CAN A WORLD MODEL PREDICT A DEEPFAKE VIDEO? A CONSISTENCY MARGIN PRIOR FOR DEEPFAKE DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Deepfake detectors often perform well on the dataset they are trained on, but their performance drops when tested on videos from different datasets. In this paper, we study whether a predictive video model can provide a signal that generalizes better across datasets. Instead of relying only on generator-specific artifacts, as many existing detectors do, we use a model trained to predict natural video and measure the difference between what it predicts and what it observes in a video.

Using V-JEPA 2.1, we keep the encoder frozen and adapt the predictor using rank-16 LoRA. We define the difference between the predictor output and the frozen encoder target as the *prediction residual*. We hypothesize that manipulated clips produce, on average, larger prediction residuals than authentic clips, and use this as our consistency-margin prior. We process these residuals using a lightweight Latent Consistency Head (LCH), which separates residual magnitude from direction for training and classification.

To test whether the direction of the consistency margin affects cross-dataset generalization, we compare standard and reversed margin objectives while keeping the remaining training setup fixed. We match the two settings on FF++ validation performance before evaluating them on unseen data. The reversed-margin setting exhibits a larger cross dataset generalization gap on average, providing evidence that the direction of the consistency margin affects cross-dataset transfer. Our complete model adds only $8.25\,\mathrm{M}$ trainable parameters, which is 2.47% of the total $334\,\mathrm{M}$ parameters. Trained only on FF++, it achieves an AUC of 90.5 on Celeb-DFv2 and performs well across several other cross-dataset benchmarks. Our ablations show that separating residual magnitude from direction, reducing dataset-specific variation, and using a low-rank projection improve cross-dataset generalization in our experiments.

1 INTRODUCTION

Deepfake detectors often achieve strong performance on the datasets they are trained on, but their performance drops when they are tested on data from different sources. XceptionNet (Chollet, 2017; Rössler et al., 2019) and SBI (Shiohara & Yamasaki, 2022), for example, achieve strong results on FF++, while cross-dataset evaluation on Celeb-DFv2 (Li et al., 2020), DFDC (Dolhansky et al., 2020), and WildDeepfake (Zi et al., 2020) remains much more difficult. Recent benchmarks show the same pattern (Chandra et al., 2025; Yan et al., 2024; 2025d; NTIRE Workshop, 2026). One reason is that detectors learn dataset-specific cues, such as compression traces, generator fingerprints, or collection biases. These cues may work well on the training dataset but may not generalize to new datasets (Cao & Ma, 2025; Wang et al., 2023b; Guo et al., 2025; Yan et al., 2023; Hu et al., 2025). This motivates us to look for a signal that is less dependent on a specific dataset or deepfake generator.

We explore this idea using the self-supervised video prediction capabilities of V-JEPA 2.1 (Bardes et al., 2026). We keep the V-JEPA encoder frozen and adapt its predictor using rank-16 LoRA on all 12 predictor blocks. We then compare the predictor output with the corresponding stop-gradient encoder target. We call the difference between them the *prediction residual* (r). Our main hypothesis

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.

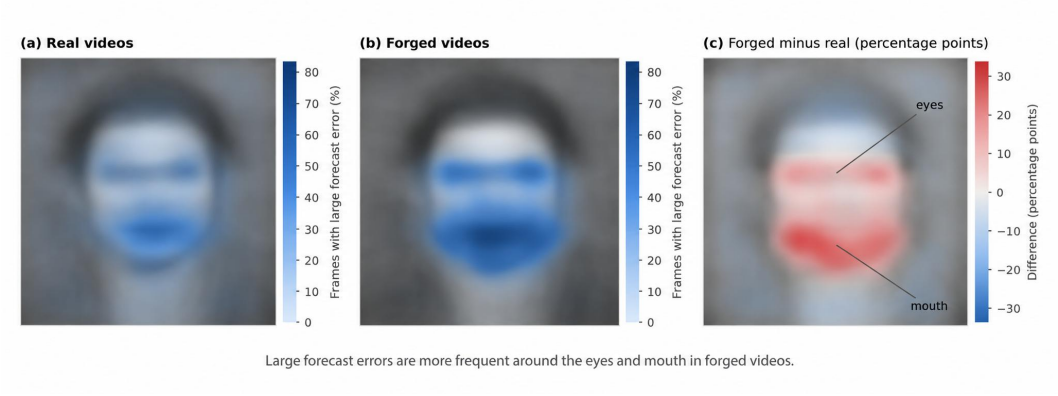

Figure 1: **Spatial distribution of prediction errors.** Average spatial maps of large forecast errors for real and forged videos, together with their difference.

is that manipulated clips should, on average, produce larger prediction residuals than authentic clips (Fig. 1). We define this as the consistency-margin prior, i.e., $\|r_{\text{fake}}\| > \|r_{\text{real}}\|$.

To use these residuals for deepfake detection, we introduce the *Latent Consistency Head (LCH)*, a lightweight detector built on top of the V-JEPA 2.1 predictor. LCH separates the residual into magnitude and direction. The residual magnitude is used by the consistency loss, while the residual direction is used to classify the video as real or fake. Additionally, we will show how LCH processes the residual using our Predictor Attention-conditioned Stop- gradient Centering (PASC), horizon-specific projections, and spatial anomaly pooling to learn features for cross-dataset detection. We show that although trained only on FF++, the detector performs well across several cross-dataset benchmarks (Celeb-DFv2, DFDC, WildDeepfake, and Deepfake-Eval-2024), indicating that prediction residuals can provide a useful signal across datasets.

We further examine whether the direction of the consistency margin affects cross-dataset generalization. We introduce the Sign-Flip Identification Protocol (SFIP), which compares a standard setting that encourages larger residual magnitudes for manipulated clips with a reversed setting that encourages larger residual magnitudes for authentic clips. The two settings keep the remaining training setup fixed and are matched on FF++ validation performance before cross-dataset evaluation. Across the evaluated cross-dataset benchmarks, the reversed-margin setting has an average generalization gap of 1.8% larger than the standard setting, indicating worse cross-dataset transfer under the reversed margin. We describe SFIP in Section 3.4. Furthermore, Our model remains parameter-efficient, adding only a small set of trainable parameters to V-JEPA 2.1. The LCH contains 6.31M trainable parameters, while predictor adaptation contributes an additional 1.48M.

Contributions.

1. **A latent prediction-residual formulation for deepfake detection.** We introduce a video-level forensic formulation based on V-JEPA 2.1 prediction residuals, extending residual-based deepfake detection from image reconstruction to self-supervised latent video prediction.
2. **Latent Consistency Head.** We propose LCH, a lightweight detector that separates residual magnitude from direction and combines PASC, horizon-specific projections, and spatial anomaly pooling to learn features for cross-dataset real/fake classification.
3. **A matched sign-flip test for the consistency margin prior.** We introduce SFIP to compare standard and reversed consistency-margin objectives at matched FF++ validation performance. Across the evaluated cross-dataset benchmarks, reversing the margin increases the average generalization gap by 1.8%, providing evidence that margin direction affects cross-dataset transfer.

2 Related Work

Frozen foundation models for forensics. Recent deepfake detectors adapt large pretrained models instead of training the full backbone. Forensics Adapter (Cui et al., 2025), LNCLIP-DF (Tan et al., 2025), DeepFake-Adapter (Shao et al., 2025), and UniFD (Ojha et al., 2023) show that frozen or

lightly adapted foundation models can transfer well across datasets but current works do not use predictive video models like V-JEPA and their prediction results for deepfake detection".

Reconstruction and residual signals. Several deepfake detectors use reconstruction error or consistency as a forensic signal (Zhao et al., 2021; Khalid & Woo, 2020; Wang et al., 2022; Cao et al., 2022; Dong et al., 2023; Wang et al., 2025). DIRE (Wang et al., 2023a), for example, detects generated images using diffusion reconstruction errors, while NPR (Tan et al., 2024a) uses neighboring-pixel relationships. Our method follows the same general idea of using prediction error, but computes it in the latent space of a self-supervised video predictor.

Cross-dataset deepfake detection. Many methods improve cross-dataset detection through augmentation, temporal modeling, feature disentanglement, or more general representations (Nguyen et al., 2024; Shiohara & Yamasaki, 2022; Yan et al., 2023; Zheng et al., 2021; Wang et al., 2023b; Guo et al., 2025; Haliassos et al., 2021). Other methods use frequency information to capture forgery artifacts (Qian et al., 2020; Tan et al., 2024b; Baru et al., 2025; Cao & Ma, 2025). However, these cues can still depend on the generator, compression pipeline, or dataset. We instead study whether prediction residuals from a self-supervised video model provide a signal that generalizes across datasets.

Relationship to existing components. The individual operations used in our detector are related to existing ideas. The Bridge uses low-rank projection (Hu et al., 2022), residual normalization is related to normalized embedding methods (Wang et al., 2017), PASC is related to grouped normalization (Wu & He, 2018), and our consistency objective uses a margin loss (Schroff et al., 2015). Our contribution is not these operations individually, but how they are combined to classify predictive-video residuals and how we test the consistency-margin direction using SFIP. Additional comparisons with recent detectors are provided in Appendix B.

3 METHOD

In this section, we now describe the proposed architecture - ConsistencyNet (Fig. 2) that utlizes the prediction residual introduced in Section 1.

