Anomaly Detection In The Wild: Can SSL Handle Strong Distribution Imbalances?

Anonymous Author(s)
Affiliation
Address
email

Abstract

Accurate anomaly detection is critical in vision-based infrastructure inspection, where it helps prevent costly failures and enhances safety. Self-Supervised Learning (SSL) offers a promising approach by learning robust representations from unlabeled data. However, its application in anomaly detection remains underexplored. This paper addresses this gap by providing a comprehensive evaluation of SSL methods for real-world anomaly detection, focusing on sewer infrastructure. Using the Sewer-ML dataset, we evaluate lightweight models such as ViT-Tiny and ResNet-18 across SSL frameworks, including BYOL, Barlow Twins, SimCLR, DINO, and MAE, under varying class imbalance levels. Through 250 experiments, we rigorously assess the performance of these SSL methods to ensure a robust and comprehensive evaluation. Our findings highlight the superiority of joint-embedding methods like SimCLR and Barlow Twins over reconstruction-based approaches such as MAE, which struggle to maintain performance under class imbalance. Furthermore, we find that the SSL model choice is more critical than the backbone architecture. Additionally, we emphasize the need for better label-free assessments of SSL representations, as current methods like RankMe fail to adequately evaluate representation quality, making cross-validation without labels infeasible. Despite the remaining performance gap between SSL and supervised models, these findings highlight the potential of SSL to enhance anomaly detection, paving the way for further research in this underexplored area of SSL applications.

1 Introduction

Self-Supervised Learning (SSL) is a machine learning paradigm where models are trained on unlabeled data by creating surrogate labels through pretext tasks that exploit inherent structures or patterns within the data. As a result, this approach enables learning meaningful representations that can be applied to various downstream tasks without the need for explicit manual labeling [4]. Because of this, SSL is particularly advantageous for semi-supervised anomaly detection problems where obtaining labeled data is costly, labor-intensive, impossible, or undesirable [1]. Despite these benefits, anomaly detection is frequently underrepresented in SSL research, with recent large-scale ablation studies often neglecting its inclusion in benchmarking [10, 2]. In fact, common benchmarks such as ImageNet [7] and CIFAR [16], are object-centric and do not accurately reflect the complexity of real-world environments, where images are more diverse and less structured [11].

Closer to the anomaly detection scenario, recent works have started to stress-test SSL on more realistic scenarios with uncurated data. Albeit still employing a classification task evaluation, it has been shown that SSL may be sensitive to the quality of the data and in particular to balance representations of the features to be learned [18, 3]. Clearly, such warning seems to go against the

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

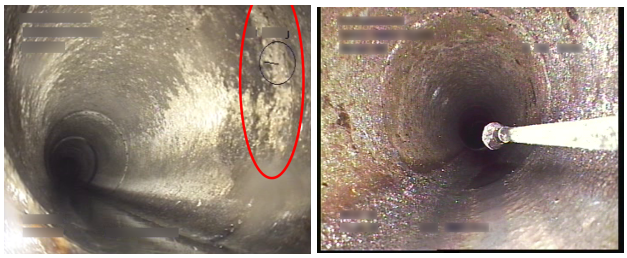


Figure 1: Samples from the Sewer-ML dataset [13]. The red circle highlights a defect associated with lateral reinstatement cuts, where improper cutting or misalignment can cause issues such as blockages, leaks, or structural weakness in the sewer system (for further details on defect types and image samples used for training and validation, refer to Appendix E).

ability of SSL to solve anomaly detection as such scenario assumes by nature that the model can learn the anomaly features from very few samples.

This gap in evaluation coupled with those recent sensitivity studies of SSL beg the following question:

Can SSL successfully learn representations on highly imbalance datasets and capture salient features to solve anomaly detection tasks?

To scientifically approach that question, we propose the use of the Sewer-ML dataset [13], a sewerage infrastructure dataset that contains 1.3 million images captured from video inspections, featuring 17 different defect classes. Figure 1 presents examples of defect and non-defect images taken from Sewer-ML. To determine the robustness of self-supervised learning in handling class imbalances, we conducted 250 binary classification experiments where we systematically vary the proportion of defect samples in the train and validation datasets as 1%, 2%, 5%, and 15%.

We summarize our key findings below:

- **Superiority of joint-embedding methods over reconstruction methods:** we observe that MAE has difficulties to maintain performance with varying class imbalance as opposed to methods like SimCLR or BarlowTwins that compete with supervised baselines.
- **Impact of SSL methodology over backbone architecture:** the choice of backbone architecture (Resnet or Transformer) is not critical compared to the choice of SSL method.
- **Need for better label-free assessment of SSL representations:** our findings indicate that methods such as RankMe fail to assess the richness of SSL representations making cross-validation without labels currently impossible on such task and dataset.

The code for this research will be open-sourced and is available at Anomaly-Detection-In-The Wild_code.zip.

2 Controlled evaluation of self-supervised anomaly detection

We will first describe our methodology in detail and then provide results and discussions at the end of this section.

2.1 Methodology

Prior to conducting the SSL ablation study, an initial hyperparameter search was performed to optimize the data augmentation pipeline for the dataset. The focus was placed on tuning image resolution and augmentation settings to achieve a balance between model performance and computational efficiency. Detailed results and configurations are provided in Appendix A.

Methods and models. Our study conducts an ablation analysis on anomaly detection using self-supervised learning methods, with a particular emphasis on their robustness to distribution imbalances. We primarily focus on joint-embedding architectures—specifically Barlow Twins, SimCLR, BYOL, and DINO [19, 6, 12, 5]—which aim to learn an embedding space by aligning representations of different augmented views of the same input while avoiding collapse. To provide a comparative perspective, we also evaluate Masked Autoencoders (MAE) [14], a self-supervised approach that reconstructs missing parts of the input data. For our backbone architectures, we use lightweight models such as ViT-Tiny [8] and ResNet-18 [15]. Please refer to Appendix B to inquire about specific training hyperparameters used for each method.

Controlled dataset imbalance. To determine the robustness of self-supervised learning in handling class imbalances, we conduct binary classification experiments with the Sewer-ML dataset [13], where we systematically vary the proportion of defect samples in the dataset. Specifically, we assess the performance of these methods under different levels of class imbalance—defect sample proportions of 1%, 2%, 5%, and 15%. As a binary setting, Sewer-ML has an approximately balanced distribution by design-45% and 47% defect proportion on the train and validation sets, respectively. We seek to assess how well self-supervised learning approaches can learn from imbalanced distributions and to evaluate their effectiveness on datasets with varying levels of class imbalance. We apply each of the specified defect proportions to both the training and testing datasets, allowing us to evaluate model performance across all combinations of class imbalance scenarios.

Evaluation metrics. In addition to reporting performance based solely on the F1 score, we also evaluate our models using the metrics proposed by the benchmark. This includes a weighted F2 metric ($F2_{CIW}$) for identifying defects and a conventional F1 score ($F1_{Normal}$) for non-defect classifications [13]. The F2 metric's weights are assigned to each defect category according to their economic significance. Additionally, they encourage the use of an F2 score to give greater emphasis to recall over precision, recognizing that overlooking a defect incurs a higher economic cost than the occurrence of a false positive.

2.2 Results

We present in Table 1 the performance of various SSL models compared to their supervised counterparts. Training was conducted on imbalanced data, while validation adhered to a balanced set, aligning with benchmark standards. Although previous studies have noted that CNN-based architectures like ResNet-18 often outperform vision transformers (ViTs) [10], our results do not indicate a significant difference in performance between these backbones architectures.

