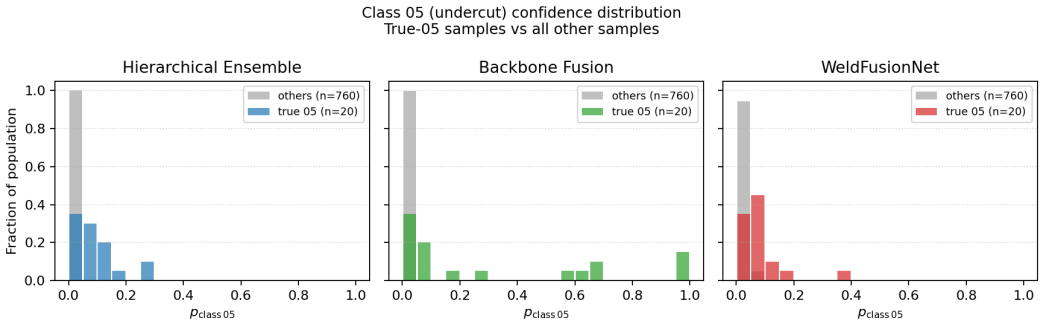

Figure 6.9: Distribution of $p_{\text{class 05}}$ on test samples. Colored: true class-05 samples (n=20). Gray: all other classes (n=760). For all three models the two distributions overlap strongly in the low-confidence region, leaving no threshold that cleanly separates them.

6.5.2 Sources of Confusion for Excessive Penetration (Class 01)

Excessive penetration occurs when the arc energy causes the weld pool to burn through the base material more than intended, creating a protruding root bead on the reverse side. From a process-monitoring perspective, this defect is difficult to separate from certain other energy-related defects because its primary physical signature, elevated heat input and deeper pool penetration, overlaps with early-stage porosity and burn-through conditions. The thermal camera captures pool temperature and geometry from above, making root-side penetration depth difficult to infer directly. Arc sound is also ambiguous: while excessive penetration

Table 6.6: Per-class F1 on the test split for the six classes most relevant to the class-05 (undercut) failure analysis. Class 05 is the primary failure mode; classes 06 and 11 are its most frequent confusion targets across models.

	Hierarchical Ensemble			Backbone Fusion			WeldFusionNet		
Class	P	R	F1	P	R	F1	P	R	F1
00 (good weld)	0.665	0.711	0.687	0.773	0.730	0.751	0.962	0.956	0.959
01 (excessive pen.)	0.706	0.300	0.421	0.873	0.775	0.821	0.976	0.512	0.672
05 (undercut)	0.000	0.000	0.000	1.000	0.350	0.519	0.500	0.050	0.091
06 (overlap)	0.667	1.000	0.800	1.000	1.000	1.000	1.000	1.000	1.000
10 (warping)	0.661	0.512	0.577	0.698	0.550	0.615	0.918	0.838	0.876
11 (crater cracks)	1.000	1.000	1.000	0.611	1.000	0.759	0.550	1.000	0.710

correlates with higher arc energy, so do other defects. This observational overlap explains the inter-class confusion between class 01 and adjacent energy-related defect types.

6.5.3 Asymmetric Performance on Crater Cracks (Class 11)

Crater cracks (class 11) attract a substantial inflow of misclassified undercut runs but are themselves recovered well by both Hierarchical Ensemble and the underlying detector structure of the other approaches (per-class F1 = 1.000 for Hierarchical Ensemble; 0.759 for Backbone Fusion; 0.710 for WeldFusionNet). Crater cracks are solidification cracks that form in the depression left at the weld termination point, arising from shrinkage stresses during the final stages of cooling. They are fundamentally a termination artifact: they form after the arc is extinguished, at the end of the weld run.

This late-formation characteristic creates an asymmetric observational pattern. When a crater crack is genuinely present, the terminal segment of the recording carries a distinctive signal that all three classifiers can detect, which explains the high recall on class 11. However, the same terminal-cooling region resembles undercut-prone runs that also exhibit irregularities at the weld toe, which explains the inverse confusion: many undercut runs are mistakenly routed to the crater-cracks class. The acoustic and thermal signals during the active welding phase do not strongly differentiate crater-cracks runs from sound runs, so the diagnostic information is concentrated in the post-arc decay.

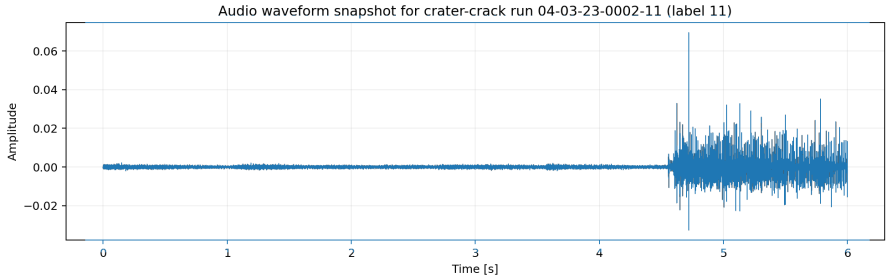

Figure 6.10: Audio waveform of a representative crater-crack (class 11) recording. The arc-active phase (left portion) exhibits amplitude characteristics similar to normal welds; the post-arc decay at the run termination (right portion) is where the crack forms, leaving only a subtle acoustic signature distinguishable from the normal termination pattern.

6.5.4 Classes with Strong Discrimination

In contrast to the harder classes, all three models achieve high per-class F1 on class 03 (porosity), class 04 (porosity with excessive penetration), class 06 (overlap), class 08 (excessive convexity), and class 09 (spatter) in offline evaluation. These defect categories have sensor signatures that are consistently and reliably discriminated from all other classes under cross-session conditions, and the reliability is consistent with their formation mechanisms. Porosity produces volumetric gas pockets with characteristic thermal signatures during pool solidification; spatter generates discrete acoustic impulses outside the nominal arc-sound envelope; overlap produces a distinctive low-energy, wide-bead thermal profile; excessive convexity creates a characteristically tall, narrow pool geometry; and the combined porosity-with-excessive-penetration class shows the joint thermal-volume signature of both component failure modes. All of these mechanisms produce spatial or temporal patterns that are clearly separable from other classes at the recording level.

6.6 Modality Contribution and Fusion Weight Analysis

The weighted probability fusion in Backbone Fusion assigns weights to audio, thermal, and video modalities. The video modality carries the highest weight, determined by grid search on the validation set, providing indirect evidence that thermal spatial structure is a more direct manifestation of weld pool geometry, and therefore of defect formation, than arc sound. Arc sound reflects process dynamics that correlate with defects but with more variability and context dependence. The thermal frame sequence, while subject to emissivity and viewpoint effects, captures the spatial state of the pool more directly.

The matched-pair audio robustness in Section 6.3.1 provides a direct empirical test for WeldFusionNet. Zeroing the audio input on the matched 20-run subset reduces multiclass macro-F1 by 0.288 while the binary screening decision remains strong (binary F1 rises on this defect-majority subset, as reported there). Because this perturbation leaves the thermal-video stream intact and a thermal-zero ablation was not run, the sustained binary performance is consistent with thermal video carrying the bulk of the binary-screening signal on its own; the data also establish that the audio stream carries non-redundant multiclass information that the video stream alone cannot reconstruct. The matched-pair noise stress in the same section reinforces this asymmetry: additive noise at either tested SNR depresses macro-F1 by at most 0.078, much less than the 0.288 drop produced by complete audio removal, indicating that the network degrades gracefully under partial acoustic contamination but depends substantially on the audio channel when it is fully absent.

6.7 Offline-to-Real-Time Performance Transfer

A key question in any monitoring system evaluation is whether offline predictive quality translates reliably to real-time operational performance. For Hierarchical Ensemble, offline multiclass macro-F1 of 0.7459 compares to real-time macro-F1 of 0.7475, and for Backbone Fusion, offline 0.8319 compares to real-time 0.8436. For these two models, however, the

real-time multiclass label is produced by a full-recording feature extraction performed at the end of the stream (Section 6.2), so near-parity with the offline numbers is expected: it confirms that the replay harness reproduces the offline multiclass pipeline, not that chunk-level inference matches full-recording quality. The more informative multiclass transfer test is WeldFusionNet, whose real-time multiclass output is genuinely event-aggregated: offline macro-F1 of 0.8178 compares to real-time 0.8016 (the latter computed on 240 sampled recordings rather than the full 780), a modest degradation indicating that averaging event-level posteriors approaches, but does not fully match, full-recording multiclass quality.

The binary F1 numbers transfer more cleanly than they did in earlier protocol mixes. WeldFusionNet maintains 0.9886 in chunked real-time (down from 0.9895 offline, a 0.001 drop), Backbone Fusion improves to 0.9698 from 0.9384 offline, and Hierarchical Ensemble improves to 0.9559 from 0.9163. The real-time improvement for Backbone Fusion and Hierarchical Ensemble reflects the benefit of averaging per-chunk defect probabilities before thresholding ($\bar{p}_i \geq \tau_r = 0.70$), which damps single-chunk errors and produces a more reliable recording-level decision than the offline classifier operating on a single aggregated feature vector.

6.8 Statistical Robustness of Performance Estimates

The point estimates reported above derive from a single held-out test partition of 780 runs. This section quantifies how much of the inter-model variation is real signal and how much could be sampling noise. Three complementary tests are applied: bootstrap confidence intervals on each metric, McNemar's test on the binary defect task, and a paired bootstrap on the multiclass macro-F1 difference between every model pair. All resampling uses the same set of 1000 paired indices generated from a fixed random seed so that comparisons across models share the same draws.

6.8.1 Bootstrap Confidence Intervals

Table 6.7 reports 95% bootstrap confidence intervals for each model on the four headline metrics, and Figure 6.11 visualizes the macro-F1 intervals.

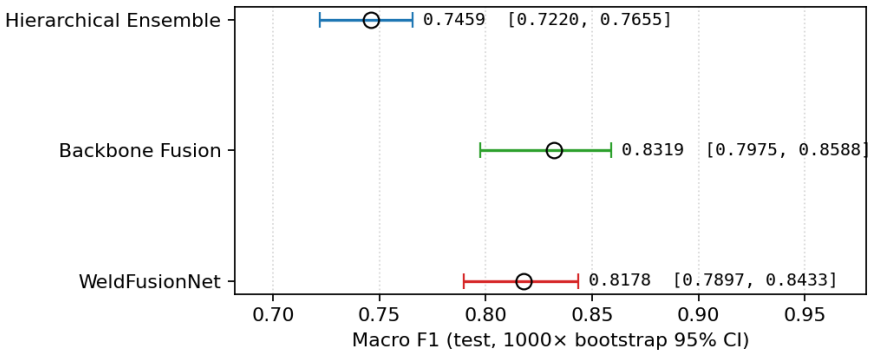

Figure 6.11: Bootstrap 95% confidence intervals on macro-F1 (1000 paired resamples).

Table 6.7: Bootstrap 95% confidence intervals on the test split (780 samples, $B = 1000$ resamples with paired indices across models). Point estimate followed by [lo, hi]. Split into two stacked panels (macro/binary F1; accuracy/weighted F1) to keep the row content readable.

Model	Macro-F1	Binary F1 (defect)
Hierarchical Ensemble	0.7459 [0.7220, 0.7655]	0.9163 [0.9000, 0.9323]
Backbone Fusion	0.8319 [0.7975, 0.8588]	0.9384 [0.9237, 0.9523]
WeldFusionNet	0.8178 [0.7897, 0.8433]	0.9895 [0.9835, 0.9951]

Model	Accuracy	Weighted F1
Hierarchical Ensemble	0.7744 [0.7436, 0.8026]	0.7551 [0.7205, 0.7871]
Backbone Fusion	0.8372 [0.8103, 0.8629]	0.8340 [0.8062, 0.8604]
WeldFusionNet	0.8872 [0.8641, 0.9090]	0.8775 [0.8519, 0.9021]

The intervals are tight (full width ≈ 0.04–0.05 macro-F1, ≈ 0.02–0.04 binary F1) and show a clear separation between Hierarchical Ensemble and the two deep-feature methods on macro-F1. The Backbone Fusion and WeldFusionNet intervals on macro-F1 overlap substantially.

6.8.2 McNemar's Test on the Binary Defect Task

McNemar's exact test (asymptotic χ^2 when $b + c \geq 25$) tells whether the per-sample disagreement between two models on the binary defect task is symmetric. Table 6.8 and Figure 6.12 report results.

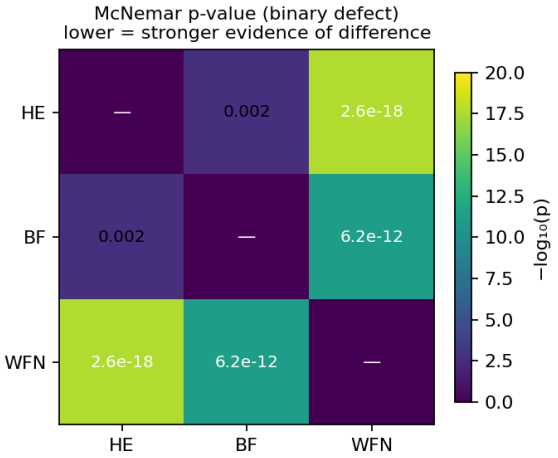

Figure 6.12: McNemar p-values (binary defect task) shown as $-\log_{10}(p)$. Larger (brighter) values indicate stronger evidence that two models differ.

WeldFusionNet differs significantly from both Backbone Fusion ($p \approx 6 \times 10^{-12}$) and Hierarchical Ensemble ($p \approx 3 \times 10^{-18}$) on the binary defect task. The Backbone Fusion vs Hierarchical Ensemble difference is also significant but at a much weaker level ($p \approx 2 \times 10^{-3}$). These results confirm that WeldFusionNet's binary-detection advantage is not a chance artifact.

Table 6.8: McNemar's test on the binary defect task (test split, 780 samples). b: model A correct & model B wrong; c: model A wrong & model B correct. Exact binomial when $b + c < 25$, asymptotic χ^2 otherwise.

Pair	b	c	Odds ratio b/c	p-value	Test
Hierarchical Ensemble vs Backbone Fusion	20	46	0.435	0.00209	asymp.
Hierarchical Ensemble vs WeldFusionNet	7	97	0.072	$2.61e-18$	asymp.
Backbone Fusion vs WeldFusionNet	10	74	0.135	$6.25e-12$	asymp.

6.8.3 Paired Bootstrap on Macro-F1 Differences

The bootstrap CIs in §6.8.1 treat each model independently. A paired bootstrap on the macro-F1 difference (same resample indices, two models per draw) gives a sharper test of whether one model is better than another at multiclass diagnosis on the same data. Table 6.9 and Figure 6.13 show the result.

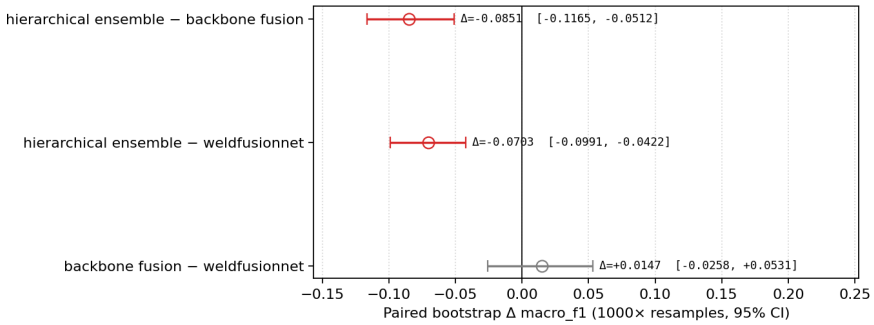

Figure 6.13: Paired bootstrap (1000 resamples) on Δ macro-F1 between every model pair. Red intervals exclude zero; gray intervals do not.

Table 6.9: Paired bootstrap (1000 resamples) on the difference in macro-F1 between model pairs. Significant differences (95% CI excludes 0) marked with *.

Pair	Δ macro-F1	95% CI lo	95% CI hi	p (two-sided)
Hierarchical Ensemble – Backbone Fusion*	−0.0851	−0.1165	−0.0512	< 0.001
Hierarchical Ensemble – WeldFusionNet*	−0.0703	−0.0991	−0.0422	< 0.001
Backbone Fusion – WeldFusionNet	+0.0147	−0.0258	+0.0531	0.428

The Hierarchical Ensemble vs Backbone Fusion and Hierarchical Ensemble vs WeldFusionNet gaps are both significant: Hierarchical Ensemble is reliably below either deep-feature method on macro-F1. *The Backbone Fusion vs WeldFusionNet macro-F1 gap is not statistically significant* ($\Delta = +0.015$, 95% CI $[-0.026, +0.053]$, $p \approx 0.43$). This qualifies the deployment recommendation: when the operational criterion is multiclass diagnosis, Backbone Fusion and WeldFusionNet are statistically tied on the current test partition, while WeldFusionNet remains the clear choice for binary screening.

6.8.4 Selection Bias from a Single Partition

A second reliability concern is selection bias: the specific composition of the test partition, determined by the session-disjoint split, may happen to include or exclude session identities that are easy or hard for certain model families. The auxiliary cross-session evaluation in the appendix (Table A.4) probes this concern by evaluating each model on a different partition cut with no session overlap with the training data. The validation-fold macro-F1 of 1.0000 reported by Backbone Fusion and Hierarchical Ensemble during training reflects grid-search hyperparameter-selection bias on the validation fold (the same fold is used to select hyperparameters and to score the resulting model) and is not a generalization estimate; the test-session values reported above are the appropriate generalization measure.

6.9 Calibration Quality

Each model emits class probabilities through a post-hoc calibration stage (§4.6). This section measures how closely those probabilities correspond to observed accuracy, a prerequisite for using them as confidence scores in human-in-the-loop triage. Table 6.10 reports Expected Calibration Error (ECE, 15 equal-width bins on the top-class confidence), Brier score, and Maximum Calibration Error (MCE) for the multiclass head and for the binary defect head.

Table 6.10: Calibration evaluation (15-bin ECE, Brier, MCE) on validation and test splits. Multiclass ECE uses top-class confidence; binary ECE uses p_{defect}.

		Multiclass (top-class)			**Binary defect**		
Model	**Split**	ECE	Brier	MCE	ECE	Brier	MCE
Hierarchical Ensemble	test	0.1165	0.3204	0.3245	0.0983	0.1083	0.4325
Hierarchical Ensemble	val	0.0564	0.1425	0.5783	0.0288	0.0222	0.6497
Backbone Fusion	test	0.1074	0.2370	0.5525	0.0883	0.0908	0.2987
Backbone Fusion	val	0.0000	0.0000	0.0000	0.1302	0.0342	0.4969
WeldFusionNet	test	0.1243	0.1884	0.4241	0.0249	0.0169	0.8401
WeldFusionNet	val	0.1216	0.1089	0.3635	0.0555	0.0525	0.7949

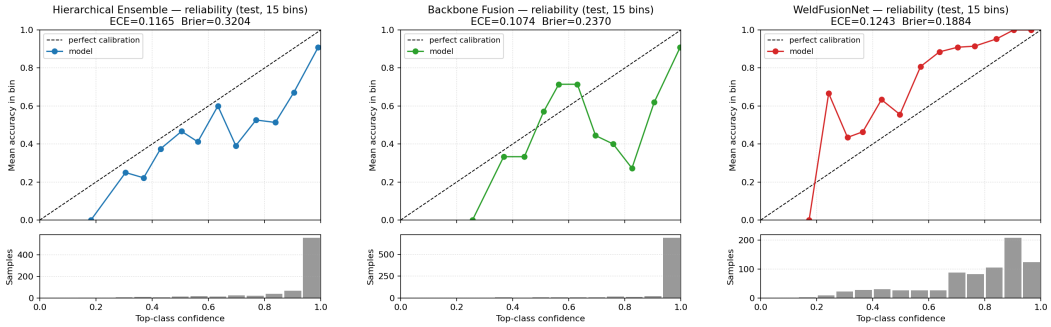

Figure 6.14: Multiclass top-class reliability diagrams on the test split (15 bins). Perfect calibration is the dashed diagonal. Bottom bars show the population per bin.

WeldFusionNet's binary head has the lowest binary ECE (0.025) and Brier (0.017) on test, consistent with the per-class temperature-scaling step in its post-hoc calibration

chain. Its binary MCE of 0.8401 nevertheless indicates that at least one confidence bin (the highest-confidence one, where predictions concentrate) carries a large local miscalibration; temperature scaling corrects average miscalibration but does not flatten every bin. The multiclass top-class confidences are systematically over-confident across all three models (multiclass test ECE $\approx$ 0.107–0.124), which is the canonical behavior of 12-class classifiers trained with focal loss or class-weighted cross-entropy. A practical consequence is that binary probabilities are better calibrated on average than multiclass top-class confidences for use as triage inputs, but the high binary MCE of WeldFusionNet indicates that threshold decisions should still be validated under the intended operating distribution before deployment rather than relied upon purely on the basis of low aggregate ECE.