3.1 PROBLEM FORMULATION

Prediction residual. For a video clip $\mathcal{V}$, we form two inputs from the same 16 frames and pass them through the V-JEPA 2.1 encoder. The first contains the complete, non-masked frames, while the second applies a spatio-temporal mask to the same frames. The encoder produces a temporal feature representation for each input, i.e., $\mathbf{H}_{\text{full}}$, $\mathbf{H}_{\text{mask}} \in \mathbb{R}^{T\times S\times 1024}$ where $T = 8$ is the number of temporal feature positions and $S = 576$ is the number of spatial tokens at each temporal position. We omit the batch dimension for clarity. We use two prediction lengths, $\delta \in \mathcal{D} = \{2, 4\}$. For a given δ, the first $T - \delta$ temporal positions of the masked representation are used as context resulting in $\mathbf{h}_{\text{ctx}}^{(\delta)}$. The V-JEPA 2.1 predictor P_ψ uses these context features to predict the representations of the final δ temporal positions - $\mathbf{h}_P^{(\delta)}$. Finally, for the same final δ temporal positions, we take the corresponding features from the non-masked encoder representation to generate $\mathbf{h}_E^{(\delta)}$. These are represented in Eq. 1 - 3 below.

$$\mathbf{h}_{\text{ctx}}^{(\delta)} = \mathbf{H}_{\text{mask}}[0 : T - \delta] \in \mathbb{R}^{(T-\delta)S\times 1024}, \tag{1}$$

$$\mathbf{h}_P^{(\delta)} = P_\psi\left(\mathbf{h}_{\text{ctx}}^{(\delta)}\right) \in \mathbb{R}^{\delta S\times 1664}, \tag{2}$$

$$\mathbf{h}_E^{(\delta)} = \mathbf{H}_{\text{full}}[T - \delta : T] \in \mathbb{R}^{\delta S\times 1024}. \tag{3}$$

Thus, $\mathbf{h}_P^{(\delta)}$ is the representation predicted from the masked context, while $\mathbf{h}_E^{(\delta)}$ is the representation of the same temporal positions obtained from the non-masked encoder input. The predictor output is 1664-dimensional, while the corresponding encoder representation is 1024-dimensional. We therefore use a low-rank Bridge $\mathcal{B} = B_2(B_1(\mathbf{h}))$ consisting of two linear projections, $B_1 = \text{Linear}(1664, 64)$ and $B_2 = \text{Linear}(64, 1024)$ to map the predictor output to the encoder feature space. The Bridge

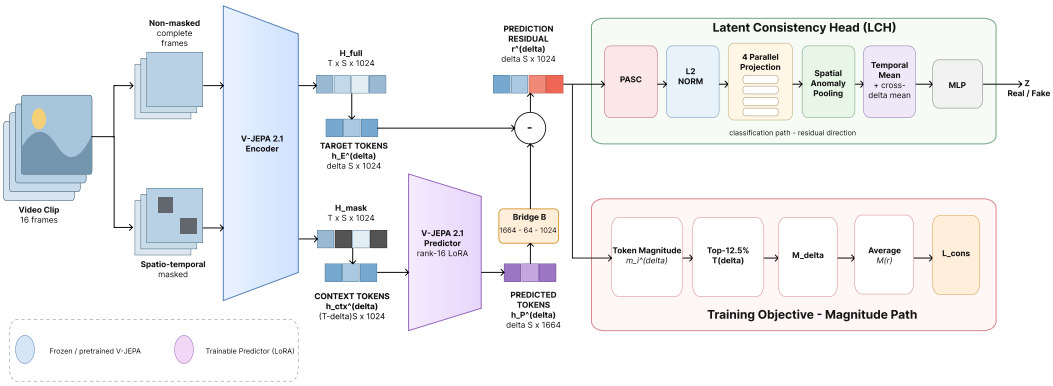

Figure 2: **Overview of ConsistencyNet.** V-JEPA 2.1 constructs prediction residuals for prediction lengths $\delta \in \{2, 4\}$. LCH processes the residual using PASC, ℓ_2 normalization, parallel projections, SAP, and temporal aggregation before real/fake classification. Residual magnitude is used separately by the consistency-margin objective.

contains 0.17 M trainable parameters. The prediction residual is then computed as defined in Eq. 4.

$$\mathbf{r}^{(\delta)} = \mathcal{B}\left(\mathbf{h}_P^{(\delta)}\right) - \mathrm{sg}\left(\mathbf{h}_E^{(\delta)}\right) \in \mathbb{R}^{\delta S \times 1024}, \tag{4}$$

where $\mathrm{sg}(\cdot)$ denotes stop-gradient. The residual therefore measures the difference between what the predictor estimates from the masked context and what the encoder represents at the corresponding positions in the complete video. A qualitative example of prediction residual is provided in Appendix C

3.2 Architecture Overview

Fig. 2 shows the overall architecture of ConsistencyNet. We use the V-JEPA 2.1 encoder and predictor to construct the prediction residual defined in Section 3.1. The encoder has 304 M parameters, 24 transformer blocks, and an output dimension of 1024. The predictor has 21.9 M base parameters and 12 transformer blocks. The prediction residual $\mathbf{r}^{(\delta)}$ is the input to our *Latent Consistency Head (LCH)*, which produces the real/fake prediction. For each prediction length $\delta \in \mathcal{D} = \{2, 4\}$, LCH applies Predictor Attention-conditioned Stop-gradient Centering (PASC) and ℓ_2 normalization, followed by four parallel projection heads and Spatial Anomaly Pooling (SAP). The resulting features are aggregated over the predicted temporal positions and then across the two prediction lengths before being passed to the final classifier. The raw residual magnitude is handled separately by the training objective described in Section 3.3.

Predictor Attention-conditioned Stop-gradient Centering (PASC). The predictor may rely on different amounts of visible context when predicting different target tokens. Centering all residual tokens with a single mean can therefore mix tokens that were predicted using different amounts of context information. PASC addresses this by grouping target tokens according to their attention to the visible context and centering the two groups separately.

We use attention from block 6 of the predictor. Let $A_{i,j}^{(6)}$ denote the attention from target token i to token j, averaged over attention heads. We define the context-attention ratio for target token i as $c_i = \sum_{j \in \mathrm{ctx}} A_{i,j}^{(6)} / \sum_j A_{i,j}^{(6)}$. Here, ctx denotes the visible context tokens. A larger c_i indicates that the prediction for token i relies more strongly on the visible context.

For each clip, we split the target tokens at the median context-attention ratio as $\mathcal{H} = \{i : c_i > \mathrm{median}(c)\}$ and $\mathcal{L} = \{i : c_i \leq \mathrm{median}(c)\}$, where $\mathcal{H}$ and $\mathcal{L}$ denote the high- and low-context

groups. Each residual token is then centered using the mean residual of its corresponding group, i.e.,

$$\tilde{\mathbf{r}}_i^{(\delta)} = \mathbf{r}_i^{(\delta)} - \frac{1}{|\mathcal{G}(i)|} \sum_{j \in \mathcal{G}(i)} \mathbf{r}_j^{(\delta)}, \qquad \mathcal{G}(i) \in \{\mathcal{H}, \mathcal{L}\}. \tag{5}$$

The grouping is label-free, and the attention used to define the groups is detached from the gradient; gradients still flow through the centered residuals. PASC adds no trainable parameters. At inference, the attention hook is removed and group centering is replaced by a single per-frame spatial mean.

Residual direction. After PASC, we normalize each centered residual token, i.e., $\hat{\mathbf{r}}_i^{(\delta)} = \tilde{\mathbf{r}}_i^{(\delta)} / \max\Big(\|\tilde{\mathbf{r}}_i^{(\delta)}\|_2, \sqrt{\varepsilon}\Big)$, where $\varepsilon = 10^{-12}$. This removes the raw residual magnitude from the classification path, so the remaining LCH components operate on the normalized residual direction.

Prediction-length-specific projections. For each prediction length $\delta \in \mathcal{D}$, we use $K = 4$ parallel heads to project the normalized residual from 1024 to 768 dimensions,

$$\mathbf{e}_i^{(\delta,k)} = \mathrm{LN}\Big(W_\delta^{(k)} \hat{\mathbf{r}}_i^{(\delta)} + \mathbf{b}_\delta^{(k)}\Big), \quad W_\delta^{(k)} \in \mathbb{R}^{768 \times 1024}, \quad k = 1, \ldots, K. \tag{6}$$

A learned softmax gate combines the four outputs,

$$\gamma_{\delta,1:K} = \mathrm{softmax}\Big(g_{\mathrm{head}}(\mathbf{e}^{(\delta,1:K)})\Big), \qquad \mathbf{e}_i^{(\delta)} = \sum_{k=1}^{K} \gamma_{\delta,k} \mathbf{e}_i^{(\delta,k)} \in \mathbb{R}^{768}. \tag{7}$$

Spatial Anomaly Pooling (SAP). For prediction length δ, each of the δ predicted temporal positions contains $S = 576$ projected spatial-token features. SAP normalizes these spatial features within each sample and uses a learned softmax gate to give more weight to informative spatial locations. It reduces the spatial dimension as $[B, \delta S, 768] \longrightarrow [B, \delta, 768]$. We denote the pooled representation at predicted temporal position t by $\mathbf{p}_t^{(\delta)}$.

