Hurricane Damage Detection from Satelite Imagery

ENV-540 | IPEO | COURSE PROJECT | FALL 25

Jan Kokla | 367628 | jan.kokla@epfl.ch
Jan Zgraggen | 344351 | jan.zgraggen@epfl.ch
Mahlia Merville-Hipeau | 345625 | mahlia.merville-hipeau@epfl.ch

https://github.com/jankokla/ipeo-hurricane-damage-detection

1 Introduction

Natural disasters, especially hurricanes, are becoming more frequent and intense [1]. After such events, it is essential to quickly identify damaged areas to send help. To do so, deep learning algorithms have become a key tool to automatically analyze images and speed up the process.

However, using these models in real-life emergencies brings a new challenge: trust. It is not enough for a model to have high accuracy; we also need to know when it is unsure. A major risk arises when a model is "confidently wrong": for example, predicting that a destroyed area is safe with very high certainty. In a rescue scenario, this kind of error can lead to dangerous situations where damaged regions are overlooked.

Therefore, this project aims to go beyond simple accuracy metrics. We evaluate deep learning models with two main goals: first, to test how well they can detect damage visually, and second, to analyze their *calibration*: checking if their confidence scores actually match reality.

2 Topic and Challenges

This section outlines the current state of damage assessment using remote sensing and details the specific challenges regarding model reliability and calibration.

2.1 Topic

Traditionally, disaster damage assessment relied on manual photo-interpretation or change detection algorithms based on handcrafted features. The advent of Deep Learning has revolutionized this field, enabling end-to-end learning of complex damage signatures from raw optical data. As noted in recent comprehensive reviews, convolutional neural networks have become the *de facto* standard for remote sensing applications due to their ability to extract hierarchical features invariant to local distortions [2].

A major catalyst for progress in this domain has been the standardization of datasets. The release of large-scale benchmarks, such as the xBD dataset, has provided the community with annotated high-resolution satellite imagery before and after disasters [3]. These resources allow researchers to frame the problem as a supervised classification task, where models are trained to categorize image patches into varying levels of destruction (e.g., no damage, minor damage, destroyed). Despite these advances, the domain suffers from inherent difficulties, including class imbalance where undamaged buildings vastly outnumber damaged ones, and visual ambiguity caused by cloud cover or off-nadir viewing angles.

2.2 Challenges

While modern deep neural networks have achieved unprecedented accuracy on benchmarks, they face a critical limitation known as *miscalibration*. Calibration refers to the alignment between a model's predicted probability (confidence) and its true correctness likelihood. Ideally, if a model predicts a class with 80% confidence, it should be correct 80% of the time.

However, seminal work by Guo et al. has demonstrated that modern neural networks, particularly deep architectures like ResNet, tend to be overconfident [4]. This means they often assign high probability scores to incorrect predictions. In the context of disaster management, this overconfidence is dangerous. Decision-makers rely on probability thresholds to automate responses: if a model is uncalibrated, these thresholds become unreliable. Therefore, the challenge lies not only in training models to detect damage but in ensuring their confidence scores are a truthful proxy for their uncertainty.

3 Methods

3.1 Dataset

To analyze the efficacy and reliability of damage detection models, this study utilizes the dataset originally curated by Cao et al. [5]. The dataset consists of high-resolution RGB satellite image patches captured by the GeoEye-1 satellite, depicting regions in Texas, USA, following the devastation of Hurricane Harvey.

The data is categorized into two binary classes: *damaged* and *undamaged*. The *damaged* class is characterized by high intra-class variability, capturing diverse destruction patterns such as roof loss, severe flooding, and structural debris.

Data Leakage

Manual inspection of the data by fellow classmates. revealed duplicates between the test/validation and test/train splits. Evaluating the model on already seen

data hinders a proper assessment of the model's generalization ability, also known as data leakage.

Under the assumption that the leakage is the result of human error during dataset manipulation, we filter duplicates from the validation and test sets by creating a hashtable of image hashes for the validation and test sets and comparing the hashes of the training set against them. This revealed 662 *damaged* and 666 *undamaged* duplicates in the test set, as well as 2 *undamaged* duplicates in the validation set.

Data partitioning

After the cleaning process, the dataset was divided into functional subsets. The counts reported below reflect the final numbers after deduplication.

- **Training Set (16,670 images):** This set is used for the primary learning phase. It remains imbalanced (11,838 *damaged* / 4,832 *undamaged*), reflecting real-world scenarios where damaged structures are less frequent than intact ones. Initially, it contained a training pool of 19,000 images (13,000 *damaged* / 6,000 *undamaged*).

- **Calibration Set (1,500 images):** To improve model reliability, this dedicated subset is used to adjust confidence scores post-training. It consists of 1,000 images sampled from the training pool and 500 from the validation pool (balanced at 750 per class). This allows the calibration algorithm to learn from both seen and unseen data.

