RiSE: Residual Subspace Expert for Generalizable Text-Centric Image Forgery Localization

Kahim Wong[1] **Kemou Li**[1] **Yiming Chen**[1] **Haiwei Wu**[2] **Jiantao Zhou**[1*]
[1] State Key Laboratory of Internet of Things for Smart City, University of Macau
[2] School of Information and Software Engineering,
University of Electronic Science and Technology of China

Abstract

As AI-assisted image editing becomes increasingly prevalent, Text-Centric Image Forgery Localization (TFL) is essential for protecting trust in financial and legal records. However, we observe that existing TFL detectors predominantly rely on Full-Parameter Fine-Tuning (FPFT) and fail to generalize to unseen forgery types because FPFT can induce low-rank feature collapse and overfit training artifacts. Meanwhile, prior detectors capture diverse forgery traces with a single unified representation, making subtle traces entangled and difficult to learn. To address these challenges, we propose **RiSE**, a Residual Subspace Expert model that preserves pre-trained priors by freezing principal SVD components and adapting only the residual subspaces with a ViT-only segmentor to avoid feature collapse. RiSE introduces forgery-aware residual experts, where each expert is trained independently on a training subset with a particular forgery type, and we show that cross-domain localization performance improves consistently as more experts are added. At inference, we introduce Latent Perturbation Confidence (LPC), which selects the most confident expert by analytically computing the latent-perturbed output. LPC enables efficient confidence estimation with a single forward pass, avoiding repeated forward passes on perturbed samples and improving expert selection. By training on synthetic forgery data, RiSE with 1 expert outperforms prior methods on real-world forgery images by 20.4% while reducing training steps by $10\times$. Scaling the number of experts to 9 yields a 35.8% gain. RiSE remains effective with DCT feature, even when the majority of the parameters is frozen with RGB-only pretraining. Code: `https://github.com/KahimWong/RiSE`.

1 Introduction

Text-Centric Image Forgery Localization (TFL) aims to identify manipulated regions in scanned or captured text-centric images. This problem is increasingly important because modern AI-assisted editing tools allow ordinary users to alter document and scene-text images with little expertise [Wong et al., 2025b,a, 2026b], threatening trust in images used for financial verification, legal review, and identity authentication. Although recent TFL methods have improved accuracy by combining RGB, frequency [Qu et al., 2023], and multi-scale cues [Dong et al., 2024], their performance still drops sharply under cross-domain evaluation, where unseen forgery patterns and acquisition artifacts differ from the training distribution [Du et al., 2025, Gushchin et al., 2026]. We argue that this failure is not merely a data scarcity issue. Unlike semantic segmentation, TFL depends on weak, non-semantic forensic traces whose statistics vary sharply across editing pipelines, recompression settings, and capture conditions. As a result, many existing TFL detectors [Qu et al., 2023, Chen et al., 2024, Wong et al., 2025c] rely on Full-Parameter Fine-Tuning (FPFT) and end up specializing to the limited

*Corresponding author: Jiantao Zhou (jtzhou@um.edu.mo).

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

forgery patterns seen during training. Their penultimate features become highly low-rank, as shown in Fig. 1, indicating representation collapse that harms cross-domain generalization.

This observation points to a different formulation of TFL: the central challenge is to control *where* task-specific adaptation happens. Recent work on residual subspace adaptation [Yan et al., 2025] shows that transferable priors from Vision Foundation Models (VFMs), typically built on the Vision Transformer [Dosovitskiy et al., 2021] (ViT) architecture (e.g., CLIP [Radford et al., 2021], DINOv2 [Oquab et al., 2024], and DINOv3 [Siméoni et al., 2025]), can be preserved by freezing the dominant SVD components of ViT linear layers and fine-tuning only residual subspaces. This prior-preserving adaptation avoids representation collapse and improves performance in AI-generated image detection [Yan et al., 2025]. However, bringing this idea to TFL requires more than inserting residual adaptation into an off-the-shelf segmentor. In TFL, the task-specific decoders and fusion modules can become a second adaptation pathway, allowing the model to memorize dataset-specific compression, synthesis, or capture artifacts outside the pre-trained backbone. At the same time, learning all forgery types in a single shared residual space entangles heterogeneous traces, causing subtle manipulation cues to be overwhelmed by dominant but less transferable patterns. This leads to two questions: (1) Can TFL be reformulated as a prior-preserving localization problem, with minimal trainable modules beyond the ViT so that adaptation remains constrained and transferable? (2) Can the limited trainable capacity be expertized, so that different residual subspaces specialize to complementary forgery cues and generalize better under domain shift?