Temporal and cross-prediction-length aggregation. For each prediction length, we first average the SAP features over its δ predicted temporal positions. We then average the resulting representations across the two prediction lengths,

$$\mathbf{F}_\delta = \frac{1}{\delta} \sum_{t=1}^{\delta} \mathbf{p}_t^{(\delta)} \in \mathbb{R}^{768}, \qquad \mathbf{F} = \frac{1}{|\mathcal{D}|} \sum_{\delta \in \mathcal{D}} \mathbf{F}_\delta, \qquad \mathcal{D} = \{2, 4\}. \tag{8}$$

Classifier. Finally, a two-layer MLP ($768 \to 384 \to 2$) maps the combined representation $\mathbf{F}$ to the real/fake logits $\mathbf{z} = \mathrm{MLP}(\mathbf{F})$. Detailed implementation information is provided in Appendix D.

3.3 Training Objective

We train ConsistencyNet using two losses. The consistency-margin loss acts on the raw prediction residual, while the classification loss supervises the real/fake prediction from LCH.

Consistency-margin prior. As introduced in Section 1, we hypothesize that manipulated clips produce larger prediction residuals than authentic clips on average, $\mathbb{E}[\|\mathbf{r}_{\mathrm{fake}}\|_2] > \mathbb{E}[\|\mathbf{r}_{\mathrm{real}}\|_2]$.

Since V-JEPA is trained to predict natural-video representations, manipulations may increase the mismatch between predicted and observed features. We therefore treat it as an empirical training prior, not a property guaranteed for every clip. Further explanation is provided in Appendix E.

Residual magnitude score. To use this prior during training, we compute the raw magnitude of each residual token as $m_i^{(\delta)} = \|\mathbf{r}_i^{(\delta)}\|_2$, where i indexes a residual token for prediction length δ.

Prediction errors may be concentrated in only a subset of spatial locations. For each prediction length δ, we therefore select the largest 12.5% of the token magnitudes and average them,

$$M_\delta\left(\mathbf{r}^{(\delta)}\right) = \frac{1}{|\mathcal{T}_\delta|} \sum_{i \in \mathcal{T}_\delta} m_i^{(\delta)}, \qquad \mathcal{T}_\delta = \mathrm{TopK}_{12.5\%}\left(\{m_i^{(\delta)}\}\right). \tag{9}$$

We then average the magnitude scores across the two prediction lengths,

$$M(\mathbf{r}) = \frac{1}{|\mathcal{D}|} \sum_{\delta \in \mathcal{D}} M_\delta\left(\mathbf{r}^{(\delta)}\right), \qquad \mathcal{D} = \{2, 4\}. \tag{10}$$

A larger $M(\mathbf{r})$ indicates a stronger prediction mismatch in the locations with the largest residuals.

Consistency-margin loss. Using the magnitude score above, we encourage manipulated clips to have a larger residual magnitude than authentic clips,

$$\mathcal{L}_{\mathrm{cons}} = \max(0,\, m + M(\mathbf{r}_{\mathrm{real}}) - M(\mathbf{r}_{\mathrm{fake}})), \qquad m = 2.0. \tag{11}$$

The loss is active when the residual magnitude of a manipulated clip is not larger than that of an authentic clip by at least the margin m.

Classification loss. We train the LCH output using weighted cross-entropy, $\mathcal{L}_{\mathrm{wCE}} = -\frac{1}{N}\sum_{n=1}^{N} w_{y_n} \log p(y_n \mid \mathcal{V}_n)$, to account for the class imbalance in FF++, where y_n is the ground-truth label and w_{y_n} is its class weight.

Total objective. The final training objective combines the classification and consistency losses,

$$\mathcal{L} = \mathcal{L}_{\mathrm{wCE}} + \lambda \mathcal{L}_{\mathrm{cons}}, \qquad \lambda = 0.1. \tag{12}$$

Thus, $\mathcal{L}_{\mathrm{wCE}}$ supervises the real/fake classifier, while $\mathcal{L}_{\mathrm{cons}}$ acts on the raw prediction residual.

3.4 Sign-Flip Identification Protocol (SFIP)

We use the Sign-Flip Identification Protocol (SFIP) to test whether the direction of the consistency margin affects cross-dataset generalization while keeping the remaining training setup fixed.

Standard and reversed objectives. The standard objective encourages manipulated clips to have a larger residual magnitude than authentic clips, $M(\mathbf{r}_{\mathrm{fake}}) > M(\mathbf{r}_{\mathrm{real}}) + m$. The reversed objective encourages the opposite relation, $M(\mathbf{r}_{\mathrm{real}}) > M(\mathbf{r}_{\mathrm{fake}}) + m$.

Matched in-domain performance. To control for in-domain differences, we match the standard and reversed settings on FF++ validation AUC before cross-dataset evaluation, yielding models with similar in-domain performance but opposite margin directions.

Cross-dataset comparison. For an out-of-domain dataset, we define the generalization gap as

$$G = \mathrm{AUC}_{\mathrm{FF++}} - \mathrm{AUC}_{\mathrm{OOD}}, \tag{13}$$

where $\mathrm{AUC}_{\mathrm{OOD}}$ is the AUC on the unseen dataset. We compare the standard and reversed settings using

$$\Delta G = G_{\mathrm{reverse}} - G_{\mathrm{standard}}. \tag{14}$$

A positive ΔG indicates that the reversed-margin model has a larger cross-dataset generalization gap at matched in-domain performance.

4 EXPERIMENTS AND ANALYSIS

4.1 TRAINING AND EVALUATION

Cross-dataset evaluation. We train ConsistencyNet on FaceForensics++ using 16-frames and evaluate the cross-dataset generalization of our model on five unseen datasets. ConsistencyNet achieves 90.5 AUC on Celeb-DF-v2, 87.1 on DFDC, 90.4 on DFDCP, and 88.2 on WildDeepfake. While individual state-of-the-art methods obtain higher AUC on specific benchmarks, their performance varies considerably across datasets. In contrast, ConsistencyNet maintains strong performance across all four established cross-dataset benchmarks, with AUC consistently above 87. On the more recent DE-2024 benchmark, ConsistencyNet achieves 66.4 AUC, outperforming the evaluated Effort and FSFM checkpoints on the same test split. These results suggest that the proposed prediction-residual representation provides consistent transfer across different manipulation sources and evaluation domains. Full training, preprocessing, and hyperparameter details are provided in Appendix F.

Table 1: **Cross-dataset generalization.** Video-level AUC (%) on unseen datasets. All methods are trained on FF++. For DE-2024*, values unavailable in the original papers are evaluated by us using the authors' released checkpoints and inference pipelines.

Method	Venue	CDF-v2	DFDC	DFDCP	WildDF	DE-2024*
Frame-based						
CDFA (Lin et al., 2024)	ECCV'24	97.2	83.8	**97.0**	84.5	—
Real appearance modelling (Tian et al., 2024)	ECCV'24	95.5	80.2	89.5	88.4	—
RepDFD (Lin et al., 2025)	AAAI'25	89.9	81.0	95.0	88.1	—
FSFM (Wang et al., 2025)	CVPR'25	91.4	83.5	89.7	87.0	65.4
Effort (Yan et al., 2025a)	ICML'25	95.6	84.3	90.9	84.8	65.7
Forensics Adapter (Cui et al., 2025)	CVPR'25	95.7	87.2	92.9	—	62.3
DFF-Adapter (Zhang et al., 2026)	AAAI'26	95.3	**90.0**	91.6	—	—
DFD-HR (Sun et al., 2026)	CVPR'26	96.0	86.5	90.1	90.7	—
DiffusionFF (Peng et al., 2026)	CVPR'26	**97.2**	85.1	92.6	—	—
Query-token forgery purification (Wang et al., 2026)	CVPR'26	96.0	86.9	92.5	**91.7**	—
TSRL (Lei et al., 2026)	CVPR'26	90.1	88.2	92.4	—	—
Video-based						
NACO (Zhang et al., 2024)	ECCV'24	89.5	76.7	—	—	—
Pixel-wise temporal frequency (Kim et al., 2025)	ICCV'25	89.7	75.2	—	—	—
DFD-FCG (Han et al., 2025b)	CVPR'25	95.0	81.8	—	87.2	—
TFCU (Guo et al., 2025)	CVPR'25	93.2	86.1	—	—	—
Video Blending + StA (Yan et al., 2025c)	CVPR'25	94.7	84.3	90.9	84.8	—
FakeRadar (Li et al., 2025)	ICCV'25	91.7	84.1	88.5	—	—
FakeSTormer (Nguyen et al., 2025)	ICCV'25	92.8	75.1	90.2	75.3	57.1
ConsistencyNet (ours)	—	90.5	87.1	90.4	88.2	**66.4**

*For DE-2024, results not reported by the corresponding papers are evaluated by us using the authors' released checkpoints and inference pipelines. ConsistencyNet is evaluated on the official DE-2024 test split.

Table 2: **Cross-manipulation evaluation on DF40 face swap.** Video-level AUC (%) on eight unseen face-swap generators.