- **Validation and Test Sets (1,500 and 2,000 images):** These sets are strictly balanced (50/50 class distribution) to provide an unbiased assessment of performance. The validation set is used for hyperparameter tuning, while the test set serves as the final benchmark for the calibrated model.

Data quality

This dataset was specifically selected for its challenging nature, which mimics realistic deployment conditions. A significant *distributional shift* exists between the training and testing distributions. While the training data consists largely of clear examples, the validation and test sets intentionally include lower-quality samples and visually ambiguous patches. These ambiguities arise from varying sensor angles, lighting conditions, and partial occlusions.

This noise effectively tests the model's robustness. It serves as an ideal testbed for our calibration analysis, as we aim to determine whether the model can express appropriate uncertainty when maximizing predictions on these ambiguous, lower-quality inputs.

Data Augmentation

Training images were augmented using a modular, severity-controlled pipeline implemented with Albumentations [6]. All images were resized to match the required input resolution of the target backbone and normalized using ImageNet statistics [5]. Augmentations were composed incrementally from four transformation groups: *photometric* augmentations (brightness/contrast, color jitter, hue–saturation shifts, and CLAHE), *geometric* augmentations (random affine transforms, resized crops, perspective distortion, and horizontal flips), *blur and noise* augmentations (motion and Gaussian blur, additive noise, ISO noise, and downscaling), *distortion* augmentations (optical and grid distortion, random fog), with an optional coarse dropout stage. Within each group, transformations were sampled stochastically using `OneOf` operator [6] to encourage diversity.

3.2 Pipeline

As illustrated in Figure 1, our methodology is divided into a sequential process that separates initial representation learning from post-hoc calibration tuning.

Training Phase: Representation Learning

The training phase focuses on evaluating the intrinsic quality of features extracted by various architectures using a **frozen backbone** strategy. Input images are passed through a pre-trained model to extract high-dimensional *embeddings*, which serve as fixed inputs to a **linear classifier**. This approach ensures that only the classification head is trainable, allowing us to compare the effectiveness of different architectural inductive biases.

To evaluate calibration versus accuracy, we use diverse backbones ranging from CNN's to foundation models. We control for model scale by selecting architectures with comparable parameter counts (22-25M). This constraint keeps computational cost manageable ($\approx$ 1.5 hours for 30 training epochs on a single GPU with frozen backbones) and ensures that observed performance differences primarily arise from architectural differences and training dynamics rather than model size. The evaluated models include:

- **ResNet-50 (Baseline)**[1]**:** A standard Convolutional Neural Network (CNN) that introduced residual connections [7]. It possesses strong inductive biases for locality and translation invariance.

- **ResNeXt-50**[2]**:** An evolution of the ResNet architecture that utilizes grouped convolutions to increase "cardinality," improving accuracy without a proportional increase in complexity [8].

- **Vision Transformer (ViT)**[3]**:** A transformer-based architecture that processes images as sequences of patches using self-attention [9]. ViTs lack inherent spatial biases, often leading to different calibration profiles than CNNs.

[1] The model identifier used is `resnet50.a1_in1k` from the `timm` library via Hugging Face.

[2] `resnext50_32x4d.a1h_in1k`.

[3] `vit_small_patch16_224.augreg_in21k_ft_in1k`.

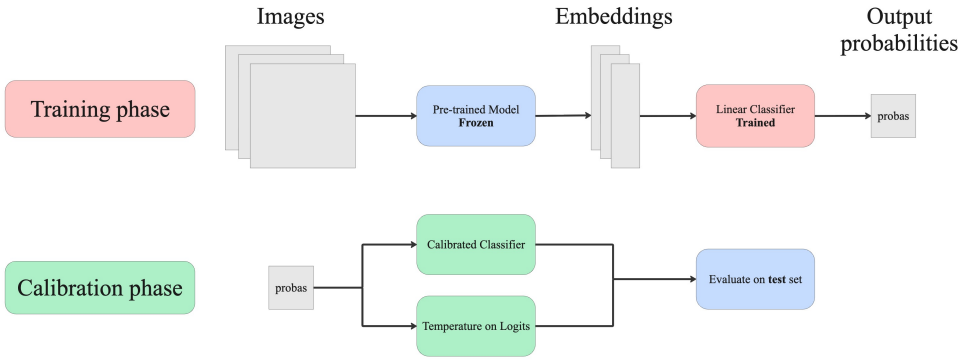

Figure 1: Overview of the global pipeline. The training phase (top) utilizes a frozen backbone to extract embeddings for a linear classifier. The calibration phase (bottom) refines these predictions to improve reliability before final evaluation.

- **DINOv3 (Foundation Model)**[4]**:** A state-of-the-art model pre-trained via self-supervised learning on massive datasets [10]. We evaluate DINOv3 to test if robust, general-purpose features generalize better to the distributional shifts in disaster imagery.

Calibration Phase: Reliability Tuning

Once the classifier is trained, we implement a multi-staged calibration strategy to align confidence scores with empirical accuracy. This stage ensures that the final predictions are statistically reliable for decision-making in humanitarian contexts.