6.10 Sanity Baselines

To bound how much the architectural choices actually contribute beyond what the input features alone make possible, five sanity baselines are trained on the same train/test split as the three production models. The baselines are: a majority-class predictor (always class 00), a stratified random predictor sampling from the training class priors, a multinomial logistic regression on the full 1196-feature input (audio + thermal handcrafted features concatenated with the ResNet18 video embeddings), a multinomial logistic regression on the 172 handcrafted features only (audio + thermal, no video embeddings), and a k-nearest-neighbors classifier (k=5, cosine distance) on the same 1196-feature input. This 1196-feature representation is a strict superset of the 172 handcrafted features but is distinct from the 417-dimensional input vector used by the Hierarchical Ensemble: the sanity baselines additionally include the ResNet18 video embeddings that Backbone Fusion uses, providing a controlled probe of how much of the discriminative signal is already present in those embeddings. Results are in Table 6.11 and Figure 6.15.

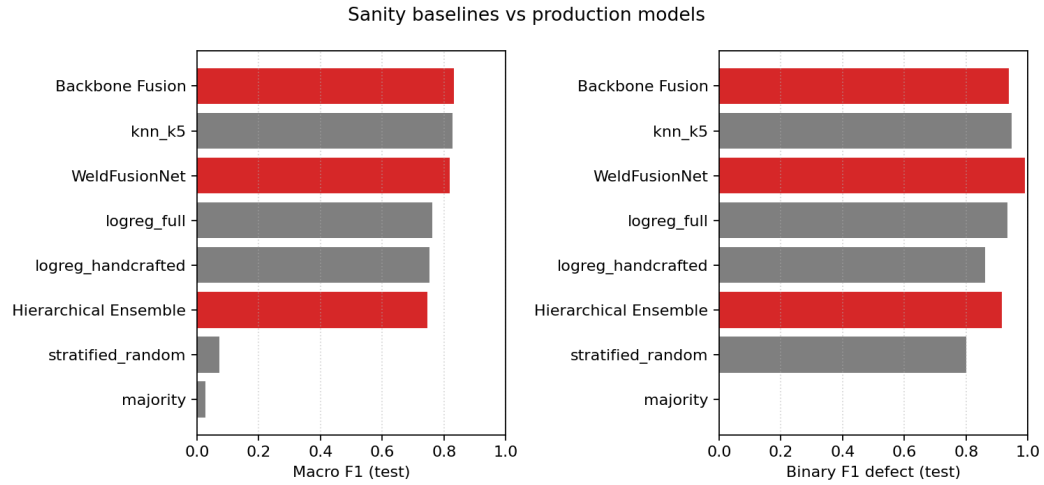

Figure 6.15: Sanity baselines (gray) against production models (red). The k-NN baseline on the 1196-feature input (172 audio+thermal handcrafted features concatenated with 1024 ResNet18 video embeddings; distinct from the 417-feature Hierarchical Ensemble input) reaches macro-F1 0.827, beating Hierarchical Ensemble and effectively tying Backbone Fusion.

Table 6.11: Sanity baselines on the test split (780 samples) compared to the three production models. Baselines train on the same train/test split. The 1196-feature input is the 172 audio+thermal handcrafted features concatenated with the 1024-dimensional ResNet18 video embeddings; it is a strict superset of the handcrafted features but is distinct from the 417-dimensional Hierarchical Ensemble input.

Kind	Method	Binary F1	Macro-F1	Accuracy	Weighted F1
baseline	Majority class (00)	0.0000	0.0282	0.2038	0.0690
baseline	Stratified random	0.8006	0.0720	0.0897	0.0933
baseline	LogReg (1196 feats)	0.9327	0.7628	0.8051	0.7924
baseline	LogReg (172 feats)	0.8620	0.7520	0.7077	0.6887
baseline	k-NN (k=5, cosine, 1196)	0.9465	0.8268	0.8449	0.8375
production	Hierarchical Ensemble	0.9163	0.7459	0.7744	0.7551
production	Backbone Fusion	0.9384	0.8319	0.8372	0.8340
production	WeldFusionNet	0.9895	0.8178	0.8872	0.8775

The sanity baselines above are oriented toward the multiclass task. For the binary defect-screening task, a complementary set of trivial baselines is reported in Tables 6.12 (quality view) and 6.13 (operational error counts): always-defect, always-normal, and binary-prior random. On this defect-majority test partition (defect = 621, normal = 159), an always-defect classifier reaches binary F1 = 0.886 with specificity = 0 and balanced accuracy = 0.5, while missing zero defects but producing 159 false alarms. The production models clear this trivial baseline by combining high recall with non-trivial specificity: WeldFusionNet misses 6 of 621 defects (FNR = 0.010) and reaches specificity = 0.956 (balanced accuracy = 0.973); Backbone Fusion misses 34 (specificity = 0.730) and Hierarchical Ensemble misses 57 (specificity = 0.711). For welding monitoring, the missed-defect count and balanced accuracy carry more operational meaning than binary F1 in isolation.

Table 6.12: Binary defect-screening baselines on the 780-run test partition (defect = 621, normal = 159) — quality view. The always-defect baseline reaches F1 = 0.886 with specificity = 0 and balanced accuracy = 0.5, demonstrating that high binary F1 alone is not evidence of a useful screening model on this defect-heavy split. Production-model rows use the same stage-1 binary-head evaluation as Table 6.1.

Kind	Method	Binary F1	Spec.	FPR	FNR	Bal. Acc.
baseline	Always defect (class ≠ 00)	0.8865	0.0000	1.0000	0.0000	0.5000
baseline	Always normal (class 00)	0.0000	1.0000	0.0000	1.0000	0.5000
baseline	Binary-prior random (p = 0.796)	0.7760	0.1635	0.8365	0.2303	0.4666
production	Hierarchical Ensemble	0.9163	0.7107	0.2893	0.0918	0.8095
production	Backbone Fusion	0.9384	0.7296	0.2704	0.0548	0.8374
production	WeldFusionNet	0.9895	0.9560	0.0440	0.0097	0.9732

The chance baselines (majority and stratified random) fall to macro-F1 0.03 and 0.07 respectively, confirming that the 12-class task is far above chance. A k-NN classifier on the 1196-feature input described above (handcrafted audio+thermal features and ResNet18 video embeddings, distinct from the 417-feature Hierarchical Ensemble input) achieves macro-F1 0.827, ahead of Hierarchical Ensemble (0.746) and within 0.005 of Backbone Fusion (0.832). Logistic regression on the same input lands at 0.763, also above Hierarchical Ensemble. Much

Table 6.13: Binary defect-screening baselines on the 780-run test partition — operational error counts view. "Missed defects" is the number of defective runs the model classified as normal (false negatives); "False alarms" is the number of normal runs the model flagged as defective (false positives). Same source as Table 6.12.

Kind	Method	Missed defects (FN)	False alarms (FP)
baseline	Always defect (class ≠ 00)	0	159
baseline	Always normal (class 00)	621	0
baseline	Binary-prior random ($p = 0.796$)	143	133
production	Hierarchical Ensemble	57	46
production	Backbone Fusion	34	43
production	WeldFusionNet	6	7

of the available discriminative signal therefore lives in the pretrained ImageNet ResNet18 video embeddings rather than in the architectural sophistication of the production models, and the gap between Hierarchical Ensemble and the better methods appears to be dominated by the way the ensemble handles the high-dimensional embedding space rather than by the feature design.

Chapter Summary

This chapter reported measured performance for all four model families under cross-session evaluation on the 780-run test partition. In the offline evaluation, WeldFusionNet leads on binary defect F1 (0.9895) and accuracy (0.8872), while Backbone Fusion leads on multiclass macro-F1 (0.8319) ahead of WeldFusionNet (0.8178) and Hierarchical Ensemble (0.7459). Under chunked real-time evaluation on 240 sampled recordings, WeldFusionNet leads on binary defect F1 (0.9886) with the lowest latency among the three offline models (72 ms mean), Backbone Fusion leads on multiclass macro-F1 (0.8436) with 127.5 ms latency, and Hierarchical Ensemble achieves 0.9559 binary F1 and 0.7475 macro-F1 with 151.7 ms latency. WeldFusionNet-RT, trained natively on 1-second windows, achieves window-level macro-F1 of 0.9853 (window-level, not directly comparable to recording-level metrics) with a per-window inference latency of 3.2 ms, making it the appropriate choice for continuous sub-second window-level scoring during the weld, subject to recording-level supervision; true defect-onset localization remains future work.

Physical interpretation identifies class 05 (undercut), class 01 (excessive penetration), and class 10 (warping) as the most challenging categories across all models, with the difficulty attributable to signal ambiguity rather than model architecture. Undercut is the dominant failure mode: Hierarchical Ensemble collapses 100% of class-05 runs to class 06 (overlap), and Backbone Fusion and WeldFusionNet push 65–75% to class 11 (crater cracks). Statistical robustness analysis (§6.8) shows that the Hierarchical Ensemble macro-F1 is reliably below both deep-feature methods (paired bootstrap CIs exclude zero), whereas the Backbone Fusion vs WeldFusionNet macro-F1 gap is not statistically significant ($\Delta = +0.015$, 95% CI $[-0.026, +0.053]$). WeldFusionNet's binary-detection lead, by contrast, is highly significant ($p \approx 6 \times 10^{-12}$ via McNemar). The sanity baselines (§6.10) show that a k-NN classifier on a

1196-feature input (172 handcrafted audio+thermal descriptors plus 1024 ResNet18 video embeddings, distinct from the 417-feature Hierarchical Ensemble input) reaches macro-F1 0.827, indicating that much of the discriminative signal lives in the pretrained ImageNet embeddings rather than in architectural sophistication. Calibration analysis (§6.9) identifies WeldFusionNet's binary head as the best-calibrated probability output (ECE = 0.025). Taken together, the results support WeldFusionNet when binary detection is the primary criterion, either Backbone Fusion or WeldFusionNet when multiclass macro-F1 governs selection (the two are statistically tied), WeldFusionNet-RT for continuous in-weld monitoring, and Hierarchical Ensemble where auditability of the feature pipeline outweighs the loss of multiclass quality.

Chapter 7

Discussion

7.1 Interpreting the Performance Tradeoff

The performance tables in Chapter 6 show consistent differentiation across all three fusion families, in both binary screening and multiclass diagnosis. The numbers themselves were established in the previous chapter; what remains to be examined is the structure of the differences, that is, what they reveal about the role of architectural choice and where the real performance limits lie.

A macro-F1 gap of comparable magnitude between the best and worst approach persists across both offline and real-time evaluations (0.086 offline, 0.096 real-time), which indicates that architectural choice has substantial impact on multiclass diagnosis under cross-session conditions. It rules out the null hypothesis that the handcrafted-feature ensemble is interchangeable with the deep-fusion approaches for 12-class defect diagnosis, although Backbone Fusion and WeldFusionNet are themselves statistically tied on macro-F1 (§6.8.3). At the same time, the offline binary F1 values (all three models above 0.91, with WeldFusionNet at 0.9895) show that defect *detection* is far less sensitive to the choice of fusion strategy than defect *classification*. This distinction has direct operational relevance: a facility that needs only a reliable go/no-go alarm has more architectural latitude than one that needs fine-grained defect-type information to guide corrective action [1, 6].

7.2 Methodological Implications

The two-stage design proved useful beyond its original motivation. It separates the question of whether a weld is defective from the question of which defect class is most plausible, which simplifies the interpretation of both strengths and failures. Without this separation, poor multiclass predictions can obscure whether the real weakness lies in defect screening or in defect differentiation among already detected defective runs.

A second methodological implication concerns evaluation realism. Grouped splitting and recording-level replay were necessary to produce estimates that approximate operational monitoring rather than static sample classification. Without leakage-aware partitions and runtime-aware evaluation, the model ranking would likely favor methods that perform well only in less constrained or partially overlapping settings. The group-level split by

configuration identity ensures that no recording configuration appears in both train and test, so the reported metrics reflect generalization to configurations the model has not encountered rather than accuracy within the training distribution.

7.3 Current Limitations

Several limitations constrain the conclusions of this work. Each is examined here in sufficient depth to characterize its potential impact.

Single dataset snapshot. The analysis is tied to one frozen snapshot of the Intel Robotic Welding Multimodal Dataset. All conclusions about model ranking, feature importance, and modality contribution are valid for the data distribution observed in this snapshot. Whether they generalize to other welding processes, other robotic cells, different base materials, or different filler transfer modes is an open question. In particular, the defect class frequencies in this dataset reflect the experimental conditions under which it was collected, not the distribution of defects that would be observed in a production environment. A model optimized for this snapshot's class distribution may require rebalancing or retraining before deployment in a facility with a different defect profile.

Class imbalance and minority class reliability. Some defect codes in the dataset are represented by substantially fewer runs than others. While the grouped split ensures that every class appears in the test partition, the per-class F1 estimates for minority classes are based on small sample counts. A single misclassification can change a minority-class F1 by 5–15 percentage points depending on its test count. The reported macro-F1 is therefore more reliable as a summary metric than as an accurate reflection of individual class performance for the least-represented categories. This limitation is inherent to datasets with naturally imbalanced class frequencies and can only be mitigated by collecting more labeled examples for underrepresented defect types.

Recording-level formulation and within-run dynamics. The present thesis treats each recording as a single classification unit and assigns a single label to each run. This recording-level formulation is appropriate for the dataset as provided, but it discards within-run temporal dynamics that could carry additional diagnostic information. A run that begins with normal welding and transitions to a defect condition partway through is labeled with the defect class but may exhibit a normal signal in its early segments. The recording-level aggregation used in the real-time protocol handles this by averaging evidence across the run, but it cannot localize the defect onset within the recording.

Summary-based feature representations. The handcrafted feature vectors used by the Hierarchical Ensemble are based on statistical summaries computed over the full recording. These summaries discard the temporal ordering of the signal and cannot represent how the audio spectrum or thermal intensity evolves over the course of the weld. Two recordings with the same overall spectral statistics but different temporal evolution would be assigned identical

handcrafted feature vectors, even if their defect patterns differ in timing or progression. The backbone and end-to-end approaches partially address this by operating on frame-level embeddings, but they too aggregate over time before classification.

Evaluation hardware and latency generalizability. The latency measurements reported in Chapter 6 were obtained on a specific CPU/GPU hardware configuration. The Hierarchical Ensemble's mean chunk-inference latency of 151.7 ms reflects STFT, MFCC extraction, and thermal statistics computation running sequentially on a single CPU core per 50-frame chunk. This comfortably clears the 250 ms boundary implied by the deployment scheduling target. On a modern server-class CPU with SIMD vectorization (e.g., Intel Xeon with AVX-512), the floating-point-intensive STFT and MFCC computation could run 2–4× faster, providing additional headroom.

Backbone Fusion achieves the second-fastest per-chunk latency at 127.5 ms. Its classifier stage uses lightweight MLP heads operating on pre-extracted ResNet18 video and log-mel spectrogram audio embeddings (trained with AdamW and class-weighted cross-entropy), not a tree-based gradient-boosting head. WeldFusionNet processes each replay inference event in 72.2 ms, the fastest of the three offline models in streaming mode, because the network runs a single GPU forward pass per window rather than aggregating chunked predictions. GPU batch inference would reduce the latency of all neural approaches further, with an estimated 10–30 ms for batch sizes of 8–16, but the relative ordering between approaches would not change. All three approaches satisfy real-time constraints for their intended operational modes, so latency discriminates between streaming and end-of-recording deployment patterns without disqualifying any single approach.

WeldFusionNet-RT split incompatibility. The window-level WeldFusionNet-RT results are reported on a configuration-grouped split, not the session-disjoint split used for the three offline models. Although both splits enforce strict group disjointness between train and test, the configuration grouping produces a different test composition. As a result, WeldFusionNet-RT's 0.9853 window-level macro-F1 is not directly comparable to the recording-level macro-F1 of the other three approaches on the 780-run session split, and the real-time comparison table reports the two regimes side-by-side only as a latency-versus-quality reference. A like-for-like cross-session evaluation of WeldFusionNet-RT is identified as immediate future work.

> **Metric-level caveat.** Recording-level metrics summarize one decision per weld run after event aggregation or post-run diagnosis. WeldFusionNet-RT metrics summarize individual overlapping windows whose labels are inherited from the parent recording. Because the evaluation unit, split criterion, and statistical dependence structure differ, window-level scores are evidence of streaming feasibility, not direct evidence that WeldFusionNet-RT outperforms recording-level models.

Asymmetric pretraining across modalities. In Backbone Fusion, the video branch uses a ResNet18 backbone pretrained on ImageNet, while the audio branch uses a log-mel spectrogram projected through a seeded random matrix with no pretraining. The choice

follows from the absence of a welding-specific pretrained acoustic backbone, but it nonetheless makes the audio–video comparison within Backbone Fusion structurally asymmetric: video benefits from external supervision, audio does not. Substituting a pretrained audio model (e.g. VGGish, YAMNet, PANNs, wav2vec) would equalize the modalities and is a natural next experiment.

Hyperparameter selection on a single validation fold. Model selection, calibration fitting, and threshold tuning all rely on the same validation partition. Although the validation set is session-disjoint from training and test, repeated use of the same fold for selection introduces a mild optimistic bias. A nested or k-fold cross-session selection scheme would tighten this and is left as future work.

No isolated component ablations. The thesis compares three architecturally distinct fusion families but does not isolate the contribution of individual components within a family (e.g. focal loss vs cross-entropy in WeldFusionNet, attention pooling vs mean pooling over video frames, gated fusion vs concatenation, fraction of MobileNetV3 frozen). The relative importance of each design choice is therefore not directly quantified.

Single-modality and modality-removed baselines. The robustness analysis (§6.3) zeroes individual modalities at inference but the model is still trained on the full multimodal input. A clean modality contribution analysis would also train audio-only, thermal-only, and process-only variants; only the WeldFusionNet no-extra-modality ablation is reported here.

Model interpretability artifacts. The Hierarchical Ensemble is discussed as the interpretable baseline of the three families, but quantitative interpretability artifacts (SHAP and permutation feature importance for the tree ensemble, Grad-CAM activations for the video branches, and learned attention weights in WeldFusionNet) are not produced in the current version. They are natural next outputs and would directly support the deployment discussion of confidence-gated human-in-the-loop triage.

Code and data release statement. The training, evaluation, and post-hoc analysis code accompanies the thesis, but a public release of the dataset is governed by the data owner. A reproducibility appendix listing exact dependency versions, random seeds, and a container image is identified as a follow-up artifact.

7.4 Critical Examination of Model Performance

The macro-F1 values reported in this thesis (0.7459–0.8319 offline, 0.7475–0.8436 chunked real-time, and 0.9853 for WeldFusionNet-RT under its separate window-level protocol) represent fair-to-strong performance for a 12-class classification problem under cross-session evaluation. A critical reading of these results must address three questions: what drives the inter-model variation, what limits performance from reaching higher levels, and what structural factors are consistent with these numbers.

7.4.1 Explaining Inter-Model Variation and Performance Levels

Several factors help explain both the inter-model gap and the absolute performance levels.

Representation capacity and learning paradigm. Under cross-session evaluation, the Backbone Fusion approach (0.8319 macro-F1) leads on multiclass discrimination, with WeldFusionNet (0.8178) within 0.014 macro-F1 and Hierarchical Ensemble (0.7459) trailing by 0.086. The relatively flat ranking between Backbone Fusion and WeldFusionNet indicates that both end-to-end learning of joint multimodal representations (audio and thermal-video in WeldFusionNet) and pretrained-backbone features combined with a tree ensemble can capture the discriminative information present in this dataset under session shift. Backbone Fusion's slight edge reflects more uniform per-class behavior: it does not collapse on any single class, while WeldFusionNet has near-zero F1 on class 05 (undercut). The 0.086 gap between Backbone Fusion and Hierarchical Ensemble suggests that the handcrafted summary statistics used by Hierarchical Ensemble omit information that the deep-feature pipelines retain, particularly for the harder classes.

Difficulty of the 12-class problem. Cross-session macro-F1 of 0.75–0.83 on 12 classes must be read against the fact that the models are evaluated on welding sessions they have never seen. The grouped split prevents any session from appearing in both train and test, creating a genuine generalization challenge. A macro-F1 above 0.83 means that Backbone Fusion correctly diagnoses the defect type in the great majority of recordings from unseen sessions; the gap to a hypothetical 1.0 is concentrated in two or three classes rather than distributed across the taxonomy.

