SIGDINO: ADAPTING DINOV3 WITH LOCAL STRUCTURAL MATCHING FOR WRITER-INDEPENDENT OFFLINE SIGNATURE VERIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Offline handwritten signature verification aims to determine the authenticity of a signature from static signature images alone. Existing methods often rely on large amounts of training data to learn discriminative signature representations. However, when only limited training data are available, their performance tends to become unstable and their generalization ability often degrades in cross-language evaluation. To address these issues, we propose SigDINO, a writer-independent offline signature verification method that adapts DINOv3, a large scale self-supervised vision model with transferable visual representations, to the signature domain. Specifically, we perform self-supervised adaptation of the pretrained model using synthetic signature images, allowing it to better capture the sparse stroke distribution of offline signatures. Intermediate stroke responsive patch tokens from the adapted transformer serve as the main local representation, while cross layer residual information from deeper layers further enhances them. These representations are then compared by the local structural matching module to compute the verification distance between signature pairs. Extensive experiments on four benchmark datasets show that SigDINO achieves state-of-the-art or competitive performance, while maintaining stable verification under limited writer settings and strong generalization in cross-language evaluation.

1 INTRODUCTION

Offline Signature Verification (OSV) aims to determine whether a questioned signature is genuine or forged using only static signature images. It is an important biometric verification task in document authentication, financial services, and identity verification (Hafemann et al., 2017b). Unlike online signature verification, offline verification cannot use dynamic cues such as pen trajectory, writing speed, or pressure (Zhao et al., 2025). Therefore, offline signature verification methods need to capture discriminative cues from stroke morphology, spatial layout, and local structural differences in signature images. The main challenge arises from skilled forgeries, which can closely imitate the overall shape, writing proportions, and spatial layout of genuine signatures, while truly discriminative differences are often hidden in subtle local structures, such as stroke trajectories, stroke junctions, stroke breaks, and subtle deformations (Zheng et al., 2025b).

Although recent OSV methods, including DetailSemNet (Shih et al., 2024), HTCSigNet (Zheng et al., 2025a), and the method of Zheng et al. (Zheng et al., 2025b), perform well when sufficient training data are available, their learned representations can become unstable under limited training data or test distribution shifts. This instability is closely related to the nature of offline signatures, where discriminative information is often concentrated in subtle local structures, while collecting genuine signatures and skilled forgeries from many writers is costly. As a result, supervised training on a limited set of writers can be sensitive to data scarcity and domain shift. Recent self-supervised OSV methods, including SWIS (Manna et al., 2022), SURDS (Chattopadhyay et al., 2022), and BYOLSiam (Udo et al., 2024), have shown that unlabeled signature images can provide useful pretraining signals for offline signature verification. Meanwhile, DINOv3 shows that large scale self-supervised vision models can learn transferable representations and generalize across a broad range of visual settings without task-specific fine tuning (Siméoni et al., 2025). These properties make DINOv3 a strong candidate backbone for offline signature verification, especially under limited-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

writer settings. However, offline signatures are sparse and highly structured, and it remains unclear how such general visual representations should be adapted to the signature domain.

To address these issues, we propose SigDINO, a writer-independent offline signature verification method based on DINOv3 (Siméoni et al., 2025). We first adapt DINOv3 to the signature domain through self-supervised adaptation using synthetic signature images, so that it better captures the sparse stroke distribution of signature images. Then, we extract patch features from different transformer layers of the adapted model and build a local structural matching module to compare effective stroke regions between signature pairs. Specifically, stroke responsive patch tokens from an intermediate layer are used as the main local representation, while their difference from deeper patch features forms a cross layer residual that further enhances this representation. The final signature authenticity is determined by the local structural matching distance. The main contributions of this work are as follows:

1. We propose a novel writer-independent offline signature verification framework, SigDINO, which improves verification stability under limited writer settings and generalization in cross-language evaluation.
2. A novel signature domain self-supervised adaptation strategy and a local structural matching module are proposed. The framework adapts DINOv3 to the signature domain using synthetic signature images and performs local structural matching between signature pairs through hierarchical patch features and cross layer residual information.
3. Comprehensive experiments on four public benchmark datasets demonstrate that the proposed framework achieves the lowest EER in benchmark evaluation and competitive performance under limited writer settings and cross-language evaluation.

2 Related Work

Early offline signature verification relied on handcrafted descriptors, including shape and texture descriptors (Hafemann et al., 2017b), HOG based features (Yilmaz et al., 2011), surroundedness and compact structural features (Kumar et al., 2012; Dutta et al., 2016), and local keypoint features (Thakare & Deshmukh, 2018). These methods depend heavily on manually designed feature representations, which limits their generalization ability. With the rise of deep learning, offline signature verification shifted toward deep representation learning (Hafemann et al., 2017a). Dey et al. (2017) formulated OSV as a pairwise similarity learning problem and used a Siamese CNN to learn discriminative signature representations. Parcham et al. (2021) proposed CBCapsNet, which combines CNNs with capsule networks to preserve spatial relationships in signature representations. However, these methods can still degrade when only a small number of writers are available for training or when distribution shifts exist between the training and test datasets.

Transformer based methods improved OSV by capturing signature structure and forgery cues. Ren et al. (2023b) proposed 2C2S, a two channel and two stream framework based on Swin Transformer that learns multiscale features from the original and center cropped streams. Li et al. (2024) proposed TransOSV to model global structure and local forgery differences. Zheng et al. (2025a) proposed HTCSigNet to combine Transformer based global modeling with CNN based local feature extraction. Shih et al. (2024) proposed DetailSemNet to emphasize local structural matching and detail enhancement for skilled forgery verification. These studies suggest that OSV benefits from stronger image representations and explicit local structural correspondences between signature pairs.

Self-supervised vision foundation models learn transferable visual representations from large scale unlabeled data. SimCLR uses contrastive learning (Chen et al., 2020), as does MoCo (He et al., 2020), whereas BEiT (Bao et al., 2021) and MAE (He et al., 2022) use masked image modeling. The DINO family learns transferable representations through self distillation (Caron et al., 2021; Oquab et al., 2023; Siméoni et al., 2025). Manna et al. (2022) proposed SWIS based on cross covariance decorrelation. Chattopadhyay et al. (2022) proposed SURDS with self-supervised attention guided reconstruction and a dual triplet loss. Udo et al. (2024) proposed BYOLSiam by combining BYOL style self-supervised pretraining with a Siamese verification framework. These works suggest that self-supervised learning can provide useful pretraining signals for offline signature verification. Building on this direction, we introduce SigDINO, which adapts DINOv3 to the signature domain to learn more transferable representations for signature images.