Robustness of joint-embedding methods. When examining individual SSL methods, SimCLR achieves the best overall results especially when paired with ResNet-18 and defect proportion is higher than 5%. When working with stronger distribution imbalances, particularly with 1%, BYOL and Resnet-18 have the best results. This might be due to the training dynamics behind these methods, in situations with moderate imbalance (like 5%), the contrastive approach of SimCLR can still adequately separate the minority class from the majority, leveraging the discriminative power of the contrastive loss. This can be counter-productive when working with extremely imbalance levels. BYOL avoids this issue by not requiring explicit negative sampling, allowing it to maintain more consistent performance even when the class imbalance becomes more extreme. Thus, while SimCLR thrives with moderate imbalance, BYOL proves more resilient in handling extreme class disparities. Another interesting insight is Barlow Twins' competitive performance under extreme imbalance. Its use of redundancy reduction loss, which both maximizes similarity between augmented views and decorrelates the learned representations, may help avoid the pitfalls of overfitting to the dominant class. DINO, however, shows a unique trend, performing poorly in 1% and 2% settings with ResNet-18, but not with ViT-Tiny. This could be attributed to DINO's focus on global feature learning through knowledge distillation, which aligns better with ViT's global attention approach.

Failure of reconstruction-based methods. Finally, the Masked Autoencoder (MAE) delivers the weakest performance across all scenarios, especially at higher imbalance levels (e.g., 45%). MAE's reliance on reconstructing multiple classes might introduce noise that hampers its ability to generalize under severe imbalance. In highly imbalanced data, MAE could face challenges in differentiating between common and rare classes, possibly due to the reconstruction bias that favors the majority class. This might also be tied to the model's neural capacity, which can struggle to produce robust representations when not sufficiently overparameterized.

Full tables with training and validation results are in Appendix C.2. Detailed analysis of each model's performance on imbalanced test distributions is provided in Appendix C.1. SSL monitoring metrics including RankMe [9] and the mean and standard deviation of features and embeddings for DINO, BYOL, and SimCLR—are discussed in Appendix D.

Performance Trends on Imbalanced Validation Sets. When evaluating the models on increasingly imbalanced validation sets (Figure 2), there is a clear and consistent decline in performance across most methods, as indicated by the F1 scores. As the validation set imbalance becomes more severe (e.g., at 1% and 2% imbalance), no method is able to sustain strong performance, particularly in

Table 1: Validation metrics for each method and architecture across different imbalance levels. The percentages represent the imbalance levels applied during training, while the validation data proportion remained unchanged.

		ResNet-18					ViT-Tiny				
		1%	**2%**	**5%**	**15%**	**45%**	**1%**	**2%**	**5%**	**15%**	**45%**
F1↑	*Supervised*	0.831	0.847	0.865	0.882	0.898	0.768	0.785	0.831	0.858	0.878
	BYOL	0.764	0.768	0.774	0.783	0.792	0.676	0.738	0.752	0.757	0.730
	Barlow Twins	0.768	0.775	0.789	0.796	0.803	0.682	0.738	0.747	0.769	0.772
	DINO	0.049	0.032	0.673	0.719	0.727	0.691	0.683	0.553	0.704	0.719
	MAE	–	–	–	–	–	0.513	0.636	0.570	0.575	0.024
	SimCLR	0.740	0.778	0.796	0.807	0.814	0.762	0.768	0.771	0.778	0.786
$F2_{CIW}$↑	*Supervised*	0.838	0.849	0.880	0.904	0.879	0.837	0.812	0.831	0.870	0.881
	BYOL	0.837	0.844	0.819	0.825	0.785	0.680	0.768	0.760	0.781	0.725
	Barlow Twins	0.817	0.805	0.817	0.822	0.780	0.704	0.749	0.786	0.784	0.758
	DINO	0.566	0.402	0.712	0.777	0.752	0.733	0.736	0.690	0.724	0.695
	MAE	–	–	–	–	–	0.629	0.640	0.704	0.702	0.009
	SimCLR	0.775	0.813	0.826	0.822	0.813	0.766	0.795	0.798	0.801	0.788
$F1_{Normal}$↑	*Supervised*	0.850	0.861	0.879	0.894	0.908	0.774	0.794	0.844	0.869	0.890
	BYOL	0.758	0.759	0.783	0.789	0.818	0.719	0.757	0.778	0.771	0.781
	Barlow Twins	0.782	0.796	0.799	0.807	0.828	0.716	0.760	0.766	0.785	0.806
	DINO	0.689	0.686	0.700	0.730	0.763	0.655	0.688	0.419	0.727	0.759
	MAE	–	–	–	–	–	0.471	0.350	0.553	0.389	0.677
	SimCLR	0.752	0.793	0.806	0.818	0.836	0.782	0.781	0.786	0.785	0.810

terms of accurately identifying the minority class. On the other hand, the F2 score remains relatively stable, likely due to the use of a weighted loss function based on defect proportions.

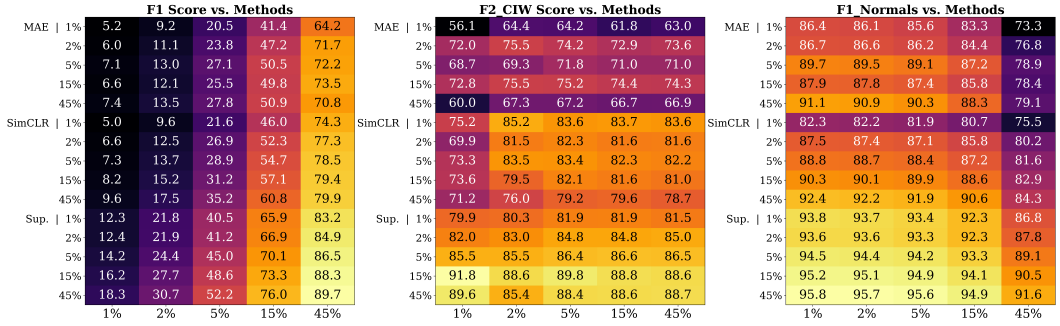


Figure 2: ResNet-18 validation performance heatmaps across imbalance levels. The x-axis represents the imbalance levels in the validation set, while the y-axis indicates the method and the imbalance level used during training.

3 Conclusions and discussion

Our study indicates that self-supervised learning (SSL) is effective for anomaly detection and remains robust even when facing significant distribution imbalances. We also find that the choice of backbone architecture is not the most critical factor in model performance, as neither ViT-Tiny nor ResNet-18 consistently outperforms the other across all cases. In contrast, the selection of the SSL methodology significantly impacts performance, with substantial variations observed among different SSL model families. Therefore, for practitioners, choosing the appropriate SSL family is more crucial than selecting a specific backbone architecture. Furthermore, there is a pressing need to accurately measure the quality of representations produced by SSL models. Our findings indicate that the RankMe metric is ineffective for this purpose; it aims to assess the richness of representations but fails to correlate with actual performance. As shown in Appendix D, there is no correlation between performance and RankMe metrics, underscoring the necessity for better methods of evaluating representations quality.

References

[1] Leman Akoglu and Jaemin Yoo. Self-supervision for tackling unsupervised anomaly detection: Pitfalls and opportunities. In *2023 IEEE International Conference on Big Data (BigData)*, pages 1047–1051, Los Alamitos, CA, USA, dec 2023. IEEE Computer Society.