Temperature Scaling Unlike mapping-based methods, Temperature Scaling acts directly on the raw logits (pre-softmax outputs). It introduces a scalar parameter T to adjust the entropy of the probability distribution (see Appendix A.1 for the derivation). We optimize T by minimizing the Negative Log Likelihood on the validation set. This method is calibration-preserving, meaning it adjusts confidence levels without altering the model's accuracy or class rankings [4].

Post-processing Calibration Following the initial scaling, we utilize the `CalibratedClassifierCV` wrapper from *scikit-learn* to map probabilities p to calibrated values $\hat{p}$. To prevent data leakage, we use the `cv='prefit'` parameter, ensuring these models are fitted exclusively on our dedicated 1,500-image calibration set.

- **Sigmoid Scaling:** A parametric method that fits a logistic regression model to the classifier's outputs [11]. It is highly effective for correcting sigmoidal distortions common in neural network outputs.
- **Isotonic Regression:** A non-parametric approach that fits a piecewise constant non-decreasing function to the probabilities [12]. It offers greater flexibility for correcting monotonic distortions but requires a larger dataset to avoid overfitting.

[4] `vit_small_patch16_dinov3.lvd1689m`.

4 Experimental Setup

4.1 Implementation Details

All models were implemented using PyTorch. We utilized the `timm` library to instantiate the backbone architectures and load pre-trained weights. Training was conducted on a single NVIDIA GeForce GTX TITAN X GPU with 12 GB of VRAM. To ensure reproducibility, we fixed the random seeds for the data loader and model initialization.

4.2 Training Strategy

Hyperparameters

All backbones were trained for *30 epochs* using the `Adam` optimizer. To accommodate high-resolution patches within GPU memory, a batch size of 16 was used along for all experiments. We tried a grid search on the learning rate with values ranging in 1×10^{-4} and 1×10^{-3}.

Loss Functions

For model optimization and performance assessment, we utilize two distinct functions targeting classification accuracy and probability calibration.

- **Binary Cross-Entropy (BCE):** Used for primary optimization, this loss is implemented via `BCEWithLogitsLoss` for numerical stability. It evaluates the divergence between predicted logits and ground truth labels (see Appendix A.1 for Eq. A.1.2).
- **Brier Score:** A strictly proper scoring rule used to evaluate calibration by measuring the mean squared difference between predicted probabilities and actual outcomes. A lower score indicates a model that avoids both misclassification and overconfidence (see Appendix A.1 for Eq. A.1.3).

4.3 Augmentation Ablation

To determine the optimal regularization strength, we evaluated four cumulative augmentation tiers on the ResNet-50 baseline: *Light* (Photometric + Geometric),

Moderate (adding Blur & Noise), *Aggressive* (adding Distortions), and *Overkill* (adding Coarse Dropout).

This progressive evaluation allowed us to identify the "Moderate" setting as the best balance between preventing overfitting and maintaining representative features. The details of these transformation groups are provided in section 3.1.

4.4 Evaluation Metrics

We evaluate classification performance using the **F1-score** to ensure a balance between precision and recall. Given our focus on trustworthiness, we also report the **Expected Calibration Error (ECE)**. This metric quantifies the misalignment between the model's confidence and its actual accuracy by binning predictions and calculating a weighted average of their differences. Lower ECE values indicate a better-calibrated model (see Appendix A.1 for Eq. A.1.4).

To further assess the quality of the predicted probabilities, we compute the **Log Loss** (Negative Log-Likelihood). Unlike hard classification metrics, Log Loss heavily penalizes "confidently wrong" predictions, making it an essential metric for measuring the information gain of our models and their suitability for decision-making in high-stakes environments (see Appendix A.1 for Eq. A.2).

Finally, we utilize **Calibration Curves** (reliability diagrams) to qualitatively inspect model behavior. These curves plot the observed frequency of a class against its predicted probability. A perfectly calibrated model follows the 45° identity line. Deviations from this diagonal reveal systematic biases, such as overconfidence (the curve lies below the diagonal) or underconfidence (the curve lies above the diagonal), providing visual confirmation of the improvements achieved through our calibration phase.

4.5 Model Selection

To determine the best-performing architecture and calibration strategy, we evaluate all candidates on the **validation set**. Following our dual focus on performance and reliability, we select the model that yields the highest **F1-score** while simultaneously achieving the lowest **ECE**. Also, we monitor the calibration curve to choose the best model.

5 Results

5.1 Data Augmentation

Among the evaluated augmentation configurations, the moderate augmentation setting (photometric, geometric and blur & noise) (see section 3.1) consistently outperformed lighter and more aggressive variants on the ResNet-50 baseline. Based on this result, the moderate augmentation strategy was adopted for all subsequent experiments. From Figure 3 in subsection B.1, we observe that the validation loss closely follows the training loss, indicating good generalization under the proposed augmentation strategy.

