AOI-SSL: Self-Supervised Framework for Efficient Segmentation of Wire-bonded Semiconductors In Optical Inspection

Supplementary Material

A. Additional pre-training implementation details

DINO For the DINO implementation, we attach to our encoders a 3-layer MLP DINO projection head [7] with a bottleneck dimension of 2048 and an output embedding dimension of 256. To maintain stability and prevent representation collapse, we employ a teacher EMA decay rate τ of 0.998 and a centering momentum λ of 0.9. The student temperature is fixed at $T_s = 0.1$, while the teacher temperature T_t follows a linear warmup from 0.04 to 0.07 during the first 30 epochs to encourage diverse feature extraction. These values are based in DINO's original implementation [7].

An additional element of DINO is its multi-crop strategy, where crops of different sizes of the same images are used to teach scale invariance to the student encoder. Our multi-crop strategy consists of 2 global crops (226×226 crops covering an area greater than 50% of the original image) and 6 local crops (96×96 crops covering an area less than 30% of the original image), with a custom "black-patch" filtering logic that re-samples crops if more than 95% of the pixels are zero-valued, ensuring the model learns from actual structural content rather than dataset artifacts. Given that the color jittering augmentation used in DINO to distinguish teacher and student crops is not applicable to our monochrome images, we use the same augmentations described in Sec. 4.2 in both teacher and student views, with the sole difference between them being that the teacher augmentation function V_t returns only global crops, while the student function V_s returns both global and local crops (the optimal number of local crops was determined by an ablation as demonstrated in Tab. 5). Moreover, note that we did not pre-train FasterViT with DINO, as it lacks a [CLS] token, and we deemed the training signal obtained only from average pooling of the token embeddings to not be adequate.

iBOT The iBOT training protocol extends the DINO training methodology by incorporating a Masked Image Modeling (MIM) loss applied to patch-level tokens, obtained by masking a specific ratio of input tokens to the student. The specific masking strategy used is blockwise masking [3] with a ratio of 0.3. Following the original iBOT [40], we compute this loss only on the two sets of global teachers and students' crops produced from an input image I: $\{x_1^{g_t}, x_2^{g_t}\}$ and $\{x_1^{g_s}, x_2^{g_s}\}$, respectively. In Eq. (10) we can see the specific formulation of the loss, where $A_g(I)$ is the collection of pairs of global crops for input image I, u_i is the teacher embedding of patch i of the input global crop x, $\hat{u}_i$ is the student embedding of the input global crop $\hat{x}$, L is the number of tokens of input crops x and $\hat{x}$ after being patchified, and P_θ^{patch} refers to a shared DINO head that projects the embeddings into a common K-dimensional probability space. Moreover, m_i is a scalar value indicating whether patch i is masked ($m_i = 1$) or not ($m_i = 0$) to ensure that unmasked tokens do not contribute to the loss.

$$A^g(I) = \{(x, \hat{x}) \mid x \in \{x_1^{g_t}, x_2^{g_t}\}, \hat{x} \in \{x_1^{g_s}, x_2^{g_s}\}\} \quad (8)$$

$$\mathcal{L}_{\text{MIM}}(x, \hat{x}) = \sum_{i=1}^{L} m_i \cdot P_\theta^{\text{patch}}(u_i)^{\text{T}} \log P_\theta^{\text{patch}}(\hat{u}_i) \quad (9)$$

$$\mathcal{L}_{\text{iBOT}}(I) = \mathcal{L}_{\text{DINO}}(I) + \frac{1}{2} \sum_{x, \hat{x} \in A^g(I)} \mathcal{L}_{\text{MIM}}(x, \hat{x}) \quad (10)$$

The aforementioned shared DINO head is also used to project the image level representations for the DINO loss, in alignment with the original iBOT implementation [40]. For the FasterViT-0 backbone, which lacks a native [CLS] token, the DINO loss is calculated on a global average pooled representation of its last feature map. To prevent early-stage divergence in FasterViT, the implementation of this method utilizes a 25-epoch "warm-up" phase where the MIM loss is removed, allowing the image-level [CLS] representation to stabilize through the DINO objective before introducing patch-level distillation. Both encoder architectures were trained for 3000 epochs using the AdamW optimizer with a base learning rate of 1.5×10^{-4}, a weight decay of 0.05, and a batch size of 300 on an A100 GPU.