[2] Haider Al-Tahan, Quentin Garrido, Randall Balestriero, Diane Bouchacourt, Caner Hazirbas, and Mark Ibrahim. UniBench: Visual Reasoning Requires Rethinking Vision-Language Beyond Scaling. (arXiv:2408.04810), August 2024. arXiv:2408.04810 [cs].

[3] Mido Assran, Randall Balestriero, Quentin Duval, Florian Bordes, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, and Nicolas Ballas. The Hidden Uniform Cluster Prior in Self-Supervised Learning. In *The Eleventh International Conference on Learning Representations*, 2023.

[4] Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, Avi Schwarzschild, Andrew Gordon Wilson, Jonas Geiping, Quentin Garrido, Pierre Fernandez, Amir Bar, Hamed Pirsiavash, Yann LeCun, and Micah Goldblum. A Cookbook of Self-Supervised Learning. (arXiv:2304.12210), June 2023.

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jegou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9630–9640, 2021.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 13–18 Jul 2020.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proceedings of the 2021 International Conference on Learning Representations (ICLR)*, 2021.

[9] Quentin Garrido, Randall Balestriero, Laurent Najman, and Yann LeCun. RankMe: Assessing the Downstream Performance of Pretrained Self-Supervised Representations by Their Rank. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[10] Micah Goldblum, Hossein Souri, Renkun Ni, Manli Shu, Viraj Uday Prabhu, Gowthami Somepalli, Prithvijit Chattopadhyay, Mark Ibrahim, Adrien Bardes, Judy Hoffman, Rama Chellappa, Andrew Gordon Wilson, and Tom Goldstein. Battle of the Backbones: A Large-Scale Comparison of Pretrained Models across Computer Vision Tasks. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

[11] Priya Goyal, Mathilde Caron, Benjamin Lefaudeux, Min Xu, Pengchao Wang, Vivek Pai, Mannat Singh, Vitaliy Liptchinsky, Ishan Misra, Armand Joulin, and Piotr Bojanowski. Self-supervised Pretraining of Visual Features in the Wild. (arXiv:2103.01988), March 2021. arXiv:2103.01988 [cs].

[12] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning. *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[13] Joakim Bruslund Haurum and Thomas B. Moeslund. Sewer-ML: A Multi-Label Sewer Defect Classification Dataset and Benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13456–13467, June 2021.

[14] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked Autoencoders Are Scalable Vision Learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16000–16009, June 2022.

[15] 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)*, pages 770–778, 2016.

[16] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, University of Toronto, 2009.

[17] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library, 2019.

[18] Huy V. Vo, Vasil Khalidov, Timothée Darcet, Théo Moutakanni, Nikita Smetanin, Marc Szafraniec, Hugo Touvron, camille couprie, Maxime Oquab, Armand Joulin, Herve Jegou, Patrick Labatut, and Piotr Bojanowski. Automatic Data Curation for Self-Supervised Learning: A Clustering-Based Approach. *Transactions on Machine Learning Research*, 2024.

[19] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stephane Deny. Barlow Twins: Self-Supervised Learning via Redundancy Reduction. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12310–12320. PMLR, 18–24 Jul 2021.

A Images resolution and data augmentations ablation

For the resolution, we trained both architectures in a supervised fashion, using no data augmentations other than resizing. As seen in Table 2, the resolution ablation study reveals a clear trend where increasing the image size leads to better model performance, as measured by validation F1 scores. However, this improvement comes at the cost of increased training time. Notably, the best trade-off between performance and runtime was achieved at a resolution of 224x224.

Table 2: Best validation F1 scores for different image sizes and architectures.

	64	**128**	**224**	**384**	**512**
ViT-Tiny	0.866	0.885	0.894	0.897	0.90
ResNet-18	0.877	0.892	0.901	0.907	0.909
Time/Epoch	28m	31m	37m	52m	69m

For the data augmentations, we explored how changes in color jitter and random cropping impacted model performance. To maintain computational efficiency, we used 25% of the dataset, running experiments with the ResNet-18 architecture. We adjusted the `ColorJitter` parameters using a variable `t_val` that ranged from 0.1 to 0.8. Specifically, brightness, contrast, and saturation were each set to `t_val`, while hue was set to half of that value (hue = `t_val` / 2). Furthermore, the minimum scale for `RandomResizedCrop` was varied from 0.08 to 0.71. As illustrated in Figure 3, the best performance, measured by F1 Score, was obtained when `t_val` was around 0.1 and 0.45 and when `min_scale` was around 0.395 and 0.605. Due to the small performance difference, and looking to introduce stronger augmentations when using SSL methodologies, we decided to use `min_scale` = 0.395 and `t_val` = 0.275 for further ablations.

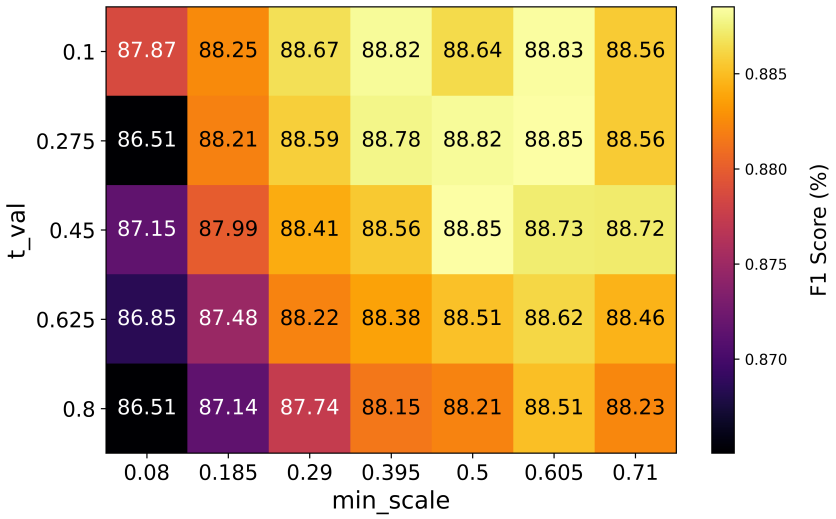


Figure 3: Val F1 Score by `min_scale` and `t_val`. This heatmap shows the performance variation in terms of F1 score, demonstrating the interaction between these two hyperparameters.

B SSL configurations

Training was conducted using PyTorch 2.0.2 [17] on a SLURM cluster equipped with NVIDIA L4 GPUs. In total, we performed 250 experiments, comprising 74 training runs and 184 validation tests. The experiments were conducted over 45 epochs varying the optimizer and its corresponding parameters depending on the method at hand. A linear warmup was applied for the first 10 epochs, followed by a cosine scheduler with no restarts. The base and final decay rates (τ) were 0.996 and 0.999, respectively, with a minimum learning rate of 1×10^{-6}.

During pre-training, a linear probe was trained, and the reported results reflect its final performance. For all experiments, the linear probe was trained using a learning rate of 0.001, with a momentum of 0.9 and no weight decay applied. However, for MAE, a reduced learning rate of 0.0001 was necessary to prevent numerical issues caused by 16-bit mixed precision. Similarly, DINO was trained using 32-bit precision due to instability on imbalanced cases.

Moreover, due to the imbalanced nature of the problem, we use a binary cross entropy loss with positive weights. The positive weights are assigned using the inverse of the defect proportion, a method commonly referred to as inverse class frequency weighting. Finally, every SSL method was trained on two augmented views of the same image and no multicrop was used.