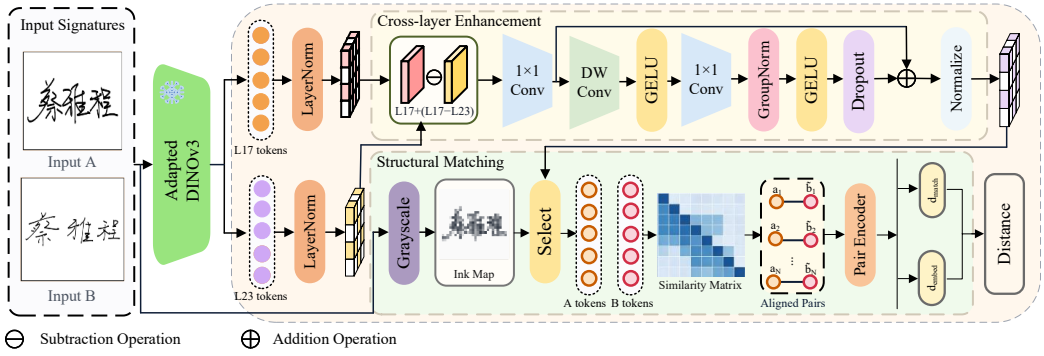


Figure 1: Overall architecture of the proposed SigDINO. The adapted DINOv3 backbone extracts hierarchical patch features, and the cross layer residual information from deeper patch features further enhances the main local stroke representation for foreground local structural matching and verification distance computation.

3 METHOD

3.1 OVERVIEW

The proposed method includes two stages. First, DINOv3 is adapted to the signature domain using a synthetic signature dataset. Second, the adapted model performs local structural matching between signature pairs. Fig. 1 presents the overall framework of SigDINO.

3.2 SIGNATURE DOMAIN SELF-SUPERVISED ADAPTATION OF DINOV3

DINOv3 learns transferable self-supervised visual representations from large scale generic image data (Siméoni et al., 2025). However, offline signature images differ markedly from natural images. They exhibit sparse strokes and extensive blank backgrounds, with discriminative visual information concentrated in stroke morphology and local structure. Directly using the generic representation may fail to fully capture these sparse and structured signature patterns. We therefore adapt DINOv3 to the signature domain to better capture the sparse stroke distribution and local structural cues of signature images, as illustrated in Fig. 2.

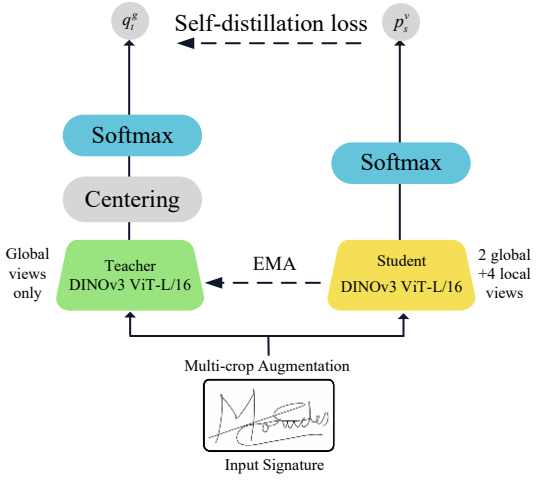


Figure 2: Signature domain self-supervised DINOv3 adaptation.

Given a signature image $\boldsymbol{x}$, we first crop the foreground signature region using a tight bounding box. We then apply random multi-crop augmentation within the cropped foreground region to generate six augmented views from the same image. The two global views are sampled as 224×224 crops to preserve the overall contour and spatial layout, while the four local views are sampled as 96×96 crops to emphasize local stroke details. Let $\mathcal{G} = \{g_1, g_2\}$ denote the two global views and $\mathcal{V}$ denote the set of all six views.

The teacher receives only the two global views and produces target distributions at the image level, whereas the student receives all six views and learns to match the teacher targets across views (Caron et al., 2021; Siméoni et al., 2025). This setting encourages the model to connect partial stroke observations with the global signature structure.

Let $\boldsymbol{q}_t^g$ denote the teacher target distribution for global view g, and let $\boldsymbol{p}_s^v$ denote the student prediction distribution for view v. Let $\mathcal{P} = \{(g, v) \mid g \in \mathcal{G}, v \in \mathcal{V}, v \neq g\}$ be the set of valid view pairs. The self-supervised objective is

$$\mathcal{L}_{\text{ssl}} = \frac{1}{|\mathcal{P}|} \sum_{(g,v)\in\mathcal{P}} H(\boldsymbol{q}_t^g, \boldsymbol{p}_s^v), \tag{1}$$

where $H(\cdot, \cdot)$ denotes cross-entropy. This objective aligns student predictions from global and local crops with the teacher targets, linking overall signature structure with partial stroke observations.

During training, the teacher is not updated by back propagation. Its parameters are updated by the exponential moving average of the student parameters (Caron et al., 2021):

$$\boldsymbol{\theta}_t \leftarrow m\boldsymbol{\theta}_t + (1 - m)\boldsymbol{\theta}_s, \tag{2}$$

where $\boldsymbol{\theta}_t$ and $\boldsymbol{\theta}_s$ are the teacher and student parameter vectors, and m is the momentum coefficient. After adaptation, we retain the teacher backbone for downstream verification and construct local representations from its patch features.

3.3 Layered Patch Feature Extraction

Offline signature verification requires preserving fine grained differences in local strokes. We therefore do not rely only on features from the final transformer layer. Features from the final layer are more globally aggregated and may dilute local stroke patterns and structural details. In contrast, patch tokens from intermediate transformer layers retain more explicit spatial structure and are better suited to representing local strokes.

Given an input signature image $\boldsymbol{x}$, let f denote the frozen adapted DINOv3 backbone, and let $f_l(\boldsymbol{x})$ denote the patch-token output extracted from its l-th transformer layer. We write this output as

$$\boldsymbol{P}_l = f_l(\boldsymbol{x}) = \left[\boldsymbol{p}_l^1, \ldots, \boldsymbol{p}_l^N\right], \tag{3}$$

where $\boldsymbol{p}_l^i$ denotes the i-th patch token and N is the number of patch tokens. We adopt patch tokens from the 17th Transformer layer (L17) as intermediate layer stroke representations because they retain more explicit spatial structure than features from the final layer and are better suited to describing stroke positions, connections, and subtle local variations.

To further strengthen the discriminative ability of local stroke representations, we incorporate a cross layer residual signal between L17 and the deeper 23rd Transformer layer (L23). Specifically, we first normalize the L17 and L23 patch tokens:

$$\widetilde{\boldsymbol{P}}_l = \mathrm{Norm}(\boldsymbol{P}_l), \quad l \in \{17, 23\}, \tag{4}$$

$$\boldsymbol{h}^i = \widetilde{\boldsymbol{p}}_{17}^i + \left(\widetilde{\boldsymbol{p}}_{17}^i - \widetilde{\boldsymbol{p}}_{23}^i\right). \tag{5}$$

Here $\boldsymbol{h}^i$ denotes the enhanced local token at the ith spatial position. As shown in Fig. 3, the cross layer residual highlights stroke responsive regions in a ChiSig example and strengthens the local stroke representation.



Figure 3: Cross layer residual enhancement on a ChiSig example. The four panels show the L17 response, L23 response, the difference between L17 and L23, and the enhanced L17 response. The residual highlights stroke responsive regions for local matching.

Because signature images contain extensive blank regions, we derive a grayscale ink map from each input and retain the corresponding stroke responsive patch tokens on the DINOv3 patch grid as foreground tokens for local matching. The resulting L17 features are enhanced with the cross layer residual and passed to the matching module, which establishes local correspondences and computes the verification distance.