Class-specific difficulty. As documented in Chapter 6, class 05 (undercut) is the primary source of misclassification across all three models under cross-session evaluation, with class 01 (excessive penetration) and class 10 (warping) as secondary failure modes. The macro-F1 is depressed by these challenging classes; without class 05 in particular, the per-class F1 distribution is notably higher for the remaining classes. This concentration suggests that targeted improvements to how these classes are represented or modeled could substantially raise the macro-F1.

Relative robustness of the binary detection stage. Binary defect F1 (0.9163–0.9895 offline; 0.9559–0.9886 chunked real-time) is consistently higher than multiclass macro-F1, which reflects that the binary defect/good boundary is more reliably separated than the fine-grained defect taxonomy. This ordering is expected and operationally reassuring: the system reliably flags defective welds even when it cannot precisely diagnose the defect type.

7.4.2 What Would Falsify These Results?

The most informative test of the reported performance would be a true out-of-distribution evaluation, applying the trained models to recordings from a different robotic welding cell, a different base material, or a different defect induction protocol. If performance degrades substantially under these conditions, the models have learned configuration-specific patterns rather than generalizable defect physics. If performance remains stable, the cross-session evaluation protocol would be validated as a reliable proxy for true generalization.

A second test is deliberate data augmentation: artificially varying the camera viewpoint,

injecting calibrated noise, or simulating thermal emissivity changes. The robustness evaluation in Section 6.3 provides a partial version of this test, and its results characterize how fragile or robust the baseline performance is under sensing perturbations.

The cross-session evaluation that defines the main test partition in Chapter 6 constitutes the closest available approximation to the first test within the present scope. The three offline models (Hierarchical Ensemble, Backbone Fusion, WeldFusionNet) are evaluated on a partition of welding sessions disjoint from those used for training; WeldFusionNet-RT is evaluated under a separate configuration-grouped window-level split, so its cross-session generalization is not directly tested by the same protocol. Binary defect F1 lands in the 0.9163–0.9895 range across the three offline-trained models, confirming that defect detection generalizes well across sessions; multiclass macro-F1 in the 0.7459–0.8319 range shows the more substantial drop in fine-grained classification quality that signals genuine distribution shift. WeldFusionNet-RT's window-level binary F1 of 0.9946 and macro-F1 of 0.9853 (window-level, not directly comparable to recording-level metrics) are reported separately and should be interpreted as performance on the configuration-grouped window-level split, not as a stronger cross-session generalization claim. The overall picture is consistent with sound methodology: the models have not trivially overfit to the specific partition, but meaningful generalization work remains. A true multi-cell or multi-material evaluation would be the definitive falsification test; the cross-session result here is a necessary but not sufficient substitute.

Multi-cell evaluation was not completed within the present scope and is identified as a high-priority direction for future work.

7.5 Robustness and Generalization Considerations

Potential robustness risks include ambient acoustic contamination, thermal viewpoint variability, class overlap in visually similar defects, and missing-modality conditions. These risks are consistent with the broader welding-monitoring literature, which repeatedly highlights process variability and sensor-context dependence as major barriers to reliable transfer [4, 5].

The robustness replay makes these risks measurable rather than speculative. The matched-pair audio-noise and missing-audio results on the production WeldFusionNet checkpoint (Table 6.5, evaluated on the same 20-run subset) characterize the model's sensitivity to acoustic contamination and to complete audio loss. The appendix provenance note (§A.4) acknowledges an earlier late-fusion auxiliary-sensor model variant but excludes its quantitative results; it is *not* used to support current claims about production WeldFusionNet robustness. WeldFusionNet is the strongest approach on binary detection and accuracy (binary F1 0.9895, multiclass accuracy 0.8872) on the full session-disjoint test partition. The relative contribution of audio ($\alpha = 0.25$) and video ($\alpha = 0.75$) in the Backbone Fusion weighting scheme provides a hypothesis about which modality loss should be most damaging.

The auxiliary cross-session split provides a second perspective on generalization. The main test partition contains 780 runs from welding sessions disjoint from training, and the auxiliary partition cut summarized in Table A.4 of the Appendix probes whether the offline performance is stable across different cross-session compositions or dependent on the specific

partition.

7.6 Implications for Practical Deployment

The deployment recommendation depends on the operational mode and the relative weight placed on binary screening versus fine-grained classification. For end-of-recording or post-weld diagnosis when binary defect detection governs selection, WeldFusionNet is the preferred choice: it leads on binary F1 (0.9895 offline, 0.9886 real-time), accuracy (0.8872), weighted F1 (0.8775), and ROC-AUC (0.9981), and its 72 ms per-replay-event inference latency is acceptable when inference runs once per weld completion. When the multiclass macro-F1 is the primary criterion, Backbone Fusion is preferred instead, since it leads on macro-F1 in both offline (0.8319) and chunked-real-time (0.8436) protocols at 127.5 ms per chunk. For continuous in-weld window-level scoring at sub-second granularity, under recording-level supervision, WeldFusionNet-RT achieves window-level macro-F1 of 0.9853 (window-level, not directly comparable to recording-level metrics) at 3.2 ms per window. Hierarchical Ensemble trails on both binary detection and multiclass macro-F1 (0.9163 binary F1, 0.7459 macro-F1 offline) but retains the auditability of a handcrafted feature pipeline.

The operationally important implication is not the specific numbers, which were reported in Chapter 6, but the structure of the tradeoff: under cross-session evaluation, no single model dominates on every metric, so the choice of preferred model depends on the operational priority of binary screening versus multiclass diagnosis and on whether decisions are emitted per chunk or per recording.

The remaining macro-F1 gap to 1.0 is concentrated in class 05 (undercut), class 01 (excessive penetration), and class 10 (warping); broader validation under additional noise conditions, sensor-placement changes, and process variants would be required before industrial deployment.

7.7 Safety, Responsibility, and Operational Risk

Deploying machine learning models in manufacturing quality assurance introduces responsibilities that go beyond predictive accuracy. Three dimensions of operational risk arise from the performance regime studied in this thesis.

7.7.1 Error Rates and Concentrated Failure Modes

WeldFusionNet achieves overall accuracy of 0.8872 on the 780-run cross-session test set, corresponding to approximately 88 expected classification errors across that partition. Backbone Fusion at 0.8372 accuracy produces approximately 127 expected errors, and Hierarchical Ensemble at 0.7744 accuracy produces approximately 176 expected errors. Error rates at this level translate into real operational risk if the system is deployed without appropriate human oversight.

The error analysis further shows that errors are concentrated: classes 05 and 01 account for a disproportionate share of misclassifications, while other classes are classified more

reliably. This concentration means that the effective error rate for undercut (class 05) in a production environment is substantially higher than the aggregate figure suggests.

In a welding quality control context, a misclassification at stage 2 (wrong defect type identified) is less severe than a missed defect at stage 1. All three approaches achieve adequate binary recall on the cross-session test partition (Hierarchical Ensemble 0.9082, Backbone Fusion 0.9452, WeldFusionNet 0.9903), meaning stage 1 misses are uncommon for WeldFusionNet in particular and acceptable for the two tree-based models. However, if a deployment operator relies on the specific defect code from stage 2 to determine the corrective action, for example adjusting shielding gas flow to address porosity, then a misclassification of porosity as a different defect type could lead to an incorrect corrective action. This potential consequence should be documented in the deployment specification and mitigated through human review for low-confidence predictions.

7.7.2 Confidence Scores as Operational Tools

One practical consequence of the probability calibration step (isotonic regression for the Hierarchical Ensemble, temperature scaling for Backbone Fusion and WeldFusionNet) is that the model's output probabilities can be used as confidence scores to trigger human review. A recording for which the top-class posterior is below a threshold (for example, $p < 0.85$) can be flagged for human inspection rather than automatically passed to the corrective action system. This human-in-the-loop design pattern substantially reduces the operational risk from misclassifications, at the cost of human review time for low-confidence predictions.

For WeldFusionNet, temperature scaling aligns the predicted probabilities more closely with observed accuracy, which makes threshold-based triage more meaningful than it would be with a miscalibrated model. This correspondence between confidence and accuracy is the defining property of a calibrated model and is essential for reliable threshold-based triage.

7.7.3 Monitoring System Failure Modes and Mitigation

In addition to individual prediction errors, a deployed monitoring system can fail in several systemic ways that are not captured by per-sample accuracy metrics.

Distribution shift. If the production distribution of welding runs drifts away from the training distribution, for example due to tool wear, material batch changes, or environmental conditions, model performance may degrade silently. Unlabeled drift detection, using statistical tests on model confidence distributions or activation statistics, can detect that the input distribution has changed even without labels, providing an early warning that the model may need retraining.

Sensor degradation. Gradual sensor degradation (camera lens fouling, microphone frequency response change) shifts the input distribution in a predictable direction. A monitoring system that checks the health of incoming modality signals, for example by tracking rolling statistics of frame brightness or audio RMS, can detect sensor issues before they translate into classification errors.

Label-definition drift. In production environments, the operational meaning of defect codes may change over time as quality engineers revise acceptance criteria or process

engineers modify defect-induction protocols. A model trained on older label conventions may no longer reflect the current quality standard. This type of drift is particularly insidious because it is invisible to the model.

These failure modes argue for a deployment architecture that includes not only the classification model but also an operational monitoring layer that tracks model confidence, sensor health, and prediction distribution over time.

7.8 Comparison of Reported Performance with Related Work

The macro-F1 values reported here (0.7459–0.8319 across three approaches on 12 classes under cross-session evaluation) fall in a range that is fair to strong relative to the welding monitoring literature, with the recognition that cross-session evaluation typically reports lower values than within-session evaluation does. This section places the reported results in context.

7.8.1 Quantitative Comparison with Prior Studies

Table 3.1 in Chapter 3 summarizes selected prior studies. Most prior acoustic and thermal monitoring studies report metrics on binary or small-scale classification problems (two to four process states) rather than on a 12-class defect taxonomy, making direct numerical comparison difficult. Zhang et al. report "real-time monitoring" capability but do not provide a numerical F1 or macro-F1. Jiao et al. report F1 and accuracy on GMAW process-state recognition but evaluate on a binary or ternary classification problem under controlled laboratory conditions. Griffin et al. report a detection rate for acoustic emission on MAG defects without specifying the number of classes or the exact evaluation protocol.

The study most directly comparable in scope is Stemmer et al., which addresses multi-class welding defect detection on the same Intel Robotic Welding Multimodal Dataset used in this thesis [41]. Their unsupervised anomaly detection approach reports anomaly scores rather than multi-class F1, which precludes direct comparison. The present supervised approaches also benefit from full label information during training, whereas the unsupervised approach of Stemmer et al. does not.

7.8.2 Structural Differences That Explain Performance Levels

Several structural differences between the present thesis and the broader related work literature are relevant to interpreting the reported numbers. Each structural advantage also carries a corresponding generalizability limitation that bounds how broadly these results can be transferred.

Dataset scale and class coverage. The present study uses 3841 labeled runs covering all 12 classes (one normal class plus eleven defect classes), with 780 cross-session test runs ensuring per-class evaluation. Most published studies use substantially smaller datasets with fewer defect types. However, this larger scale is specific to the defect-induction protocol of this robotic cell. Facilities with different defect profiles, fewer induction cycles, or different

process variants may find that the effective labeled set for their conditions is much smaller, limiting transferability of the trained models and reducing the class coverage advantage.

Controlled induction protocol. The Intel Robotic Welding Multimodal Dataset was collected under controlled defect-induction conditions, producing reliably labeled examples with known defect causes. The corresponding limitation is that models trained on controlled-induction data are optimized for defect signatures that appear when defects are intentionally induced. Opportunistically collected production data, where defects emerge from gradual process drift, material inconsistency, or tool wear, may exhibit different signal characteristics. Performance on such data is an open question that the present evaluation cannot answer.

Supervised versus unsupervised learning. Because all three approaches are fully supervised, they can exploit the label structure directly. This advantage is conditional on label availability: the approach assumes that every training recording has been assigned a correct defect code. In production environments where defects are unlabeled or retrospectively labeled (and therefore noisy), this supervision advantage is partially or fully lost.

Cross-session evaluation. The grouped split by session identity measures cross-session generalization and produces lower but more honest estimates than within-session splits. However, "cross-session" here means different welding-session identity groups within the same dataset, not a different welding cell or process. The evaluation therefore quantifies within-dataset generalization, not cross-facility generalization, and the auxiliary cross-session cut summarized in the Appendix gives a concrete indication of how sensitive the results are to alternative partition compositions.

7.8.3 Implications of the Comparison

The macro-F1 of 0.8319 achieved by Backbone Fusion under cross-session evaluation, and the binary F1 of 0.9895 achieved by WeldFusionNet under the same protocol, are credible results for a 12-class problem on an unseen-session split. They provide evidence that multimodal sensor fusion (audio and thermal video across all four models) captures defect-discriminative information that transfers across welding sessions, while also confirming that concentrated per-class difficulty (class 05 undercut in particular) remains the primary target for further improvement.

7.9 Resource-Aware Deployment Considerations

Beyond the latency results reported in Chapter 6, the three approaches differ in several resource dimensions that are relevant to deployment planning.

7.9.1 Memory and Storage Requirements

Backbone Fusion uses a ResNet18 backbone (approximately 11.7 million parameters) and log-mel spectrogram-based audio embeddings, plus two MLP classifier heads. WeldFusionNet at 1.08 million parameters is substantially more compact. The Hierarchical Ensemble requires storing the full feature extraction pipeline alongside the XGBoost and ExtraTrees model states.

For deployment on edge hardware with constrained memory, WeldFusionNet's smaller parameter count is advantageous given that it also achieves the best binary classification and accuracy. Backbone Fusion's ResNet18 backbone requires more memory but achieves the best multiclass macro-F1 and is better supported by hardware acceleration frameworks.

7.9.2 Training Data Requirements

The Hierarchical Ensemble is the most data-efficient approach in this study: handcrafted features provide strong inductive bias, and tree ensembles can generalize with less data than deep networks. If the labeled dataset were substantially smaller (for example, 100–200 runs), the Hierarchical Ensemble would likely outperform the neural approaches. This data-efficiency advantage is relevant for deployment in facilities where labeled defect data is scarce.

WeldFusionNet requires the most data among the three approaches because it must learn representations from scratch for the non-pretrained components. Backbone Fusion's use of ImageNet-pretrained features mitigates this requirement but the MLP heads still require sufficient labeled data.

7.9.3 Maintenance and Updating

In production settings, the distribution of welding runs may drift over time as tooling wears, materials change, or process parameters are adjusted. For the Hierarchical Ensemble, retraining involves re-running feature extraction and refitting tree ensembles, which is fast. For the neural approaches, retraining involves fine-tuning or retraining from scratch, requiring GPU access. Backbone Fusion is the most straightforward to update because the backbone weights (pretrained on ImageNet) do not need to be retrained; only the classifier heads need updating. WeldFusionNet's compact size makes it faster to retrain in full than Backbone Fusion, which may offset the need for backbone retraining in practice.

Chapter Summary

The central finding of this chapter is that architectural choice matters substantially for multiclass diagnosis but much less for binary detection under cross-session evaluation. All three offline-trained models reach binary F1 above 0.91 on the good/defect boundary (WeldFusionNet 0.9895, Backbone Fusion 0.9384, Hierarchical Ensemble 0.9163), while macro-F1 spans 0.7459–0.8319 with Backbone Fusion leading. WeldFusionNet retains the lead on binary detection, accuracy, and weighted F1, while Backbone Fusion leads on macro-F1; the choice between them depends on the operational priority of binary screening versus fine-grained classification. The class that drives most of the remaining error, undercut (05), has an identifiable physical explanation in its subtle thermal and acoustic signatures, which resemble crater cracks at the weld toe, and represents the clearest target for future improvement together with class 01 (excessive penetration). Chapter 8 draws the final conclusions and outlines the most productive directions for future work.

Chapter 8

Conclusion and Future Work

This thesis examined whether multimodal thermal-acoustic analysis can support real-time welding-defect classification under conditions that approximate deployment constraints. The investigation covered one industrial dataset, three architecturally distinct offline fusion approaches, and a window-trained streaming variant (WeldFusionNet-RT). Within that scope the experimental evidence supports a positive answer, with qualifications concerning which models are viable, under what hardware conditions, and where the remaining failure modes are concentrated.

The first research question asked whether thermal and acoustic features can support reliable binary defect detection at recording level. Under cross-session evaluation, all three offline-trained approaches exceed 0.91 binary F1 on the 780-run test set: WeldFusionNet reaches 0.9895, Backbone Fusion 0.9384, and Hierarchical Ensemble 0.9163. Under the chunked real-time replay protocol (frame-indexed sliding context per Table 5.5: a 250-frame video context with 125-frame stride for WeldFusionNet, internally reduced to a fixed 25×44 audio tensor and 8 sampled video frames; 50-frame chunks for Backbone Fusion and Hierarchical Ensemble), the corresponding binary F1 values are 0.9886, 0.9698, and 0.9559. The binary screening stage is therefore reliable in both offline and streaming evaluation modes.

The second question asked which multimodal fusion strategy offers the best tradeoff between multiclass quality and real-time constraints. Backbone Fusion achieves the strongest offline multiclass performance (macro-F1 0.8319), followed by WeldFusionNet (0.8178) and Hierarchical Ensemble (0.7459). The 0.086 macro-F1 gap between Backbone Fusion and Hierarchical Ensemble is substantial, whereas the 0.014 gap between Backbone Fusion and WeldFusionNet (paired bootstrap $\Delta = +0.015$, not statistically significant; §6.8.3) falls within estimation uncertainty. Under chunk-based real-time evaluation, Backbone Fusion (0.8436 macro-F1, 127.5 ms per chunk), Hierarchical Ensemble (0.7475 macro-F1, 151.7 ms per chunk), and WeldFusionNet (0.8016 macro-F1, 72.2 ms per window) all satisfy the 250 ms streaming constraint.

The third question asked how model selection should proceed when offline and operational criteria disagree. No single model dominates every metric in this study. WeldFusionNet leads on binary defect F1, accuracy, weighted F1, and ROC-AUC; Backbone Fusion leads on multiclass macro-F1 in both offline and chunked-real-time protocols; Hierarchical Ensemble

trails on every multiclass aggregate but retains an auditable handcrafted feature pipeline. Model selection must therefore start from the operational priority. When binary detection or accuracy is decisive, WeldFusionNet is the appropriate choice. When multiclass macro-F1 governs selection, Backbone Fusion is preferable. For continuous sub-second window-level scoring during the weld, within the limits of recording-level labels, WeldFusionNet-RT applies. Where auditability of the feature pipeline outweighs multiclass quality, Hierarchical Ensemble remains defensible.

The main contribution of the thesis is the comparison of three offline fusion approaches under a unified session-disjoint protocol on the 780-run cross-session test partition, complemented by a separate streaming-native WeldFusionNet-RT experiment evaluated under a configuration-grouped window-level protocol. Multimodal weld-defect classification achieves strong binary detection (binary F1 0.91–0.99) and fair-to-strong multiclass discrimination (macro-F1 0.75–0.83) across all 12 classes (one normal class plus eleven defect classes) on the cross-session test set produced by a session-disjoint grouped split of the full 3841-recording dataset. WeldFusionNet-RT, trained natively on 1-second windows, reaches window-level binary F1 of 0.9946 and macro-F1 of 0.9853 (window-level, not directly comparable to recording-level metrics) with a per-window GPU latency of 3.2 ms on its own configuration-grouped window-level test partition. This result shows that native window-level training removes the mismatch between full-recording training and window-level inference that affects offline models applied to streaming contexts, although other forms of distribution shift remain; its window-level metrics are not directly comparable to the recording-level metrics of the offline models. Within the main session-disjoint split and the chunked-real-time replay subset, the three offline models preserve the same multiclass ordering, which indicates that the within-protocol evaluation is stable. The sanity baseline analysis (§6.10) shows that both representation choice and model architecture contribute to the observed differences, with much of the multiclass signal already present in the pretrained ResNet18 video embeddings.

Beyond the specific model comparison, the thesis shows that multiple fusion paradigms can reach credible performance on weld-defect classification under cross-session evaluation, and that the evaluation protocol matters as much as model architecture for obtaining reliable conclusions. The protocol established here combines grouped splitting by welding session, adequate test coverage, and multi-metric assessment.

8.1 Limitations

The conclusions above must be read against the following scope boundaries; each one constrains the strength of the claims made in this thesis.

- **Single dataset, single cell.** All recordings originate from one robotic welding cell with fixed sensor placement. No external industrial validation has been performed. Performance under a different cell, base material, filler wire, or shielding-gas composition is unknown.
- **Replay-based real-time, not live deployment.** “Real-time” in this thesis denotes a sliding-window replay protocol over pre-recorded runs, not a live in-the-loop production

deployment. Latency and throughput figures are measured on stored data, not on a torch-attached sensor pipeline. A live deployment would additionally introduce I/O latency, sensor synchronization drift, and operator-feedback loops not modeled here.