To ensure reproducibility, the remainder of the section contains the specific configurations adopted for each method.

B.1 BYOL

```
optimization:
  optimizer: lars
  batch_size: 256
  lr: 0.2
  momentum: 0.9
  weight_decay: 1.5e-06
  exclude_bias_and_norm: true
  warmup_start_lr: 0.1
ssl_settings:
  proj_hidden_dim: 2048
  pred_hidden_dim: 2048
  proj_output_dim: 256
  normalize_projector: false
```

B.2 DINO

```
optimization:
  optimizer: adamw
  batch_size: 128
  lr: 0.0005
  momentum: 0.9
  weight_decay: 0.0001
  warmup_start_lr: 3.0e-05
ssl_settings:
  proj_hidden_dim: 2048
  proj_output_dim: 256
  num_prototypes: 8192
  clip_grad: true
  freeze_prototyper: 1
  use_bn_in_head: false
  norm_prototyper: false
  student_temperature: 0.1
  warmup_teacher_temperature: 0.04
  teacher_temperature: 0.07
  warmup_temperature_epochs: 15
```

B.3 MAE

```
optimization:
  optimizer: adamw
  batch_size: 256
  lr: 0.001
  momentum: 0.9
  weight_decay: 0.05
  warmup_start_lr: 0.0005
ssl_settings:
```

```
  mask_ratio: 0.75
  norm_pix_loss: true
  decoder_embed_dim: 192
  decoder_depth: 12
  decoder_num_heads: 12
```

B.4 SimCLR

```
optimization:
  optimizer: lars
  batch_size: 256
  lr: 0.3
  momentum: 0.9
  weight_decay: 1.0e-06
  warmup_start_lr: 0.15
ssl_settings:
  proj_hidden_dim: 2048
  proj_output_dim: 256
  temperature: 0.2
```

B.5 Barlow Twins

```
optimization:
  optimizer: lars
  batch_size: 256
  lr: 0.2
  momentum: 0.9
  weight_decay: 1.5e-06
  warmup_start_lr: 0.1
ssl_settings:
  proj_hidden_dim: 2048
  proj_output_dim: 256
  lambda: 0.0051
  scale_loss: 0.024
```

B.6 Supervised

```
optimization:
  optimizer: adamw
  batch_size: 256
  lr: 0.0005
  momentum: 0.9
  weight_decay: 0.0001
  warmup_start_lr: 3.0e-05
```

C Complementary tables

When evaluating SSL methods trained and tested on various class imbalances, several important observations emerged (refer to Figure 4). DINO, when trained on low defect proportions (1% and 2%), struggled significantly in identifying defects, despite performing well at detecting non-defects, suggesting a limitation in handling extreme class imbalances. Additionally, the supervised baseline consistently outperformed all SSL methods across every class imbalance, with the performance gap widening as the class imbalance in the validation set increased. This highlights the need for further exploration of SSL methods in real-world anomaly detection scenarios. Interestingly, Barlow Twins demonstrated the strongest resilience in highly imbalanced settings, particularly when both training and evaluation involved significant class imbalances. Specifically, Barlow Twins variants trained on 1% and 2% defect proportions achieved the best performance among SSL methods under these conditions, underscoring its robustness in scenarios with severe class imbalances.

When evaluating precision across varying validation imbalances, a degradation trend is evident as class imbalance increases (refer to Figure 5). Notably, BYOL, when trained on a nearly balanced setting (45% defect proportion), shows a larger performance gap compared to its 15% counterpart, indicating the potential influence of training balance. Meanwhile, the recall score remains consistently strong across all validation imbalance proportions, likely due to the weighted loss choice, demonstrating solid recall performance even under significant class imbalances.

C.1 Imbalanced validation setting

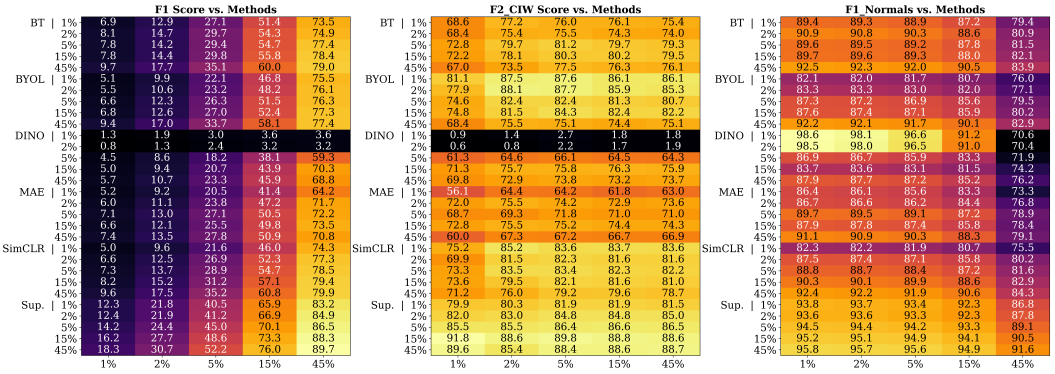


Figure 4: ResNet-18's F1, F2$_{CIW}$, and F1$_{Normal}$ validation score heatmaps across imbalance levels. The x-axis represents the imbalance levels in the validation set, while the y-axis indicates the method and the imbalance level used during training.

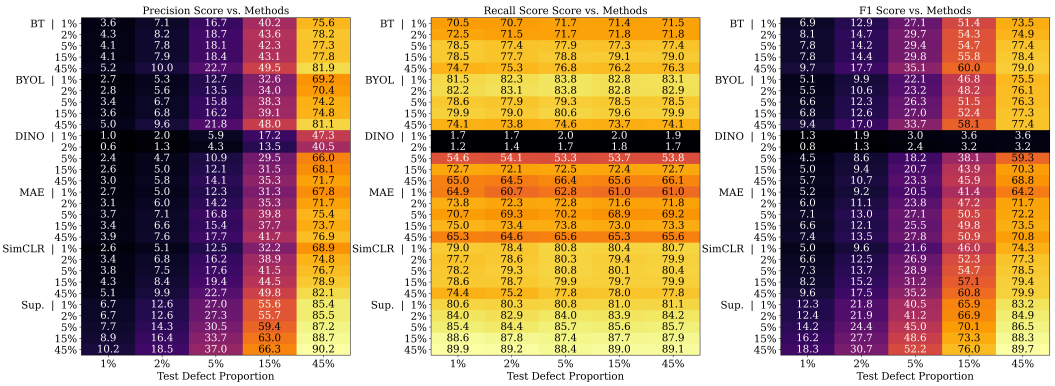


Figure 5: ResNet-18's precision, recall, and f1 validation score heatmaps across imbalance levels. The x-axis represents the imbalance levels in the validation set, while the y-axis indicates the method and the imbalance level used during training.

C.2 Balanced validation setting

Overall, the training performance across the models show a clear linear degradation as the proportion of defects in the dataset decreases. This decline is primarily reflected in the precision scores, which drop significantly at lower defect proportions, such as 1% and 2%. The decreasing precision indicates that as fewer defects are present in the training set, the models fail to generalize properly, as the limited number of defect examples hinders their ability to accurately distinguish between defective and non-defective instances, leading to an increase in false positives. However, this trend is less evident in the validation set, where the precision remains relatively stable. This is due to the balanced nature of the validation set. On the other hand, recall remains fairly consistent across both the training and validation sets, even as the defect proportion decreases. This behavior can be explained by the use of a weighted loss function, which adjusts weights based on defect proportion. As a result, the model becomes biased towards positive predictions, preserving recall but impacting precision.