5.2 Backbones Architectures

The validation results highlight clear differences between backbone architectures. As shown in Table 1, ViT and DINOv3 substantially outperform the convolutional baselines in terms of F1 score and log loss. However, ViT exhibits slightly better calibration than DINOv3, with a lower ECE (0.007 vs 0.020). Based on this trade-off between accuracy and calibration, subsequent experiments focus on ViT and DINOv3.

Table 1: Comparison of backbone architectures on validation performance.

Model Backbone	F1 ↑	ECE ↓	Log Loss ↓
Convolutional			
ResNet-50	0.909	0.061	0.269
ResNeXt-50	0.889	0.044	0.296
Transformers			
ViT-Small	0.934	0.007	0.164
DINOv3	0.938	0.020	0.166

5.3 Loss Functions

The effect of different training losses on calibration was also examined. Although Brier loss is theoretically expected to improve calibration by directly penalizing miscalibrated probabilities, the empirical results for DINOv3 and ViT do not support this assumption. In both cases, models trained with BCE achieved lower ECE than those trained with Brier loss. Detailed results are reported in Table 2 in Appendix A.3.

5.4 Calibration

5.4.1 Temperature

Temperature scaling was then applied to DINOv3 and ViT to assess its effect on calibration. As shown in Table 3 in Appendix A.3, DINOv3's calibration improved (ECE 0.011 vs 0.020), and this configuration achieved the highest validation F1 score observed so far (0.944). In contrast, ViT degraded under temperature scaling, with both log loss and ECE increasing to 0.170 and 0.022, respectively. The corresponding reliability diagrams are presented in Figure 2. Although ViT achieves a slightly lower ECE than temperature-scaled DINOv3 (0.007 vs 0.011), the latter attains marginally better F1 and log loss. Therefore, subsequent analysis focuses on DINOv3 with temperature scaling.

5.4.2 Post-Hoc Methods

Finally, post-hoc calibration methods, specifically sigmoid and isotonic regression, were fitted on the calibration set. As illustrated in Figure 8 and seen from Table 4, the sigmoid-adjusted model performs almost identically to the temperature-scaled DINOv3 baseline, while isotonic regression substantially degrades both ECE and log loss, likely due to the non-parametric nature of the method overfitting on the calibration set. Given the negligible gains from the sigmoid regressor and the deterioration under isotonic regression, the subsequent

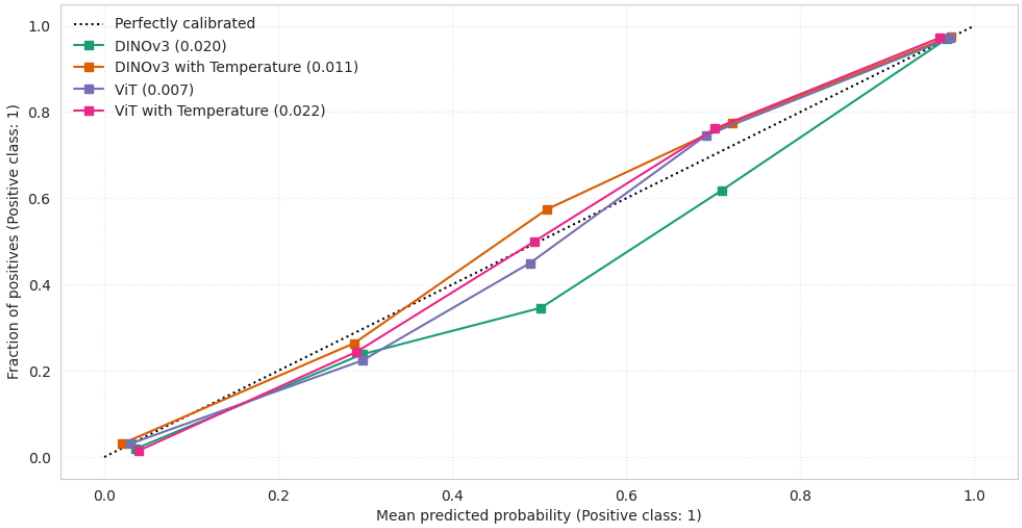

Figure 2: Reliability diagrams for DINOv3 and ViT with and without temperature scaling. The dashed identity line denotes perfect calibration, and bracketed values indicate the corresponding ECE scores.

analysis adopts the temperature-scaled DINOv3 model without any additional post-hoc wrapper.

5.5 Final Results

Prior to final evaluation on the test set, the DINOv3 model (with temperature scaling) was trained for 50 epochs to ensure full convergence, as initial results suggested 30 epochs were insufficient. During this phase, we conducted a grid search across several learning rates ($1e-4$, $3e-4$, $1e-3$, and $3e-3$). The training dynamics and detailed performance metrics are documented in Figure 6 (Appendix B.1) and Table 5 (Appendix A.3). Interestingly, the model using the highest learning rate ($3e-3$) performed best, achieving an F1 score of 0.948 and an ECE of 0.008. Upon test set evaluation, the top-performing DINOv3 model with temperature scaling achieved an **F1 score of 0.943** and an **ECE of 0.057**.