- **Point estimates from fixed evaluation partitions.** The main offline model-selection metrics (binary and 12-class quality for the three offline models) are computed on the 780-run session-disjoint test partition. The recording-level real-time replay uses a 240-run balanced subset of that partition (20 per class); WeldFusionNet-RT is reported on 64,221 windows from a separate configuration-grouped test partition; matched audio-noise and missing-modality robustness uses a 20-run paired subset; and the auxiliary cross-session sensitivity probe (Table A.4) uses a 103-run configuration-grouped held-out split. Bootstrap 95% confidence intervals, paired bootstrap deltas, and McNemar's test (§6.8) quantify uncertainty on the main partition, but each partition itself is fixed. A multi-cell or multi-material partition would still be required for population-level generalization claims.

- **WeldFusionNet-RT uses a different protocol.** WeldFusionNet-RT is trained and evaluated on 1-second windows, not full recordings. Its window-level macro-F1 (0.9853) is *not* directly comparable to the recording-level macro-F1 of the three offline models (0.7459–0.8319). It is presented as a separate streaming-native experiment, not as a multiclass winner.

- **Modality scope.** All four models consume only audio and thermal video. The dataset's auxiliary process-sensor stream (pressure, CO_2 flow, feed rate, weld current, wire consumed, voltage) is not part of any reported model input in this thesis; whether incorporating it would change the model ranking or close the gap on class 05 (undercut) remains open for future work.

- **Robustness scope.** Audio-noise robustness is measured by paired matched-subset replay on the same 20 runs across three conditions (clean, SNR=20 dB, SNR=10 dB). A separate missing-audio condition is also reported on the same 20-run subset. Other stress conditions (thermal-video corruption, viewpoint perturbation, sensor-feed failure) were not evaluated.

- **No external benchmark or competitor reproduction.** Comparisons in this thesis are internal between the three offline models and WeldFusionNet-RT. No prior published method has been reproduced on the same data under the same split protocol.

The system is therefore best described as *real-time-feasible under replay on one cross-session test partition*, not as a deployment-ready industrial monitoring solution.

8.2 Deployment Flow

Figure 8.1 summarizes the deployment path implied by the experimental evidence. The diagram is a release-gate view rather than an implemented production architecture: the

current thesis validates replay inference and model comparison, while live sensor integration, drift monitoring, and operator-feedback handling remain future engineering steps.

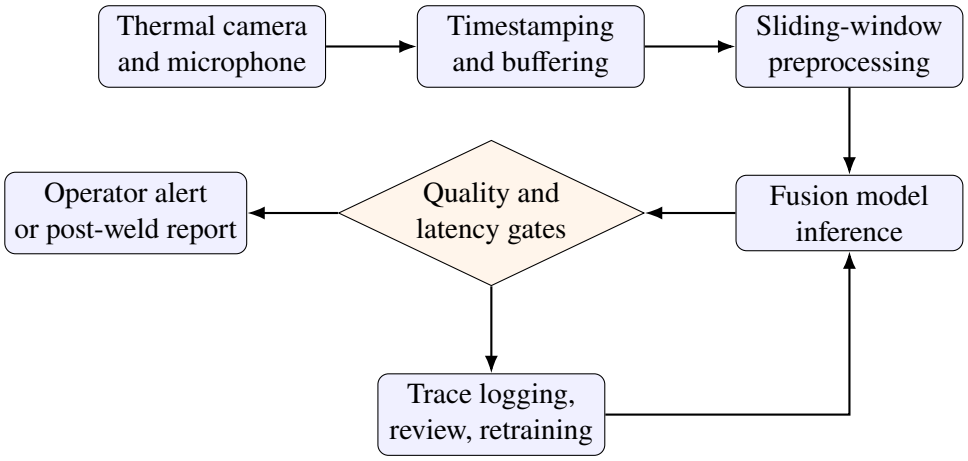

Figure 8.1: Deployment flow for a live welding-monitoring system. The thesis implements and evaluates the preprocessing, fusion inference, and replay-level latency stages; sensor integration, operator workflow, drift monitoring, and retraining gates remain release tasks.

8.3 Broader Impact and Release Plan

The broader value of thermal-acoustic weld monitoring is not only higher aggregate accuracy. A reliable alarm can reduce operator exposure to uncertain weld states by supporting earlier intervention, and can reduce avoidable scrap or rework when a defective run is identified before downstream processing. These impacts remain conditional on human supervision and target-cell validation: false alarms can interrupt production, and missed defects can create misplaced confidence. For that reason, any deployment should retain operator authority, log all decisions for audit, and use the system as decision support rather than as an autonomous acceptance rule.

The release plan is to publish the training, evaluation, artifact-generation, and manuscript-reproduction code together with configuration files, random seeds, and generated result tables. Raw dataset redistribution is not claimed here; the Intel Robotic Welding Multimodal Dataset is referenced as an external source [45, 41]. Reproducibility should therefore be based on documented download instructions, checksums or manifest entries where available, and scripts that regenerate the reported tables and figures from a local authorized dataset copy.

8.4 Future Work

The following directions address specific weaknesses and open questions identified by the present experiments. Each direction is motivated by a concrete gap in the current evidence.

1. Out-of-distribution generalization validation. The most pressing open question is whether the performance obtained on the 780-run cross-session test set (binary F1 0.91–0.99, macro-F1 0.75–0.83) generalizes beyond the current dataset. The session-disjoint split

ensures that no welding session appears in both train and test, which provides cross-session generalization evidence, but all recordings originate from a single robotic welding cell under fixed sensor placement. The natural test is to collect recordings from a different cell, a different base material or filler wire, or a different shielding-gas composition, and to apply the evaluation protocol of this thesis identically: grouped split by welding session, recording-level aggregation, macro-F1 as the primary metric. Substantial performance degradation under these conditions would indicate cell-specific or session-specific learning; stability would provide stronger evidence of generalizable defect discrimination.

2. Multi-cell and multi-material falsification. Bootstrap confidence intervals, McNemar's test, and paired bootstrap deltas (§6.8) confirm that Hierarchical Ensemble is reliably below the deep-feature methods on macro-F1, that the Backbone Fusion versus WeldFusionNet macro-F1 gap is not significant, and that WeldFusionNet's binary-detection advantage is highly significant. The remaining uncertainty in the macro-F1 ordering of the two leading models is consequently not a sampling-noise issue but a genuine population question. A multi-cell or multi-material evaluation would directly test whether either advantage transfers to a different defect-induction regime; per-cell macro-F1 computed with the same evaluation pipeline would provide the most informative additional evidence.

3. Temporal defect localization. The present thesis assigns a single label to each recording and does not attempt to identify when within the recording a defect begins. A natural extension is temporal localization: estimating the approximate onset time of the defect within the run. This problem is harder than recording-level classification because it requires either fine-grained temporal supervision (labels for individual frames or windows) or weakly supervised methods that infer localization from recording-level labels. It also has clear practical value: a monitoring system that reports the approximate onset time of a defect within the weld is more actionable than one that reports only that the weld is defective.

4. Modality fallback and graceful degradation. The missing-modality robustness results characterize how performance degrades when a sensing channel fails, but no adaptive fallback mechanism has been implemented. Future work should develop a switching or interpolation strategy that detects when a modality is unavailable, for example through anomaly detection on the input signal level, and adjusts the fusion weights or switches to a single-modality model accordingly. The modality weight asymmetry in Backbone Fusion ($\alpha_{\text{video}} = 0.75$) provides a starting point: when the thermal camera fails, the system should fall back to the audio-only classifier with appropriately recalibrated thresholds rather than attempt to fuse a zeroed video embedding.

5. Model compression for edge deployment. The neural approaches (WeldFusionNet at 1.08M parameters, Backbone Fusion using ResNet18 and MobileNetV3) are suitable for deployment on commodity hardware but may be too large for embedded edge devices with tight memory and compute budgets. Knowledge distillation, in which a smaller student network is trained to match the outputs of a larger teacher [46], and structured pruning offer

well-established paths for reducing model size while preserving most of the classification quality [47]. A tenfold parameter reduction, to approximately 0.1M parameters, would make these models deployable on microcontrollers or FPGA-based industrial controllers without GPU support. WeldFusionNet's already compact parameter count of 1.08M provides a strong starting point for further compression.

6. Continuous learning and model drift management. In production settings, the distribution of welding runs changes over time as materials, tooling, and process parameters evolve, and a model trained on a historical snapshot may gradually degrade as the deployment distribution drifts. Maintaining system reliability over the full operational lifetime requires continuous learning strategies: online retraining on new labeled examples, periodic re-evaluation on held-out segments, and automated drift detection. The grouped split protocol established in this thesis provides a template for partitioning new data without leakage when the training set is extended.

7. Extended augmentation policies and test-time augmentation. The production WeldFusionNet checkpoint reported in this thesis is trained with a modest augmentation policy: additive Gaussian noise on the audio feature tensor ($\sigma = 0.03$) and label smoothing on the multiclass head. WeldFusionNet-RT additionally applies pitch shift (±2 semitones, $p = 0.30$) and brightness jitter on the video frames. Stronger policies were not evaluated, including SpecAugment-style time and frequency masking on the MFCC matrix, spatial crops and horizontal flips on the thermal-video frames, thermal-emissivity jitter on the recording-level thermal feature vector, within-window time-shifts at sub-second granularity, and mixup-style label-aware blending. Test-time augmentation, in which probabilities are averaged over multiple augmented views of each window, is an inexpensive complementary direction that may tighten confidence calibration on the harder classes (§6.5: class 05 undercut). A controlled augmentation sweep followed by a test-time-augmentation pass at inference is a natural next experiment for raising minority-class F1 without changing the model architecture.

8. Explainability and interpretable defect signatures. The three fusion approaches evaluated in this thesis operate as black-box classifiers: they assign labels with high accuracy but do not explain which features or signal segments drove each decision. In industrial deployment, operator trust and regulatory compliance often require that the system justify a defect classification by identifying, for example, which frequency band or thermal region triggered the diagnosis. Post-hoc explanation methods such as SHAP for the Hierarchical Ensemble's tabular features [48] and Grad-CAM for the convolutional components of WeldFusionNet and Backbone Fusion [49] are candidate approaches for attributing predictions to input features. Explainability analysis also helps identify spurious correlations: if the model's decisions are driven predominantly by features that correlate with recording configuration identity rather than defect physics, this would indicate a form of dataset bias that accuracy metrics alone cannot detect. Interpretability analysis is therefore both a practical deployment requirement and a diagnostic tool for validating the generalizability of

learned representations.

8.5 Broader Implications

The specific findings of this thesis carry lessons that extend beyond weld-defect classification.

Operational priority determines model selection more than accuracy rank. The three offline-trained models differ by up to 0.086 macro-F1 offline, and no single model dominates on every metric. WeldFusionNet (72.2 ms per window) leads on binary detection and accuracy; Backbone Fusion (127.5 ms per chunk) leads on macro-F1; Hierarchical Ensemble (151.7 ms per chunk) trails but provides an auditable feature pipeline. All three satisfy the 250 ms streaming constraint and are appropriate for online classification. The model with the best aggregate accuracy is therefore not automatically the right choice for every operational priority. In any monitoring system integrated into a production control loop, the relative weight of binary detection versus fine-grained classification quality, together with the target latency budget for each use case, must be specified before model selection begins, because these priorities act as discriminating filters that a single-metric ranking cannot capture.

Granular class taxonomy exposes failure modes invisible to binary evaluation. Evaluating all 12 classes (the eleven defect classes alongside the normal weld), rather than collapsing to binary good/defect, was essential for discovering the failure concentration at undercut (class 05), excessive penetration (class 01), and warping (class 10). Binary F1 above 0.91 for all three offline-trained models would suggest deployment-oriented feasibility if it were the only metric considered. The multiclass evaluation reveals that this aggregate figure conceals per-class F1 values as low as 0.000 (Hierarchical Ensemble on class 05) and 0.091 (WeldFusionNet on class 05) for the hardest class, and that these classes have specific physical explanations rooted in their transient, cross-class-similar signal signatures. Prior welding monitoring studies that reported only binary or ternary classification results would have missed this structure entirely. Adopting fine-grained taxonomies, even when the operational system ultimately collapses to fewer categories, is important for correctly characterizing where a system will and will not be reliable.

Evaluation protocol determines whether rankings are stable. The offline macro-F1 ordering Backbone Fusion > WeldFusionNet > Hierarchical Ensemble is consistent between offline and chunked-real-time evaluation, which indicates that the ordering reflects genuine architectural differences rather than evaluation artifacts. The binary detection ordering WeldFusionNet > Backbone Fusion > Hierarchical Ensemble is similarly consistent across protocols. The two orderings disagree, and the divergence is itself informative: binary detection and multiclass diagnosis impose different demands on the feature representations, and no single architecture dominates both. Reporting both rankings alongside the primary metric is a practical step toward more honest performance claims in manufacturing AI. The cross-session split that defines the main test partition further illustrates the cost of distributional assumptions that are almost always present in research evaluations but rarely

reported: the resulting macro-F1 values (0.75–0.83) are lower than what would typically be observed under within-session evaluation.

Two-stage architecture as a diagnostic design pattern. The hierarchical binary-then-multiclass formulation proved useful beyond its primary purpose of mirroring industrial monitoring logic. It decouples two failure modes with different operational consequences and different remediation paths: missing a defect versus misidentifying its type. The binary stage (F1 0.91–0.99 offline; 0.96–0.99 chunked real-time) shows that all three fusion approaches can serve as adequate go/no-go alarms even where their fine-grained classification quality differs. The separation also clarifies where future improvement effort should be directed: the binary detection ceiling is already high, so additional labeled data or model improvement would primarily benefit the multiclass stage, specifically the classes that currently depress the macro-F1 (class 05 in particular). This diagnostic clarity would be lost in a single-stage formulation that predicts one of twelve labels (eleven defect classes plus the good weld) without an intermediate detection decision.

Appendix A

Supplementary Appendix

A.1 Appendix Scope

This appendix collects supplementary technical material that supports the interpretation of the main chapters without interrupting their narrative flow. The appendix contains:

- the complete feature dictionary used in the baseline analysis,
- a robustness provenance note referenced in Chapters 6 and 7,
- a cross-session comparison table,
- supplementary confusion matrices from the offline evaluation,
- per-class multimodal profiles for all twelve classes (the normal weld plus eleven defect categories),
- a reproducibility note (commit hash, seeds, environment, run commands).

A.2 Reproducibility Note

This section records the minimum information needed to reproduce every quantitative result reported in this thesis. A full dependency manifest, Docker recipe, and run scripts are tracked in the project repository and are not duplicated here.

Source revision. All results in this manuscript correspond to repository commit `90b0fc4` on branch `addressing-gaps`. The repository contains the full training, evaluation, and figure-generation pipeline.

Random seeds. Every reported run uses the project-level seed `seed = 42` declared in `src/configs/base.yaml`, propagated to NumPy, PyTorch (CPU and CUDA), and Python's `random` module at the start of each training entrypoint. Per-model overrides (where used) are recorded in the model-specific config under `src/configs/`; no reported result was selected over a seed sweep.

Software environment. Experiments were executed under Python 3.12 on Linux (kernel 6.17). Dependency versions are pinned in `requirements.lock.txt`; the unpinned high-level dependency set is in `requirements.txt`. Core libraries: PyTorch (CUDA build), torchaudio, torchvision, scikit-learn, XGBoost, NumPy, pandas, librosa, and Matplotlib. GPU training and latency measurements were performed on a single NVIDIA CUDA-enabled accelerator; CPU-only execution is supported but materially slower.

Canonical run commands. The four reported models are trained and evaluated through the following entrypoints (each consumes its YAML config under `src/configs/`; outputs land under `results/`):

- Hierarchical Ensemble: `python -m src.models.train_hierarchical_ensemble -config src/configs/hierarchical_ensemble.yaml`
- Backbone Fusion: `python -m src.models.train_backbone_fusion -config src/configs/backbone_fusion.yaml`
- WeldFusionNet: `python -m src.models.train_fusionnet -config src/configs/fusionnet.yaml`
- WeldFusionNet-RT: `python -m src.models.train_weldfusionnet_rt -config src/configs/weldfusionnet_rt.yaml`

Recording-level replay (latency and chunked real-time metrics) is launched via `python -m src.inference.realtime` with the corresponding model checkpoint. The grouped session-disjoint split used by the three offline models is fixed and serialized under `results/splits/`; WeldFusionNet-RT uses a separate `config_id`-grouped split also stored there. Split files are deterministic given the canonical seed and the snapshot dataset, so re-running the split generation reproduces identical partition memberships.

Determinism caveats. A small residual non-determinism remains on the GPU code paths due to cuDNN benchmarking and TF32 matmul kernels; this affects the third or fourth decimal of reported metrics but does not change relative model ranking. The recording-level latency numbers depend on the host hardware and should be reproduced on a comparable accelerator; the offline classification metrics are hardware-independent up to the determinism caveat above.

A.3 Full Feature Dictionary

Table A.1 lists the full set of extracted descriptors used by the audio-thermal baseline models. The purpose of including this table is to document the analytical content of the representation, not to foreground software implementation details.

Table A.1: Extracted feature dictionary used by the fusion models.

Feature name	Modality	Operational meaning
audio_missing	Audio	Missing audio indicator (1 if unavailable)
audio_duration_sec	Audio	Duration of loaded waveform in seconds
audio_abs_mean	Audio	Absolute waveform amplitude statistic
audio_abs_std	Audio	Absolute waveform amplitude statistic
audio_abs_min	Audio	Absolute waveform amplitude statistic
audio_abs_max	Audio	Absolute waveform amplitude statistic
audio_rms_mean	Audio	Root-mean-square energy statistic
audio_rms_std	Audio	Root-mean-square energy statistic
audio_rms_min	Audio	Root-mean-square energy statistic
audio_rms_max	Audio	Root-mean-square energy statistic
audio_rms_p10	Audio	Root-mean-square energy statistic
audio_rms_p90	Audio	Root-mean-square energy statistic
audio_rms_skew	Audio	Root-mean-square energy statistic
audio_zcr_mean	Audio	Zero-crossing rate statistic
audio_zcr_std	Audio	Zero-crossing rate statistic
audio_zcr_min	Audio	Zero-crossing rate statistic
audio_zcr_max	Audio	Zero-crossing rate statistic
audio_zcr_p10	Audio	Zero-crossing rate statistic
audio_zcr_p90	Audio	Zero-crossing rate statistic
audio_zcr_skew	Audio	Zero-crossing rate statistic
audio_centroid_mean	Audio	Spectral centroid statistic
audio_centroid_std	Audio	Spectral centroid statistic
audio_centroid_min	Audio	Spectral centroid statistic
audio_centroid_max	Audio	Spectral centroid statistic
audio_centroid_p10	Audio	Spectral centroid statistic
audio_centroid_p90	Audio	Spectral centroid statistic
audio_centroid_skew	Audio	Spectral centroid statistic
audio_bandwidth_mean	Audio	Spectral bandwidth statistic
audio_bandwidth_std	Audio	Spectral bandwidth statistic
audio_bandwidth_min	Audio	Spectral bandwidth statistic
audio_bandwidth_max	Audio	Spectral bandwidth statistic
audio_bandwidth_p10	Audio	Spectral bandwidth statistic
audio_bandwidth_p90	Audio	Spectral bandwidth statistic
audio_bandwidth_skew	Audio	Spectral bandwidth statistic
audio_rolloff_mean	Audio	Spectral rolloff statistic
audio_rolloff_std	Audio	Spectral rolloff statistic
audio_rolloff_min	Audio	Spectral rolloff statistic
audio_rolloff_max	Audio	Spectral rolloff statistic
audio_rolloff_p10	Audio	Spectral rolloff statistic
audio_rolloff_p90	Audio	Spectral rolloff statistic