Method	Venue	Blend	e4s	FD	FSGAN	IS	MS	SS	UF	Avg.
RECCE (Cao et al., 2022)	CVPR'22	83.2	68.3	84.8	94.9	84.8	92.5	76.8	89.8	84.4
SLADD (Chen et al., 2022)	CVPR'22	88.2	76.5	82.5	94.3	87.9	95.4	79.4	87.8	86.5
ProDet (Cheng et al., 2024)	NeurIPS'24	92.9	77.1	74.7	92.8	83.7	97.5	84.4	90.8	86.7
Effort (Yan et al., 2025a)	ICML'25	87.3	98.3	92.6	95.7	93.6	95.3	92.6	96.2	94.0
Forensics Adapter (Cui et al., 2025)	CVPR'25	87.2	96.3	94.1	96.8	94.6	97.4	91.4	94.2	94.0
TSRL (Lei et al., 2026)	CVPR'26	90.2	98.3	94.8	97.5	93.7	93.3	90.4	95.4	94.2
Video Blending + StA (Yan et al., 2025c)	CVPR'25	90.6	98.0	91.2	96.4	93.7	94.6	93.1	96.0	94.2
DFD-HR (Sun et al., 2026)	CVPR'26	95.6	**99.3**	97.0	98.4	98.1	98.6	97.0	**98.8**	97.8
Query-token forgery purification (Wang et al., 2026)	CVPR'26	96.5	99.1	**97.7**	**98.6**	**98.7**	**99.0**	**97.7**	**98.8**	**98.3**
ConsistencyNet (ours)	—	**97.2**	97.1	94.3	96.4	93.1	95.9	93.7	96.6	95.5

Blend: BlendFace; FD: FaceDancer; IS: InSwapper; MS: MobileSwap; SS: SimSwap; UF: UniFace.

Cross-manipulation evaluation. Finally, we evaluate transfer to eight unseen face-swap generators from DF40(Table 2). ConsistencyNet achieves 95.5 average AUC across eight unseen face-swap generators, with AUC above 93 on every method and the highest performance on BlendFace (97.2). The gap to Query-token forgery purification decreases from 5.5 AUC on Celeb-DFv2 (Table 1) to 2.8 AUC on DF40 on average. This consistent performance across diverse unseen generators supports the transferability of the learned consistency representation beyond the observed training manipulations. during

4.2 Analysis of Learned Consistency Representations

From prediction residuals to discriminative representations. To examine how LCH transforms the V-JEPA prediction residual, we visualize video-level representations from the unseen Celeb-DFv2 test set using t-SNE. Fig. 3 compares the raw prediction residual before LCH with the final consistency representation used by the classifier. The raw residuals show substantial overlap between authentic and manipulated videos. After LCH, the representations become more organized along a real-to-forged direction, with substantially greater separation between the two classes. Importantly, Celeb-DFv2 is not used during training, so this visualization illustrates how the learned consistency representation transfers to an unseen dataset.

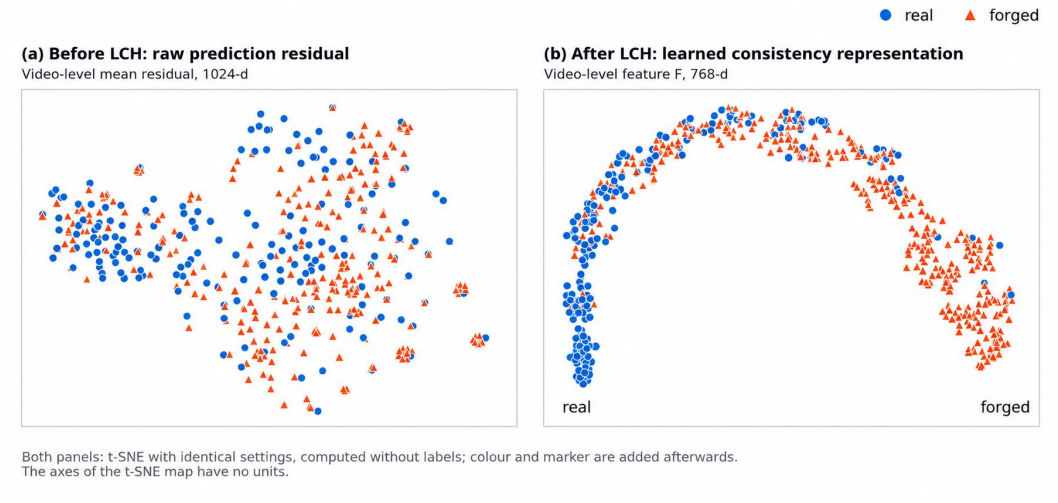

Figure 3: **Prediction residuals before and after LCH on unseen Celeb-DFv2 videos.** t-SNE visualization of (a) raw video-level prediction residuals and (b) the final consistency representations used for classification. Real and forged samples largely overlap before LCH, while the learned representations show clearer class separation after LCH.

4.3 Ablation Study

Training objective. We first study how the consistency prior is imposed. Replacing the pairwise margin with random groups (A1) reduces the average AUC by 2.5%, despite improving Celeb-DF-v2 from 90.5 to 91.2. Applying the prior only to class means (A2) gives a larger drop of 3.0%. Reversing the margin to enforce higher residuals for real clips than forged clips (A3) reduces the average by 2.2%. These results support applying the prior at the pair level with forged clips above real clips.

Architecture. We next examine the role of the residual and predictor. Removing the margin loss (B1) reduces the average AUC by 4.3%, showing that the consistency constraint is important for cross-dataset transfer. Without the margin, using predicted features directly (B2) performs better than using residuals, but remains 2.0% below the full model. Freezing the predictor (B3) causes the largest degradation, with the average AUC dropping from 89.0 to 76.6. This indicates that adapting the predictor together with the residual-based objective is important for generalization.

Tail fraction. Finally, we vary the fraction of largest residual norms used to compute the clip-level consistency score. The full model uses the largest 12.5%. Reducing this fraction to 6.25% (C1) lowers the average AUC by 2.9%, while increasing it to 25% (C2) lowers AUC by 2.6%. Thus, focusing on the largest 12.5% provides the best average performance among the tested settings.

Margin. The full model uses $m = 2$ and $\lambda = 0.1$. Reducing the margin to $m = 1$ results in a 2.4% AUC drop, while increasing it to $m = 4$ or increasing the margin weight to $\lambda = 0.2$ results in a 2.2% drop. The default setting therefore provides the strongest cross-dataset performance.

Prediction lengths. The full model uses prediction lengths $\delta \in \{2, 4\}$. Using only $\delta = 2$ results in a 2.0% AUC drop, while using only $\delta = 4$ reduces AUC by 2.7%. Adding the longer prediction length $\delta = 6$ results in the largest drop of 3.0%. This suggests that combining $\delta = 2$ and $\delta = 4$ provides a better balance between available context and prediction length.

Table 3: **Ablation study.** Video-level AUC (%) on four unseen datasets. Avg. is the mean across datasets and Δ is the change from the full model.

#	Variant	CDF-v2	DFDC	DFDCP	WildDF	Avg.	Δ
—	**ConsistencyNet**	90.5	87.1	90.4	88.2	**89.0**	—
	A. Training objective						
A1	No prior: random groups	**91.2**	82.1	89.1	83.7	86.5	−2.5
A2	Prior on class means	86.0	85.2	87.7	84.9	86.0	−3.0
A3	Reversed prior: real above forged	88.8	84.7	88.5	85.2	86.8	−2.2
	B. Architecture						
B1	No margin ($\lambda = 0$)	81.6	82.7	**90.7**	83.7	84.7	−4.3
B2	Prediction instead of residual	86.5	85.4	88.8	87.4	87.0	−2.0
B3	Frozen predictor	81.5	75.0	81.2	68.6	76.6	−12.4
	C. Tail fraction						
C1	Largest 6.25%	86.2	83.5	88.5	86.2	86.1	−2.9
C2	Largest 25%	88.4	85.8	84.8	86.4	86.4	−2.6
	D. Margin						
D1	Margin $m = 1$	87.0	84.3	88.5	86.7	86.6	−2.4
D2	Margin $m = 4$	88.7	84.9	87.1	86.5	86.8	−2.2
D3	Margin weight $\lambda = 0.2$	87.0	84.0	88.8	87.5	86.8	−2.2
	E. Prediction lengths						
E1	One length, $\delta \in \{2\}$	88.5	84.0	89.0	86.5	87.0	−2.0
E2	One length, $\delta \in \{4\}$	86.5	84.0	88.5	86.0	86.3	−2.7
E3	Three lengths, $\delta \in \{2, 4, 6\}$	86.5	83.5	88.0	86.0	86.0	−3.0

5 Discussion and Limitations

Our results suggest that prediction residuals from a self-supervised video predictor provide a useful signal for cross-dataset deepfake detection. The ablations and matched sign-flip experiments further show how these residuals are used during training matters for generalization. Overall, the results demonstrate strong and consistent performance across diverse datasets and evaluation settings, supporting the ability of prediction-based consistency cues to generalize across manipulation domains.

Limitations. Our system is visual-only and operates on face-cropped videos. We do not evaluate audio-visual consistency, adversarial robustness, or demographic fairness. The method also requires a predictive video backbone, which limits direct comparison with encoder-only foundation models. Future work could extend this study to other predictive video backbones, emerging deepfake benchmarks, and more comprehensive evaluations of robustness and fairness.

AI USE STATEMENT

We used Large Language Models (LLMs) solely to assist with language polishing and refinement during the preparation of this manuscript. Their use was limited to improving grammar, sentence structure, and readability. All research ideas, methodology, experiments, data analysis, results, and conclusions are the original work of the authors. The authors take full responsibility for the accuracy and integrity of the content presented in this paper.