3.4 Foreground Local Structural Matching

Based on the stroke responsive patch tokens selected in the previous section, we compute a local structural matching distance between a reference signature and a query signature. Since two signatures rarely have strict one-to-one spatial correspondences, tokens at the same position may correspond to different stroke regions. We therefore establish bidirectional correspondences between the two sets of stroke responsive patch tokens, allowing local stroke regions to match flexibly under displacement, scale variation, and deformation. For ease of exposition, we write the forward direction explicitly and obtain the reverse direction by symmetry. The local structural matching module is illustrated in Fig. 4.

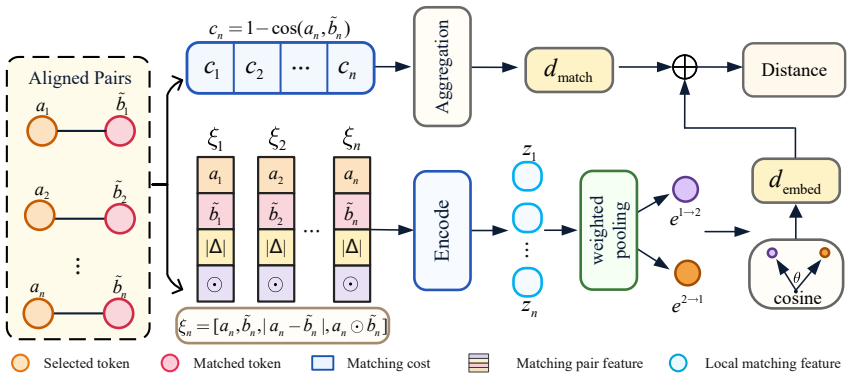


Figure 4: Computation of the local structural matching distance. The matching branch compares locally matched tokens, while the embedding branch summarizes relations between matched token pairs into compact matching embeddings.

Given a pair of signature images $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$, we represent the selected tokens of each image as

$$\mathcal{F}^{(1)} = \{(\boldsymbol{a}_i, w_i^{(1)})\}_{i=1}^{n_1}, \qquad \mathcal{F}^{(2)} = \{(\boldsymbol{b}_j, w_j^{(2)})\}_{j=1}^{n_2}. \tag{6}$$

where $\boldsymbol{a}_i$ and $w_i^{(1)}$ denote the normalized feature and normalized foreground weight of the i-th selected token in Input A, respectively, while $\boldsymbol{b}_j$ and $w_j^{(2)}$ denote the corresponding quantities in Input B. The foreground weights are normalized over the selected tokens and used to weight the local matching costs.

We first compute the similarity matrix between the two token sets:

$$S_{ij} = \langle \boldsymbol{a}_i, \boldsymbol{b}_j \rangle . \tag{7}$$

For matching from $\boldsymbol{x}^{(1)}$ to $\boldsymbol{x}^{(2)}$, the matching weights are computed as

$$A_{ij}^{1\to2} = \frac{\exp(S_{ij}/\tau) w_j^{(2)}}{\sum_{k=1}^{n_2} \exp(S_{ik}/\tau) w_k^{(2)}}, \tag{8}$$

where τ is the matching temperature. The matched representation of token $\boldsymbol{a}_i$ is then obtained by aggregating selected tokens from the other signature:

$$\widetilde{\boldsymbol{b}}_i = \sum_{j=1}^{n_2} A_{ij}^{1\to2} \boldsymbol{b}_j. \tag{9}$$

The reverse direction from $\boldsymbol{x}^{(2)}$ to $\boldsymbol{x}^{(1)}$ is computed in the same way. The directional token matching cost from $\boldsymbol{x}^{(1)}$ to $\boldsymbol{x}^{(2)}$ is defined as

$$d_{\text{match}}^{1\to2} = \sum_{i=1}^{n_1} w_i^{(1)} \left[1 - \cos\left(\boldsymbol{a}_i, \widetilde{\boldsymbol{b}}_i \right) \right], \tag{10}$$

We then average the two directional costs to obtain the token matching distance:

$$d_{\text{match}} = \frac{1}{2}\left(d_{\text{match}}^{1\to 2} + d_{\text{match}}^{2\to 1}\right). \tag{11}$$

This term measures the weighted cosine discrepancy between each selected token and its matched representation, reflecting local structural consistency between two signatures.

In addition to the explicit token matching distance, we further construct local matching embeddings from the matched token pairs. For each selected token, we form a matching pair feature as

$$\boldsymbol{\xi}_i^{1\to 2} = \left[\boldsymbol{a}_i; \widetilde{\boldsymbol{b}}_i; \left|\boldsymbol{a}_i - \widetilde{\boldsymbol{b}}_i\right|; \boldsymbol{a}_i \odot \widetilde{\boldsymbol{b}}_i\right]. \tag{12}$$

where $\odot$ denotes elementwise multiplication. The matching pair features $\boldsymbol{\xi}_i$ are encoded into local matching features $\boldsymbol{z}_i$, which are then aggregated and projected to obtain the directional matching embeddings $\boldsymbol{e}^{1\to 2}$ and $\boldsymbol{e}^{2\to 1}$. The embedding distance is computed as

$$d_{\text{embed}} = 1 - \cos\left(\boldsymbol{e}^{1\to 2}, \boldsymbol{e}^{2\to 1}\right). \tag{13}$$

The final local structural matching distance, denoted as d, combines the token matching distance and the embedding distance:

$$d = \beta_{\text{match}} d_{\text{match}} + \beta_{\text{embed}} d_{\text{embed}}, \tag{14}$$

where β_{match} and β_{embed} are the weights of the matching and embedding terms. In our experiments, $\beta_{\text{match}} = 0.80$ and $\beta_{\text{embed}} = 0.20$.

3.5 Training Objective and Inference

The downstream verification stage uses d as the discriminative score. Given a signature pair label $y \in \{0, 1\}$, where $y = 1$ denotes a genuine pair and $y = 0$ denotes a forged pair, we optimize d with a contrastive loss with two margins:

$$\mathcal{L}_{\text{dm}}(d, y) = y\left[\max(d - m_p, 0)\right]^2 + (1 - y)\left[\max(m_n - d, 0)\right]^2, \tag{15}$$

where m_p and m_n are the positive and negative margins, with $m_p < m_n$. The loss encourages genuine pairs to have distances below m_p and forged pairs to have distances above m_n. During training, an additional double margin contrastive loss is applied to difficult signature pairs selected according to their pairwise distances.

At inference time, we use the distance d between a reference signature and a query signature as the verification score and make the final verification decision by thresholding.

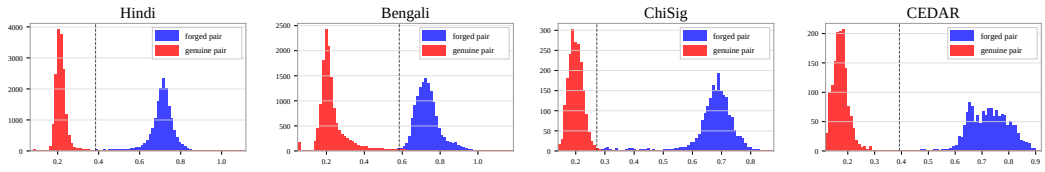


Figure 5: Distance distributions of genuine and forged signature pairs on four datasets. Genuine pairs are concentrated at lower distances, while forged pairs receive higher distances.