To answer these questions, we propose **RiSE**, a Residual Subspace Expert model for TFL built on a ViT-only segmenter. We instantiate RiSE on EoMT [Kerssies et al., 2025], which turns a ViT into a segmenter using only query embeddings and a lightweight mask module, thereby minimizing additional adaptation pathways beyond the backbone. On top of this architecture, RiSE divides the training set into non-overlapping subsets with a particular forgery type and trains one residual expert on each subset independently. All experts share the dominant principal subspaces of the VFM backbone, while each expert learns only a tiny residual subspace tailored to a more homogeneous forgery regime. We show that this offline image-level routing is more stable than learnable routing [Shazeer et al., 2017], and that scaling the number of experts consistently improves cross-domain performance as shown in Fig. 2 while parameter growth remains sublinear because the principal components are shared among experts. To select the final prediction among experts, we further propose Latent Perturbation Confidence (LPC), which addresses the limitation that existing confidence metrics [Hendrycks and Gimpel, 2017, Borges et al., 2024] often become unreliable when forged regions are small. LPC estimates the expected latent-perturbed output analytically and favors experts whose forged predictions remain stable under perturbations. As a result, LPC enables single-pass confidence estimation, avoiding repeated forward passes on perturbed samples and improving expert selection for more accurate localization. Trained on DocTamper [Qu et al., 2023] synthetic forgery data, a single RiSE expert generalizes effectively to real-world forgery images, achieving a substantial 20.4% performance improvement while reducing training steps by $10\times$ compared with existing TFL methods. Scaling to 9 experts further boosts the performance gain to 35.8%.

Our contributions are fourfold:

- We identify uncontrolled task-specific adaptation as a key cause of poor cross-domain TFL, show that FPFT induces low-rank feature collapse, and reformulate TFL as a prior-preserving localization problem with a ViT-only residual adaptation architecture.
- We introduce residual subspace experts for TFL, each optimized on a training subset with a specific forgery type, so that different residual subspaces specialize to complementary forgery cues and improve cross-domain localization as more experts are added while keeping parameter growth sublinear.
- We develop a plug-and-play confidence estimator, LPC, that analytically computes expected latent-perturbed outputs to measure expert confidence effectively and efficiently.
- Extensive experiments demonstrate that RiSE consistently improves cross-domain localization performance by 35.8% across 4 test sets.

2 Related Work

General image forgery localization increasingly focuses on robust trace modeling across manipulation types. Representative methods such as MVSS-Net [Chen et al., 2021] and PSCC-Net [Liu et al.,

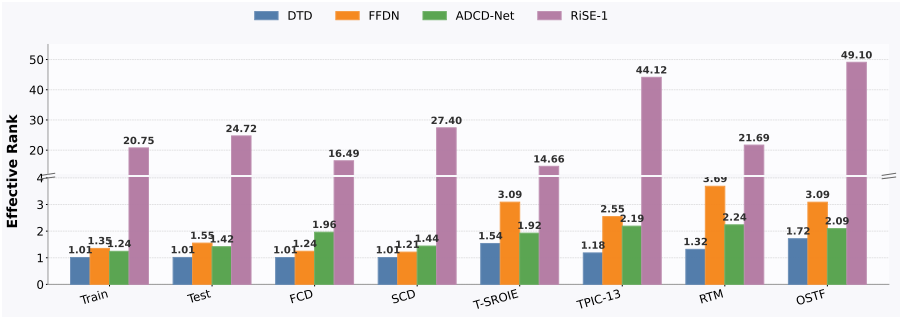


Figure 1: Effective-rank comparison of penultimate features across 8 datasets. Prior TFL detectors (DTD, FFDN, and ADCD-Net) concentrate most representation variance in a very small number of directions, while RiSE-1 (one expert RiSE) maintains a substantially higher effective rank, indicating a less collapsed and more expressive representation.

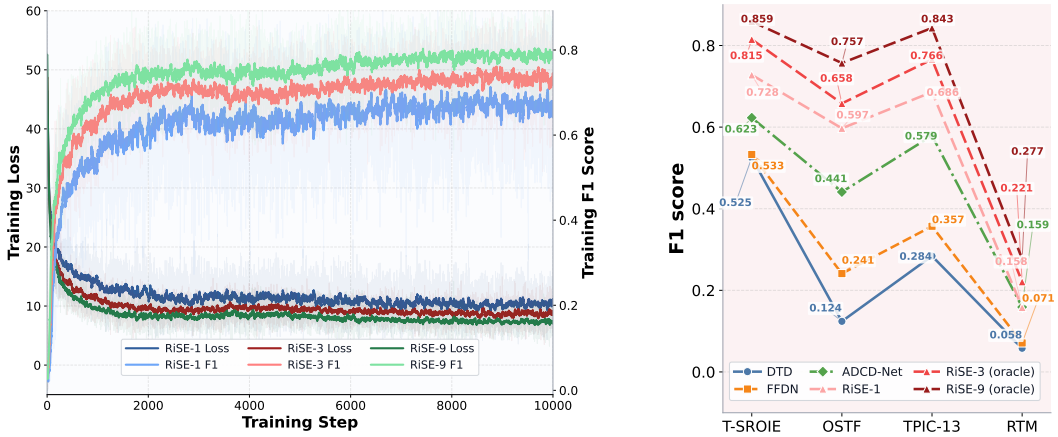
2022] use multi-scale cues, CAT-Net [Kwon et al., 2022] exploits JPEG DCT statistics, and later work including IFOSN [Wu et al., 2022], TruFor [Guillaro et al., 2023], FOCAL [Wu et al., 2025], and Mesorch [Zhu et al., 2025] improves generalization through modeling Online Social Networks (OSN) transmission, noise fingerprint, contrastive feature, and designing CNN–Transformer structure. Nevertheless, general image forgery detectors often struggle on text-centric images because forged regions can blend into the uniform backgrounds and structured text layouts.