Feature name	Modality	Operational meaning
audio_rolloff_skew	Audio	Spectral rolloff statistic
audio_mfcc1_mean	Audio	MFCC coefficient summary statistic
audio_mfcc1_std	Audio	MFCC coefficient summary statistic
audio_mfcc2_mean	Audio	MFCC coefficient summary statistic
audio_mfcc2_std	Audio	MFCC coefficient summary statistic
audio_mfcc3_mean	Audio	MFCC coefficient summary statistic
audio_mfcc3_std	Audio	MFCC coefficient summary statistic
audio_mfcc4_mean	Audio	MFCC coefficient summary statistic
audio_mfcc4_std	Audio	MFCC coefficient summary statistic
audio_mfcc5_mean	Audio	MFCC coefficient summary statistic
audio_mfcc5_std	Audio	MFCC coefficient summary statistic
audio_mfcc6_mean	Audio	MFCC coefficient summary statistic
audio_mfcc6_std	Audio	MFCC coefficient summary statistic
audio_mfcc7_mean	Audio	MFCC coefficient summary statistic
audio_mfcc7_std	Audio	MFCC coefficient summary statistic
audio_mfcc8_mean	Audio	MFCC coefficient summary statistic
audio_mfcc8_std	Audio	MFCC coefficient summary statistic
audio_mfcc9_mean	Audio	MFCC coefficient summary statistic
audio_mfcc9_std	Audio	MFCC coefficient summary statistic
audio_mfcc10_mean	Audio	MFCC coefficient summary statistic
audio_mfcc10_std	Audio	MFCC coefficient summary statistic
audio_mfcc11_mean	Audio	MFCC coefficient summary statistic
audio_mfcc11_std	Audio	MFCC coefficient summary statistic
audio_mfcc12_mean	Audio	MFCC coefficient summary statistic
audio_mfcc12_std	Audio	MFCC coefficient summary statistic
audio_mfcc13_mean	Audio	MFCC coefficient summary statistic
audio_mfcc13_std	Audio	MFCC coefficient summary statistic
audio_delta_mfcc1_mean	Audio	Delta-MFCC summary statistic
audio_delta_mfcc1_std	Audio	Delta-MFCC summary statistic
audio_delta2_mfcc1_mean	Audio	Delta-delta MFCC summary statistic
audio_delta2_mfcc1_std	Audio	Delta-delta MFCC summary statistic
audio_delta_mfcc2_mean	Audio	Delta-MFCC summary statistic
audio_delta_mfcc2_std	Audio	Delta-MFCC summary statistic
audio_delta2_mfcc2_mean	Audio	Delta-delta MFCC summary statistic
audio_delta2_mfcc2_std	Audio	Delta-delta MFCC summary statistic
audio_delta_mfcc3_mean	Audio	Delta-MFCC summary statistic

Feature name	Modality	Operational meaning
audio_delta_mfcc3_std	Audio	Delta-MFCC summary statistic
audio_delta2_mfcc3_ mean	Audio	Delta-delta MFCC summary statistic
audio_delta2_mfcc3_std	Audio	Delta-delta MFCC summary statistic
audio_delta_mfcc4_ mean	Audio	Delta-MFCC summary statistic
audio_delta_mfcc4_std	Audio	Delta-MFCC summary statistic
audio_delta2_mfcc4_ mean	Audio	Delta-delta MFCC summary statistic
audio_delta2_mfcc4_std	Audio	Delta-delta MFCC summary statistic
audio_delta_mfcc5_ mean	Audio	Delta-MFCC summary statistic
audio_delta_mfcc5_std	Audio	Delta-MFCC summary statistic
audio_delta2_mfcc5_ mean	Audio	Delta-delta MFCC summary statistic
audio_delta2_mfcc5_std	Audio	Delta-delta MFCC summary statistic
audio_delta_mfcc6_ mean	Audio	Delta-MFCC summary statistic
audio_delta_mfcc6_std	Audio	Delta-MFCC summary statistic
audio_delta2_mfcc6_ mean	Audio	Delta-delta MFCC summary statistic
audio_delta2_mfcc6_std	Audio	Delta-delta MFCC summary statistic
audio_delta_mfcc7_ mean	Audio	Delta-MFCC summary statistic
audio_delta_mfcc7_std	Audio	Delta-MFCC summary statistic
audio_delta2_mfcc7_ mean	Audio	Delta-delta MFCC summary statistic
audio_delta2_mfcc7_std	Audio	Delta-delta MFCC summary statistic
audio_delta_mfcc8_ mean	Audio	Delta-MFCC summary statistic
audio_delta_mfcc8_std	Audio	Delta-MFCC summary statistic
audio_delta2_mfcc8_ mean	Audio	Delta-delta MFCC summary statistic
audio_delta2_mfcc8_std	Audio	Delta-delta MFCC summary statistic
audio_delta_mfcc9_ mean	Audio	Delta-MFCC summary statistic
audio_delta_mfcc9_std	Audio	Delta-MFCC summary statistic
audio_delta2_mfcc9_ mean	Audio	Delta-delta MFCC summary statistic
audio_delta2_mfcc9_std	Audio	Delta-delta MFCC summary statistic

Feature name	Modality	Operational meaning
audio_delta_mfcc10_mean	Audio	Delta-MFCC summary statistic
audio_delta_mfcc10_std	Audio	Delta-MFCC summary statistic
audio_delta2_mfcc10_mean	Audio	Delta-delta MFCC summary statistic
audio_delta2_mfcc10_std	Audio	Delta-delta MFCC summary statistic
audio_delta_mfcc11_mean	Audio	Delta-MFCC summary statistic
audio_delta_mfcc11_std	Audio	Delta-MFCC summary statistic
audio_delta2_mfcc11_mean	Audio	Delta-delta MFCC summary statistic
audio_delta2_mfcc11_std	Audio	Delta-delta MFCC summary statistic
audio_delta_mfcc12_mean	Audio	Delta-MFCC summary statistic
audio_delta_mfcc12_std	Audio	Delta-MFCC summary statistic
audio_delta2_mfcc12_mean	Audio	Delta-delta MFCC summary statistic
audio_delta2_mfcc12_std	Audio	Delta-delta MFCC summary statistic
audio_delta_mfcc13_mean	Audio	Delta-MFCC summary statistic
audio_delta_mfcc13_std	Audio	Delta-MFCC summary statistic
audio_delta2_mfcc13_mean	Audio	Delta-delta MFCC summary statistic
audio_delta2_mfcc13_std	Audio	Delta-delta MFCC summary statistic
audio_contrast1_mean	Audio	Spectral contrast summary statistic
audio_contrast1_std	Audio	Spectral contrast summary statistic
audio_contrast2_mean	Audio	Spectral contrast summary statistic
audio_contrast2_std	Audio	Spectral contrast summary statistic
audio_contrast3_mean	Audio	Spectral contrast summary statistic
audio_contrast3_std	Audio	Spectral contrast summary statistic
audio_contrast4_mean	Audio	Spectral contrast summary statistic
audio_contrast4_std	Audio	Spectral contrast summary statistic
audio_contrast5_mean	Audio	Spectral contrast summary statistic
audio_contrast5_std	Audio	Spectral contrast summary statistic
audio_contrast6_mean	Audio	Spectral contrast summary statistic
audio_contrast6_std	Audio	Spectral contrast summary statistic
audio_contrast7_mean	Audio	Spectral contrast summary statistic

Feature name	Modality	Operational meaning
audio_contrast7_std	Audio	Spectral contrast summary statistic
audio_flatness_mean	Audio	Spectral flatness statistic
audio_flatness_std	Audio	Spectral flatness statistic
audio_flatness_min	Audio	Spectral flatness statistic
audio_flatness_max	Audio	Spectral flatness statistic
audio_rms_peak_ratio	Audio	Peak-to-mean RMS energy ratio for transient bursts
thermal_missing	Thermal	Missing thermal/video indicator (1 if unavailable)
thermal_n_samples	Thermal	Number of sampled video frames used for features
thermal_intensity_mean	Thermal	Frame-intensity distribution statistic
thermal_intensity_std	Thermal	Frame-intensity distribution statistic
thermal_intensity_min	Thermal	Frame-intensity distribution statistic
thermal_intensity_max	Thermal	Frame-intensity distribution statistic
thermal_texture_mean	Thermal	Frame-level texture (pixel standard deviation) statistic
thermal_texture_std	Thermal	Frame-level texture (pixel standard deviation) statistic
thermal_texture_min	Thermal	Frame-level texture (pixel standard deviation) statistic
thermal_texture_max	Thermal	Frame-level texture (pixel standard deviation) statistic
thermal_hot_ratio_mean	Thermal	Ratio of pixels above per-frame 95th percentile
thermal_hot_ratio_std	Thermal	Ratio of pixels above per-frame 95th percentile
thermal_hot_ratio_min	Thermal	Ratio of pixels above per-frame 95th percentile
thermal_hot_ratio_max	Thermal	Ratio of pixels above per-frame 95th percentile
thermal_motion_mean	Thermal	Inter-frame absolute-difference statistic
thermal_motion_std	Thermal	Inter-frame absolute-difference statistic
thermal_motion_min	Thermal	Inter-frame absolute-difference statistic
thermal_motion_max	Thermal	Inter-frame absolute-difference statistic
thermal_entropy_mean	Thermal	Frame entropy statistic
thermal_entropy_std	Thermal	Frame entropy statistic
thermal_entropy_min	Thermal	Frame entropy statistic
thermal_entropy_max	Thermal	Frame entropy statistic
thermal_gradient_mean	Thermal	Sobel gradient-magnitude statistic
thermal_gradient_std	Thermal	Sobel gradient-magnitude statistic

Feature name	Modality	Operational meaning
thermal_gradient_min	Thermal	Sobel gradient-magnitude statistic
thermal_gradient_max	Thermal	Sobel gradient-magnitude statistic
thermal_hist_spread_mean	Thermal	Histogram dynamic-range spread statistic
thermal_hist_spread_std	Thermal	Histogram dynamic-range spread statistic
thermal_hist_spread_min	Thermal	Histogram dynamic-range spread statistic
thermal_hist_spread_max	Thermal	Histogram dynamic-range spread statistic
thermal_edge_density_mean	Thermal	Canny edge-density statistic
thermal_edge_density_std	Thermal	Canny edge-density statistic
thermal_edge_density_min	Thermal	Canny edge-density statistic
thermal_edge_density_max	Thermal	Canny edge-density statistic

A.4 Supplementary Robustness Material

The matched-pair audio-noise and missing-audio robustness results are reported in full in Chapter 6 (Table 6.5); this section records only the provenance of an earlier, superseded robustness experiment.

Earlier model variant: provenance note. An earlier internal experiment used a *late-fusion two-stage* model variant that consumed a 6-channel auxiliary sensor stream alongside audio and thermal video. That variant is *not* the production WeldFusionNet checkpoint reported in this thesis and its quantitative missing-modality results are excluded here because the model input and architecture differ. The directly-comparable missing-audio robustness result for the production WeldFusionNet is reported in Section 6.3.1 (matched 20-run subset, Δ macro-F1 $= -0.288$ under audio zeroing). The sensor-stream-zeroed condition has no analogue on the production WeldFusionNet because the production model does not consume that stream.

A.5 WeldFusionNet Regularization Sweep: Supplementary Material

This section provides supporting material for the WeldFusionNet regularization sweep summarized in Figure 5.13. The sweep compares the production baseline configuration against three alternative settings (reg-A, reg-B, reg-C) that vary dropout, weight decay, label smoothing, and augmentation intensity while leaving the model architecture, optimizer family, schedule, and partitions unchanged. Table A.2 lists the configuration deltas relative to the

baseline. The selected (validation-best) checkpoint for each variant and the corresponding validation macro-F1 at selection are reported in Table A.3. These results are reported as model-selection diagnostics; the production checkpoint reported in Chapter 6 is the baseline configuration.

Table A.2: WeldFusionNet regularization sweep: configuration deltas relative to the production baseline. All other hyperparameters (architecture, learning-rate schedule, optimizer family, fusion structure, loss composition, partitions) are held fixed.

Variant	Dropout	Weight decay	Label smoothing	Aug. intensity
baseline (production)	0.30	1×10^{-4}	0.00	mild
reg-A	0.40	3×10^{-4}	0.05	mild
reg-B	0.45	5×10^{-4}	0.05	mild + thermal jitter
reg-C	0.50	7×10^{-4}	0.05	medium

Table A.3: WeldFusionNet regularization sweep: validation-best checkpoint per variant. Selection is governed by validation macro-F1; epoch index refers to the validation-best epoch within the variant's run. The test column reports cross-session test macro-F1 at the selected checkpoint and is provided only for transparency; the production checkpoint reported in Chapter 6 is the baseline.

Variant	Selected epoch	Val macro-F1	Test macro-F1	Test defect-F1
baseline (production)	—	0.918	0.818	0.990
reg-A	9	0.960	0.817	0.955
reg-B	17	0.959	0.828	0.954
reg-C	9	0.927	0.810	0.959

Figure A.1 shows the per-epoch training and validation curves for the reg-B variant, the configuration with the highest test macro-F1 in the sweep. Figure A.2 shows the candidate-level selection diagnostic from the parallel Hierarchical Ensemble regularization sweep, included here for reproducibility of the model-selection procedure described in Chapter 5. Train scores are reported as fit diagnostics; validation and held-out test behavior govern selection interpretation.

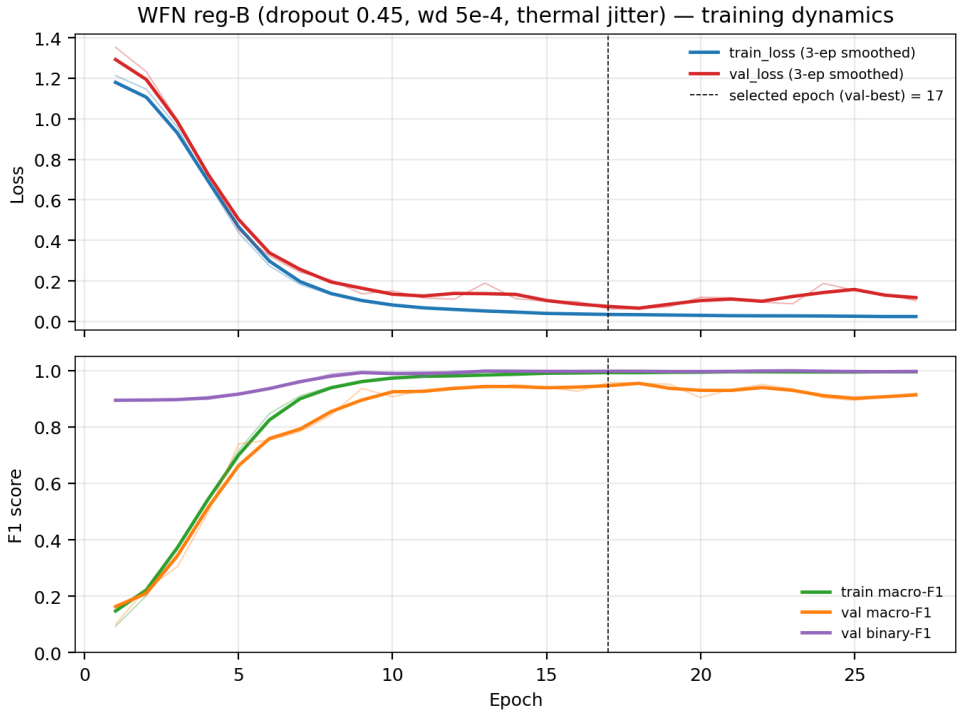

Figure A.1: Per-epoch training and validation curves for the WeldFusionNet reg-B configuration (dropout 0.45, weight decay 5×10^{-4}, thermal jitter): training and validation loss (top), and training macro-F1, validation macro-F1, and validation defect F1 (bottom). The dashed vertical line marks the validation-best selected epoch.

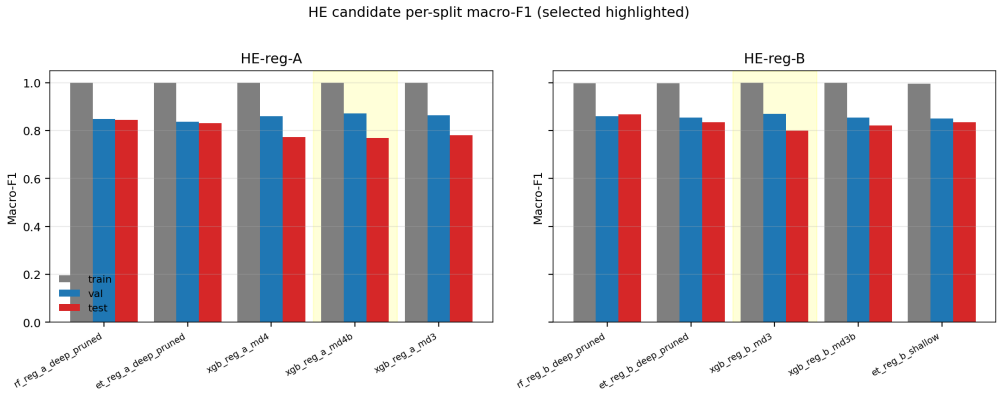

Figure A.2: Candidate-level train, validation, and test macro-F1 for the Hierarchical Ensemble regularization sweep (reg-A and reg-B). The highlighted band marks the selected candidate used for the corresponding final evaluation. Train scores approach 1.0 by construction (the ensemble is fitted on the training partition) and are shown only as a fit diagnostic; validation and held-out test behavior govern selection interpretation.

A.6 Cross-Session Supplementary Tables

Table A.4: Auxiliary cross-session evaluation under an alternative configuration-grouped split (103-run held-out partition drawn from the dataset's designated held-out subset; 3,258-run training partition retrained from scratch). **Note:** this split disagrees with the main session-disjoint split on macro-F1 leadership: Hierarchical Ensemble leads here, in contrast to Backbone Fusion leading on the main split. The two splits agree on binary screening being strong across all three models but disagree on multiclass ranking, indicating that the macro-F1 ranking is sensitive to the choice of grouping criterion. The main session-disjoint result (Chapter 6) remains the headline comparison; this table is reported as a sensitivity probe, not as a competing claim.

Model	Binary F1	Binary Acc	Macro-F1	Multiclass Acc
Hierarchical Ensemble	0.953823	0.923655	0.869676	0.834793
Backbone Fusion	0.933532	0.888611	0.769669	0.729662
WeldFusionNet	0.948469	0.913642	0.844682	0.809762

A.7 Supplementary Confusion Matrices

Table A.5 lists the top misclassification pairs per model on the offline final 12-class test set; it is the tabular companion to the confusion matrices below and is referenced from the error analysis in Chapter 6.

Table A.5: Offline final 12-class top misclassification pairs (ranked by count and class share).

Model	True Label	Predicted Label	Misclassified Count	Share of True Class
Hierarchical Ensemble	01	02	37	0.4625
Hierarchical Ensemble	10	00	37	0.4625
Hierarchical Ensemble	05	06	20	1.0000
Hierarchical Ensemble	01	00	19	0.2375
Hierarchical Ensemble	00	10	14	0.0881
Hierarchical Ensemble	00	02	12	0.0755
Backbone Fusion	10	00	21	0.2625
Backbone Fusion	00	10	16	0.1006
Backbone Fusion	10	02	15	0.1875
Backbone Fusion	05	11	13	0.6500
Backbone Fusion	00	07	12	0.0755
Backbone Fusion	01	02	12	0.1500
WeldFusionNet	01	02	35	0.4375
WeldFusionNet	05	11	15	0.7500
WeldFusionNet	10	07	12	0.1500
WeldFusionNet	03	10	4	0.0667
WeldFusionNet	05	00	4	0.2000
WeldFusionNet	00	10	2	0.0126

The following confusion matrices complete the visual error analysis for all three models for the offline evaluation. Each model is shown in absolute-count and row-normalized forms.

A.7.1 Offline Confusion Matrices

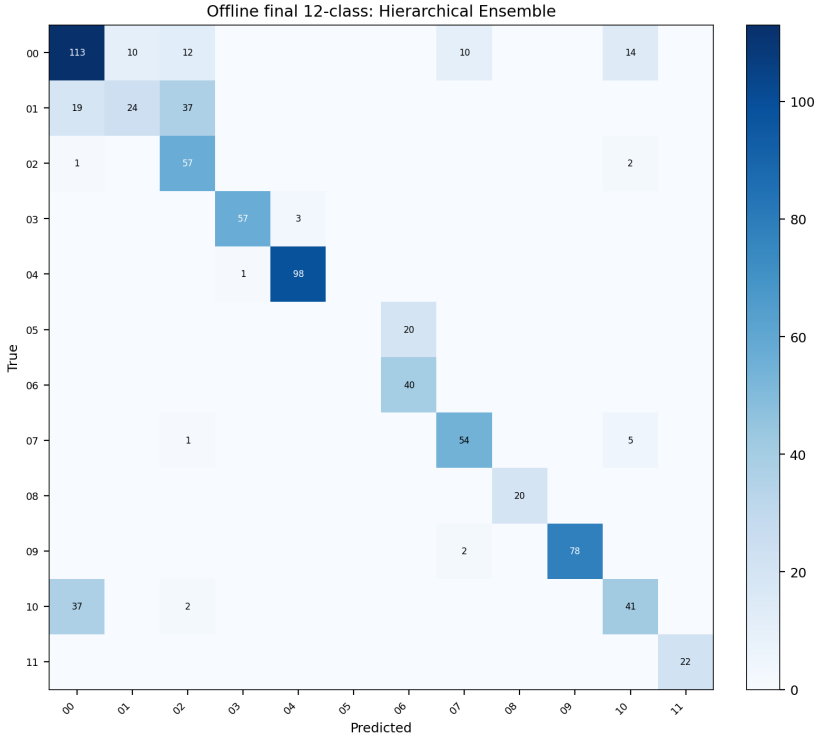

Figure A.3: Offline final 12-class confusion matrix: Hierarchical Ensemble (absolute counts).