4 Experiments

4.1 Experimental Setup

Datasets and Evaluation Protocol. We conduct experiments on the CEDAR, BHSig-Bengali, BHSig-Hindi, and ChiSig datasets. CEDAR is an English signature dataset with 55 writers, each providing 24 genuine signatures and 24 skilled forgeries (Kalera et al., 2004). BHSig-Bengali and BHSig-Hindi contain Bengali and Hindi signatures, respectively, with 100 and 160 writers. Each

writer in both datasets provides 24 genuine signatures and 30 skilled forgeries (Pal et al., 2016). ChiSig is a Chinese signature dataset with 1,658 writers, each providing 5 genuine signatures and 5 skilled forgeries (Yan et al., 2022). Following the data partition protocol of DetailSemNet (Shih et al., 2024), the train/test writer splits for CEDAR, BHSig-Bengali, BHSig-Hindi, and ChiSig are 50/5, 50/50, 100/60, and 829/829, respectively. For benchmark evaluation, the training and test writers come from the same dataset but do not overlap. For cross-language evaluation, the model is trained on one dataset and directly evaluated on another without adaptation to the evaluation dataset. The 5-writer training setting uses only five writers for downstream training, while the ablation studies isolate the effects of the main design components.

Evaluation Metrics and Implementation Details. We report Accuracy (Acc), Equal Error Rate (EER), False Acceptance Rate (FAR), and False Rejection Rate (FRR), with EER serving as the primary evaluation metric. The proposed method is implemented in PyTorch, and all experiments are conducted on a single NVIDIA RTX 3090 GPU. Training consists of two stages. Stage 1: Signature domain self-supervised adaptation, where DINOv3 is adapted for 10 epochs using GPDS1000 synthetic signature images (Ferrer et al., 2016), without identity or genuine/forgery labels. Stage 2: Verification training, where the verification model is optimized using AdamW (Loshchilov & Hutter, 2017) with a learning rate of 1×10^{-4}, a weight decay of 0.02, and an effective batch size of 32. Benchmark evaluation uses 1,100 training steps, whereas the 5-writer setting uses 100 steps.

Table 1: Signature verification comparison on BHSig-Bengali and BHSig-Hindi (%). The best results in each column are shown in bold, and "–" denotes unavailable results.

Method	BHSig-Bengali				BHSig-Hindi			
	Acc(%)↑	FRR(%)↓	FAR(%)↓	EER(%)↓	Acc(%)↑	FRR(%)↓	FAR(%)↓	EER(%)↓
SigNet	86.11	13.89	13.89	13.89	84.64	15.36	15.36	15.36
IDN	95.32	5.24	4.12	4.68	93.04	8.99	4.93	6.96
SURDS	87.34	5.42	19.89	–	89.50	8.98	12.01	–
SDINet	94.42	7.86	3.30	5.39	95.00	3.77	6.24	5.11
CaC	96.04	3.96	3.96	3.96	94.03	5.97	5.97	5.97
AVN	93.80	5.07	7.33	6.14	94.32	5.91	5.46	5.65
2C2S	93.25	8.11	5.37	6.75	90.68	5.16	8.66	9.32
TransOSV	90.05	9.95	9.95	9.95	96.61	3.39	3.39	3.39
SigGCN	95.99	3.95	4.06	4.00	87.88	11.27	12.96	12.17
CWSA	95.70	6.37	2.21	4.11	95.45	5.04	4.05	4.66
HTCSigNet	91.17	7.83	9.20	8.52	95.26	4.65	4.61	4.63
DetailSemNet	98.19	4.04	0.95	2.11	**98.24**	3.59	**1.07**	2.07
Ours	**99.02**	**1.20**	**0.77**	**1.03**	98.11	**2.02**	1.76	**1.93**

4.2 Performance Comparison

Tables 1, 2, and 3 summarize the benchmark results. The compared baselines cover Siamese, adversarial, transformer, graph based, and structural matching OSV methods, including SigNet (Dey et al., 2017), Inverse Discriminative Networks (Wei et al., 2019), SWIS (Manna et al., 2022), SURDS (Chattopadhyay et al., 2022), SDINet (Li et al., 2021b), CAC (Lu et al., 2021), AVN (Li et al., 2021a), 2C2S (Ren et al., 2023b), TransOSV (Li et al., 2024), SigGCN (Ren et al., 2023a), CWSA (Zhang et al., 2024), HTCSigNet (Zheng et al., 2025a), DetailSemNet (Shih et al., 2024), and OCR4HSV (Lin et al., 2024). Overall, the proposed method achieves the lowest EER on all four datasets. The improvement is especially clear on CEDAR, where the EER decreases from 0.58% for DetailSemNet to 0.07% for our method. Consistent gains are also observed on the BHSig datasets, indicating robust discrimination across different writing systems.

Table 2: Comparison on CEDAR (%).

Method	Acc(%)↑	FRR(%)↓	FAR(%)↓	EER(%)↓
SigNet	–	–	–	4.63
IDN	–	2.17	5.87	3.62
SDINet	–	0.73	3.42	1.75
CWSA	98.33	1.52	1.81	1.67
DetailSemNet	99.53	0.58	0.36	0.58
Ours	**99.93**	**0.14**	**0.00**	**0.07**

Table 3: Comparison on ChiSig (%).

Method	Acc(%)↑	FRR(%)↓	FAR(%)↓	EER(%)↓
DetailSemNet	94.40	–	–	5.85
CWSA	95.71	–	–	–
OCR4HSV	95.92	5.26	2.89	–
Ours	**99.49**	**0.30**	**0.71**	**0.61**

Fig. 5 shows the distance distributions of genuine and forged signature pairs across the four datasets.

4.3 CROSS-LANGUAGE EVALUATION

Tab. 4 summarizes the cross-language evaluation results across CEDAR, BHSig-Bengali, and BHSig-Hindi, following the comparison protocol adopted by prior OSV studies (Dey et al., 2017; Wei et al., 2019; Lu et al., 2021; Chattopadhyay et al., 2022; Li et al., 2024). For each transfer direction, our method is trained on one dataset and directly tested on another, without using samples from the evaluation dataset for adaptation. Our method achieves a lower EER than DetailSemNet (Shih et al., 2024) in all six transfer directions. When trained on BHSig-Hindi, our method achieves EERs of 3.26% on BHSig-Bengali and 7.75% on CEDAR, indicating effective transfer across the evaluated datasets.

Table 4: Cross-language evaluation results across CEDAR, BHSig-Bengali, and BHSig-Hindi (%). The best results are shown in bold, and "–" denotes unavailable results.