Text-centric image forgery localization has progressed from early benchmarking to DCT-aware and robustness-oriented detectors [Wong et al., 2026a]. DTD [Qu et al., 2023] introduced the DocTamper benchmarking and showed that DCT perception with multi-scale decoding can effectively localize tampered regions. FFDN [Chen et al., 2024] and ADCD-Net [Wong et al., 2025c] further improved localization through RGB–DCT fusion, with ADCD-Net explicitly addressing DCT misalignment and text-background disparity. TIFDM [Dong et al., 2024] strengthens trace enhancement and multi-scale aggregation under degradations, CAFTB [Song et al., 2025] combines spatial and noise-domain cues with cross-attention fusion, and DITL^2 improves performance through learning from both general and text-centric image forgeries [Li et al., 2025]. However, most existing TFL detectors rely on FPFT, which overfits training forgery patterns and induces low-rank feature collapse, harming cross-domain generalization. In contrast, RiSE preserves transferable priors via residual subspace adaptation and uses forgery-aware residual experts for stronger specialization and generalization.

3 Method

Motivation. Recent TFL methods such as DTD [Qu et al., 2023], FFDN [Chen et al., 2024], and ADCD-Net [Wong et al., 2025c] rely on FPFT. As shown in Fig. 1, these models produce strongly low-rank features, overfit to training forgery traces, and perform worse on cross-domain data [Du et al., 2025]. Motivated by residual subspace adaptation [Yan et al., 2025], we use a ViT-only segmenter [Kerssies et al., 2025] to keep the architecture minimal and preserve transferable VFM priors. RiSE maintains an effective rank that is roughly $4.7 - 37.3\times$ higher than these prior methods across the 8 datasets, suggesting that residual subspace adaptation better preserves transferable knowledge and avoids feature collapse. The effective rank computation is detailed in Appendix C. Our second motivation is the scaling behavior of residual experts. For TFL, we argue that modeling diverse forgery patterns and image context with a single space could lead to entangled and noisy representations [Tu et al., 2026]. Therefore, we propose to split the training data into subsets with a particular forgery type and train a residual expert on each subset independently. Adding residual experts consistently improves training optimization in Fig. 2 (a), as well as enhancing cross-domain localization in Fig. 2 (b). For RiSE with multiple experts, we report the oracle performance, which assumes access to the ground-truth mask to choose the best expert for each sample and therefore serves as an upper bound on expert selection.

Overview. RiSE is a residual subspace expert model for TFL built on a ViT-only segmenter. Let the training set be $\mathcal{D} = \{(\mathbf{X}_i, \mathbf{Y}_i, d_i)\}_{i=1}^{N}$, where $\mathbf{X}_i \in \mathbb{R}^{3\times H\times W}$ is a forgery image, $\mathbf{Y}_i \in \{0,1\}^{H\times W}$ is the ground-truth mask, and $d_i \in \{1,\ldots,N_e\}$ is the expert-assignment label of the i-th sample.



(a) (b)

Figure 2: More residual experts improve (a) training optimization and (b) cross-domain localization. RiSE-x denotes the RiSE with x residual experts. In (b), RiSE-3 and RiSE-9 report oracle expert selection, i.e., the upper bound obtained by choosing the best expert with the ground-truth mask.

We denote $s_i \in \{1, \ldots, N_s\}$ and $k_i \in \{1, \ldots, N_c\}$ as the synthesis-type index and feature-cluster index of sample i, respectively. We derive the expert-assignment label by $d_i = (s_i - 1)N_c + k_i$. As illustrated in Fig. 3 (a), RiSE partitions the training set by the Cartesian product of N_s forgery synthesis types and N_c feature-cluster indices, yielding $N_e = N_s \times N_c$ non-overlapping sub-training sets $\mathcal{D} = \bigcup_{j=1}^{N_e} \mathcal{D}_j$. Details of the expert assignment are provided in Appendix B.