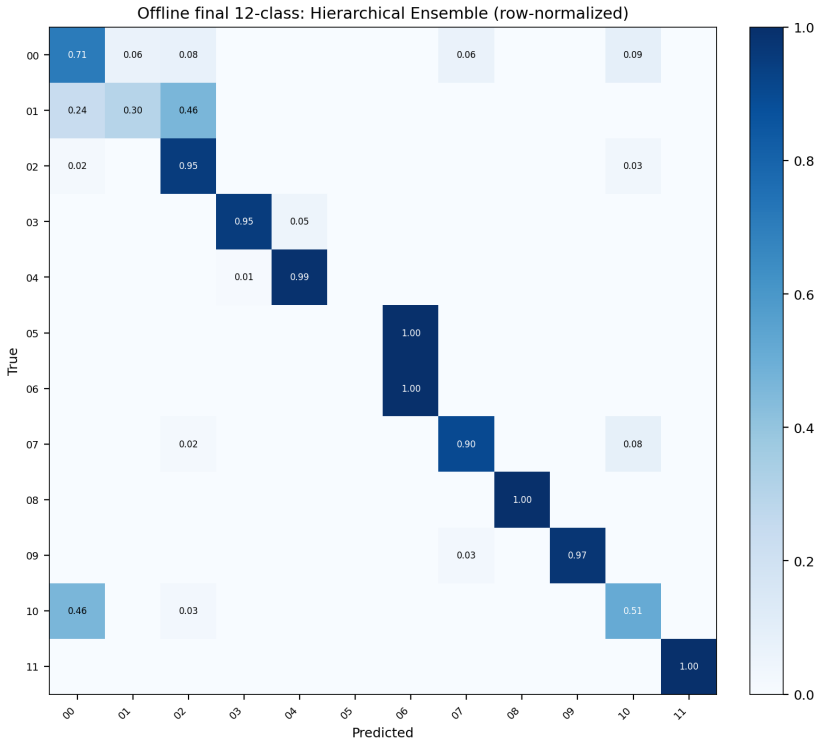

Figure A.4: Offline final 12-class confusion matrix: Hierarchical Ensemble (row-normalized).

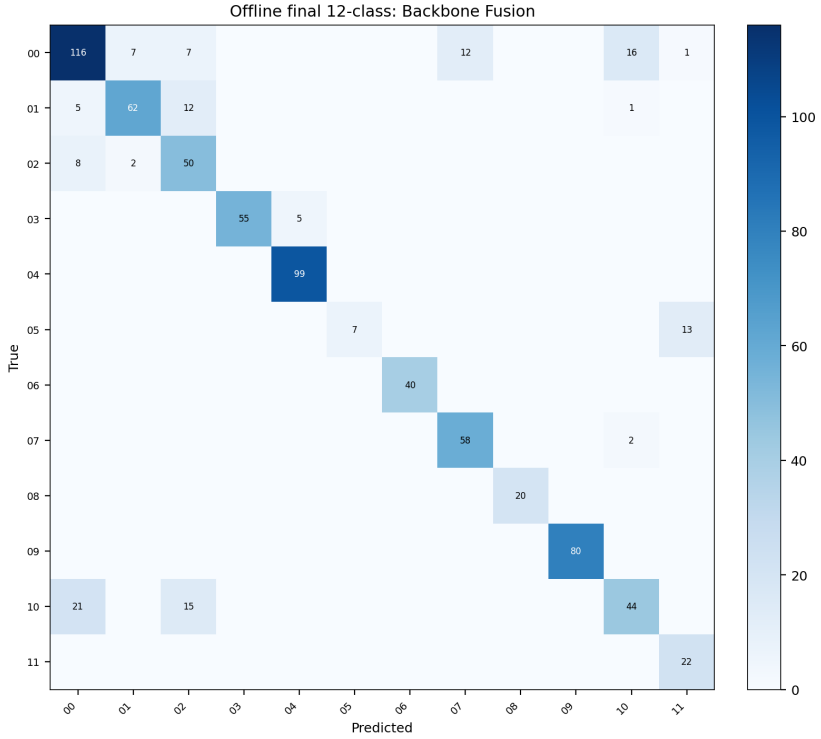

Figure A.5: Offline final 12-class confusion matrix: Backbone Fusion (absolute counts).

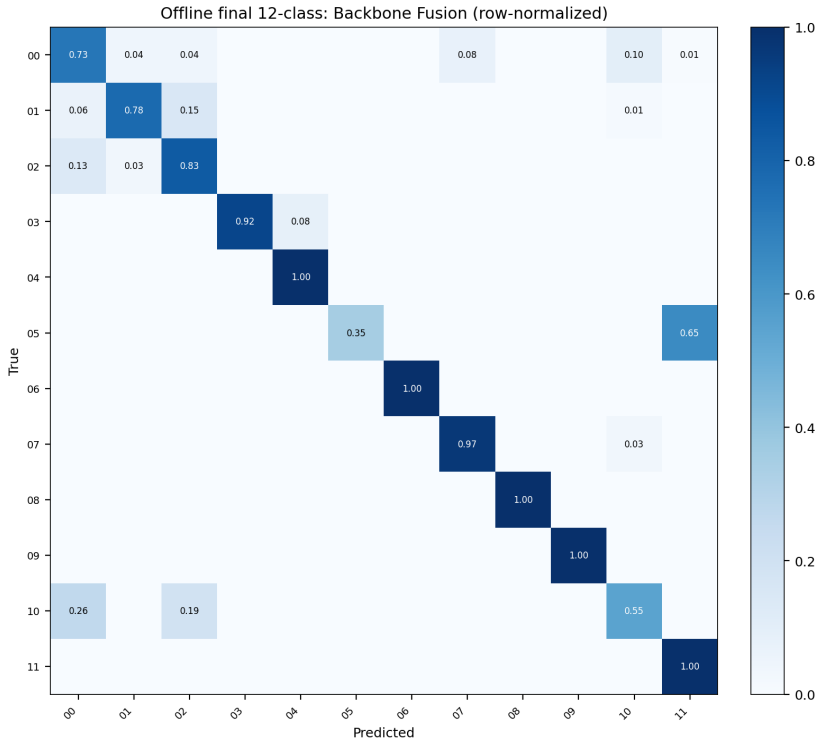

Figure A.6: Offline final 12-class confusion matrix: Backbone Fusion (row-normalized).

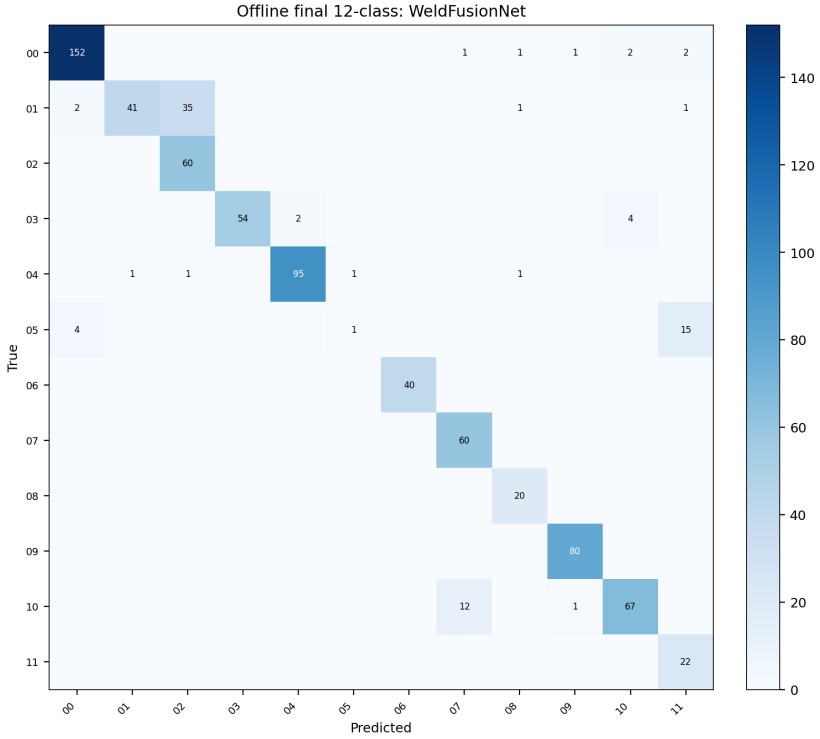

Figure A.7: Offline final 12-class confusion matrix: WeldFusionNet (absolute counts).

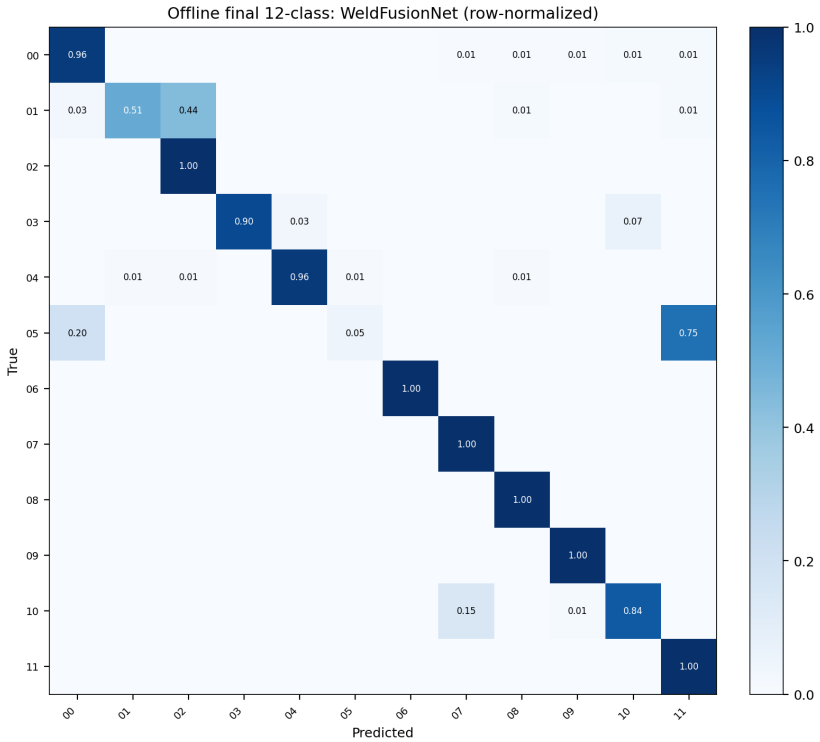

Figure A.8: Offline final 12-class confusion matrix: WeldFusionNet (row-normalized).

A.8 Per-Class Multimodal Defect Profiles

The following profiles provide a four-panel characterization for each of the twelve classes (the normal weld plus eleven defect classes). Each figure shows (top row, left to right) the thermal frame at the recording midpoint using the Inferno colormap, the raw audio waveform with RMS envelope overlay, and the log-mel spectrogram (128 Mel bands, 8 kHz ceiling). The three lower panels show class-averaged sensor traces (mean ± 1 SD, up to 80 training recordings per class) grouped by Gas/Pressure, Electrical, and Wire Feed channel pairs.

Class 00 — Good Weld. Stable arc energy transfer with uniform pool geometry and consistent wire feed. Thermal frames show a symmetric, moderate-intensity pool; audio is rhythmic and steady; sensor channels remain within nominal ranges throughout.

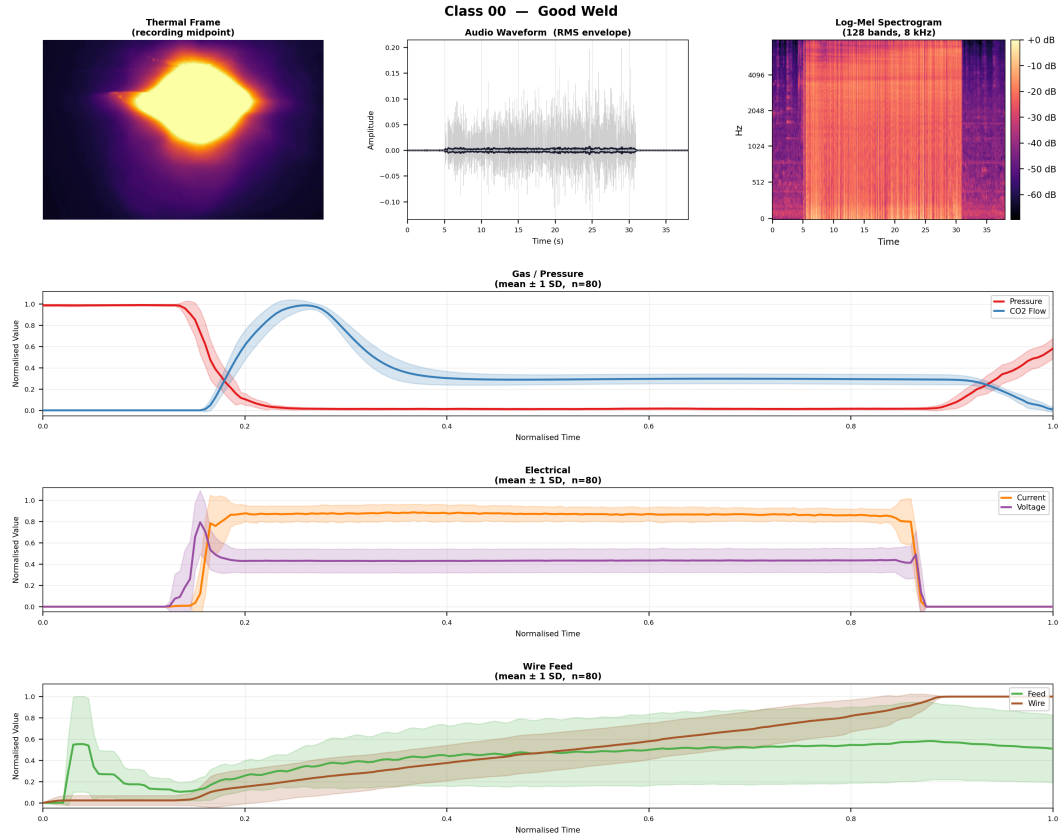

Figure A.9: Multimodal profile for class 00 (Good Weld).

Class 01 — Excessive Penetration. Excessive heat input causes the weld pool to penetrate too deeply through the base metal. Thermal frames show an elongated, high-intensity bright region; audio energy is elevated; current and voltage traces read above nominal setpoints.

Class 02 — Burn Through. Severe over-penetration causes a hole to form in the base material. Thermal frames show abrupt blowout zones of extreme intensity; audio contains sharp transient spikes; pressure and current signals drop suddenly at failure onset.

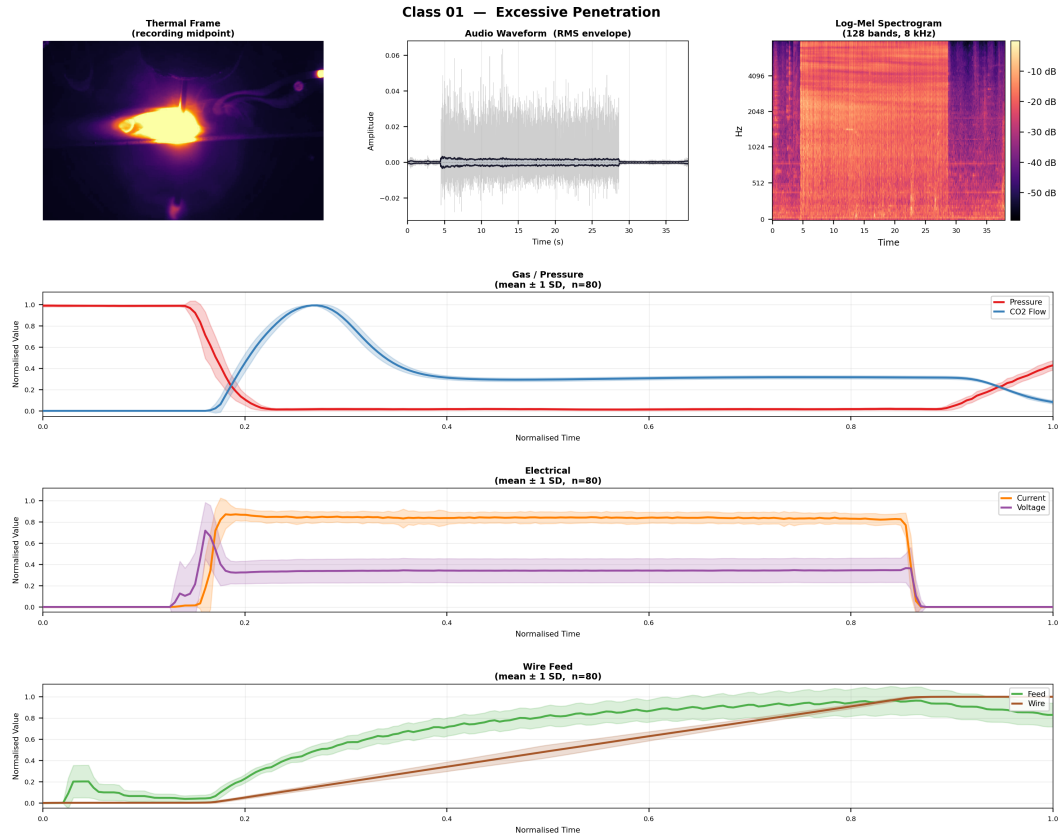

Figure A.10: Multimodal profile for class 01 (Excessive Penetration).

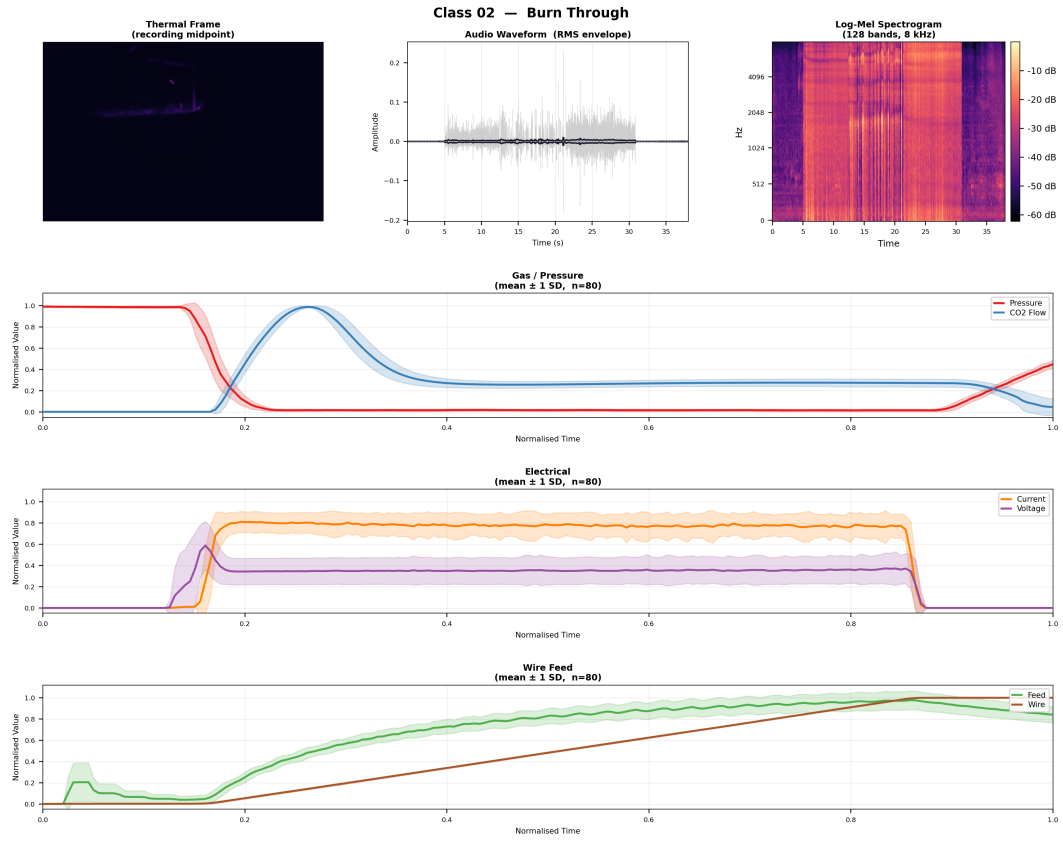

Figure A.11: Multimodal profile for class 02 (Burn Through).