REFERENCES

Adrien Bardes, Mahmoud Assran, et al. V-JEPA 2.1: Distilled video world models from a giant teacher. *arXiv preprint arXiv:2603.14482*, 2026. Distilled-from-Giant 384 px ViT-L variant.

Lalith Bharadwaj Baru, Rohit Boddeda, Shilhora Akshay Patel, and Sai Mohan Gajapaka. Wavelet-driven generalizable framework for deepfake face forgery detection. In *WACV Workshops*, 2025.

Junyi Cao, Chao Ma, Taiping Yao, Shen Chen, Shouhong Ding, and Xiaokang Yang. End-to-end reconstruction-classification learning for face forgery detection. In *CVPR*, 2022.

Zhongyi Cao and Andy J. Ma. Towards generalized face anti-spoofing from a frequency shortcut view. In *WACV*, 2025.

Nuria Alina Chandra, Ryan Murtfeldt, Lin Qiu, Arnab Karmakar, Hannah Lee, Emmanuel Tanumihardja, Kevin Farhat, Ben Caffee, Sejin Paik, Changyeon Lee, Jongwook Choi, Aerin Kim, and Oren Etzioni. Deepfake-Eval-2024: A multi-modal in-the-wild benchmark of deepfakes circulated in 2024. *arXiv preprint arXiv:2503.02857*, 2025.

Liang Chen, Yong Zhang, Yibing Song, Lingqiao Liu, and Jue Wang. Self-supervised learning of adversarial example: Towards good generalizations for deepfake detection. In *CVPR*, pp. 18710–18719, 2022.

Jikang Cheng, Zhiyuan Yan, Ying Zhang, Yuhao Luo, Zhongyuan Wang, and Chen Li. Can we leave deepfake data behind in training deepfake detector? In *NeurIPS*, 2024.

François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, pp. 1251–1258, 2017.

Xinjie Cui, Yuezun Li, Ang Pu, Haipeng Hu, and Siwei Lyu. Forensics adapter: Adapting CLIP for generalizable face forgery detection. In *CVPR*, 2025.

Brian Dolhansky, Russel Howes, Bradley Pflaum, Nicole Baram, and Cristian Ferrer. The DeepFake detection challenge dataset. *arXiv preprint arXiv:2006.07397*, 2020.

Shichao Dong, Jin Wang, Renhe Ji, Jiajun Liang, Haoqiang Fan, and Zheng Ge. Implicit identity leakage: The stumbling block to improving deepfake detection generalization. In *CVPR*, 2023.

Yufeng Guo et al. Face forgery video detection via temporal forgery cue unraveling. In *CVPR*, 2025.

Alexandros Haliassos, Rodrigo Mira, Stavros Petridis, and Maja Pantic. Leveraging real talking faces via self-supervision for robust forgery detection. In *CVPR*, 2022.

Alexandros Haliassos et al. Lips don't lie: A generalisable and robust approach to face forgery detection. In *CVPR*, 2021.

Yue-Hua Han, Tai-Ming Huang, Kai-Lung Hua, and Jun-Cheng Chen. Towards more general video-based deepfake detection through facial component guided adaptation for foundation model. In *CVPR*, 2025a.

Yue-Hua Han, Tai-Ming Huang, Kai-Lung Hua, and Jun-Cheng Chen. Towards more general video-based deepfake detection through facial component guided adaptation for foundation model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22995–23005, 2025b.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022.

Yiwen Hu, Junyi Chen, and Tianlin Wang. PLADA: Progressive label-aware domain adaptation for cross-domain deepfake detection. *arXiv preprint*, 2025.

Hasam Khalid and Simon S. Woo. OC-FakeDect: Classifying deepfakes using one-class variational autoencoder. In *CVPR Workshops*, 2020.

Taehoon Kim, Jongwook Choi, Yonghyun Jeong, Haeun Noh, Jaejun Yoo, Seungryul Baek, and Jongwon Choi. Beyond spatial frequency: Pixel-wise temporal frequency-based deepfake video detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11198–11207, 2025.

Zhanhe Lei, Zhongyuan Wang, Jikang Cheng, Baojin Huang, Yuhong Yang, Zhen Han, Chao Liang, and Dengpan Ye. Tutor-student reinforcement learning: A dynamic curriculum for robust deepfake detection. In *CVPR*, 2026.

Hanzhe Li, Jiaran Zhou, Yuezun Li, Baoyuan Wu, Bin Li, and Junyu Dong. FreqBlender: Enhancing deepfake detection by blending frequency knowledge. In *NeurIPS*, 2024.

Yuezun Li et al. Celeb-DF: A large-scale challenging dataset for deepfake forensics. In *CVPR*, 2020.

Zhaolun Li, Jichang Li, Yinqi Cai, Junye Chen, Xiaonan Luo, Guanbin Li, and Rushi Lan. FakeRadar: Probing forgery outliers to detect unknown deepfake videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13382–13392, 2025.

Kaiqing Lin, Yuzhen Lin, Weixiang Li, Taiping Yao, and Bin Li. Standing on the shoulders of giants: Reprogramming visual-language model for general deepfake detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 5262–5270, 2025. doi: 10.1609/aaai.v39i5.32559.

Yuzhen Lin, Wentang Song, Bin Li, Yuezun Li, Jiangqun Ni, Han Chen, and Qiushi Li. Fake it till you make it: Curricular dynamic forgery augmentations towards general deepfake detection. In *ECCV*, pp. 87–103, 2024.

Dat Nguyen, Marcella Astrid, Anis Kacem, Enjie Ghorbel, and Djamila Aouada. Vulnerability-aware spatio-temporal learning for generalizable deepfake video detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10786–10796, 2025.

Dat Nguyen et al. LAA-Net: Localized artifact attention network for quality-agnostic and generalizable deepfake detection. In *CVPR*, 2024.

NTIRE Workshop. NTIRE 2026 deepfake detection challenge, 2026. Deepfake Detection Challenge at CVPR 2026 Workshops.

Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models. In *CVPR*, 2023.

Siran Peng, Haoyuan Zhang, Li Gao, Tianshuo Zhang, Xiangyu Zhu, Bao Li, Weisong Zhao, and Zhen Lei. DiffusionFF: A diffusion-based framework for joint face forgery detection and fine-grained artifact localization. In *CVPR*, 2026.

Yuyang Qian et al. Thinking in frequency: Face forgery detection by mining frequency-aware clues. In *ECCV*, 2020.

Andreas Rössler et al. FaceForensics++: Learning to detect manipulated facial images. In *ICCV*, 2019.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *CVPR*, pp. 815–823, 2015.

Rui Shao, Tianxing Wu, Liqiang Nie, and Ziwei Liu. DeepFake-Adapter: Dual-level adapter for deepfake detection. *International Journal of Computer Vision*, 133(6):3613–3628, 2025. doi: 10.1007/s11263-024-02274-6.

Kaede Shiohara and Toshihiko Yamasaki. Detecting deepfakes with self-blended images. In *CVPR*, 2022.

Jiamu Sun, Zhiyuan Yan, Ke-Yue Zhang, Taiping Yao, and Shouhong Ding. DFD-HR: Generalizable deepfake detection via hierarchical routing learning. In *CVPR*, pp. 13984–13995, 2026.

Ke Sun, Shen Chen, Taiping Yao, Hong Liu, Xiaoshuai Sun, Shouhong Ding, and Rongrong Ji. DiffusionFake: Enhancing generalization in deepfake detection via guided stable diffusion. In *NeurIPS*, 2024.

Chuangchuang Tan, Huan Liu, Yao Zhao, Shikui Wei, Guanghua Gu, Ping Liu, and Yunchao Wei. Rethinking the up-sampling operations in CNN-based generative network for generalizable deepfake detection. In *CVPR*, 2024a.

Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu, Ping Liu, and Yunchao Wei. Frequency-aware deepfake detection: Improving generalizability through frequency space domain learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 5052–5060, 2024b. doi: 10.1609/aaai.v38i5.28310.

Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu, Ping Liu, and Yunchao Wei. LNCLIP-DF: Layer normalization tuning for cross-dataset deepfake detection. In *CVPR*, 2025.

Jiahe Tian, Cai Yu, Xi Wang, Peng Chen, Zihao Xiao, Jiao Dai, Yesheng Chai, and Jizhong Han. Real appearance modeling for more general deepfake detection. In *ECCV*, pp. 402–419, 2024.

Changshuo Wang, Jiangming Wang, Ke-Yue Zhang, Taiping Yao, Shouhong Ding, Shunli Wang, Ran Yi, and Lizhuang Ma. Beyond [CLS] token: Query-driven token-level forgery purification for generalizable deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 42922–42931, 2026.

Feng Wang, Xiang Xiang, Jian Cheng, and Alan L. Yuille. NormFace: L2 hypersphere embedding for face verification. In *ACM Multimedia*, 2017.

Gaojian Wang, Feng Lin, Tong Wu, Zhenguang Liu, Zhongjie Ba, and Kui Lu. FSFM: Foundation model for unified face spoofing and forgery detection. In *CVPR*, 2025.

Junke Wang et al. Comprehensive self-consistency learning for deepfake detection. *arXiv preprint*, 2022.