6 Discussion and Conclusion

This study evaluated the interplay between architectural biases and calibration strategies for disaster damage assessment. Our results highlight several key insights regarding the development of trustworthy AI for humanitarian aid.

Augmentation and Distributional Shifts The intentional noise gap between the training set and the validation/test sets simulated realistic deployment conditions. Our ablation study confirmed that moderate augmentation, incorporating photometric and noise-based transforms, is essential to mitigate this distributional shift. By forcing the models to learn invariant features rather than sensor-specific noise, we significantly reduced the generalization gap.

Transformers vs. Convolutional Backbones We observed a clear superiority of Transformer-based architectures (ViT and DINOv3) over standard CNNs. As shown in Table 1, both outperformed the ResNet baselines in F1 score and calibration. The self-attention mechanism appears better suited for capturing the global context of structural damage than the local inductive biases of traditional convolutional layers.

Inherent Calibration of Foundation Models Our results suggest that self-supervised Foundation Models (FMs) are already quite well-calibrated out of the box. While supervised networks often overfit to labels, the general-purpose features learned by DINOv3 provide a robust basis for uncertainty estimation. Interestingly, despite expectations that Brier loss would improve calibration, models trained with standard BCE achieved lower ECE scores (see Table 2).

Limited Impact of Calibration Methods A key finding is that calibration interventions provided only marginal gains for these high-performing backbones. While Temperature Scaling improved DINOv3's ECE from 0.020 to 0.011, it actually degraded ViT's calibration (Table 3). Similarly, post-hoc Sigmoid scaling provided a negligible ECE reduction of only 0.002, while Isotonic regression significantly worsened performance (Table 4). This suggests that for robust backbones like DINOv3, complex post-hoc mapping may be unnecessary or even detrimental.

Final Conclusion Our best-performing model: DINOv3 trained with a learning rate of $3e-3$ and temperature scaling, attained an **F1 score of 0.943** and an **ECE of 0.057** on the test set. This study demonstrates that while accuracy is vital, monitoring calibration is paramount for disaster response. By ensuring confidence scores are a truthful proxy for reality, we can deploy automated systems that assist rescuers more reliably, reducing the risk of overlooked damage in critical humanitarian scenarios.

References

[1] J. K. Summers, A. Lamper, C. McMillion, and L. C. Harwell. Observed changes in the frequency, intensity, and spatial patterns of nine natural hazards in the united states from 2000 to 2019. *Sustainability*, 14(7):4158, March 2022.

[2] Lei Ma, Yu Liu, Xueliang Zhang, Yuanxin Ye, Gaofei Yin, and Brian Alan Johnson. Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 152:166–177, 2019.

[3] Ritwik Gupta, Bryce Goodman, Nirav Patel, Ricky Hosfelt, Sandra Sajeev, Eric Heim, Jure Doshi, Kaleb Lucas, Howie Choset, and Matthew Gaston. xbd: A dataset for assessing building damage from satellite imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 10–17, 2019.

[4] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.

[5] Quoc Cao and Yoonsuck Choe. Building damage annotation on post-hurricane satellite imagery based on convolutional neural networks. *Natural Hazards*, 103:3361–3381, 2020.

[6] Alexander Buslaev, Vladimir I. Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A. Kalinin. Albumentations: Fast and flexible image augmentations. *Information*, 11(2):125, 2020.

[7] 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*, pages 770–778, 2016.

[8] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[10] Oriane Simeoni et al. Dinov3: Learning state-of-the-art visual features with self-supervised learning. *arXiv preprint*, 2025. Meta AI Research.

[11] John Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in large margin classifiers*, volume 10, pages 61–74, 1999.

[12] Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699, 2002.

A Appendix

A.1 Mathematical Formulations

A.1.1 Temperature Scaling

Temperature scaling adjusts the predicted probability $\hat{p}_i$ for class i by dividing the raw logits z by a scalar T:

$$\hat{p}_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)} \tag{1}$$

This transformation preserves class rankings while adjusting the distribution's entropy to match empirical accuracy [4].

A.1.2 Binary Cross-Entropy (BCE)

The BCE loss for a batch of size N is defined as:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} [y_i \cdot \log(\sigma(x_i)) + (1 - y_i) \cdot \log(1 - \sigma(x_i))] \tag{2}$$

where y_i is the ground truth, x_i is the raw logit, and σ is the sigmoid function.

A.1.3 Brier Score

The Brier Score measures the accuracy of probabilistic predictions:

$$BS = \frac{1}{N}\sum_{i=1}^{N}(\hat{p}_i - y_i)^2 \tag{3}$$

where $\hat{p}_i$ is the predicted probability for sample i.