Table 3: Comprehensive performance metrics for ResNet-18 across both training and validation phases. The percentages represent the imbalance levels applied during training, while the validation default distribution remained unchanged.

		Train					Val				
		1%	**2%**	**5%**	**15%**	**45%**	**1%**	**2%**	**5%**	**15%**	**45%**
Precision↑	*Supervised*	0.055	0.114	0.282	0.614	0.918	0.834	0.840	0.861	0.877	0.893
	BYOL	0.027	0.053	0.143	0.372	0.782	0.715	0.715	0.747	0.750	0.801
	Barlow Twins	0.032	0.067	0.167	0.410	0.805	0.749	0.769	0.764	0.772	0.811
	DINO	0.024	0.043	0.107	0.309	0.719	0.571	0.422	0.664	0.693	0.738
	SimCLR	0.028	0.066	0.168	0.433	0.808	0.715	0.762	0.770	0.786	0.817
Recall↑	*Supervised*	0.893	0.895	0.910	0.928	0.927	0.827	0.854	0.869	0.887	0.903
	BYOL	0.805	0.822	0.807	0.811	0.777	0.820	0.829	0.802	0.820	0.784
	Barlow Twins	0.771	0.782	0.791	0.802	0.777	0.787	0.781	0.815	0.821	0.796
	DINO	0.663	0.670	0.669	0.733	0.711	0.026	0.017	0.683	0.747	0.716
	SimCLR	0.748	0.794	0.808	0.808	0.793	0.766	0.795	0.824	0.829	0.812
F1↑	*Supervised*	0.104	0.202	0.431	0.739	0.922	0.831	0.847	0.865	0.882	0.898
	BYOL	0.052	0.099	0.243	0.510	0.779	0.764	0.768	0.774	0.783	0.792
	Barlow Twins	0.061	0.123	0.276	0.543	0.791	0.768	0.775	0.789	0.796	0.803
	DINO	0.046	0.080	0.185	0.434	0.715	0.049	0.032	0.673	0.719	0.727
	SimCLR	0.054	0.123	0.279	0.564	0.800	0.740	0.778	0.796	0.807	0.814
$F2_{CIW}$↑	*Supervised*	0.868	0.877	0.886	0.904	0.900	0.838	0.849	0.880	0.904	0.879
	BYOL	0.836	0.841	0.825	0.831	0.786	0.837	0.844	0.819	0.825	0.785
	Barlow Twins	0.828	0.822	0.828	0.825	0.788	0.817	0.805	0.817	0.822	0.780
	DINO	0.757	0.719	0.733	0.761	0.699	0.566	0.402	0.712	0.777	0.752
	SimCLR	0.792	0.821	0.834	0.830	0.807	0.775	0.813	0.826	0.822	0.813
$F1_{Normal}$↑	*Supervised*	0.916	0.922	0.933	0.939	0.936	0.850	0.861	0.879	0.894	0.908
	BYOL	0.826	0.821	0.849	0.846	0.821	0.758	0.759	0.783	0.789	0.818
	Barlow Twins	0.865	0.873	0.879	0.870	0.834	0.782	0.796	0.799	0.807	0.828
	DINO	0.832	0.812	0.820	0.808	0.769	0.689	0.686	0.700	0.730	0.763
	SimCLR	0.849	0.869	0.878	0.880	0.839	0.752	0.793	0.806	0.818	0.836

Table 4: Comprehensive performance metrics for ViT-Tiny across both training and validation phases. The percentages represent the imbalance levels applied during training, while the validation data proportion remained unchanged.

		Train					Val				
		1%	2%	5%	15%	45%	1%	2%	5%	15%	45%
Precision↑	*Supervised*	0.030	0.065	0.209	0.520	0.879	0.735	0.757	0.816	0.842	0.873
	BYOL	0.024	0.055	0.152	0.356	0.755	0.687	0.723	0.751	0.737	0.770
	Barlow Twins	0.024	0.058	0.136	0.375	0.780	0.681	0.729	0.733	0.753	0.792
	DINO	0.018	0.041	0.052	0.307	0.714	0.625	0.651	0.474	0.692	0.734
	MAE	0.010	0.022	0.063	0.154	0.223	0.469	0.503	0.530	0.479	0.232
	SimCLR	0.033	0.063	0.155	0.373	0.776	0.752	0.746	0.753	0.747	0.788
Recall↑	*Supervised*	0.807	0.808	0.855	0.880	0.890	0.804	0.816	0.847	0.875	0.882
	BYOL	0.657	0.746	0.749	0.770	0.690	0.666	0.753	0.752	0.777	0.694
	Barlow Twins	0.677	0.731	0.754	0.779	0.744	0.682	0.748	0.762	0.787	0.752
	DINO	0.679	0.710	0.674	0.715	0.695	0.773	0.719	0.662	0.717	0.703
	MAE	0.496	0.500	0.588	0.705	0.013	0.567	0.864	0.617	0.719	0.013
	SimCLR	0.761	0.783	0.788	0.805	0.776	0.772	0.792	0.790	0.812	0.784
F1↑	*Supervised*	0.059	0.120	0.335	0.654	0.884	0.768	0.785	0.831	0.858	0.878
	BYOL	0.046	0.102	0.252	0.487	0.721	0.676	0.738	0.752	0.757	0.730
	Barlow Twins	0.046	0.107	0.231	0.506	0.762	0.682	0.738	0.747	0.769	0.772
	DINO	0.035	0.077	0.096	0.430	0.704	0.691	0.683	0.553	0.704	0.719
	MAE	0.020	0.043	0.114	0.253	0.025	0.513	0.636	0.570	0.575	0.024
	SimCLR	0.062	0.116	0.259	0.509	0.776	0.762	0.768	0.771	0.778	0.786
$F2_{CIW}$↑	*Supervised*	0.824	0.825	0.848	0.869	0.868	0.837	0.812	0.831	0.870	0.881
	BYOL	0.711	0.773	0.776	0.789	0.726	0.680	0.768	0.760	0.781	0.725
	Barlow Twins	0.708	0.759	0.787	0.804	0.760	0.704	0.749	0.786	0.784	0.758
	DINO	0.722	0.744	0.715	0.729	0.680	0.733	0.736	0.690	0.724	0.695
	MAE	0.624	0.394	0.711	0.683	0.014	0.629	0.640	0.704	0.702	0.009
	SimCLR	0.792	0.809	0.807	0.815	0.785	0.766	0.795	0.798	0.801	0.788
$F1_{Normal}$↑	*Supervised*	0.849	0.862	0.903	0.912	0.904	0.774	0.794	0.844	0.869	0.890
	BYOL	0.839	0.847	0.869	0.840	0.789	0.719	0.757	0.778	0.771	0.781
	Barlow Twins	0.834	0.858	0.850	0.851	0.813	0.716	0.760	0.766	0.785	0.806
	DINO	0.785	0.791	0.507	0.810	0.764	0.655	0.688	0.419	0.727	0.759
	MAE	0.615	0.709	0.692	0.461	0.695	0.471	0.350	0.553	0.389	0.677
	SimCLR	0.869	0.862	0.867	0.847	0.816	0.782	0.781	0.786	0.785	0.810