Definition 1 (RiSE Expert)**.** For expert index $j \in \{1, \ldots, N_e\}$, define the j-th sub-training set as

$$\mathcal{D}_j := \{(\mathbf{X}_i, \mathbf{Y}_i, d_i) \in \mathcal{D} : d_i = j\}.$$

Let Θ_{sh} denote the shared parameter set and let $\Theta_{\text{sp}}^{(j)}$ denote the expert-specific parameter set of expert j. Then the j-th RiSE expert is the segmentation model

$$E_j(\cdot; \Theta_{\text{sh}}, \Theta_{\text{sp}}^{(j)}) : \mathbb{R}^{3\times H\times W} \to [0,1]^{H\times W},$$

which is optimized only on $\mathcal{D}_j$, i.e.,

$$(\Theta_{\text{sp}}^{(j)})^\star \in \arg\min_{\Theta_{\text{sp}}^{(j)}} \mathbb{E}_{(\mathbf{X},\mathbf{Y},d)\sim\mathcal{D}_j}\left[\mathcal{L}(E_j(\mathbf{X}; \Theta_{\text{sh}}, \Theta_{\text{sp}}^{(j)}), \mathbf{Y})\right].$$

In practice, $\Theta_{\text{sp}}^{(j)}$ corresponds to the residual subspace parameters, while Θ_{sh} contains the dominant shared pre-trained components and other shared modules. The RiSE expert differs significantly from a conventional Mixture-of-Experts (MoE) [Shazeer et al., 2017, Kong et al., 2025], where the entire model is jointly trained and combined by a learned router. RiSE instead assigns samples to experts offline during training and selects one expert prediction at test time. The detailed architecture of a single expert is shown in Fig. 3 (b). All experts share the same macro architecture while having expert-specific residual parameters in the ViT linear layers. At inference, as shown in Fig. 4 (a), all experts perform inference on the same input in parallel, producing expert-wise probability maps $\{\hat{\mathbf{P}}_j\}_{j=1}^{N_e}$ with $\hat{\mathbf{P}}_j \in [0,1]^{H\times W}$. RiSE then estimates a confidence score c_j for each expert and selects the prediction of the most confident expert, $j^* = \arg\max_j c_j$.

3.1 RiSE Architecture and Training

RiSE partitions the training set into subsets with a particular forgery type, trains experts independently, and selects the most confident expert at inference. We now describe the ViT-only architecture and training details of a single RiSE expert. As shown in Fig. 3 (b), an expert consists of three modules: a ViT-based VFM backbone, an optional DCT module, and a mask module. Given an RGB image $\mathbf{X}$, let P denote the ViT patch size, and $H_p = H/P$ and $W_p = W/P$. RiSE first extracts visual

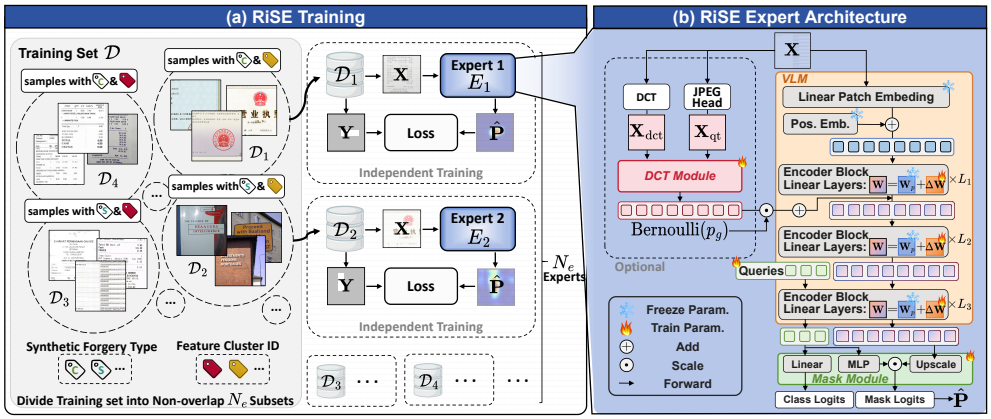


Figure 3: Training overview of RiSE. (a) The training set is partitioned into non-overlapping subsets using forgery synthesis type and feature-cluster index, and each subset trains one expert independently. (b) Each expert uses the same ViT-only segmenter with a shared, optional DCT branch, a shared frozen principal subspace, and expert-specific residual parameters.

tokens $\mathbf{Z}_{\text{rgb}} \in \mathbb{R}^{H_p W_p \times d}$ from the first L_1 blocks of the ViT. Optionally, a DCT branch extracts JPEG compression traces. The branch first applies an 8×8 block DCT to the luminance channel of $\mathbf{X}$ to obtain coefficients $\mathbf{X}_{\text{dct}} \in \mathbb{N}^{H \times W}$ and the quantization table $\mathbf{X}_{\text{qt}} \in \mathbb{N}^{8 \times 8}$ from the JPEG header, encodes $(\mathbf{X}_{\text{dct}}, \mathbf{X}_{\text{qt}})$ into frequency-aware tokens $\mathbf{Z}_{\text{dct}} \in \mathbb{R}^{H_p W_p \times d}$, and fuses $\mathbf{Z}_{\text{dct}}$ with $\mathbf{Z}_{\text{rgb}}$ to form $\mathbf{Z}_{\text{fuse}} \in \mathbb{R}^{H_p W_p \times d}$ before the intermediate L_2 blocks. The learnable query tokens $\mathbf{Q} \in \mathbb{R}^{N_q \times d}$ are prepended before the last L_3 blocks, so that the last VFM block produces both dense tokens $\mathbf{Z}_{\text{vfm}} \in \mathbb{R}^{H_p W_p \times d}$ and query embeddings $\mathbf{Q}_{\text{vfm}} \in \mathbb{R}^{N_q \times d}$. The mask module takes $(\mathbf{Z}_{\text{vfm}}, \mathbf{Q}_{\text{vfm}})$ and predicts the forgery probability map $\hat{\mathbf{P}} \in [0,1]^{H \times W}$. All experts share the architecture in Fig. 3 (b) and differ only in the expert-specific residual parameters inserted into selected ViT linear layers. Table 1 summarizes the training and sharing status of the modules.