Zhendong Wang, Jianmin Bao, Wengang Zhou, Weilun Wang, Hezhen Hu, Hong Chen, and Houqiang Li. DIRE for diffusion-generated image detection. In *ICCV*, 2023a.

Zhendong Wang et al. AltFreezing for more general video face forgery detection. In *CVPR*, 2023b.

Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, 2018.

Zhiyuan Yan, Taiping Yao, Shen Chen, Yandan Zhao, Xinghe Fu, Junwei Zhu, Donghao Luo, Chengjie Wang, Shouhong Ding, Yunsheng Wu, and Li Yuan. DF40: Toward next-generation deepfake detection. In *Advances in Neural Information Processing Systems*, volume 37, 2024. Datasets and Benchmarks Track.

Zhiyuan Yan, Jiangming Wang, Zhendong Wang, Peng Jin, Ke-Yue Zhang, Shen Chen, Taiping Yao, Shouhong Ding, Baoyuan Wu, and Li Yuan. Effort: Efficient orthogonal modeling for generalizable ai-generated image detection. *Proceedings of the International Conference on Machine Learning*, 2025a.

Zhiyuan Yan, Yandan Zhao, Shen Chen, Mingyi Guo, Xinghe Fu, Taiping Yao, Shouhong Ding, Yunsheng Wu, and Li Yuan. Generalizing deepfake video detection with plug-and-play: Video-level blending and spatiotemporal adapter tuning. In *CVPR*, 2025b.

Zhiyuan Yan, Yandan Zhao, Shen Chen, Mingyi Guo, Xinghe Fu, Taiping Yao, Shouhong Ding, Yunsheng Wu, and Li Yuan. Generalizing deepfake video detection with plug-and-play: Video-level blending and spatiotemporal adapter tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12615–12625, 2025c.

Zhiyuan Yan et al. UCF: Uncovering common features for generalizable deepfake detection. In *ICCV*, 2023.

Zhiyuan Yan et al. GenD: A generalizable deepfake detection benchmark. *arXiv preprint arXiv:2508.06248*, 2025d.

Yongqi Yang, Zhihao Qian, Ye Zhu, Olga Russakovsky, and Yu Wu. D3: Scaling up deepfake detection by learning from discrepancy. In *CVPR*, 2025.

Daichi Zhang, Zihao Xiao, Shikun Li, Fanzhao Lin, Jianmin Li, and Shiming Ge. Learning natural consistency representation for face forgery video detection. In *ECCV*, 2024.

Tianxiang Zhang, Peipeng Yu, Zhihua Xia, Longchen Dai, Xiaoyu Zhou, and Hui Gao. Fine-grained DINO tuning with dual supervision for face forgery detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 12780–12788, 2026. doi: 10.1609/aaai.v40i15.38275.

Tianchen Zhao et al. Learning self-consistency for deepfake detection. In *ICCV*, 2021.

Yinglin Zheng et al. Exploring temporal coherence for more general video face forgery detection. In *ICCV*, 2021.

Bojia Zi et al. WildDeepfake: A challenging real-world dataset for deepfake detection. In *ACM Multimedia*, 2020.

A Appendix

B Extended Related Work

Recent approaches to cross-dataset deepfake detection. Recent work addresses cross-dataset generalization from several complementary directions. D3 (Yang et al., 2025) uses multiple generators together with a discrepancy-learning branch. DiffusionFake (Sun et al., 2024) incorporates features from a frozen diffusion model to improve source–target disentanglement. FreqBlender (Li et al., 2024) constructs pseudo-forgeries through frequency-domain blending. Video-level blending with a spatiotemporal adapter (Plug-and-Play VLB) (Yan et al., 2025b) targets video generalization through augmentation and parameter-efficient adaptation. DFD-FCG (Han et al., 2025a) adapts a pretrained image encoder using facial-component-guided features.

ConsistencyNet approaches the problem from a different perspective. Rather than introducing an external diffusion model, synthesizing additional pseudo-forgeries, or relying primarily on frequency-domain cues, we use the latent prediction residual of a pretrained self-supervised video model. Specifically, the detector compares the V-JEPA predictor output, after dimensional alignment through the Bridge, with the corresponding stop-gradient encoder target.

Consistency-based forensic signals. RealForensics (Haliassos et al., 2022) is related in its use of self-supervised consistency for forgery detection. It learns audio–visual correspondence as a consistency signal, whereas ConsistencyNet is visual-only and derives its signal from disagreement between predicted and target latent video representations. These approaches therefore exploit different forms of consistency and are complementary rather than directly equivalent.

C Visualization of Prediction Residuals

Real and fake prediction residuals. Figure 4 visualizes the prediction residual for an example real clip and an example fake clip. The prediction-error maps show the spatial distribution of the residual magnitude for each clip. The final panel shows the difference between the two maps, computed as forged minus real. Positive values therefore indicate regions where the prediction error is larger for the forged clip.

The example illustrates the consistency-margin prior introduced in Section 3.4: the forged clip exhibits larger prediction errors in several facial regions, producing positive values in the forged-minus-real

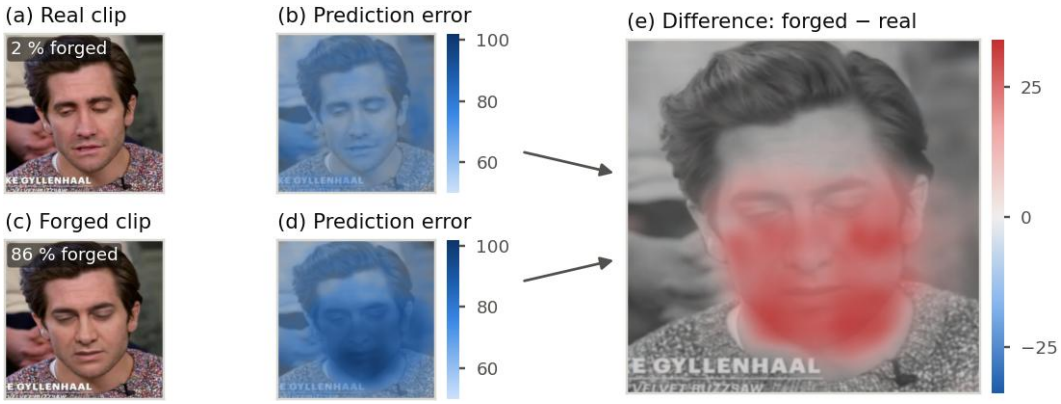

Figure 4: **Visualization of prediction residuals for real and forged clips.** (a) Real clip and (b) its prediction-error map; (c) forged clip and (d) its prediction-error map. (e) Difference between the forged and real prediction-error maps, computed as forged minus real. Positive values indicate regions with larger prediction error for the forged clip.

difference map. This visualization is illustrative and is not intended as a quantitative estimate of the margin across the dataset.

D Implementation Details

This section provides additional implementation details for ConsistencyNet. Unless stated otherwise, the settings below are shared across all experiments. The V-JEPA 2.1 encoder remains frozen. Within the predictor, the transformer base weights remain frozen, while the LoRA adapters and selected input/output components are trainable. The Bridge, Latent Consistency Head (LCH), and classifier are also trained.

Input clips and latent representation. Each input consists of 16 face-aligned RGB frames at 384×384 resolution. V-JEPA 2.1 maps the clip to $T' = 8$ temporal feature positions, with $S = 576$ spatial tokens at each position. The encoder therefore produces

$$\mathbf{H} \in \mathbb{R}^{B \times T' \times S \times 1024}, \qquad T' = 8, \quad S = 576, \tag{15}$$

where B denotes the batch size.

We use prediction lengths $\mathcal{D} = \{2, 4\}$. For prediction length δ, the first $T' - \delta$ temporal positions provide the predictor context and the final δ positions define the prediction target. Thus, each prediction length contains δS target tokens: 1152 tokens for $\delta = 2$ and 2304 tokens for $\delta = 4$.

Frozen encoder and predictor adaptation. The V-JEPA 2.1 encoder contains 24 transformer blocks and produces 1024-dimensional latent features. All encoder parameters remain frozen during training.

The V-JEPA predictor contains 12 transformer blocks and produces 1664-dimensional target predictions. Its transformer base weights remain frozen. We apply rank-16 LoRA to the query–key–value and output projections of every predictor block. The predictor input embedding, output projection, final layer normalization, and mask token are additionally trainable.

The predictor LoRA modules contribute 442,368 trainable parameters.

Low-rank Bridge. The predictor and encoder operate at different feature dimensions: the predictor output is 1664-dimensional, whereas the encoder target is 1024-dimensional. We therefore map the predictor output into the encoder feature space using a two-layer low-rank Bridge,

$$\mathcal{B} = B_2 \circ B_1, \tag{16}$$

with

$$B_1 : \mathbb{R}^{1664} \to \mathbb{R}^{64}, \qquad B_2 : \mathbb{R}^{64} \to \mathbb{R}^{1024}. \tag{17}$$

Both linear layers include biases. The Bridge therefore contains

$$(1664 \times 64 + 64) + (64 \times 1024 + 1024) = 173{,}120 \tag{18}$$

trainable parameters.

The bottleneck is substantially smaller than a direct $1664 \to 1024$ projection, which would require $1.70\,\mathrm{M}$ parameters including bias.