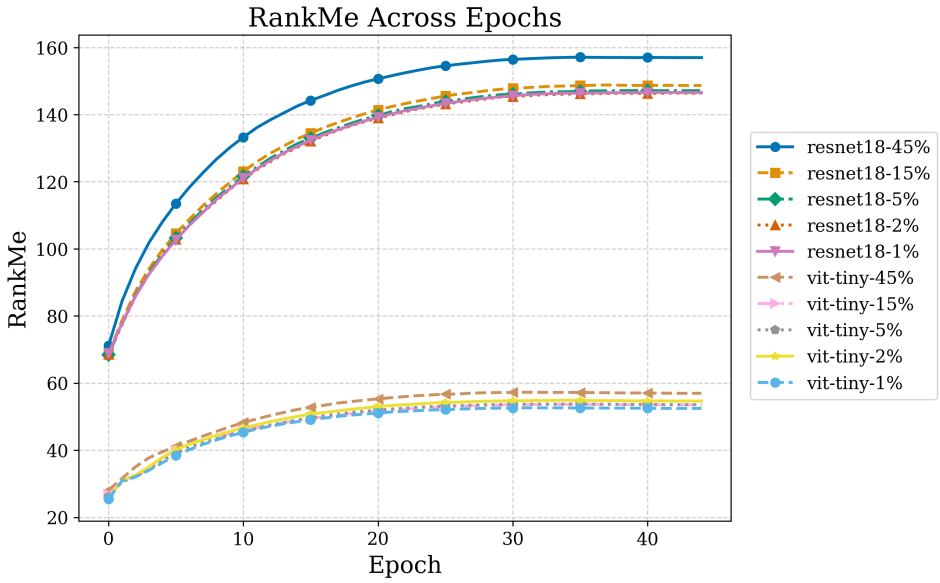
D Self-supervised monitoring metrics

When evaluating SimCLR, BYOL, and DINO, significant differences emerge in the behavior of ResNet-18 versus ViT-Tiny architectures. ResNet-18 consistently achieved higher RankMe values and demonstrated steady improvement in feature standard deviation, indicating more stable and robust feature representations. In contrast, ViT-Tiny models showed strong initial performance that often degraded over time, particularly under SimCLR and BYOL, suggesting challenges in maintaining feature consistency (see Figures 6, 7). However, under DINO, ViT-Tiny's performance was more stable, aligning closely with ResNet-18's in terms of RankMe values (see Figure 8).

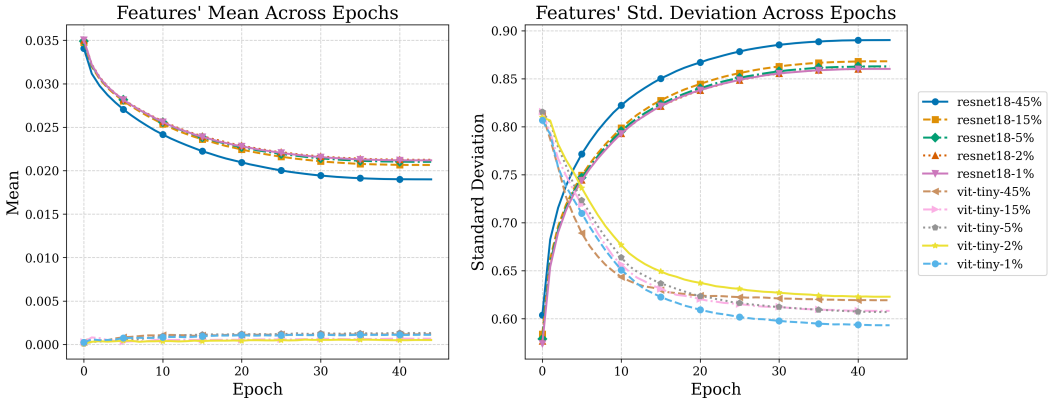
Overall, these findings suggest that ResNet architectures offer superior stability and consistency across different self-supervised learning algorithms, whereas ViT-Tiny models may require tailored optimization techniques to sustain their initial performance levels. This underscores the importance of continuously monitoring self-supervised metrics to ensure robust feature learning and representation stability.

Furthermore, scatter plots comparing RankMe with train and validation F1 scores—grouped by SSL method, model type, and defect proportion—show no correlation (Figures 9, 10, indicating that RankMe is not a reliable predictor of downstream performance. This challenges RankMe's intended role and highlights a gap in its effectiveness within the SSL domain. Nonetheless, distinct RankMe differences between ResNet-18 and ViT-Tiny architectures persist, reinforcing the architectural distinctions previously observed.

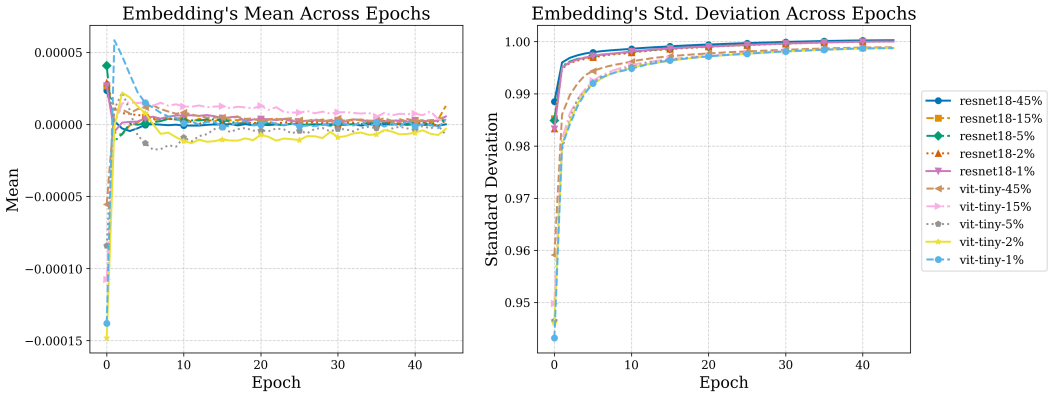
D.1 SimCLR



(a) RankMe metric across epochs.



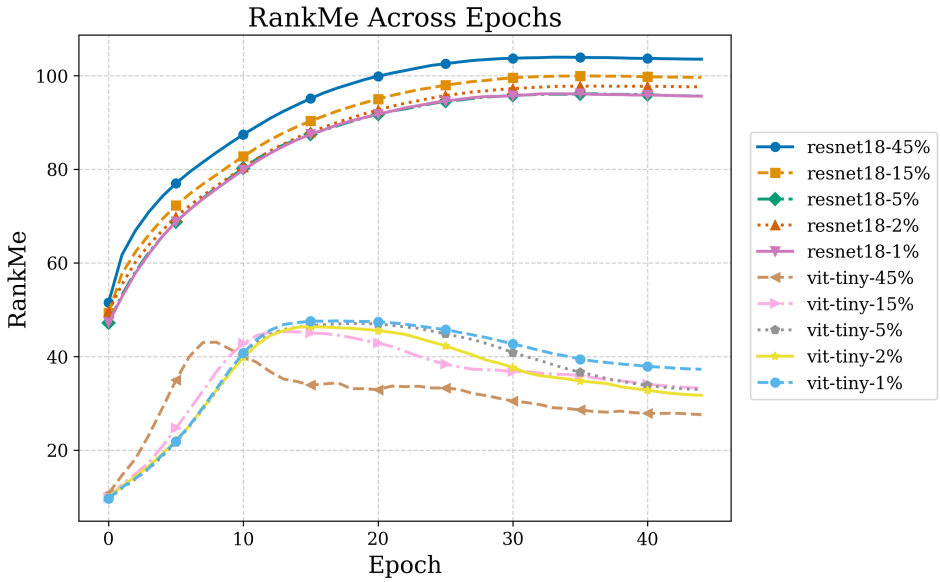
(b) Features' mean and standard deviation across epochs.