Table 1. Module details of a single RiSE expert. "Shared" indicates the module is shared by all experts.

Module	Sub-module	Trainable	Shared	#Params (M)
DCT Module	-	✓	✓	3.82
VFM	Patch Embedding		✓	0.79
	Prefix Tokens / RoPE		✓	0.01
	Principal Weights ($\mathbf{W}_{n-r}$)		✓	301.99
	Residual Weights ($\Delta\mathbf{W}$)	✓		0.44 / expert
Query	-	✓	✓	0.01
Mask Module	-	✓	✓	11.56

VFM backbone. The VFM encodes the input image into visual tokens and processes them with ViT blocks. Residual adaptation applies to any linear layer $\mathbf{W} \in \mathbb{R}^{d_1 \times d_2}$. We consider the square case $d_1 = d_2 = n$, the attention $\texttt{q/k/v/o}$ and MLP $\texttt{up/down}$ linear projections. Given a pre-trained weight matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$, RiSE applies SVD decomposition $\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, where $\mathbf{U} \in \mathbb{R}^{n \times n}$ and $\mathbf{V} \in \mathbb{R}^{n \times n}$ are orthogonal matrices of left and right singular vectors, respectively, and $\mathbf{\Sigma} \in \mathbb{R}^{n \times n}$ is diagonal. We keep the rank-$(n-r)$ approximation $\mathbf{W}_{n-r} = \mathbf{U}_{n-r}\mathbf{\Sigma}_{n-r}\mathbf{V}_{n-r}^\top$, with $\mathbf{U}_{n-r} \in \mathbb{R}^{n \times (n-r)}$, $\mathbf{\Sigma}_{n-r} \in \mathbb{R}^{(n-r) \times (n-r)}$, and $\mathbf{V}_{n-r} \in \mathbb{R}^{n \times (n-r)}$, as the frozen principal components. Each expert shares the same frozen $\mathbf{W}_{n-r}$ and learns an individual residual component $\Delta\mathbf{W} = \mathbf{W} - \mathbf{W}_{n-r} = \mathbf{U}_r\mathbf{\Sigma}_r\mathbf{V}_r^\top$, where $\mathbf{U}_r \in \mathbb{R}^{n \times r}$, $\mathbf{\Sigma}_r \in \mathbb{R}^{r \times r}$, and $\mathbf{V}_r \in \mathbb{R}^{n \times r}$. This residual adaptation preserves the dominant pre-trained knowledge while letting each expert learn specific forgery traces, and the number of parameters grows sublinearly as more experts are added. Before the last L_3 encoder blocks, we prepend learnable query tokens $\mathbf{Q}$ so that the backbone produces both query embeddings and feature tokens for downstream localization.

DCT module. As demonstrated in prior TFL studies [Chen et al., 2024, Wong et al., 2025c], DCT traces improve robustness to recompression. However, such traces can also become a shortcut, causing the model to over-rely on DCT cues instead of RGB evidence. RiSE encodes the DCT coefficients $\mathbf{X}_{\text{dct}}$ and quantization table $\mathbf{X}_{\text{qt}}$ with the DCT encoder ϕ_{dct} in [Qu et al., 2023], and obtains DCT features $\mathbf{Z}_{\text{dct}} = \phi_{\text{dct}}(\mathbf{X}_{\text{dct}}, \mathbf{X}_{\text{qt}})$. The DCT features then conditionally fuse with the RGB features using an image-level Bernoulli gate, $\mathbf{Z}_{\text{fuse}} = \mathbf{Z}_{\text{rgb}} + g\,\mathbf{Z}_{\text{dct}}$ with $g \sim \text{Bernoulli}(p_g)$,

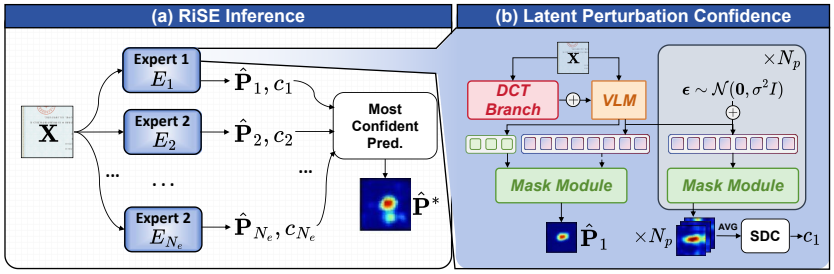


Figure 4: Inference overview of RiSE. (a) RiSE evaluates all experts in parallel on the same input and selects the final localization map from the most confident expert. (b) Latent Perturbation Confidence perturbs the latent features after the VFM backbone, averages the resulting mask predictions, and scores the averaged output with SDC for expert selection.