MAE An additional detail regarding our MAE implementation is the protocol used to select the masking ratio. We ablate this hyperparameter to determine the best pre-training configuration in Tab. 6. The results demonstrate that high masking ratios (0.7) provide the strongest self-supervised signal for the AOI dataset, while extreme ratios (0.95) lead to a significant drop in representational quality.

Table 5. Impact of number of local crops on DINO pre-training, evaluated using image-level retrieval (mIoU) on a ViT-Tiny encoder pre-trained on the AOI dataset. Evaluations adopt $k = 3$ nearest neighbors and 50 pre-training epochs. The best result is highlighted in **bold**.

Number of Local Crops	mIoU (%)
1	18.1
3	24.0
6	**27.4**
9	26.4
12	25.5

B. Fine-tuning Hyperparameters

Table 7 contains the hyperparameters used to fine-tune the transformer based encoders. We fixed the β parameters of the AdamW optimizer to the values suggested in DINO [7], and the batch size to maximize GPU saturation (NVIDIA RTX 2080). The other parameters were experimentally selected to maximize performance in the validation set.

Table 6. Effect of masking ratio on MAE pre-training, evaluated using image-level retrieval performance (mIoU) with $k = 3$ nearest neighbors (best result in **bold**).

Masking Ratio	mIoU (%)
0.20	16.8
0.40	31.8
0.70	**32.3**
0.95	22.0

Table 7. Fine-tuning hyperparameters for transformer-based encoder–decoder segmentation.

Hyperparameter	ViT-Tiny	FasterViT-0
Learning Rate	1.0×10^{-3}	5.0×10^{-4}
Layer-wise LR Decay	0.75	0.75
AdamW Weight Decay	5.0×10^{-2}	5.0×10^{-2}
Warmup Epochs	5	5
AdamW β_1, β_2	0.9, 0.95	0.9, 0.95
Batch Size	64	32

Table 8. **Ablation on Loss Functions and Background Class.** Impact of various loss formulations and the inclusion of a dedicated background class on segmentation performance. Results were generated using an MAE-pretrained ViT-Tiny encoder paired with a randomly-initialized UPerNet segmentation head. Best results per class are highlighted in **bold**.

Loss Function	Background	mIoU	Epoxy	Wire	Wedge	Ball
Binary Cross Entropy	✗	**53.7**	**66.0**	**43.7**	44.3	**60.8**
	✓	51.3	65.4	40.9	37.4	61.6
Soft Jaccard	✗	41.3	39.4	37.0	**51.6**	37.1
	✓	35.5	37.0	34.5	44.4	26.3
Soft Dice	✗	40.9	41.0	37.7	49.9	35.1
	✓	44.0	50.3	38.0	50.5	37.1

C. Loss Ablations

The results of the different losses tested are included in Tab. 8. We additionally ablate the effect of including the background class. The background-less BCE loss consistently outperforms all other losses except in the wedge class. We hypothesize that this dominance of BCE is explained by the additional complexity introduced by the multi-label DICE and Jaccard's optimization objective, which our low-parameter encoders and limited computational budget struggle to solve efficiently. Moreover, we believe that the smaller size and finer details of wedge bonds explain the superior performance of the DICE and Jaccard losses on this class, as they focus on precise mask-prediction overlap rather than simple localization.

D. Retrieval Failure Modes

While the proposed retrieval strategies achieve competitive performance without labeled parametric fine-tuning and with significantly higher throughput, they do not yet match the accuracy of the best fine-tuned fully parametric models in the generalized settings of Tab. 2. These performance gaps are particularly pronounced in the *epoxy* and *wire* classes.