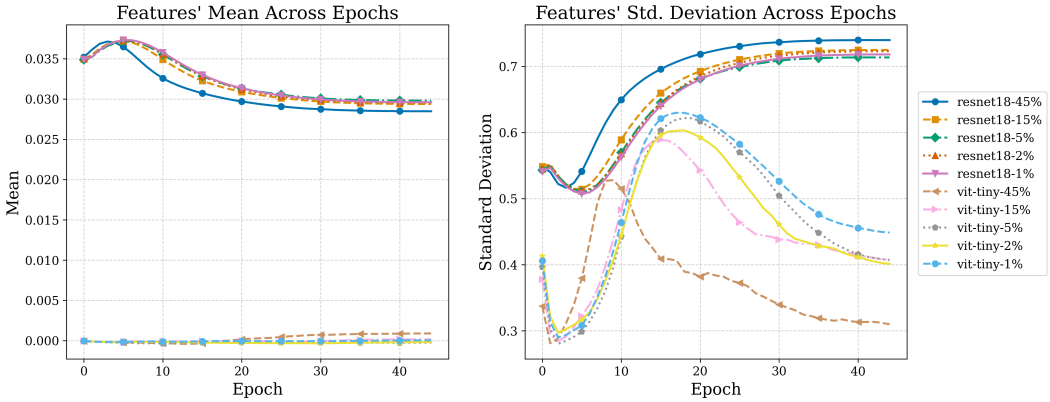
(c) Embedding's mean and standard deviation across epochs.

Figure 6: Self-supervised monitoring metrics for SimCLR.

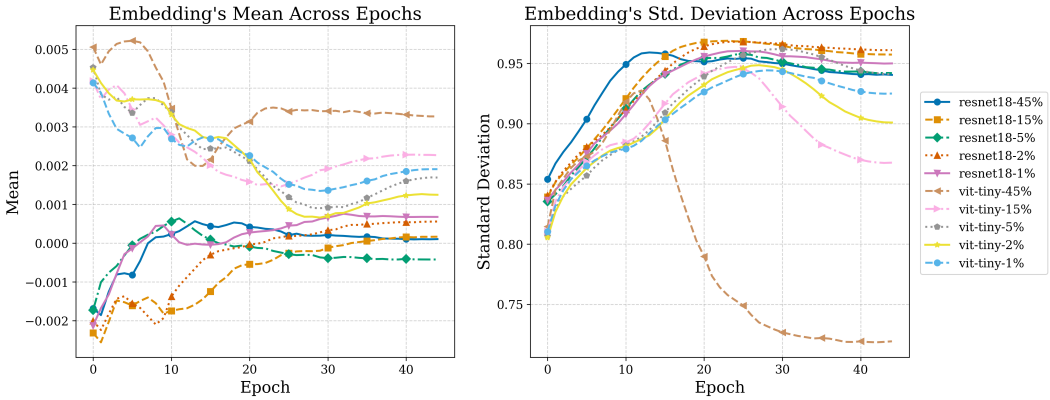
D.2 BYOL



(a) RankMe metric across epochs.



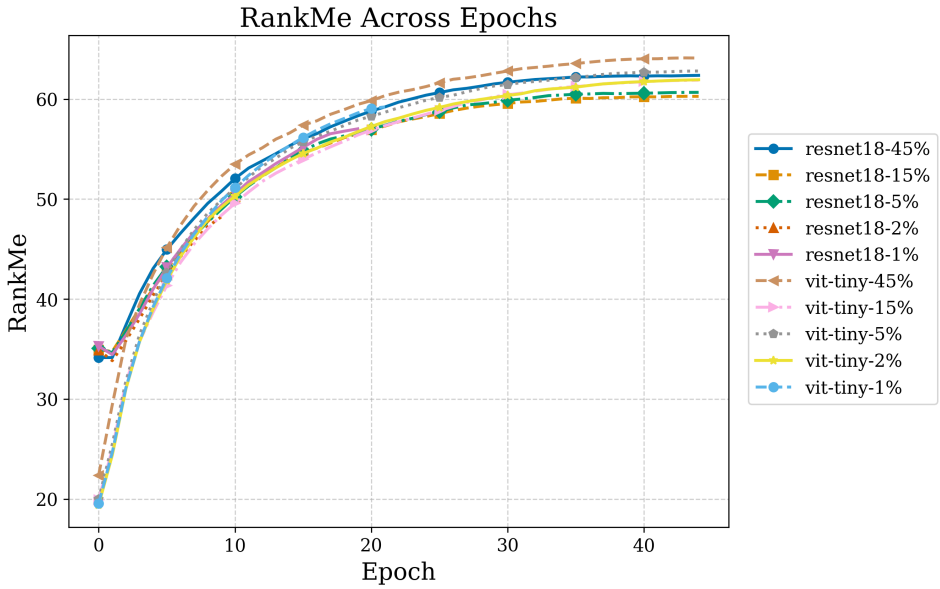
(b) Features' mean and standard deviation across epochs.



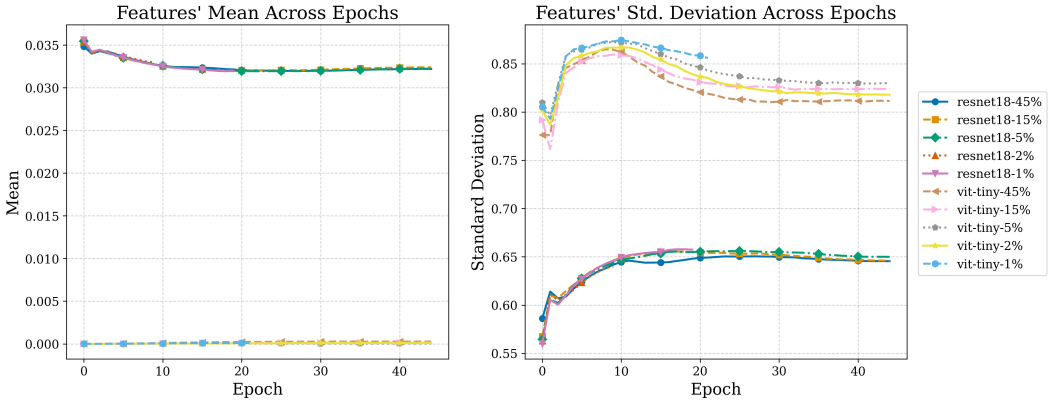
(c) Embedding's mean and standard deviation across epochs.

Figure 7: Self-supervised monitoring metrics for BYOL.