Train	BHSig-H		BHSig-B		CEDAR	
Test	BHSig-B	CEDAR	BHSig-H	CEDAR	BHSig-H	BHSig-B
SigNet	39.35	40.43	35.43	50.00	44.39	35.85
IDN	25.88	50.00	25.70	50.00	49.64	49.99
CaC	14.66	29.49	30.41	33.71	39.08	38.07
SURDS	27.74	–	32.99	–	–	–
TransOSV	18.66	–	17.17	–	–	–
DetailSemNet	7.46	14.05	15.91	7.32	16.35	8.40
Ours	**3.26**	**7.75**	**11.75**	**5.87**	**10.16**	**7.75**

4.4 PERFORMANCE UNDER THE 5-WRITER TRAINING SETTING

In practical applications, collecting labeled signatures from many writers is costly. We therefore evaluate writer-independent OSV in the 5-writer training setting, where five writers are used for training and the remaining writers are reserved for testing. Tab. 5 compares our method with four representative writer-independent baselines: SigNet (Dey et al., 2017), IDN (Wei et al., 2019), HTC-SigNet (Zheng et al., 2025a), and DetailSemNet (Shih et al., 2024).

Table 5: Signature verification comparison under the 5-writer training setting (%). The best results in each column are shown in bold.

Method	Bengali		Hindi		CEDAR		ChiSig	
	Acc(%)↑	EER(%)↓	Acc(%)↑	EER(%)↓	Acc(%)↑	EER(%)↓	Acc(%)↑	EER(%)↓
SigNet	83.61	16.49	77.05	22.99	75.50	25.32	82.76	18.00
IDN	66.38	34.01	77.92	22.26	55.18	46.83	65.55	34.90
HTCSigNet	82.12	18.75	81.52	18.68	73.60	26.64	61.86	38.31
DetailSemNet	79.45	22.93	85.04	15.41	64.96	35.59	57.08	43.15
Ours	**94.26**	**5.85**	**90.90**	**9.18**	**89.75**	**10.55**	**89.68**	**10.62**

The proposed method consistently outperforms representative baselines across all four datasets. On BHSig-Bengali, BHSig-Hindi, CEDAR, and ChiSig, it achieves Acc values of 94.26%, 90.90%, 89.75%, and 89.68%, with corresponding EERs of 5.85%, 9.18%, 10.55%, and 10.62%, respectively. In contrast, all competing methods report EERs above 15% under the same setting. These results indicate that our method provides more stable verification performance when downstream training is limited to five writers, while the proposed matching distance remains effective under this restricted training setting.

4.5 ABLATION STUDIES

We examine the contribution of the key components in Tab. 6 and Tab. 7. Tab. 6 studies the backbone, local matching design, and stroke token layer under the benchmark evaluation setting. Tab. 7 focuses on the backbone and signature domain adaptation under the 5-writer training setting. For both tables, Avg. denotes the arithmetic mean of the EER values across Bengali, Hindi, CEDAR, and ChiSig. Here, "signature SSL" denotes signature domain self-supervised adaptation on GPDS1000 synthetic signature images, while "Raw DINOv3" denotes the pretrained DINOv3 backbone without signature domain self-supervised adaptation.

Backbone choice accounts for most of the performance difference. Under the benchmark evaluation setting, ViT-L + signature SSL yields an average EER of 18.87%, raw DINOv3 lowers it to 2.30%, and the full model reaches 0.91%. This suggests that DINOv3's generic visual representations provide the main gain, while signature domain adaptation further improves performance on this basis. The same trend appears in the 5-writer training setting. ViT-L + signature SSL, raw DINOv3, and the full model obtain average EERs of 38.69%, 16.60%, and 9.05%, respectively. This comparison shows that performance does not depend only on exposure to signature domain images. Self-supervised adaptation on GPDS1000 gives ViT-L access to signature data, but the weaker backbone still struggles to form a reliable signature-pair distance. DINOv3, trained on large scale and diverse image data, provides a more transferable structural representation, and this domain adaptation further calibrates the representation to sparse signature images, bringing an additional gain over raw DINOv3. The local matching components also matter. Removing the cross layer residual raises the average EER from 0.91% to 1.21%, indicating that deeper patch features provide useful complementary structure for foreground matching. The matching distance directly measures token-level correspondence after soft matching, whereas the embedding distance summarizes the matched local relations into a more stable discriminative representation. Their combination therefore preserves explicit local correspondence while improving the robustness of the final signature-pair distance. Layer selection also has a clear effect on local matching. L17 performs best, L14 is close but slightly worse, and L20 and L23 degrade more noticeably. This suggests that shallower features retain useful low-level stroke information but offer weaker discrimination, whereas deeper features begin to lose local position and stroke-connection cues. Overall, L17 provides the best balance between local spatial structure and discriminative representation for foreground token matching.

Table 6: Ablation study on representation backbone, local matching design, and stroke token layer (%).

Group	Variant	Bengali	Hindi	CEDAR	ChiSig	Avg.
Representation backbone						
	ViT-L + signature SSL	21.30	29.55	10.65	13.96	18.87
	Raw DINOv3	4.46	2.70	1.01	1.02	2.30
Local matching design						
	Without cross layer residual	1.65	2.38	0.14	0.66	1.21
	Matching distance only	2.66	2.43	0.65	2.99	2.18
	Embedding distance only	2.20	2.31	0.14	1.37	1.51
Stroke token layer						
	L14 patch tokens	1.12	1.94	0.14	0.71	0.98
	L20 patch tokens	1.43	2.33	0.29	1.47	1.38
	L23 patch tokens	2.93	3.47	0.22	2.13	2.19
Full model						
	Ours	**1.03**	**1.93**	**0.07**	**0.61**	**0.91**

Table 7: Backbone and signature domain adaptation ablation under the 5-writer training setting (%).

Variant	Bengali	Hindi	CEDAR	ChiSig	Avg.
ViT-L + signature SSL	34.82	45.16	44.50	30.26	38.69
Raw DINOv3	20.92	11.51	17.58	16.38	16.60
Ours	**5.85**	**9.18**	**10.55**	**10.62**	**9.05**

5 Conclusion

This paper proposes SigDINO for writer-independent offline signature verification. Its main contribution is to adapt DINOv3 to the signature domain and use intermediate patch tokens as the main local representation, enhanced by cross layer residual information for foreground local structural matching between signature pairs. Experiments on four public datasets show strong benchmark performance, competitive cross-language performance, and robust results under limited writer settings. Ablation studies further indicate that the generic DINOv3 representation provides the main gain, while signature domain adaptation and foreground local structural matching bring additional improvements. The relatively large backbone increases inference cost and may limit latency-sensitive deployment. Future work will explore lightweight architectures and efficient matching strategies.

REFERENCES

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEiT: BERT pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9630–9640. IEEE, 2021.

Soumitri Chattopadhyay, Siladittya Manna, Saumik Bhattacharya, and Umapada Pal. SURDS: Self-supervised attention-guided reconstruction and dual triplet loss for writer independent offline signature verification. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pp. 1600–1606. IEEE, 2022.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pp. 1597–1607. PMLR, 2020.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9620–9629. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00950.

Sounak Dey, Anjan Dutta, J. Ignacio Toledo, Suman K. Ghosh, Josep Lladós, and Umapada Pal. SigNet: Convolutional siamese network for writer independent offline signature verification. *arXiv preprint arXiv:1707.02131*, 2017.

Anjan Dutta, Umapada Pal, and Josep Lladós. Compact correlated features for writer independent signature verification. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pp. 3422–3427. IEEE, 2016.