Figure 6 illustrates the primary failure modes underlying these limitations. As shown in the first row, a prominent issue is the lack of spatial continuity on thin, elongated structures. Unlike convolutional decoders, which maintain better local connectivity, our

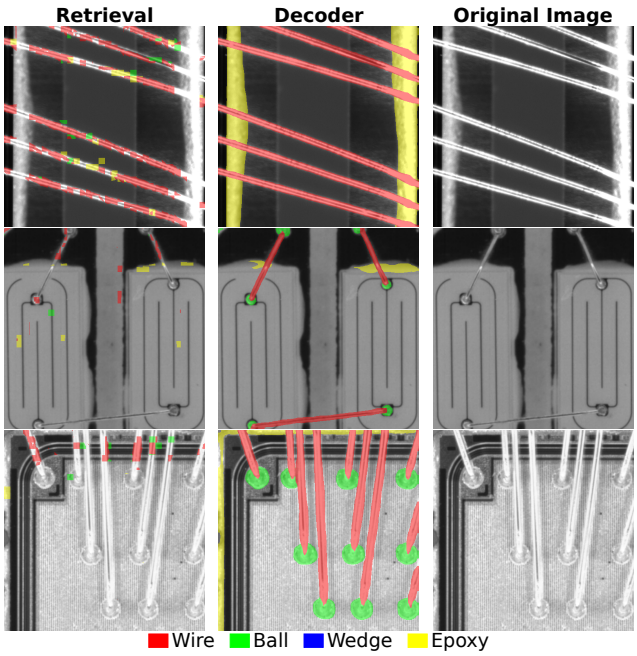

Figure 6. **Retrieval Failure Cases**. Problematic cases where the patch-level retrieval strategy fails to produce high quality segmentation masks compared to the parametric decoders. Exactly as in Tab. 2, the retrieval strategy used is patch-level retrieval with features produced by an MAE pre-trained ViT-Tiny, while the decoder results employ an MAE pre-trained FasterViT-0 encoder attached to an UPerNet decoder.

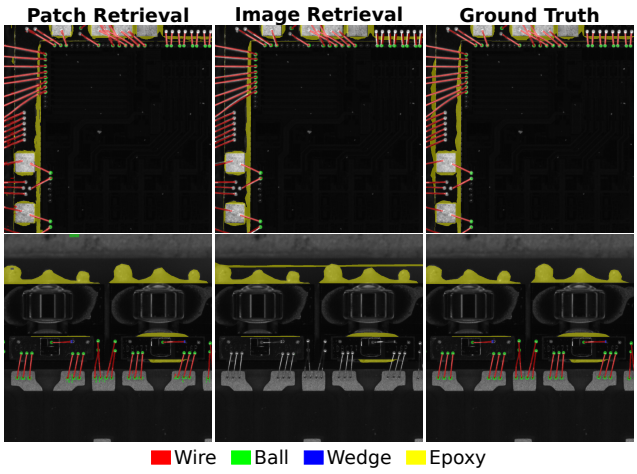

Figure 7. **Visual Comparison of Retrieval Strategies.** Patch-level retrieval (left) showcases superior spatial alignment and recall compared to image-level baselines (center), accurately capturing component boundaries despite layout variations. These results were generated using an MAE pre-trained ViT encoder as the retrieval backbone.

Table 9. **Comprehensive Method Comparison.** Comparison of our fine-tuned and retrieval methods (MAE, iBOT, and DINO pre-trained) against supervised, ImageNet pre-trained baselines. *AOI Dataset* refers to our custom AOI pre-training dataset. The throughput was measured on an NVIDIA RTX 2080 GPU. The best results are in **bold**, second best are underlined.