where $p_g = 0.5$ is the default DCT fusion probability during training, and $p_g = 1.0$ at inference. This random gate balances DCT and RGB learning by randomly dropping DCT traces during training, reducing reliance on DCT shortcuts. Unlike ADCD-Net [Wong et al., 2025c], which uses a learnable DCT fusion score, our design does not assume DCT-grid alignment and avoids extra computation from alignment-breaking augmentations and score-learning modules. We observe that RiSE remains robust to such auxiliary cues despite freezing the principal components of an RGB-pretrained VFM, and future work could extend this fusion to noise prints [Guillaro et al., 2023], boundary traces [Liu et al., 2024], or edit traces [Wu et al., 2026].

Mask module. The mask module follows a Mask2Former-style design [Cheng et al., 2022]. With the VFM outputs $(\mathbf{Q}_{\text{vfm}}, \mathbf{Z}_{\text{vfm}})$, the class head predicts per-query logits $\mathbf{C} = \phi_{\text{cls}}(\mathbf{Q}_{\text{vfm}}) \in \mathbb{R}^{N_q \times 2}$. For mask prediction, we define the transformed query features $\mathbf{Q}_{\text{mask}} = \phi_{\text{mlp}}(\mathbf{Q}_{\text{vfm}}) \in \mathbb{R}^{N_q \times d}$ and the transformed dense features $\mathbf{Z}_{\text{mask}} = \phi_{\text{up}}(\mathbf{Z}_{\text{vfm}}) \in \mathbb{R}^{H_p \times W_p \times d}$. The mask head then predicts mask logits $\mathbf{M} \in \mathbb{R}^{N_q \times H_p \times W_p}$, where each $\mathbf{M}[q] \in \mathbb{R}^{H_p \times W_p}$ is obtained by the dot product between $\mathbf{Q}_{\text{mask}}[q]$ and $\mathbf{Z}_{\text{mask}}$. We then select $q^* = \arg\max_q(\mathbf{C}[q:0])$ and output $\hat{\mathbf{P}} = \sigma(\text{Up}(\mathbf{M}[q^*])) \in [0,1]^{H \times W}$, where $\text{Up}(\cdot)$ bilinearly upsamples the mask to $H \times W$, and σ is the sigmoid function.

Training objective. We adopt the same Mask2Former [Cheng et al., 2022] losses: the class logits use Cross-Entropy (CE), and the mask logits use Binary Cross-Entropy (BCE) and Dice loss. The total loss is $\mathcal{L} = \lambda_{\text{CE}}\mathcal{L}_{\text{CE}} + \lambda_{\text{BCE}}\mathcal{L}_{\text{BCE}} + \lambda_{\text{Dice}}\mathcal{L}_{\text{Dice}}$.

3.2 RiSE Inference

As suggested by the expert-scaling trend in Fig. 2 (b), adding residual experts improves cross-domain oracle F1. For a test set $\mathcal{D}_T = \{(\mathbf{X}_i, \mathbf{Y}_i)\}_{i=1}^{N_t}$ and evaluation metric m (e.g. F1 score), we define the oracle score $\text{Oracle}(\mathcal{D}_T, \{E_j\}_{j=1}^{N_e}) = \frac{1}{N_t}\sum_{i=1}^{N_t} \max_{j \in \{1,\ldots,N_e\}} m\big(E_j(\mathbf{X}_i), \mathbf{Y}_i\big)$ for RiSE with multiple experts. Because the ground-truth mask is unavailable at test time, this oracle score serves only as an upper bound on RiSE performance. As shown in Fig. 4 (a), RiSE evaluates all experts in parallel on an input $\mathbf{X}$, producing expert-wise probability maps $\{\hat{\mathbf{P}}_j\}_{j=1}^{N_e}$. We then estimate a confidence score c_j for each expert and select the most confident one, $j^* = \arg\max_j c_j$, as the final output. We evaluate three confidence measurements for this purpose, the existing Maximum Softmax Probability (MSP) [Hendrycks and Gimpel, 2017] and Soft Dice Confidence (SDC) [Borges et al., 2024], as well as our proposed Latent Perturbation Confidence (LPC). For clarity, we flatten the probability map $\hat{\mathbf{P}}$ into a vector $\hat{\mathbf{p}} \in [0,1]^{N_{\text{pix}}}$, where $N_{\text{pix}} = HW$.