Class 03 — Porosity. Gas entrapment within the solidifying weld creates internal voids. Thermal frames show irregular pool shape or surface pitting; audio contains periodic dull pops; CO_2 flow and pressure signals exhibit instability episodes.

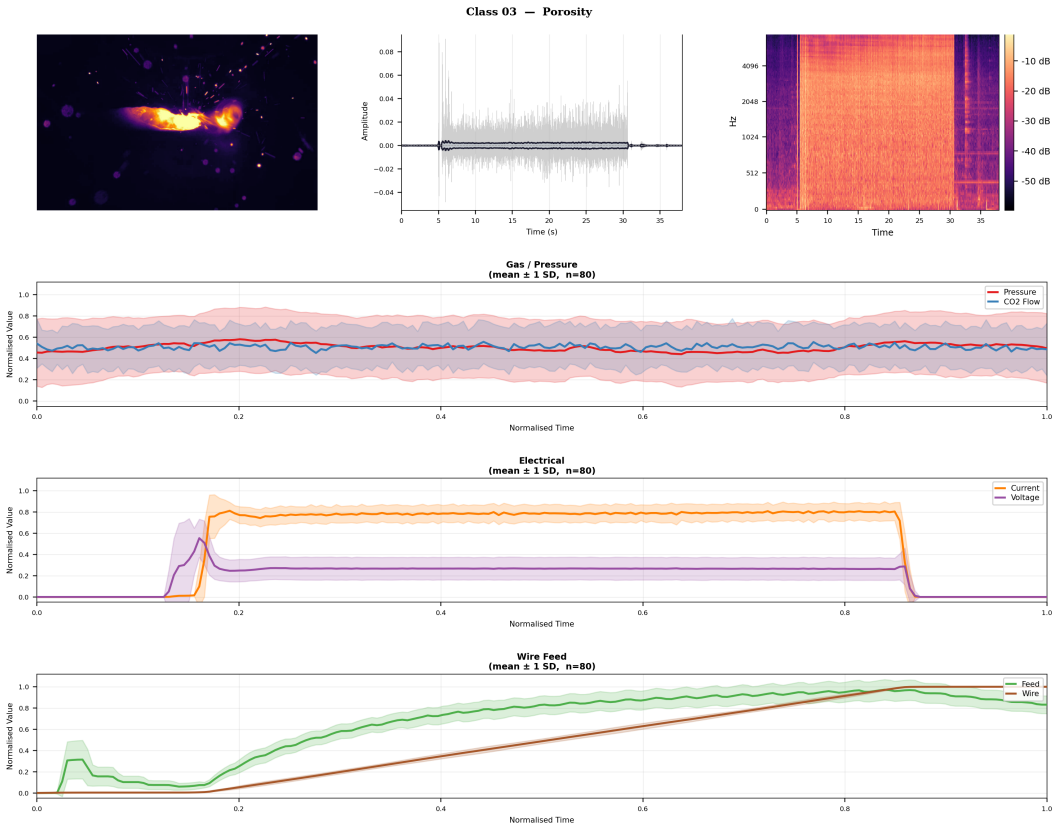

Figure A.12: Multimodal profile for class 03 (Porosity).

Class 04 — Porosity with Excessive Penetration. Gas entrapment combined with excessive heat input produces both internal voids and a deeply penetrated pool. Thermal frames show an elongated bright region with surface pitting; audio mixes the dull pops of porosity with elevated baseline energy; CO_2 flow is unstable while current and voltage read above nominal.

Class 05 — Undercut. Base metal at the weld toe is eroded, forming a groove along the bead edge. Thermal frames show a subtly asymmetric pool with edge erosion; audio and sensor signatures are near-normal, making this the hardest class to discriminate (see also Section 6.5).

Class 06 — Overlap. Excess filler metal rolls over the base plate without bonding. Thermal frames show a wide, low-intensity pool extending beyond the joint line; audio energy is reduced; wire-feed traces are elevated relative to current.

Class 07 — Lack of Fusion. Insufficient heat prevents filler metal from fusing with the base material. Thermal frames show a cold pool with low intensity at the fusion line; audio

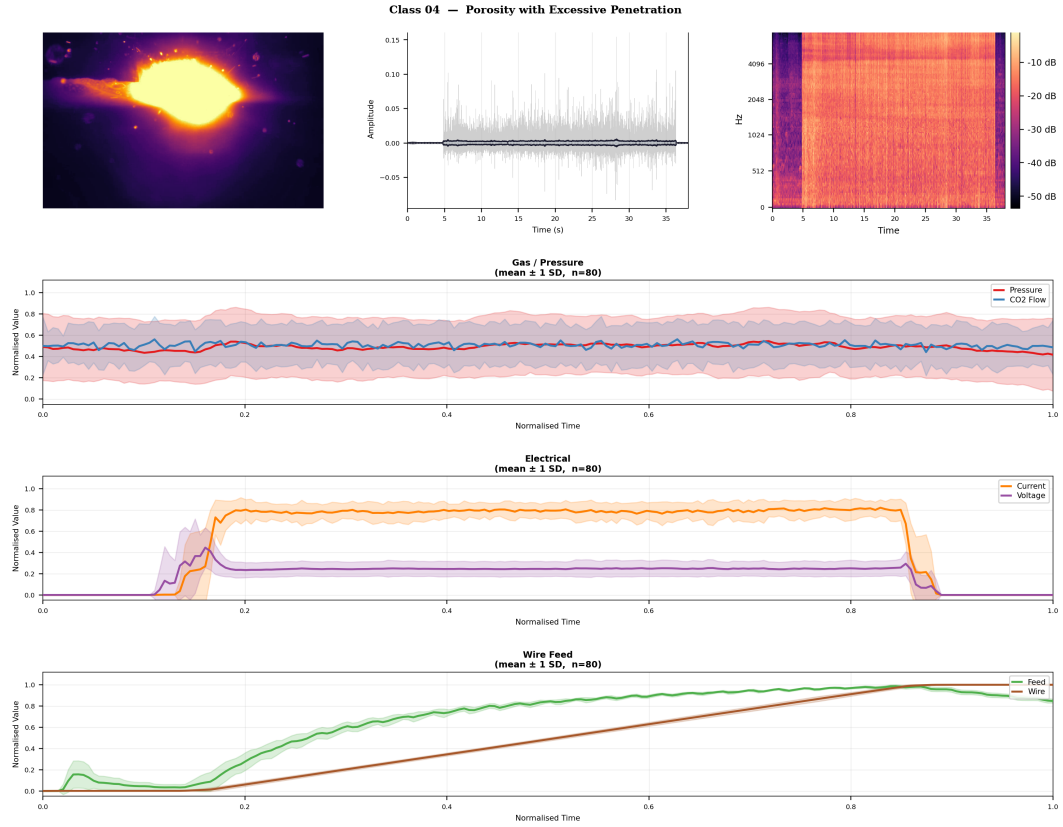

Figure A.13: Multimodal profile for class 04 (Porosity with Excessive Penetration).

energy is reduced; current traces read consistently below nominal.

Class 08 — Excessive Convexity. The weld bead is excessively tall and rounded due to surplus filler deposit. Thermal frames show a narrow, highly convex bright peak; audio and sensor signals reflect elevated wire feed and current above setpoint.

Class 09 — Spatter. Molten metal droplets are expelled from the pool and deposit on surrounding surfaces. Thermal frames show scattered bright particles around the pool; audio has crackling high-frequency bursts; wire-consumption fluctuates irregularly.

Class 10 — Warping. Uneven thermal expansion and cooling causes the welded part to distort out of plane. Thermal frames show a pool that drifts spatially relative to the nominal joint line as the workpiece deforms; audio is largely indistinguishable from a normal weld; sensor traces show small but persistent deviations in current and feed consistent with the changing geometry.

Class 11 — Crater Cracks. Rapid arc termination without proper crater fill creates a void that cracks upon cooling. Audio is distinctive only at the final arc-off transient; thermal frames during active welding appear near-normal, with cracks forming post-arc (see also Section 6.5.3).

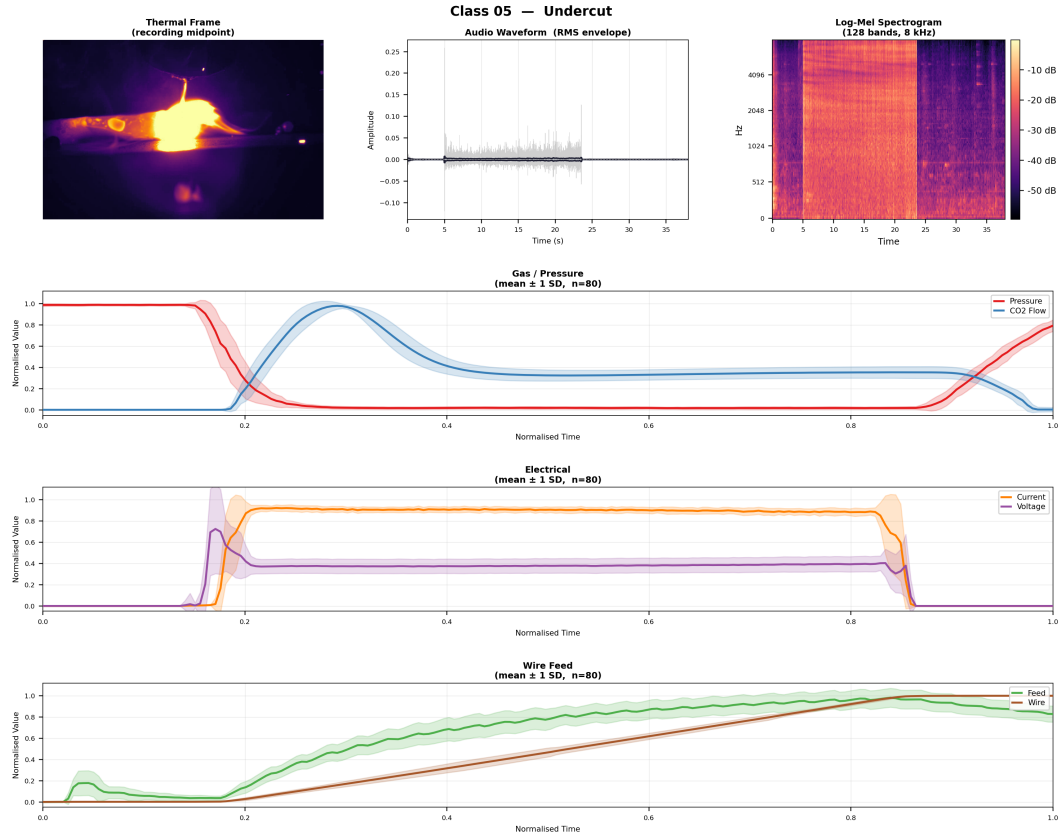

Figure A.14: Multimodal profile for class 05 (Undercut). Signal signatures are deliberately close to those of class 00, consistent with the classification difficulty reported in Chapter 6.

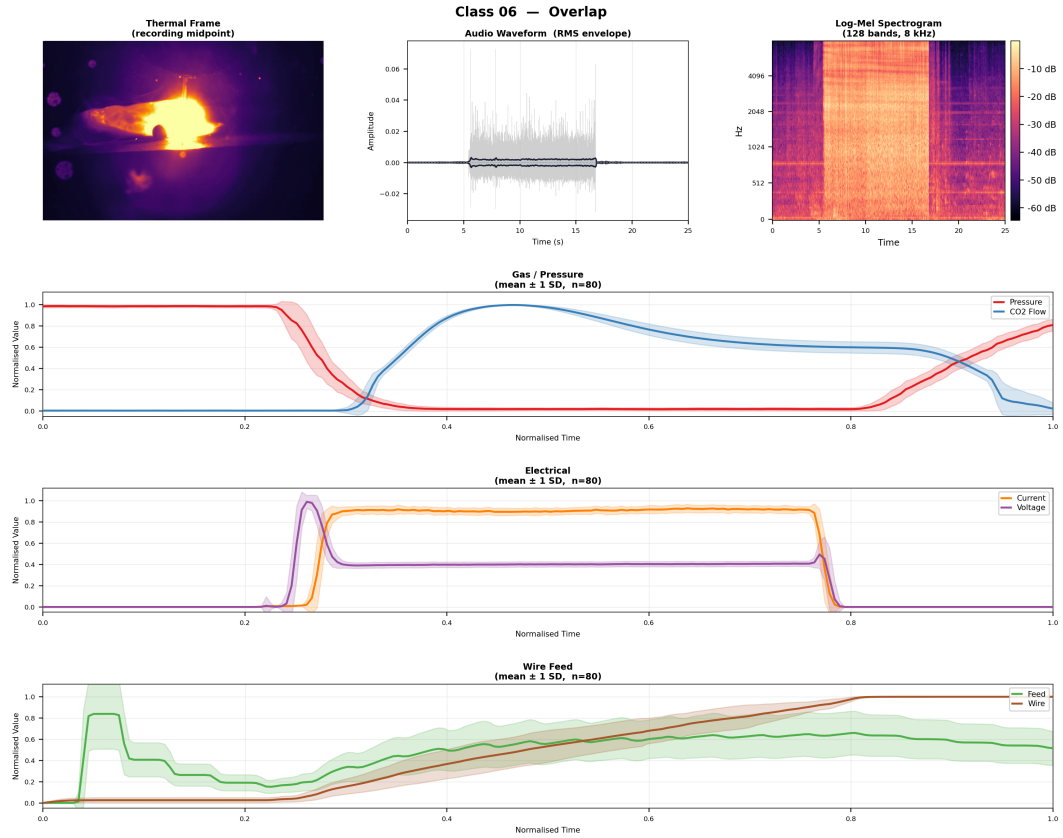

Figure A.15: Multimodal profile for class 06 (Overlap).

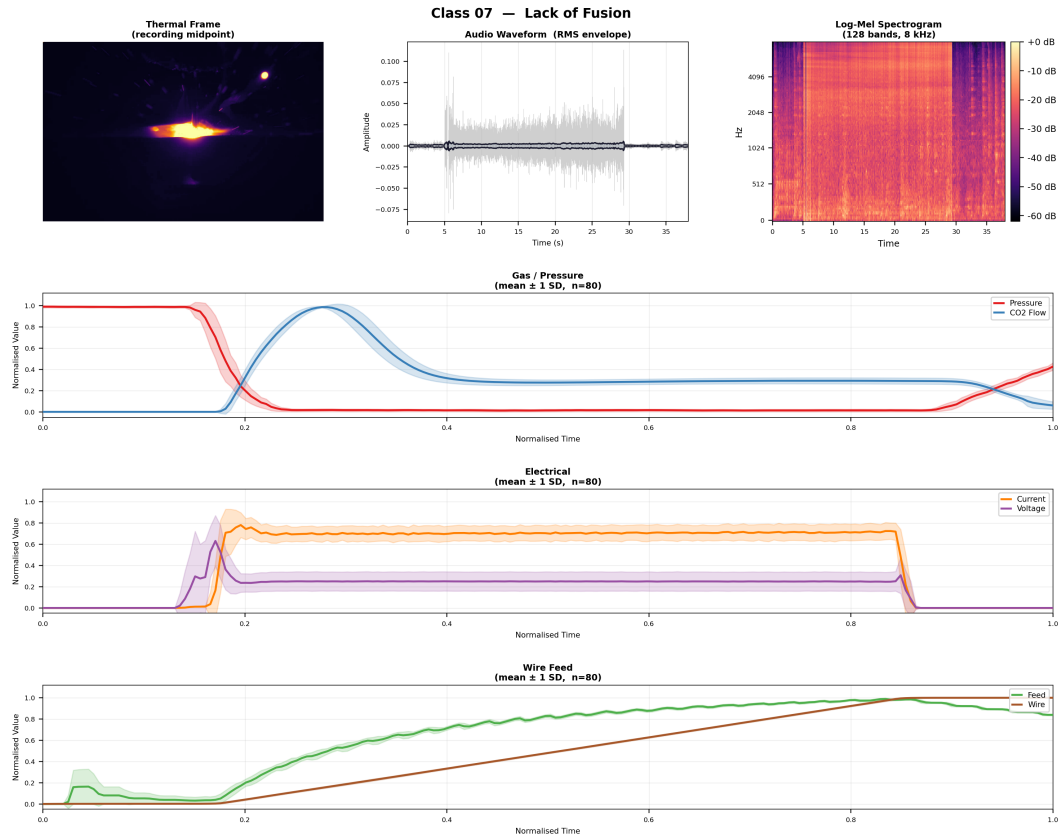

Figure A.16: Multimodal profile for class 07 (Lack of Fusion).

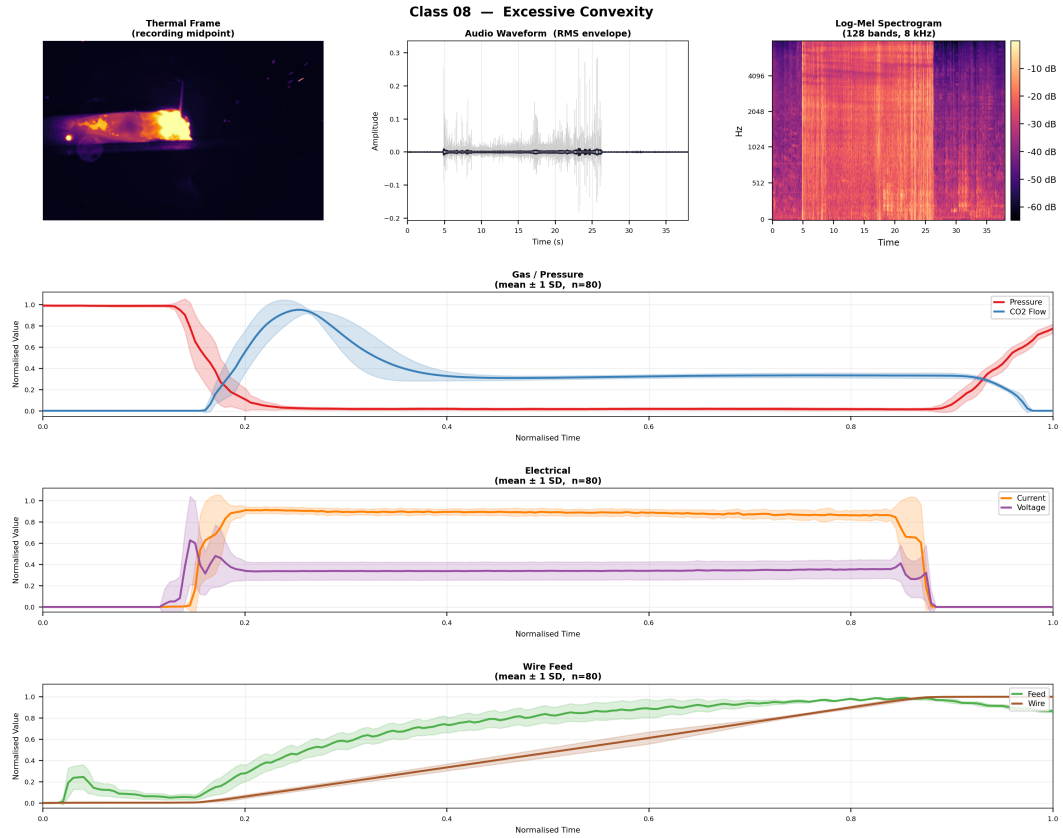

Figure A.17: Multimodal profile for class 08 (Excessive Convexity).