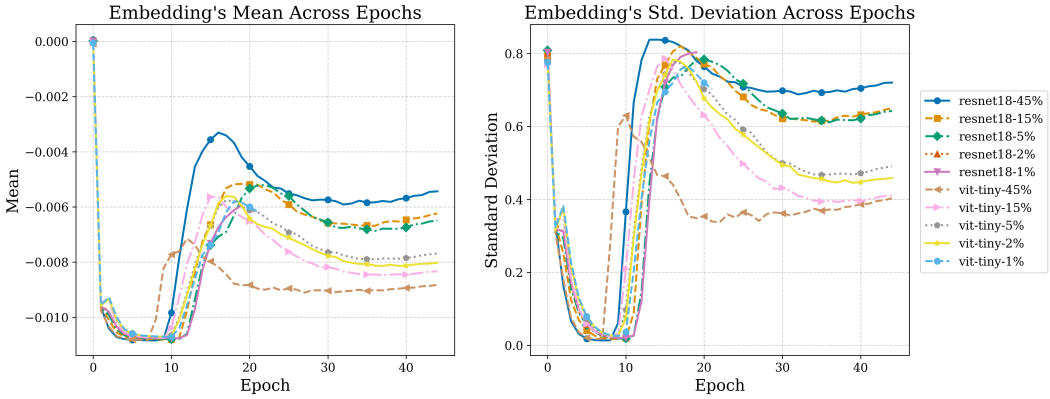
D.3 DINO



(a) RankMe metric across epochs.



(b) Features' mean and standard deviation across epochs.



(c) Embedding's mean and standard deviation across epochs.

Figure 8: Self-supervised monitoring metrics for DINO.

D.4 RankMe scatter plots

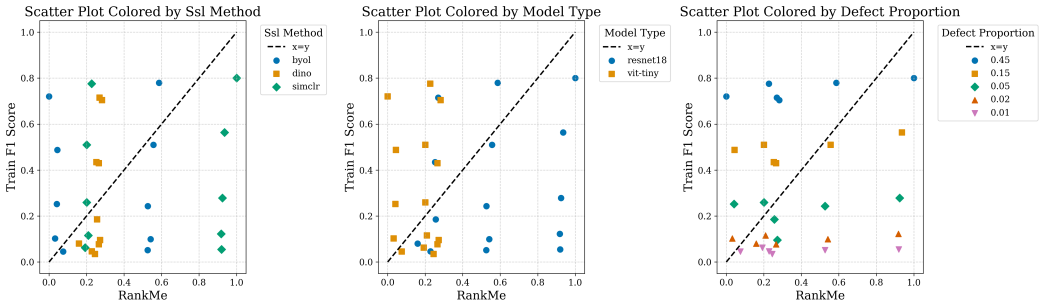


Figure 9: Train F1 scores plotted against re-scaled RankMe metrics. RankMe metrics were re-scaled to enable easier comparisons with F1 scores.

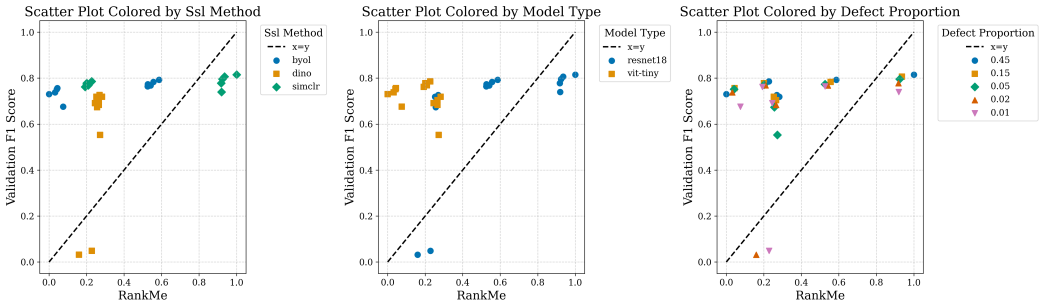


Figure 10: Validation F1 scores plotted against re-scaled RankMe metrics. RankMe metrics were re-scaled to enable easier comparisons with F1 scores.

E Sewer-ML samples

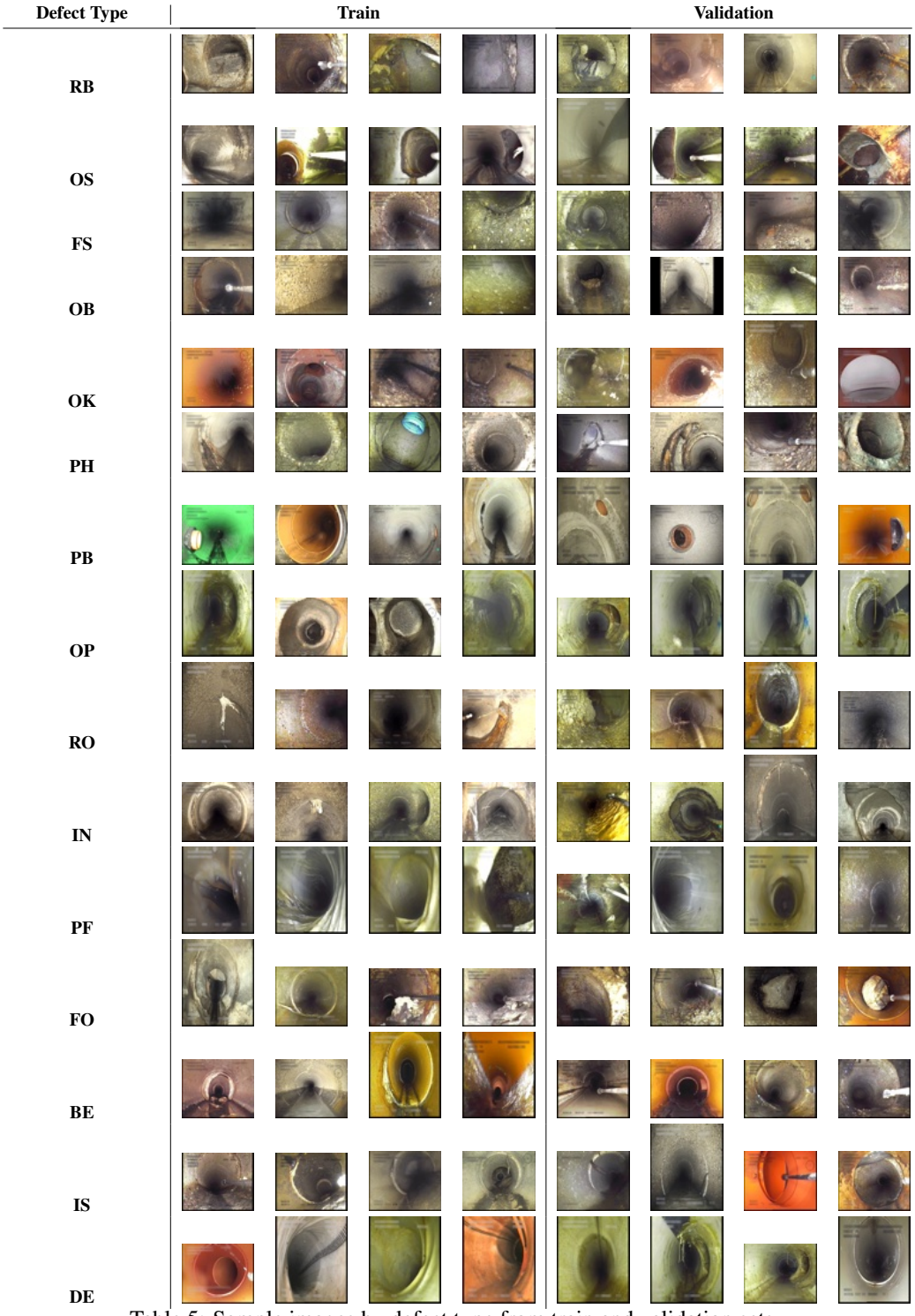


Table 5: Sample images by defect type from train and validation sets.