MSP. As a simple baseline, we compute the average Maximum Softmax Probability (MSP) across all pixels [Hendrycks and Gimpel, 2017], $\text{MSP}(\hat{\mathbf{p}}) = \frac{1}{N_{\text{pix}}}\sum_{n=1}^{N_{\text{pix}}} \max\{\hat{\mathbf{p}}(n), 1 - \hat{\mathbf{p}}(n)\}$. However, MSP can be dominated by the easy pristine pixels and underestimate the influence of tiny tampered regions. This motivates the usage of SDC, which focuses on the foreground confidence.

SDC. To address the extreme class imbalance, Soft Dice Confidence (SDC) [Borges et al., 2024] does not consider true negative predictions and therefore ignores the huge easy pristine area: $\mathrm{SDC}(\hat{\mathbf{p}}) = 2\hat{\mathbf{p}}^\top\hat{\mathbf{y}}/(\|\hat{\mathbf{p}}\|_1 + \|\hat{\mathbf{y}}\|_1)$, where $\hat{\mathbf{y}} = \mathbb{1}[\hat{\mathbf{p}} > \tau] \in \{0,1\}^{N_{\mathrm{pix}}}$, in which τ is the binarization threshold. SDC favors experts that consistently produce high probabilities for the forged area. Although SDC improves over MSP, it still lags behind oracle performance.

LPC. Our proposed LPC builds on SDC with a simple intuition: an expert should be trusted when its confidences remain stable under perturbed inputs. We therefore compute SDC on the prediction obtained by averaging over multiple Test-Time Augmentations (TTA), and Appendix D provides a justification for why TTA averaging stabilizes SDC. Formally, let $T(\cdot)$ be an augmentation (e.g. flipping, rotation, noise), and the TTA-averaged prediction be $\bar{\mathbf{p}} = \frac{1}{M}\sum_{m=1}^{M} T_m^{-1}(E(T_m(\mathbf{X})))$, where T_m^{-1} maps the prediction back to the original coordinates, and for non-spatial perturbations such as additive Gaussian noise, T_m^{-1} is the identity map. We then compute SDC on $\bar{\mathbf{p}}$ to obtain the TTA-SDC score. However, this input-space TTA is computationally expensive, yielding M forward passes per image. To avoid the cost of TTA, we propose to apply perturbations directly to the latent space, that is the VFM output feature $\mathbf{Z}_{\mathrm{vfm}}$:

$$\mathrm{LPC}(\mathbf{Z}_{\mathrm{vfm}}) = \mathrm{SDC}\big(\bar{\mathbf{p}}_{\mathrm{lat}}\big), \qquad \bar{\mathbf{p}}_{\mathrm{lat}} = \frac{1}{M}\sum_{m=1}^{M} T_m^{-1}\big(\phi_{\mathrm{mask}}\big(\mathbf{Q}_{\mathrm{vfm}}, T_m(\mathbf{Z}_{\mathrm{vfm}})\big)\big)\,,$$

where ϕ_{mask} is the mask module. Compared with image-space TTA, LPC avoids repeated VFM forwards by perturbing only the latent features. For Gaussian latent perturbations,

$$T_m(\mathbf{Z}_{\mathrm{vfm}}) = \mathbf{Z}_{\mathrm{vfm}} + \boldsymbol{\epsilon}_m, \qquad \boldsymbol{\epsilon}_m \sim \mathcal{N}(\mathbf{0}, \sigma^2 I).$$

The averaged output can then be written as the expected latent-perturbed output:

$$\bar{\mathbf{p}}_{\mathrm{lat}} = \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\sigma^2 I)}\big[\phi_{\mathrm{mask}}(\mathbf{Q}_{\mathrm{vfm}}, \mathbf{Z}_{\mathrm{vfm}} + \boldsymbol{\epsilon})\big].$$

We find that this expectation admits an analytic approximation, without explicitly sampling the M perturbations. Concretely, we propagate the mean and variance through the mask module using Assumed Density Filtering (ADF) [Gast and Roth, 2018] and details are provided in Appendix E. As a result, LPC requires only a single forward pass through both the VFM and the mask module.

4 Experiments

Dataset. We adapt the *Doc Protocol* in ForensicHub [Du et al., 2025] for the main experiments. All models are trained only on the DocTamper training split [Qu et al., 2023] and are evaluated on seven test sets: three in-domain DocTamper test sets, *FCD*, *SCD*, and *Test*, and four cross-domain test sets, *T-SROIE* [Wang et al., 2022b], *OSTF* [Qu et al., 2025], *TPIC-13*[Wang et al., 2022a], and *RTM* [Luo et al., 2025]. Details of the datasets are provided in Appendix Table 5.

Competitors. We compare RiSE with six general image forgery localization methods: MVSS-Net [Chen et al., 2021], PSCC-Net [Liu et al., 2022], CAT-Net [Kwon et al., 2022], TruFor [Guillaro et al., 2023], IML-ViT [Ma et al., 2023], and Mesorch [Zhu et al., 2025], as well as five recent TFL methods: DTD [Qu et al., 2023], FFDN [Chen et al., 2024], CAFTB [Song et al., 2025], TIFDM [Dong et al., 2024], and ADCD-Net [Wong et al., 2025c].