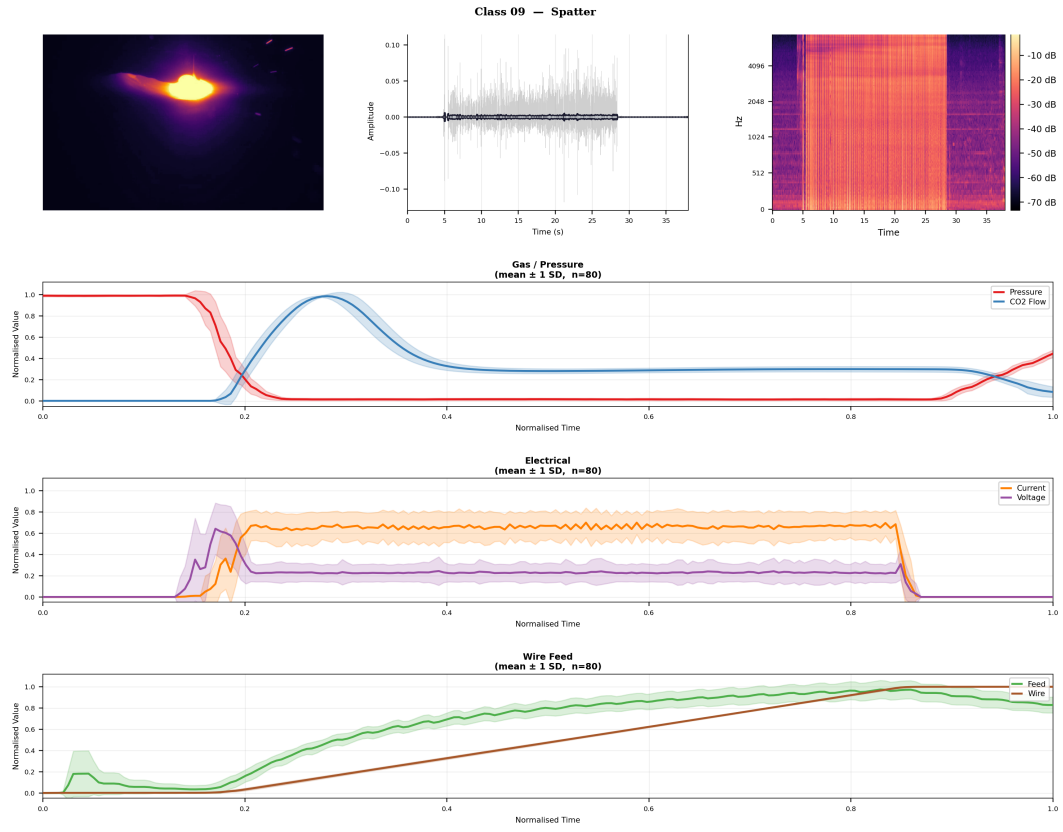

Figure A.18: Multimodal profile for class 09 (Spatter).

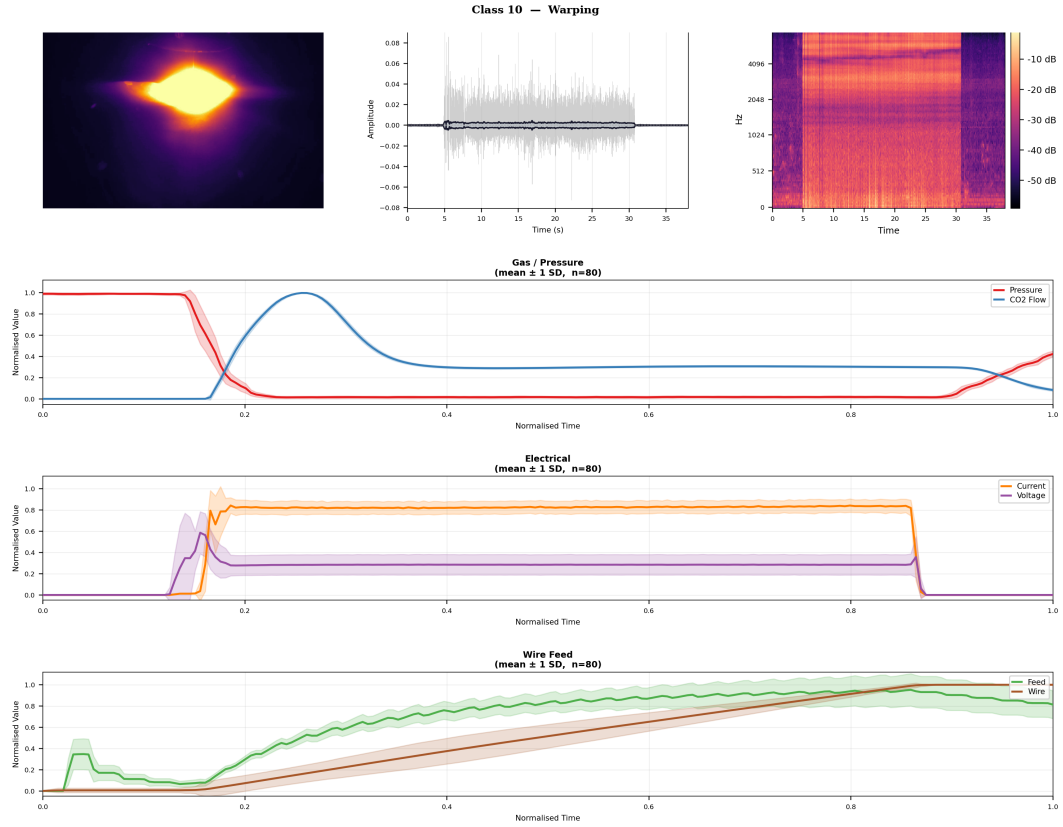

Figure A.19: Multimodal profile for class 10 (Warping).

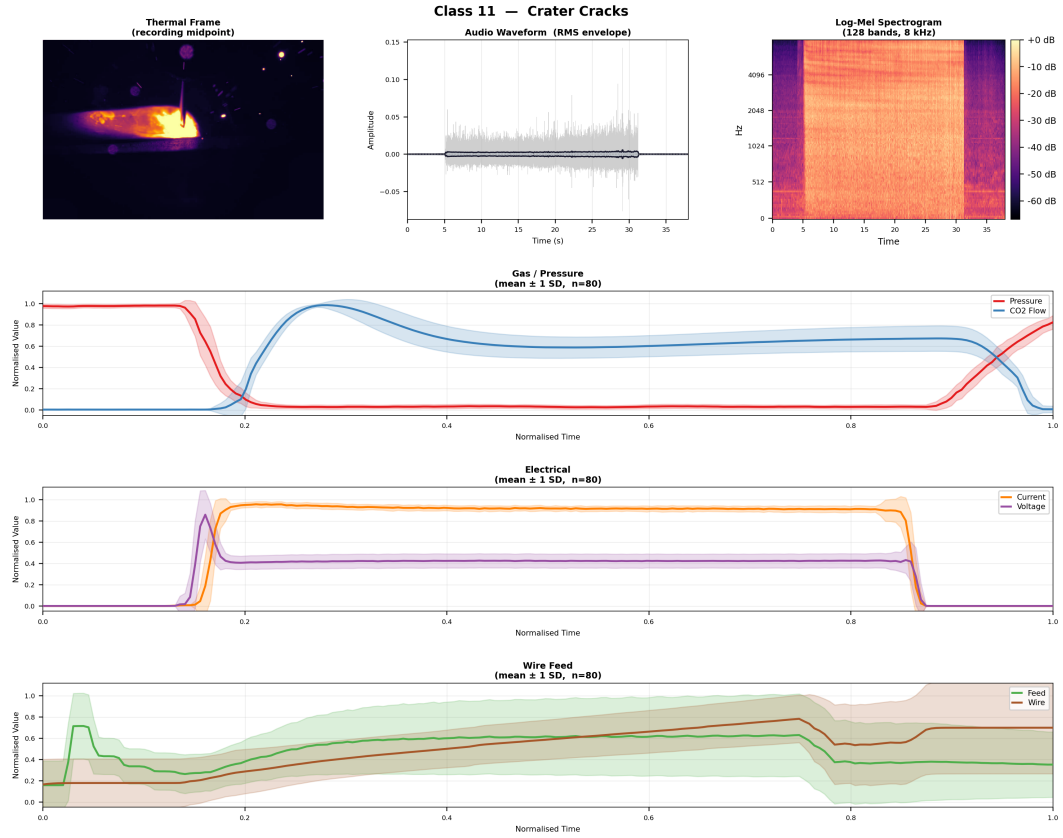

Figure A.20: Multimodal profile for class 11 (Crater Cracks). The near-normal thermal appearance during active welding explains the elevated confusion with class 00 at the binary detection stage.

Bibliography

[1] Y.-M. Zhang. “An analysis of welding process monitoring and control”. In: *Real-Time Weld Process Monitoring*. 2008, pp. 1–11. DOI: 10.1533/9781845694401.1. URL: https://doi.org/10.1533/9781845694401.1 (cit. on pp. 1, 5, 16, 19, 21, 23, 26, 39, 84).

[2] M. Curiel, Y. Falchuk, and F. Lorenzi. “Advances in Robotic Welding for Metallic Materials”. In: *Metals* 13.4 (2023), p. 711. DOI: 10.3390/met13040711. URL: https://doi.org/10.3390/met13040711 (cit. on pp. 1, 16).

[3] International Organization for Standardization. *ISO 5817:2014 Welding – Fusion-welded joints in steel, nickel, titanium and their alloys (beam welding excluded) – Quality levels for imperfections*. 2014. URL: https://www.iso.org/standard/54952.html (visited on 03/05/2026) (cit. on pp. 1, 5).

[4] Reza Hamzeh, Stefan Thomas, Alexander Polzer, Weigang Xu, and Christian Heinzel. “A Sensor-Based Monitoring System for Real-Time Quality Control: Semi-Automatic Arc Welding Case Study”. In: *Procedia Manufacturing* 51 (2020), pp. 524–531. DOI: 10.1016/j.promfg.2020.10.074. URL: https://doi.org/10.1016/j.promfg.2020.10.074 (cit. on pp. 1, 5, 6, 8, 18–21, 89).

[5] Solomon Habtamu Tessema and Dariusz Bismor. “Quality Monitoring of Hybrid Welding Processes: A Comprehensive Review”. In: *Archives of Control Sciences* 34.4 (2024), pp. 833–861. DOI: 10.24425/acs.2024.153104. URL: https://doi.org/10.24425/acs.2024.153104 (cit. on pp. 1, 5, 8, 16, 18, 20, 89).

[6] Jeffrey Dean and Luiz Andre Barroso. “The Tail at Scale”. In: *Communications of the ACM* 56.2 (2013), pp. 74–80. DOI: 10.1145/2408776.2408794. URL: https://doi.org/10.1145/2408776.2408794 (cit. on pp. 4, 8, 21, 23, 36, 84).

[7] International Organization for Standardization. *ISO 6520-1:2007 Welding and allied processes – Classification of geometric imperfections in metallic materials – Part 1: Fusion welding*. 2007. URL: https://www.iso.org/standard/40229.html (visited on 03/05/2026) (cit. on p. 5).

[8] American Welding Society. *AWS D1.1/D1.1M:2020 Structural Welding Code—Steel*. 24th edition. 2020. URL: https://pubs.aws.org/Download_PDFS/D1_1_D1_1M_2020_PV.pdf (visited on 03/14/2026) (cit. on p. 5).

[9] Sadek C. A. Alfaro and Fernand Díaz Franco. "Exploring Infrared Sensoring for Real-Time Welding Defects Monitoring in GTAW". In: *Sensors* 10.6 (2010), pp. 5962–5974. DOI: 10.3390/s100605962. URL: https://doi.org/10.3390/s100605962 (cit. on pp. 6, 16, 19, 23, 26, 39).

[10] Yanxin Cui, Yonghua Shi, Tao Zhu, and Shuwan Cui. "Welding penetration recognition based on arc sound and electrical signals in K-TIG welding". In: *Measurement* 163 (2020), p. 107966. DOI: 10.1016/j.measurement.2020.107966. URL: https://doi.org/10.1016/j.measurement.2020.107966 (cit. on pp. 6, 17, 19, 21, 23, 25).

[11] Ziquan Jiao, Tongshuai Yang, Xingyu Gao, Shanben Chen, and Wenjing Liu. "Welding Penetration Monitoring for Ship Robotic GMAW Using Arc Sound Sensing Based on Improved Wavelet Denoising". In: *Machines* 11.9 (2023), p. 911. DOI: 10.3390/machines11090911. URL: https://doi.org/10.3390/machines11090911 (cit. on pp. 6, 17, 19, 21, 23, 25).

[12] James Griffin, Steven Jones, Bama Perumal, and Carl Perrin. "Investigating the Detection Capability of Acoustic Emission Monitoring to Identify Imperfections Produced by the Metal Active Gas (MAG) Welding Process". In: *Acoustics* 5.3 (2023), pp. 714–745. DOI: 10.3390/acoustics5030043. URL: https://doi.org/10.3390/acoustics5030043 (cit. on pp. 6, 17, 19, 25).

[13] S. Davis and P. Mermelstein. "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28.4 (1980), pp. 357–366. DOI: 10.1109/TASSP.1980.1163420. URL: https://doi.org/10.1109/TASSP.1980.1163420 (cit. on pp. 6, 11, 12, 26, 39).

[14] Brian McFee, Colin Raffel, Dawen Liang, Daniel P. W. Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. "librosa: Audio and Music Signal Analysis in Python". In: *Proceedings of the 14th Python in Science Conference*. 2015, pp. 18–24. DOI: 10.25080/Majora-7b98e3ed-003. URL: https://doi.org/10.25080/Majora-7b98e3ed-003 (cit. on pp. 6, 26, 39, 59).

[15] Tadas Baltrusaitis, Chaitanya Ahuja, and Louis-Philippe Morency. "Multimodal Machine Learning: A Survey and Taxonomy". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.2 (2019), pp. 423–443. DOI: 10.1109/TPAMI.2018.2798607. URL: https://doi.org/10.1109/TPAMI.2018.2798607 (cit. on pp. 6, 7, 18).

[16] Di Wu, Yiming Huang, Huabin Chen, and Yinshui He. "VPPAW penetration monitoring based on fusion of visual and acoustic signals using t-SNE and DBN model". In: *Materials & Design* 123 (2017), pp. 1–14. DOI: 10.1016/j.matdes.2017.03.033. URL: https://doi.org/10.1016/j.matdes.2017.03.033 (cit. on pp. 6, 18, 19, 21).

[17] Haibo He and Edwardo A. Garcia. "Learning from Imbalanced Data". In: *IEEE Transactions on Knowledge and Data Engineering* 21.9 (2009), pp. 1263–1284. DOI: 10.1109/TKDE.2008.239. URL: https://doi.org/10.1109/TKDE.2008.239 (cit. on p. 7).

[18] Marina Sokolova and Guy Lapalme. "A systematic analysis of performance measures for classification tasks". In: *Information Processing & Management* 45.4 (2009), pp. 427–437. DOI: 10.1016/j.ipm.2009.03.002. URL: https://doi.org/10.1016/j.ipm.2009.03.002 (cit. on p. 8).

[19] Jerome H. Friedman. "Greedy Function Approximation: A Gradient Boosting Machine". In: *The Annals of Statistics* 29.5 (2001), pp. 1189–1232. DOI: 10.1214/aos/1013203451 (cit. on p. 8).

[20] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016, pp. 785–794. DOI: 10.1145/2939672.2939785 (cit. on pp. 8, 30).

[21] Leo Breiman. "Random Forests". In: *Machine Learning* 45.1 (2001), pp. 5–32. DOI: 10.1023/A:1010933404324. URL: https://doi.org/10.1023/A:1010933404324 (cit. on p. 9).

[22] Pierre Geurts, Damien Ernst, and Louis Wehenkel. "Extremely randomized trees". In: *Machine Learning* 63.1 (2006), pp. 3–42. DOI: 10.1007/s10994-006-6226-1 (cit. on pp. 9, 30).

[23] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-Based Learning Applied to Document Recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791 (cit. on pp. 9, 17).

[24] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, 2016. ISBN: 9780262035613. URL: https://mitpress.mit.edu/9780262035613/deep-learning/ (cit. on pp. 10, 17, 31, 39).

[25] 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 (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90 (cit. on pp. 10, 31).

[26] Andrew Howard et al. "Searching for MobileNetV3". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 1314–1324. DOI: 10.1109/ICCV.2019.00140 (cit. on pp. 10, 33).

[27] Sinno Jialin Pan and Qiang Yang. "A Survey on Transfer Learning". In: *IEEE Transactions on Knowledge and Data Engineering* 22.10 (2010), pp. 1345–1359. DOI: 10.1109/TKDE.2009.191 (cit. on pp. 10, 39).

[28] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. "How transferable are features in deep neural networks?" In: *Advances in Neural Information Processing Systems*. Vol. 27. 2014, pp. 3320–3328 (cit. on pp. 10, 11, 39).

[29] Tamas Czimmermann, Gastone Ciuti, Mario Milazzo, Maurizio Chiurazzi, Stefano Roccella, Calogero Maria Oddo, and Paolo Dario. "Visual-Based Defect Detection and Classification Approaches for Industrial Applications: A Survey". In: *Sensors* 20.5 (2020), p. 1459. DOI: 10.3390/s20051459 (cit. on pp. 11, 17–20).

[30] Rui Zhao, Ruqiang Yan, Zhenghua Chen, Kezhi Mao, Peng Wang, and Robert X. Gao. "Deep Learning and Its Applications to Machine Health Monitoring". In: *Mechanical Systems and Signal Processing* 115 (2019), pp. 213–237. DOI: 10.1016/j.ymssp.2018.05.050 (cit. on pp. 11, 18–20).

[31] Alan V. Oppenheim and Ronald W. Schafer. *Discrete-Time Signal Processing*. 3rd. Pearson Education, 2009. ISBN: 9780131988422 (cit. on p. 11).

[32] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *Proceedings of the 32nd International Conference on Machine Learning (ICML)*. 2015, pp. 448–456 (cit. on pp. 12, 13, 33, 39).

[33] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958 (cit. on pp. 13, 33, 39).

[34] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*. 2015 (cit. on p. 13).

[35] Ilya Loshchilov and Frank Hutter. "SGDR: Stochastic Gradient Descent with Warm Restarts". In: *International Conference on Learning Representations*. 2017. URL: https://openreview.net/forum?id=Skq89Scxx (cit. on pp. 13, 34).

[36] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. "Focal Loss for Dense Object Detection". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 2980–2988. DOI: 10.1109/ICCV.2017.324 (cit. on pp. 14, 34, 39).

[37] Rich Caruana. "Multitask Learning". In: *Machine Learning* 28.1 (1997), pp. 41–75. DOI: 10.1023/A:1007379606734 (cit. on p. 14).

[38] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. "On Calibration of Modern Neural Networks". In: *Proceedings of the 34th International Conference on Machine Learning*. 2017, pp. 1321–1330 (cit. on pp. 14, 15, 32).

[39] Alexandru Niculescu-Mizil and Rich Caruana. "Predicting Good Probabilities with Supervised Learning". In: *Proceedings of the 22nd International Conference on Machine Learning*. 2005, pp. 625–632. DOI: 10.1145/1102351.1102430 (cit. on pp. 15, 30).

[40] Dumitru Bacioiu, Gordon Melton, Mayorkinos Papaelias, and Robert Shaw. "Automated defect classification of Aluminium 5083 TIG welding using machine vision and multi-layer perceptron". In: *NDT & E International* 107 (2019), p. 102121. DOI: 10.1016/j.ndteint.2019.102121 (cit. on pp. 18, 19).

[41] Georg Stemmer, Jose A. Lopez, Juan A. Del Hoyo Ontiveros, Arvind Raju, Tara Thimmanaik, and Sovan Biswas. *Unsupervised Welding Defect Detection Using Audio and Video*. 2024. DOI: 10.48550/arXiv.2409.02290. arXiv: 2409.02290 [cs.RO]. URL: https://arxiv.org/abs/2409.02290 (cit. on pp. 18–20, 42, 92, 98).

[42] Bin Luo, Hongrui Wang, Haitao Liu, Bing Li, and Fangyu Peng. "Early fault detection of machine tools based on deep learning and dynamic identification". In: *IEEE Transactions on Industrial Electronics* 66.1 (2019), pp. 509–518. DOI: 10.1109/TIE.2018.2807414 (cit. on p. 18).

[43] Ben D. Fulcher and Nick S. Jones. "Highly comparative feature-based time-series classification". In: *IEEE Transactions on Knowledge and Data Engineering* 26.12 (2014), pp. 3026–3037. DOI: 10.1109/TKDE.2014.2316504. URL: https://doi.org/10.1109/TKDE.2014.2316504 (cit. on p. 25).

[44] Anthony Bagnall, Jason Lines, Aaron Bostrom, James Large, and Eamonn Keogh. "The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances". In: *Data Mining and Knowledge Discovery* 31 (2017), pp. 606–660. DOI: 10.1007/s10618-016-0483-9. URL: https://doi.org/10.1007/s10618-016-0483-9 (cit. on p. 25).

[45] Intel Labs. *Intel Robotic Welding Multimodal Dataset*. Hugging Face dataset page (gated access). 2024. URL: https://huggingface.co/datasets/IntelLabs/Intel_Robotic_Welding_Multimodal_Dataset (visited on 03/05/2026) (cit. on pp. 42, 98).

[46] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. "Distilling the Knowledge in a Neural Network". In: *arXiv preprint arXiv:1503.02531* (2015) (cit. on p. 99).

[47] Song Han, Huizi Mao, and William J. Dally. "Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding". In: *International Conference on Learning Representations*. 2016 (cit. on p. 100).

[48] Scott M. Lundberg and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems*. Vol. 30. 2017 (cit. on p. 100).

[49] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization". In: *International Journal of Computer Vision* 128.2 (2020), pp. 336–359 (cit. on p. 100).