A.1.4 Expected Calibration Error (ECE)

The ECE measures the difference between confidence and accuracy by partitioning predictions into M bins based on confidence. It is calculated as:

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} \left|\text{acc}(B_m) - \text{conf}(B_m)\right| \tag{4}$$

where:

- n is the total number of samples.
- $|B_m|$ is the number of samples in bin m.
- $\text{acc}(B_m)$ is the average accuracy of bin m.
- $\text{conf}(B_m)$ is the average confidence of bin m.

A.2 Log Loss (Negative Log-Likelihood)

The Log Loss for a binary classification task is calculated as:

$$\text{LogLoss} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \ln(p_i) + (1 - y_i)\ln(1 - p_i)] \tag{5}$$

where N is the number of samples, y_i is the binary label, and p_i is the predicted probability for the positive class.

A.3 Results of the Experiments

Table 2: Calibration performance of DINOv3 and ViT trained with BCE and Brier loss.

Model / Loss	F1 ↑	ECE ↓	Log Loss ↓
DINOv3			
BCE	0.938	0.020	0.166
Brier Loss	0.935	0.029	0.181
ViT			
BCE	0.934	0.007	0.164
Brier Loss	0.939	0.027	0.171

Table 3: Validation performance of DINOv3 and ViT with and without temperature scaling.

Model / Method	F1 ↑	ECE ↓	Log Loss ↓
DINOv3			
Raw	0.938	0.020	0.166
+ Temperature	0.944	0.011	0.153
ViT			
Raw	0.934	0.007	0.164
+ Temperature	0.937	0.022	0.170

Table 4: Validation performance of DINOv3 with temperature scaling and post-hoc regressors.

Model / Method	F1 ↑	ECE ↓	Log Loss ↓
DINOv3 w. temp.	0.944	0.011	0.153
+ Sigmoid	0.946	0.009	0.153
+ Isotonic	0.936	0.021	0.308

Table 5: Validation performance of DINOv3 across different learning rates.

Model / Method	F1 ↑	ECE ↓	Log Loss ↓
lr $1e-4$	0.938	0.019	0.163
lr $3e-4$	0.946	0.012	0.152
lr $1e-3$	0.948	0.018	0.148
lr $3e-3$	0.948	0.008	0.146

B Ethical Considerations

Humanitarian Impact and Bias

Models may underperform on underrepresented rural or informal housing, risking inequitable aid distribution. Calibration mitigates "confidently wrong" predictions that could mislead emergency responders in time-critical scenarios.

Use of Generative AI

LLMs were used strictly for structural refinement, grammatical flow, and LaTeX formatting. All experimental design, data analysis, and conclusions are the original work of the authors and were manually verified for accuracy.

B.1 Training Loss & F1 Curves

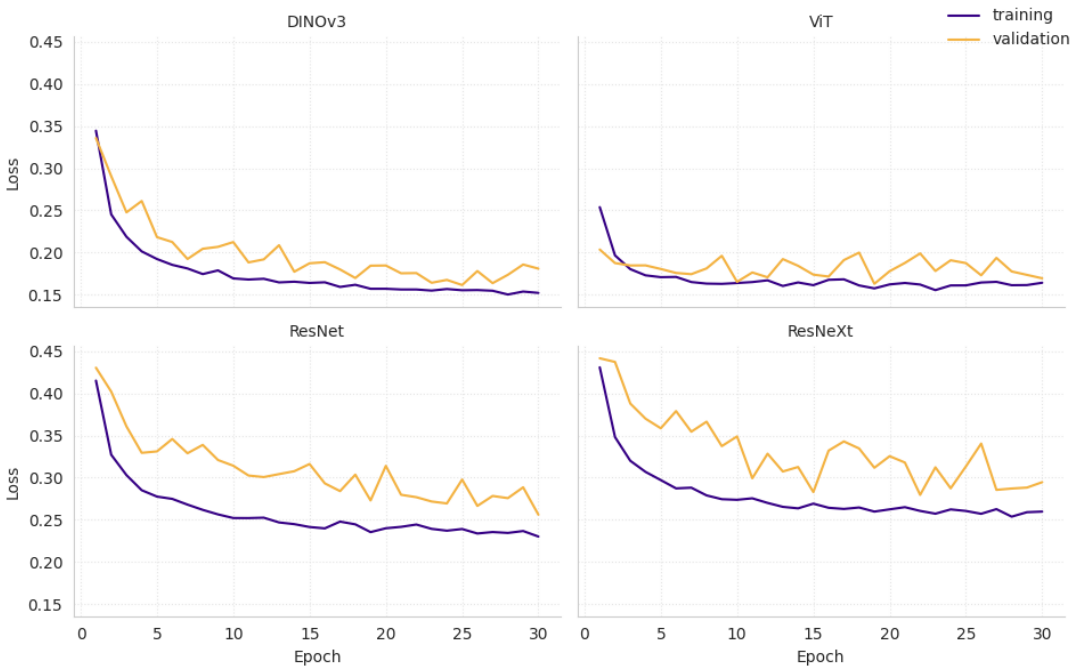

Figure 3: Training and validation loss curves for different backbone architectures.