Miguel A. Ferrer, Moises Diaz, Cristina Carmona-Duarte, and Aythami Morales. A behavioral handwriting model for static and dynamic signature synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1041–1053, 2016.

Luiz G. Hafemann, Robert Sabourin, and Luiz S. Oliveira. Learning features for offline handwritten signature verification using deep convolutional neural networks. *Pattern Recognition*, 70:163–176, 2017a.

Luiz G. Hafemann, Robert Sabourin, and Luiz S. Oliveira. Offline handwritten signature verification: Literature review. In *2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pp. 1–8. IEEE, 2017b.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9726–9735. IEEE, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15979–15988. IEEE, 2022.

Meenakshi K. Kalera, Sargur N. Srihari, and Aihua Xu. Offline signature verification and identification using distance statistics. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(7):1339–1360, 2004. doi: 10.1142/S0218001404003630.

Rajesh Kumar, J. D. Sharma, and Bhabatosh Chanda. Writer-independent off-line signature verification using surroundedness feature. *Pattern Recognition Letters*, 33(3):301–308, 2012.

Huan Li, Ping Wei, and Ping Hu. Avn: An adversarial variation network model for handwritten signature verification. *IEEE Transactions on Multimedia*, 24:594–608, 2021a.

Huan Li, Ping Wei, and Ping Hu. Static-dynamic interaction networks for offline signature verification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 1893–1901, 2021b.

Huan Li, Ping Wei, Zeyu Ma, Changkai Li, and Nanning Zheng. TransOSV: Offline signature verification with transformers. *Pattern Recognition*, 145:109882, 2024.

Chao-Qun Lin, Da-Han Wang, Yan-Fei Su, De-Wu Ge, and Xu-Yao Zhang. OCR4HSV: A multi-task learning approach for handwritten signature verification. In *International Conference on Pattern Recognition*, pp. 287–302. Springer, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Xi Lu, Linlin Huang, and Fei Yin. Cut and compare: End-to-end offline signature verification network. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 3589–3596. IEEE, 2021.

Siladittya Manna, Soumitri Chattopadhyay, Saumik Bhattacharya, and Umapada Pal. SWIS: Self-supervised representation learning for writer independent offline signature verification. In *2022 IEEE International Conference on Image Processing (ICIP)*, pp. 1411–1415. IEEE, 2022.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Srikanta Pal, Alireza Alaei, Umapada Pal, and Michael Blumenstein. Performance of an off-line signature verification method based on texture features on a large indic-script signature dataset. In *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, pp. 72–77. IEEE, 2016. doi: 10.1109/DAS.2016.48.

Ebrahim Parcham, Mahdi Ilbeygi, and Mohammad Amini. CBCapsNet: A novel writer-independent offline signature verification model using a CNN-based architecture and capsule neural networks. *Expert Systems with Applications*, 185:115649, 2021.

Chengkai Ren, Jian Zhang, Hongwei Wang, and Shuguang Shen. Vision graph convolutional network for writer-independent offline signature verification. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7. IEEE, 2023a.

Jian-Xin Ren, Yu-Jie Xiong, Hongjian Zhan, and Bo Huang. 2C2S: A two-channel and two-stream transformer based framework for offline signature verification. *Engineering Applications of Artificial Intelligence*, 118:105639, 2023b.

Meng-Cheng Shih, Tsai-Ling Huang, Yu-Heng Shih, Hong-Han Shuai, Hsuan-Tung Liu, Yi-Ren Yeh, and Ching-Chun Huang. DetailSemNet: Elevating signature verification through detail-semantic integration. In *European Conference on Computer Vision*, pp. 449–466. Springer, 2024.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.

B. S. Thakare and H. R. Deshmukh. A combined feature extraction model using SIFT and LBP for offline signature verification system. In *2018 3rd International Conference for Convergence in Technology (I2CT)*, pp. 1–7. IEEE, 2018.

Abundance Udo, Jianhui Guo, Aloysious Barlea, and Samia Ben Amarat. BYOLSiam: Self-supervised pre-training for offline signature verification. In *2024 International Conference on Image Processing, Computer Vision and Machine Learning (ICICML)*, pp. 1854–1859. IEEE, 2024. doi: 10.1109/ICICML63543.2024.10957965.

Ping Wei, Huan Li, and Ping Hu. Inverse discriminative networks for handwritten signature verification. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5757–5765. IEEE, 2019.

Kaihong Yan, Ying Zhang, Haoran Tang, Chengkai Ren, Jian Zhang, Gaoang Wang, and Hongwei Wang. Signature detection, restoration, and verification: A novel chinese document signature forgery detection benchmark. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 5159–5168. IEEE, 2022.

Mustafa Berkay Yilmaz, Berrin Yanikoglu, Caglar Tirkaz, and Alisher Kholmatov. Offline signature verification using classifier combination of hog and lbp features. In *2011 international joint conference on Biometrics (IJCB)*, pp. 1–7. IEEE, 2011.

Yongliang Zhang, Jiahang Wang, Zipeng Chen, and Ziwen Li. Cross layer weakly supervised data augmentation network for offline signature verification. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2024.

Xingbiao Zhao, Lidong Zheng, Panli Yuan, and Yuchen Zheng. Fusing deep and hand-crafted features by deep canonically correlated contractive autoencoder for offline signature verification. *Pattern Recognition*, 168:111834, 2025.

Lidong Zheng, Da Wu, Shengjie Xu, and Yuchen Zheng. HTCSigNet: A hybrid transformer and convolution signature network for offline signature verification. *Pattern Recognition*, 159:111146, 2025a.

Lidong Zheng, Xingbiao Zhao, Shengjie Xu, Yuanyuan Ren, and Yuchen Zheng. Learning discriminative representations by a canonical correlation analysis-based siamese network for offline signature verification. *Engineering Applications of Artificial Intelligence*, 139:109640, 2025b.

Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. iBOT: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021.

A Additional Experimental Details

This appendix provides implementation details, dataset splits, and control experiments that support the main results.

A.1 The Use of Large Language Models (LLMs)

We used LLMs only to polish and correct the grammar of this paper. This involved rephrasing sentences, checking grammar, and improving the overall flow for better readability and clarity. All research ideation, experiments, and analyses were developed by the authors, with no involvement from LLMs in these stages.

A.2 Datasets and Splits

We conduct experiments on four public offline signature datasets. Tab. 8 summarizes the dataset sizes and data splits used in our experiments.

CEDAR contains 55 writers, each with 24 genuine signatures and 24 skilled forgeries. BHSig-Bengali and BHSig-Hindi contain 100 and 160 writers, respectively, with 24 genuine signatures and 30 skilled forgeries per writer. ChiSig contains 1,658 writers, each with 5 genuine signatures and 5 skilled forgeries. Following the data split protocol of DetailSemNet (Shih et al., 2024), the train/test writer splits for CEDAR, BHSig-Bengali, BHSig-Hindi, and ChiSig are 50/5, 50/50, 100/60, and 829/829, respectively. Benchmark evaluation trains and tests on disjoint writers within the same dataset. Cross-language evaluation uses a model trained on one source dataset and evaluates it directly on another target dataset, without any training or fine tuning on the target dataset. For 5-writer training, only five writers are used for downstream training, while the remaining writers are used for testing.