Model	Pre-training	Crops/s	mIoU	Epoxy	Wire	Wedge	Ball
Supervised Baselines							
MobileNetV3 + DeepLab		**309.8**	29.9	60.6	19.3	2.8	36.8
ResNet18 + DeepLab	ImageNet	87.3	43.5	66.7	50.3	10.3	46.7
ResNet18 + U-Net++		86.9	52.4	<u>75.4</u>	59.1	9.2	65.8
Ours (Frozen LVD142M DINOv2)							
ViT-S Patch Retrieval	DINOv2	90.9	47.8	43.4	48.7	**45.8**	53.4
Ours (AOI Dataset)							
FasterViT-0 + UPerNet	**MAE**	163.3	**60.3**	**79.3**	**66.7**	19.3	**75.8**
FasterViT-0 + UPerNet	iBOT		<u>53.7</u>	67.5	<u>60.0</u>	19.7	67.7
ViT-Tiny + UPerNet	MAE		53.5	73.7	43.0	26.5	<u>70.7</u>
ViT-Tiny + UPerNet	iBOT	218.1	41.6	71.8	39.3	5.9	49.5
ViT-Tiny + UPerNet	DINO		40.0	68.7	31.6	1.3	54.2
ViT-T + Patch Retrieval	MAE		48.1	38.4	47.8	<u>44.4</u>	61.8
ViT-T + Patch Retrieval	iBOT	<u>266.7</u>	45.9	46.2	43.7	41.0	52.8
ViT-T + Patch Retrieval	DINO		41.5	41.5	41.1	36.6	46.9

retrieval approach suffers from fragmentation in these regions. We attribute this to the non-overlapping patchification of the ViT-Tiny backbone: when a narrow object does not align perfectly with the patchification grid, individual patches often become dominated by background noise or adjacent classes, diluting the feature representation. Adopting smaller patch sizes or overlapping patches may mitigate this discretization artifact.

Furthermore, the retrieval mechanism exhibits sensitivity to low-contrast images, as demonstrated in the final two rows of Fig. 6. Since the majority of training samples share a consistent contrast profile, extreme outliers reduce the relative similarity and attention scores between the memory bank and the query images. This shift effectively invalidates the class thresholds established during training, resulting in sparse or entirely empty classification masks. To enhance robustness, future work could incorporate contrast-based data augmentations during memory bank construction, ensuring that the reference distribution better accounts for such environmental variations.

A final notable failure mode of our retrieval strategies is their performance within the *epoxy* class in the generalized setting of the main method comparison. As seen in Tab. 9, although most fine-tuned models achieve robust results for this category, retrieval-based strategies exhibit performance degradation. We hypothesize that this stems from the structural characteristics of epoxy, specifically its relatively large size and high geometric irregularity. Effectively capturing such morphological variance likely requires a significantly larger patch memory bank to achieve sufficient representational coverage of the diverse boundary contours. Consequently, this limitation could also benefit from the introduction of augmentations in the construction of the memory bank.

E. Extended Results

Detailed results for the exhaustive suite of pre-training and adaptation strategies are provided in Tab. 9. This includes an additional MobileNetV3 [21] baseline, incorporated to benchmark the throughput–accuracy tradeoff of our proposed retrieval techniques against a faster lightweight architecture. Although MobileNetV3 achieves moderately higher throughput, its mIoU is nearly 50% lower than that of the best performing retrieval strategy, further solidifying the adequacy of retrieval in AOI. These expanded metrics also underscore the significant performance difference between the pre-training regimes mentioned in Sec. 4.2.

Finally, our expanded qualitative results (Fig. 7) help to elucidate the significant performance gap between image and patch-level retrieval. While image-level retrieval identifies similar device types successfully, it fails to account for sub-pixel positional shifts. In contrast, patch-level retrieval aligns predictions with local precision, leading to significantly greater overlap with ground truth, particularly visible in the spatial alignment of the wires. Moreover, it also displays higher recall, as it is able to identify individual object patches at fine-grained resolutions without requiring full image-level similarity.