Prediction residual construction. For each prediction length, we subtract the corresponding encoder target from the bridged predictor output,

$$\mathbf{r}^{(\delta)} = \mathcal{B}\left(\mathbf{h}_P^{(\delta)}\right) - \mathrm{sg}\left(\mathbf{h}_E^{(\delta)}\right), \tag{19}$$

where $\mathrm{sg}(\cdot)$ denotes stop-gradient.

The resulting residual tensor has shape

$$\mathbf{r}^{(\delta)} \in \mathbb{R}^{B \times \delta S \times 1024}. \tag{20}$$

Hence,

$$\mathbf{r}^{(2)} \in \mathbb{R}^{B \times 1152 \times 1024}, \qquad \mathbf{r}^{(4)} \in \mathbb{R}^{B \times 2304 \times 1024}. \tag{21}$$

The stop-gradient operation prevents the encoder target from being updated by the detector losses. Gradients therefore update the trainable predictor components, Bridge, and LCH while the pretrained encoder remains fixed.

Predictor Attention-conditioned Stop-gradient Centering. Raw prediction residuals may contain systematic offsets associated with how strongly individual target tokens depend on visible context. PASC uses predictor attention to define two context-dependence groups and centers their residuals separately.

We extract self-attention from predictor block 6. Let $A_{i,j}^{(6,h)}$ denote the attention from target token i to token j in head h. We average over the H_a attention heads,

$$A_{i,j}^{(6)} = \frac{1}{H_a} \sum_{h=1}^{H_a} A_{i,j}^{(6,h)}. \tag{22}$$

For target token i, its context-attention ratio is

$$c_i = \frac{\sum_{j \in \mathrm{ctx}} A_{i,j}^{(6)}}{\sum_j A_{i,j}^{(6)}}. \tag{23}$$

Because the attention weights are post-softmax, c_i measures the fraction of the token's attention mass assigned to visible context tokens.

Within each clip, we compute the median context-attention ratio and define

$$\mathcal{H} = \{i : c_i > \mathrm{median}(c)\}, \qquad \mathcal{L} = \{i : c_i \le \mathrm{median}(c)\}. \tag{24}$$

The two groups correspond to relatively high- and low-context target tokens within the same clip. No real/fake label is used to construct these groups.

For each residual token, PASC subtracts the mean residual of its assigned group:

$$\hat{\mathbf{r}}_i^{(\delta)} = \mathbf{r}_i^{(\delta)} - \boldsymbol{\mu}_{\mathcal{G}(i)}^{(\delta)}, \tag{25}$$

where

$$\boldsymbol{\mu}_g^{(\delta)} = \frac{1}{|g|} \sum_{j \in g} \mathbf{r}_j^{(\delta)}, \qquad g \in \{\mathcal{H}, \mathcal{L}\}. \tag{26}$$

The attention values used to determine group membership are detached from the gradient. The centering operation itself remains differentiable with respect to the residual features. PASC therefore introduces no trainable parameters.

Training and inference centering. PASC requires predictor attention only during training. At inference time, the attention hook is removed. We reshape the residual tokens according to their predicted temporal positions and subtract one spatial mean independently from each predicted frame.

For temporal position t,

$$\boldsymbol{\mu}_t^{(\delta)} = \frac{1}{S}\sum_{s=1}^{S} \mathbf{r}_{t,s}^{(\delta)}, \tag{27}$$

and inference-time centering becomes

$$\tilde{\mathbf{r}}_{t,s}^{(\delta)} = \mathbf{r}_{t,s}^{(\delta)} - \boldsymbol{\mu}_t^{(\delta)}. \tag{28}$$

Thus, predictor attention is not required during deployment while the remaining LCH operations are unchanged.

Magnitude–direction decomposition. After centering, each residual token is separated into magnitude and direction. The normalized direction is

$$\hat{\mathbf{r}}_i^{(\delta)} = \frac{\tilde{\mathbf{r}}_i^{(\delta)}}{\max\left(\|\tilde{\mathbf{r}}_i^{(\delta)}\|_2, \sqrt{\varepsilon}\right)}, \qquad \varepsilon = 10^{-12}. \tag{29}$$

The normalized direction is provided to LCH for classification. Raw residual magnitude is used separately by the consistency-margin objective described in Section 3.3; it is not concatenated with the classification representation.

Prediction-length-specific projection heads. For each prediction length $\delta \in \{2, 4\}$, LCH uses $K = 4$ parallel projection heads. Head k computes

$$\mathbf{e}_i^{(\delta,k)} = \mathrm{LN}\left(W_\delta^{(k)} \hat{\mathbf{r}}_i^{(\delta)} + \mathbf{b}_\delta^{(k)}\right), \tag{30}$$

where

$$W_\delta^{(k)} \in \mathbb{R}^{768\times 1024}, \qquad k = 1, \ldots, 4. \tag{31}$$

Each projection therefore produces a 768-dimensional representation while preserving the token structure. The four heads are instantiated independently for each prediction length.

Projection-head gating. The four projections are combined using a learned softmax gate rather than a uniform average. We use one trainable scalar logit per projection head for each prediction length:

$$\boldsymbol{\gamma}_\delta = \mathrm{softmax}\left(\boldsymbol{a}_\delta\right), \qquad \boldsymbol{a}_\delta \in \mathbb{R}^4. \tag{32}$$

The resulting token representation is

$$\mathbf{e}_i^{(\delta)} = \sum_{k=1}^{4} \gamma_{\delta,k} \mathbf{e}_i^{(\delta,k)} \in \mathbb{R}^{768}. \tag{33}$$

This gate contributes $4 \times 2 = 8$ trainable parameters.

Spatial Anomaly Pooling. For prediction length δ, the projected representation contains δS spatial tokens:

$$\mathbf{E}^{(\delta)} \in \mathbb{R}^{B\times \delta S\times 768}. \tag{34}$$

We reshape it as

$$\mathbf{E}^{(\delta)} \to \mathbb{R}^{B\times \delta\times S\times 768}. \tag{35}$$

SAP first normalizes the spatial-token features within each sample. A learned scalar scoring layer then assigns a score to each spatial token:

$$q_{t,s}^{(\delta)} = \mathbf{w}_{\mathrm{SAP}}^\top \mathbf{e}_{t,s}^{(\delta)} + b_{\mathrm{SAP}}, \tag{36}$$

with $\mathbf{w}_{\mathrm{SAP}} \in \mathbb{R}^{768}$. The spatial weights are obtained by

$$\alpha_{t,s}^{(\delta)} = \mathrm{softmax}_s\left(q_{t,s}^{(\delta)}\right). \tag{37}$$

The pooled representation is

$$\mathbf{p}_t^{(\delta)} = \sum_{s=1}^{S} \alpha_{t,s}^{(\delta)} \mathbf{e}_{t,s}^{(\delta)}. \tag{38}$$

A separate SAP scoring layer is used for each prediction length, giving

$$2(768+1) = 1{,}538 \tag{39}$$

trainable SAP parameters.

SAP therefore maps

$$[B, \delta, S, 768] \longrightarrow [B, \delta, 768]. \tag{40}$$

Temporal aggregation. For each prediction length, SAP outputs one representation for each predicted temporal position. We average these representations:

$$\mathbf{F}_\delta = \frac{1}{\delta} \sum_{t=1}^{\delta} \mathbf{p}_t^{(\delta)} \in \mathbb{R}^{768}. \tag{41}$$

Cross-length aggregation. We combine the two prediction-length representations using a learned softmax gate. One scalar logit is assigned to each prediction length,

$$\boldsymbol{\beta} = \mathrm{softmax}\left(\mathbf{b}_{\mathrm{len}}\right), \qquad \mathbf{b}_{\mathrm{len}} \in \mathbb{R}^2. \tag{42}$$

The final consistency representation is

$$\mathbf{F} = \beta_2 \mathbf{F}_2 + \beta_4 \mathbf{F}_4 \in \mathbb{R}^{768}. \tag{43}$$

The cross-length gate contributes two trainable parameters.

Classifier. The final classifier is a two-layer MLP,

$$768 \to 384 \to 2. \tag{44}$$

Given the aggregated consistency representation $\mathbf{F}$, it produces

$$\mathbf{z} = \mathrm{MLP}(\mathbf{F}) \in \mathbb{R}^2, \tag{45}$$

corresponding to the real and fake logits.

Gradient flow. The encoder target is detached when the prediction residual is constructed, and all pretrained encoder parameters remain frozen. Gradients from the classification objective propagate through the classifier, cross-length aggregation, temporal aggregation, SAP, projection heads, Bridge, and trainable predictor components. The consistency-margin objective acts on raw residual magnitude and propagates through the Bridge and predictor trainable components while leaving the stop-gradient encoder target unchanged.

This separates the pretrained target representation from the task-specific adaptation while allowing the lightweight trainable modules to shape the prediction residual for forensic detection.

Table 4: **Parameter accounting for ConsistencyNet.** Trainable and frozen parameters are counted from the stated architecture. LCH uses four parallel projection heads for each prediction length, together with learned projection-head, SAP, and cross-length gates.