Table 8: Dataset statistics and writer splits used in our experiments.

Dataset	Writers	Genuine / writer	Skilled forgeries / writer	Train / test writers
CEDAR	55	24	24	50 / 5
BHSig-Bengali	100	24	30	50 / 50
BHSig-Hindi	160	24	30	100 / 60
ChiSig	1,658	5	5	829 / 829

A.3 Implementation Details

We initialize DINOv3 ViT-L/16 from its publicly released pretrained weights and adapt it to the signature domain using GPDS1000 synthetic signature images without writer identity or authenticity labels. For each signature, we generate two global crops of size 224×224 and four local crops of size 96×96, covering 50% to 100% and 10% to 40% of the image area, respectively. The student processes all six crops, whereas the EMA teacher processes only the two global crops. The adaptation uses the image level DINO objective and does not include the iBOT patch objective.

During adaptation, all parameters of the student backbone and DINO head are optimized for 10 epochs using AdamW with a batch size of 16, a fixed learning rate of 1×10^{-5}, and a weight decay of 1×10^{-4}. The DINO head uses an output dimension of 65,536. The student and teacher temperatures are fixed at 0.1 and 0.04, respectively, and the center momentum is set to 0.9. The teacher EMA momentum increases linearly from 0.996 to 1.0, and gradient norms are clipped to 3.0. The EMA teacher backbone obtained after the final epoch is used for all downstream verification experiments.

For downstream verification, the adapted backbone is frozen. For benchmark evaluation, we use a micro-batch size of 8 and accumulate gradients over 4 steps, resulting in an effective batch size of 32. The head learning rate is 1×10^{-4}, and the weight decay is 0.02. For 5-writer training, we use the same micro-batch size and gradient accumulation, 100 optimization steps, a head learning rate of 2×10^{-5}, and the same weight decay. The positive and negative margins in the downstream double margin loss are set to $m_p = 0.25$ and $m_n = 0.75$, respectively. We use L17 tokens as the main local stroke representation and introduce a cross layer residual from L23. We retain all patch

tokens associated with foreground cells and compute the local structural matching distance from the token matching distance and the embedding distance, weighted 0.80 and 0.20. We report Acc and EER following the evaluation protocol used in the main experiments.

To provide additional context to the pair encoder, each selected token is accompanied by its normalized two-dimensional patch coordinate and foreground weight. Following bidirectional soft matching, the encoder also uses token-level matching confidence and normalized matching uncertainty as correspondence cues. The confidence is the cosine similarity between a token and its softly aligned counterpart, whereas the uncertainty is derived from the entropy of the corresponding soft matching distribution. Coordinates provide spatial context but do not enter token similarity computation or soft matching logits. Foreground weights modulate matching and aggregation, while confidence and uncertainty characterize correspondence reliability. The two directions are encoded separately and then combined through foreground-weighted pooling and projection to obtain directional matching embeddings. To visualize score separation under the 5-writer training setting, Fig. 6 shows the distance distributions of genuine and forged signature pairs across the four datasets.

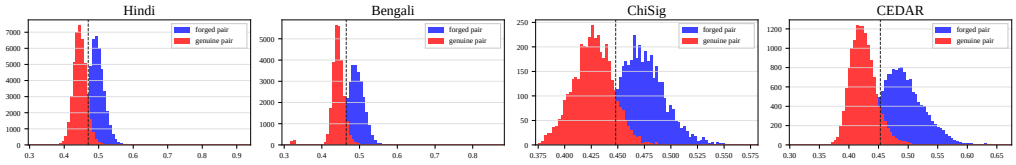


Figure 6: Distance distributions of genuine and forged signature pairs under the 5-writer training setting. The panels show BHSig-Hindi, BHSig-Bengali, ChiSig, and CEDAR from left to right. Genuine pairs generally exhibit lower distances than forged pairs, although the two distributions partially overlap.

A.4 First-Stage SSL Recipe Ablation

To examine how the first stage SSL recipe affects downstream verification, we compare three configurations under the same downstream protocol. Tab. 9 reports Acc and EER under benchmark evaluation.

For context, the official DINOv3 recipe uses two global crops and eight local crops, together with image level DINO and patch level iBOT objectives (Siméoni et al., 2025). We use this design as the reference point, but examine a reduced crop design for signature-domain adaptation because signature images contain sparse foreground strokes and extensive blank backgrounds. The ablation varies two aspects of the first-stage SSL recipe, namely the number of local crops and the inclusion of the patch-level objective. 2G+6L uses two global crops and six local crops with the image-level DINO objective only. This control evaluates the effect of reducing the number of local views relative to the DINOv3 configuration. 2G+4L+iBOT uses two global crops and four local crops and additionally introduces an iBOT-style masked patch objective. This control evaluates whether patch-level prediction provides an additional benefit when the number of local views is reduced. 2G+4L (Ours) uses two global crops and four local crops with the DINO objective only, corresponding to the first-stage SSL configuration adopted in our method. All three configurations use the same GPDS1000 adaptation data, image preprocessing, and downstream verification protocol. The comparison therefore isolates the effects of the local-crop count and the SSL objective.

Table 9: First-stage SSL recipe ablation under benchmark evaluation (Acc/EER%). The best results in each column are shown in bold.

Method	Bengali		Hindi		CEDAR	
	Acc(%)↑	EER(%)↓	Acc(%)↑	EER(%)↓	Acc(%)↑	EER(%)↓
2G+6L	98.12	1.91	97.71	2.30	99.92	0.14
2G+4L+iBOT	98.35	1.72	97.67	2.34	99.85	0.29
2G+4L (Ours)	**99.02**	**1.03**	**98.11**	**1.93**	**99.93**	**0.07**

Among the three configurations, 2G+4L (Ours) achieves the highest Acc and the lowest EER on all three datasets. We therefore use this configuration for first-stage SSL adaptation.

A.5 Self-Supervised Backbone Comparison

We compare several self-supervised backbones after GPDS1000 adaptation under the same downstream matching and evaluation pipeline. The compared backbones differ in pretraining data and learning objectives. MAE uses masked image reconstruction (He et al., 2022), iBOT combines masked patch prediction with self-distillation (Zhou et al., 2021), and MoCo v3 uses contrastive learning (Chen et al., 2021). DINOv2 and DINOv3 use self-distillation and were pretrained on larger curated image collections (Oquab et al., 2023; Siméoni et al., 2025).

Under the same benchmark evaluation protocol, Tab. 10 compares adapted MAE ViT-L/16, iBOT ViT-L/16, DINOv2 ViT-L/14, and MoCo v3 ViT-B/16 with Ours.

Table 10: Comparison with alternative adapted self-supervised backbones under the benchmark evaluation protocol (Acc/EER%). The best results in each column are shown in bold.