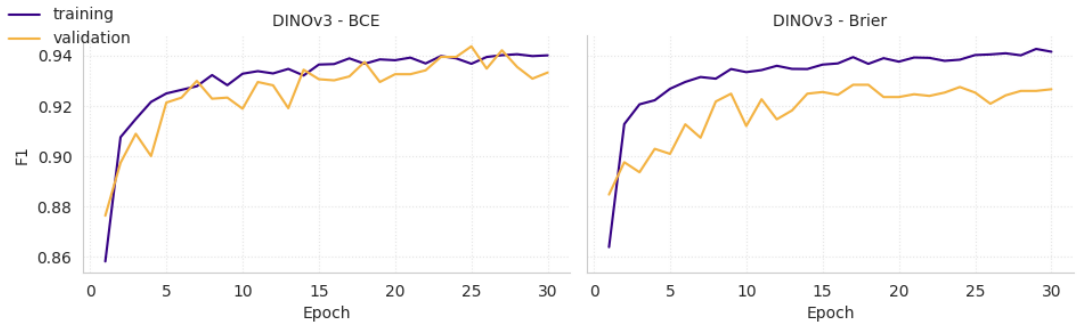

Figure 4: Training and validation F1 dynamics for different loss functions.

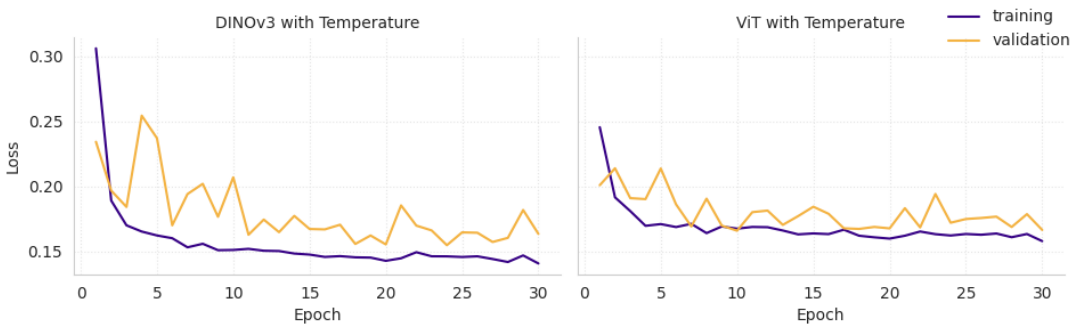

Figure 5: Training and validation BCE loss dynamics with temperature scaling.

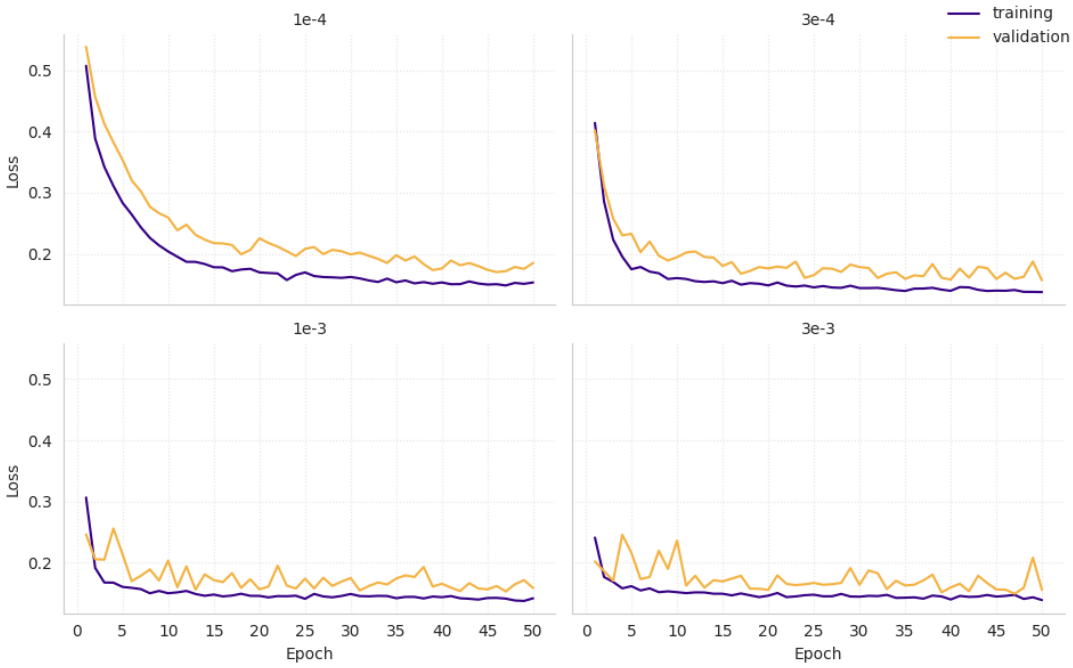

Figure 6: Training and validation BCE loss dynamics with different learning rates.