Component	Layer	Trainable	Frozen
Encoder	V-JEPA 2.1 ViT-L (24 blocks)	—	304,680,960
Predictor	Base weights (12 blocks)	—	21,937,664
	Predictor LoRA on query–key–value and output projections (A: 147,456; B: 294,912)	442,368	—
	Input embedding, $1024 \to 384$	393,600	—
	Output projection, $384 \to 1664$	640,640	—
	Final layer normalization and mask token	1,152	—
	Predictor total	*1,477,760*	*21,937,664*
Bridge $\mathcal{B}$	$1664 \to 64 \to 1024$	173,120	—
LCH	PASC / context-conditioned centering	0	—
	ℓ_2 residual normalization	0	—
	Four parallel projection heads per prediction length, $1024 \to 768$, with LayerNorm	6,309,888	—
	Learned projection-head gate	8	—
	Spatial Anomaly Pooling, including learned spatial gate	1,538	—
	Temporal aggregation	0	—
	LCH total, two prediction lengths	*6,311,434*	—
Cross-length aggregation	Learned two-way softmax gate	2	—
Classifier	MLP $768 \to 384 \to 2$	296,066	—
Total	**334,877,006 parameters**	**8,258,382 (2.47%)**	**326,618,624**

End-to-end tensor shapes. For clarity, the principal transformations for one clip are

$$\text{encoder features} : [B, 8, 576, 1024], \tag{46}$$
$$\text{predictor target features} : [B, \delta \times 576, 1664], \tag{47}$$
$$\text{bridged prediction} : [B, \delta \times 576, 1024], \tag{48}$$
$$\text{prediction residual} : [B, \delta \times 576, 1024], \tag{49}$$
$$\text{parallel projections} : [B, 4, \delta \times 576, 768], \tag{50}$$
$$\text{head-aggregated features} : [B, \delta \times 576, 768], \tag{51}$$
$$\text{SAP output} : [B, \delta, 768], \tag{52}$$
$$\text{temporal aggregation} : [B, 768], \tag{53}$$
$$\text{cross-length aggregation} : [B, 768], \tag{54}$$
$$\text{classifier output} : [B, 2]. \tag{55}$$

For $\delta = 2$, the residual contains 1152 target tokens; for $\delta = 4$, it contains 2304 target tokens.

Parameter accounting. Table 4 reports the parameter count of each component in the instantiated architecture. LCH uses four parallel projection heads for each prediction length. The learned projection-head, SAP, and cross-length gates are included explicitly in the trainable parameter count.

E From V-JEPA Prediction to the Consistency-Margin Prior

V-JEPA learns to predict masked latent representations from visible spatio-temporal context. Abstractly, its self-supervised prediction objective can be written as

$$\mathcal{L}_{\text{JEPA}}(v) = \mathbb{E}_{m \sim \mathcal{M}} \left[\left\| P_\psi(E_\phi(v \odot m), m) - \text{sg}(E_\phi(v))_m \right\|_2^2 \right], \tag{56}$$

where E_ϕ denotes the encoder, P_ψ the predictor, $\mathcal{M}$ the masking distribution, m a sampled mask, and $\text{sg}(\cdot)$ the stop-gradient operator. The predictor is therefore trained to estimate latent target representations from visible context rather than reconstructing pixels directly.

ConsistencyNet uses the discrepancy between the predicted and target latent representations as a forensic signal. For prediction length δ, we compute

$$\mathbf{r}^{(\delta)} = \mathcal{B}\Big(\mathbf{h}_P^{(\delta)}\Big) - \mathrm{sg}\Big(\mathbf{h}_E^{(\delta)}\Big), \tag{57}$$

where $\mathcal{B}$ aligns the predictor representation with the encoder feature space. The residual therefore measures the agreement between a target representation predicted from visible context and the corresponding encoder representation.

Consistency-margin prior. As introduced in Section 1, we hypothesize that fake videos produce larger prediction residuals than real videos on average, $\mathbb{E}[\|\mathbf{r}_{\mathrm{fake}}\|_2] > \mathbb{E}[\|\mathbf{r}_{\mathrm{real}}\|_2]$. This relation is used as an empirical training prior rather than a guaranteed property of V-JEPA. Residual magnitude can also depend on video content, compression, sampling, manipulation type, and dataset shift. We therefore do not require the ordering to hold for every individual clip.

This distinction motivates the Sign-Flip Identification Protocol (SFIP) in Section 3.4. SFIP reverses the direction of the margin while keeping the remaining training setup fixed and matches the two settings on FF++ validation AUC before cross-dataset evaluation. Across the evaluated cross-dataset benchmarks, the reversed-margin setting exhibits a larger generalization gap on average, providing empirical evidence that the direction of the consistency margin is relevant to cross-dataset transfer in our training setup.

F Training and Evaluation Details

Training and in-domain evaluation. We train ConsistencyNet only on FaceForensics++ (FF++). Each training sample consists of a 16-frame face clip. Faces are detected and aligned before being processed by the V-JEPA input pipeline. FF++ validation is used for in-domain evaluation and to match the standard and reversed margin settings in SFIP.

Cross-dataset evaluation. We evaluate the FF++-trained model without further fine-tuning or adaptation on Celeb-DFv2 (Li et al., 2020), DFDC (Dolhansky et al., 2020), DFDCP, WildDeepfake (Zi et al., 2020), and Deepfake-Eval-2024. No samples or labels from these datasets are used during training. We report video-level AUC (%), obtained by aggregating predictions from 16-frame clips into a video-level score.

Inference protocol. At inference time, predictor-attention extraction for PASC is disabled, and the prediction residual is centered using a per-frame spatial mean. All remaining LCH operations are unchanged. Clip-level predictions are aggregated to obtain the final video-level score.

Training hyperparameters. Table 5 summarizes the model, objective, optimization, data, and compute settings used in the main experiments.

LoRA initialization. For each adapted predictor projection, LoRA computes

$$\mathbf{y} = W\mathbf{x} + \frac{\alpha}{r} BA\mathbf{x}, \tag{58}$$

where the pretrained weight W remains frozen. We use rank $r = 16$ and $\alpha = 16$, giving $\alpha/r = 1$.

Following the reference LoRA initialization (Hu et al., 2022), A is initialized with Kaiming-uniform weights. We initialize $B \sim \mathcal{N}(0, 10^{-6})$ rather than setting B exactly to zero. With $B = 0$, the initial LoRA update BA is zero and the gradient with respect to A is also zero at the first optimization step; consequently, only B is updated initially. The choice of which LoRA factor is initialized at zero can affect fine-tuning dynamics.

Our small non-zero initialization allows gradients to reach both factors from the first optimization step while keeping the initial LoRA perturbation negligible relative to the pretrained weights. For the query–key–value projection of the first predictor block, the initial product BA has a standard deviation of approximately 1.2×10^{-7}, compared with 6.0×10^{-2} for the corresponding frozen weights.

Table 5: **Training hyperparameters of ConsistencyNet.** The same configuration is used across all reported runs.

Group	Hyperparameter	Value
Model	Backbone	V-JEPA 2.1 ViT-L/16 (Bardes et al., 2026), distilled from ViT-G; encoder frozen
	Input	16-frame face clip, 384×384; tubelet size 2, yielding $T' = 8$ temporal positions with 576 spatial tokens per position
	Prediction lengths δ	$\{2, 4\}$
	Predictor adaptation	LoRA on the query–key–value and output projections of all 12 predictor blocks; rank $r = 16$, $\alpha = 16$
	LoRA initialization	A: Kaiming-uniform; $B \sim \mathcal{N}(0, 10^{-6})$
	Predictor layers trained in full	Input embedding, output projection, final layer normalization, and mask token
	Bridge $\mathcal{B}$	Linear $1664 \to 64 \to 1024$
	Classifier	MLP $768 \to 384 \to 2$
Objective	Loss	Class-weighted cross-entropy $+\lambda \mathcal{L}_{\text{margin}}$
	Cross-entropy class weights	Real : fake $= 3 : 1$
	Label smoothing	0.1
	Margin	Pairwise; $m = 2.0$, $\lambda = 0.1$
	Tail fraction of n	12.5%
Optimization	Optimizer	Adam; $\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$
	Learning rate	1×10^{-4}; 5 warm-up epochs followed by cosine decay to 10^{-6}
	Weight decay	5×10^{-4}
	Gradient clipping	Global norm 1.0
	Epochs / steps	50 / 4,600
	Batch	384 clips (16 GPUs $\times$ 24 clips/GPU)
	Precision	BF16 mixed precision
	Checkpoint	Final epoch; no checkpoint selection
Data	Training set	FaceForensics++
	Augmentation	Clip-consistent augmentation: random crop with scale $[0.85, 1.0]$ and center shift $\pm 10\%$; horizontal flip ($p = 0.5$); affine transformation with scale $[0.85, 1.15]$, rotation $\pm 5^\circ$, translation $\pm 5\%$, and shear $\pm 3^\circ$ ($p = 0.5$); one of color, gamma, brightness–contrast, or hue–saturation jitter ($p = 0.5$); illumination gradient ($p = 0.5$); Gaussian or motion blur ($p = 0.2$); JPEG compression with quality $[20, 100]$ ($p = 0.5$)
	Normalization	ImageNet mean and standard deviation
Compute	Hardware	2 nodes $\times$ 8 NVIDIA A100 80 GB GPUs
	Training time	9.2 hours

Non-zero initialization of both LoRA factors has also been reported to reduce sensitivity to the learning rate. We use this initialization consistently throughout our experiments. We did not perform a controlled comparison against the standard $B = 0$ initialization; therefore, we treat it as an implementation choice rather than an empirical contribution.