Method	Bengali		Hindi		CEDAR	
	Acc(%)↑	EER(%)↓	Acc(%)↑	EER(%)↓	Acc(%)↑	EER(%)↓
MAE ViT-L/16	93.82	6.26	88.40	11.61	95.97	4.28
iBOT ViT-L/16	97.47	2.59	96.09	3.93	99.52	0.58
DINOv2 ViT-L/14	98.08	2.02	95.93	4.22	99.56	0.65
MoCo v3 ViT-B/16	94.38	5.70	94.34	5.67	98.47	1.81
Ours	**99.02**	**1.03**	**98.11**	**1.93**	**99.93**	**0.07**

Tab. 11 further examines the effect of DINOv3 backbone scale by comparing ViT-B/16 and ViT-S/16 with Ours. The adapted DINOv3 ViT-L/16 used by SigDINO achieves the strongest overall performance among the compared configurations.

Table 11: Comparison with different adapted DINOv3 backbone scales under the benchmark evaluation protocol (Acc/EER%). The best results in each column are shown in bold.

Method	Bengali		Hindi		CEDAR		ChiSig	
	Acc(%)↑	EER(%)↓	Acc(%)↑	EER(%)↓	Acc(%)↑	EER(%)↓	Acc(%)↑	EER(%)↓
DINOv3 ViT-B/16	97.06	3.01	95.94	4.07	99.34	0.65	99.28	0.76
DINOv3 ViT-S/16	93.88	6.70	95.06	4.95	98.83	1.74	97.76	2.36
Ours	**99.02**	**1.03**	**98.11**	**1.93**	**99.93**	**0.07**	**99.49**	**0.61**

For the 5-writer training setting, Tab. 12 further reports the Acc/EER values of the same backbone controls together with the full SigDINO result. In this setting, only 5 writers are used for training, while the remaining writers are reserved for testing.

Table 12: Additional backbone comparison under the 5-writer training setting (Acc/EER%). The best results in each column are shown in bold.

Method	Bengali		Hindi		CEDAR		ChiSig	
	Acc(%)↑	EER(%)↓	Acc(%)↑	EER(%)↓	Acc(%)↑	EER(%)↓	Acc(%)↑	EER(%)↓
MAE ViT-L/16	79.44	25.45	70.38	29.76	77.47	22.64	80.47	19.82
iBOT ViT-L/16	86.86	13.13	80.15	19.90	84.22	15.82	**90.96**	**9.38**
MoCo v3 ViT-B/16	88.19	11.88	83.62	13.76	85.08	15.00	89.65	10.58
DINOv3 ViT-B/16	93.55	6.48	84.40	15.92	86.85	13.29	88.05	12.05
DINOv3 ViT-S/16	87.43	12.60	82.70	17.34	80.11	19.95	83.73	16.53
Ours	**94.26**	**5.85**	**90.90**	**9.18**	**89.75**	**10.55**	89.68	10.62

These results indicate that backbone choice remains a major factor after signature-domain adaptation. Among the large ViT models, iBOT performs substantially better than MAE, despite their

similar nominal backbone scale. DINOv3 ViT-B/16 also transfers better than ViT-S/16, supporting the role of backbone capacity. The strongest overall result is obtained by the adapted DINOv3 ViT-L/16 configuration, which combines a higher-capacity backbone with a self-supervised representation learned from substantially broader pretraining data. Since all configurations use the same GPDS1000 adaptation data and downstream matching pipeline, this comparison identifies DINOv3 ViT-L/16 as the most effective initialization among the evaluated backbones.

A.6 Input Resolution and Token Density

Several OSV baselines use 224×224 signature images, whereas SigDINO uses 448×448 inputs by default. We therefore conduct an input resolution control experiment to separate the effect of image resolution from the denser patch token grid used by the ViT backbone.

With a 16×16 patch size, a 448×448 input produces 784 patch tokens, whereas a native 224×224 input produces only 196 patch tokens. Native 224 input therefore provides substantially fewer local tokens for stroke responsive patch token selection and local structural matching. We also evaluate a $224 \rightarrow 448$ setting, where the image is first resized to 224×224 and then bilinearly upsampled to 448×448 before patch embedding. This keeps the image content comparable to a 224 input while producing the denser patch token grid used by the 448 setting.

Tab. 13 shows that $224 \rightarrow 448$ achieves a lower average EER than 224 and remains closer to the 448 setting. The improvement of $224 \rightarrow 448$ over native 224 suggests that the denser patch token grid is an important factor beyond pixel resolution alone.

Table 13: Input resolution control under benchmark evaluation (EER%). Best EER values are shown in bold.

Input setting	Bengali	Hindi	CEDAR	ChiSig	Avg.
448	**1.03**	1.93	**0.07**	**0.61**	**0.91**
224→448	1.78	**1.84**	0.58	1.68	1.47
224	2.52	3.68	**0.07**	3.55	2.46

To visualize the token-level similarities underlying local structural matching, Fig. 7 shows cosine similarity matrices for genuine and forged pairs.

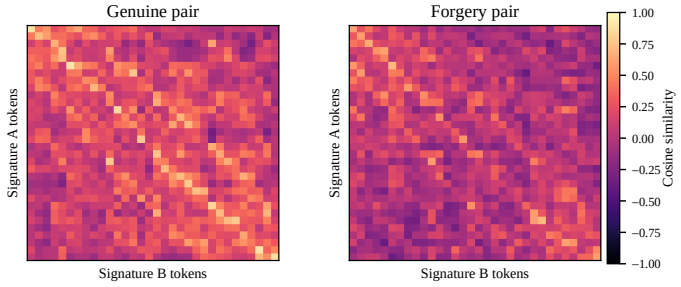


Figure 7: Token-level cosine similarity matrices for genuine and forged pairs. The genuine pair exhibits more structured high-similarity patterns.

Figure 8 visualizes token correspondences produced by the local structural matching module for an example genuine pair. The figure is placed at the top of the following page to preserve its readability.

A.7 Sensitivity Analysis of the Local Structural Matching Distance Weights

The final local structural matching distance is defined as a weighted sum of the token matching distance and the embedding distance, with coefficients 0.80 and 0.20 in the full SigDINO model. To analyze this coefficient choice, we evaluate the adopted 0.80/0.20 ratio, alternative mixed ratios, and settings that retain only one distance term. Tab. 14 summarizes the results under the same downstream verification setting.



Figure 8: Visualization of local structural matching for a genuine signature pair. The two panels show the signatures in the pair. Matching colors and indices identify corresponding stroke responsive patch tokens selected by the local structural matching module.

Table 14: Weight sensitivity of the local structural matching distance (EER%). Weight pairs are reported as $\beta_{\text{match}}/\beta_{\text{embed}}$. Best EER values are shown in bold.

Dataset	1.00 / 0.00	0.90 / 0.10	0.80 / 0.20	0.50 / 0.50	0.00 / 1.00
Bengali	2.66	1.96	**1.03**	1.39	2.20
Hindi	2.43	**1.81**	1.93	1.97	2.31
CEDAR	0.65	0.14	**0.07**	**0.07**	0.14
ChiSig	2.99	**0.51**	0.61	1.07	1.37
Avg.	2.18	1.10	**0.91**	1.13	1.51

The results show that, on average, combining the two distance terms is more effective than using either term alone. The 0.80/0.20 setting achieves the lowest average EER, while nearby settings show small dataset-dependent variations. The results support assigning most of the weight to the token matching distance, which directly compares local token discrepancies, while the embedding distance aggregates relations between matched token pairs.