B.2 Calibration Curves

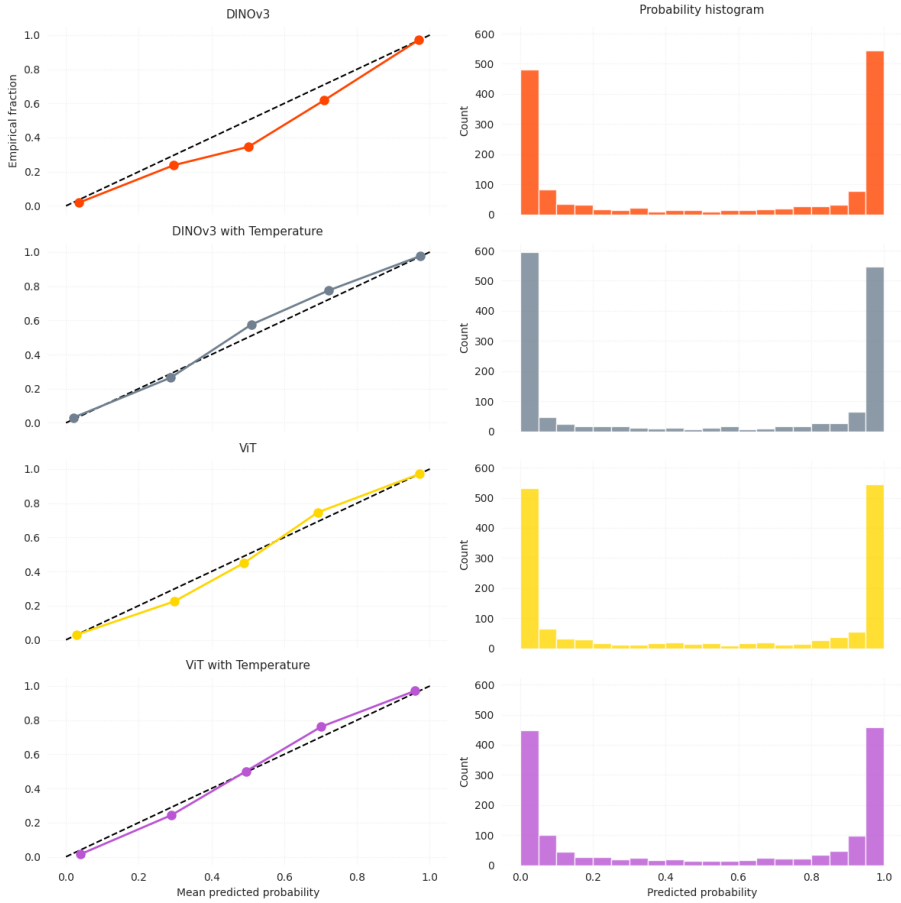

Figure 7: Calibration curves of models with and without temperature scaling.

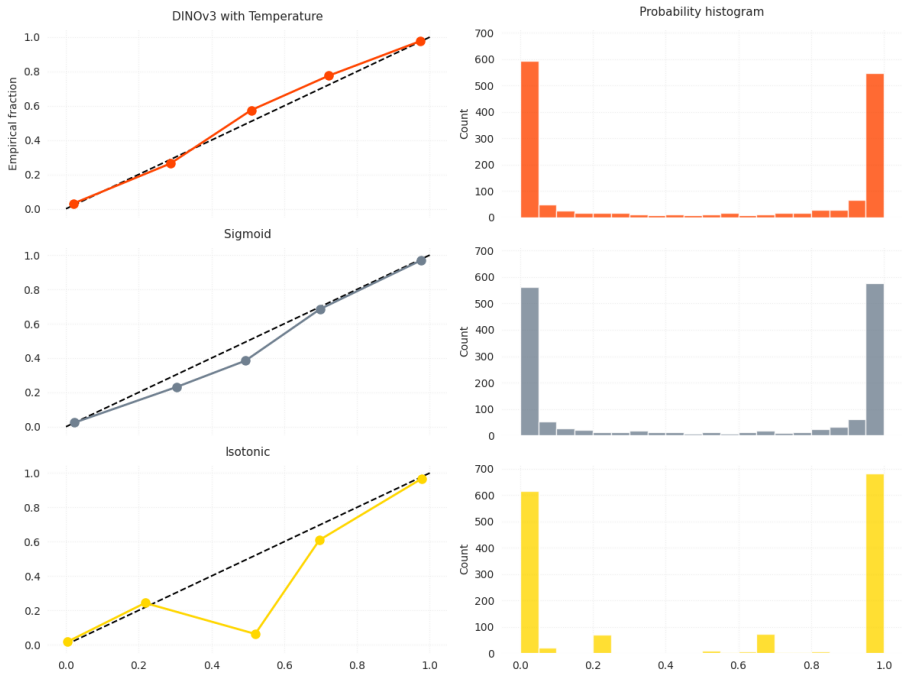

Figure 8: Calibration curves for post-hoc regressors applied to DINOv3 with temperature scaling.