Implementation. RiSE uses DINOv3 ViT-L/16 [Siméoni et al., 2025] as the VFM backbone, with $L_1 = 10$, $L_2 = 10$, and $L_3 = 4$ blocks. The RiSE design choices are detailed in the ablation study in Table 3. We train RiSE for 10,000 steps with the AdamW optimizer, using a learning rate of $3e-4$ decayed to $1e-5$ via cosine annealing, and a batch size of 16. Detailed training and inference efficiency are provided in Appendix Table 16.

4.1 Main Results

Table 2 highlights the superiority of RiSE over prior TFL methods, especially under domain shift. Compared with ADCD-Net, RiSE-1 improves cross-domain F1 by 20.4%, RiSE-3 increases this gain to 27.8%, and RiSE-9 further enlarges the cross-domain F1 to 35.8%. Prior non-DCT detectors

Table 2. F1 scores on 7 TFL test sets. Number colors run from blue (lowest) to orange (highest) across this table; alternating rows have a light-gray background. Best is **bold** and second-best is underlined. AvgI: average F1 on the 3 in-domain test sets. AvgC: average F1 on the 4 cross-domain test sets. Avg: average F1 on all 7 test sets. RiSE-x: x experts without DCT traces. RiSE-9-DCT: 9 experts with DCT traces.

Methods	Venue	In-Domain (Synthetic Data)				Cross-Domain (Real Data)					Avg
		FCD	*SCD*	*Test*	AvgI	*T-SROIE*	*OSTF*	*TPIC-13*	*RTM*	AvgC	
TruFor	CVPR'23	0.261	0.312	0.252	0.275	0.213	0.046	0.104	0.034	0.099	0.175
MVSS-Net	ICCV'23	0.207	0.371	0.381	0.320	0.187	0.037	0.113	0.027	0.091	0.189
Mesorch	AAAI'25	0.423	0.459	0.465	0.449	0.294	0.139	0.241	0.041	0.178	0.294
IML-ViT	arXiv	0.469	0.512	0.449	0.476	0.427	0.210	0.256	0.076	0.242	0.343
PSCC-Net	TCSVT'22	0.386	0.393	0.497	0.425	0.517	0.441	0.550	0.126	0.408	0.416
Cat-Net	IJCV'22	0.760	0.641	0.764	0.722	0.609	0.178	0.343	0.063	0.298	0.480
TIFDM	TCE'24	0.090	0.257	0.259	0.202	0.058	0.006	0.013	0.018	0.024	0.100
CAFTB	TOMM'24	0.292	0.377	0.328	0.332	0.262	0.119	0.301	0.033	0.179	0.244
DTD	CVPR'23	0.686	**0.739**	0.803	0.743	0.525	0.124	0.284	0.058	0.247	0.460
FFDN	ECCV'24	**0.877**	**0.739**	**0.821**	**0.813**	0.533	0.241	0.357	0.071	0.300	0.520
ADCD-Net	ICCV'25	0.803	0.718	0.791	0.771	0.623	0.441	0.579	0.159	0.450	0.588
RiSE-1	Ours	0.744	0.590	0.654	0.662	0.728	0.597	0.686	0.158	0.542	0.594
RiSE-3	Ours	0.764	0.588	0.675	0.676	0.760	0.625	0.740	0.173	0.575	0.618
RiSE-9	Ours	0.791	0.579	0.692	0.687	0.785	**0.692**	**0.795**	0.175	0.611	0.644
RiSE-9-DCT	Ours	0.864	0.662	0.775	0.767	**0.808**	0.654	0.784	**0.218**	**0.616**	**0.681**

fall short in both domains. PSCC-Net reaches a relatively strong cross-domain F1 of 0.408, yet the in-domain F1 remains only 0.425. Adding DCT cues improves in-domain fitting for Cat-Net, DTD, FFDN, and ADCD-Net, all with in-domain F1 above 0.7, but the best cross-domain F1 still peaks at only 0.450 (ADCD-Net). RiSE-9-DCT obtains an in-domain F1 of 0.767, a cross-domain F1 of 0.616, and an overall F1 of 0.681. Relative to ADCD-Net, RiSE-9-DCT offers 36.8% higher cross-domain F1 and 15.7% higher overall F1. Compared with RiSE-9, RiSE-9-DCT increases AvgI from 0.687 to 0.767 and increases AvgC from 0.611 to 0.616. RTM is particularly challenging because it contains manual manipulations, and the oracle F1 of RiSE-9 in Fig. 2 (b) can further raise the RTM F1 to 0.277, encouraging more advanced expert-selection methods in future studies.

4.2 Ablation Study

Table 3. Ablation on RiSE-9 design. $*$: LoRA. FT: fine-tune blocks. A: attention. M: MLP. Shared: shared mask module. N_q: number of queries. p_g: DCT fusion probability. Router: expert routing. †: two-stage MoE. **RiSE-9 uses #16 setting. RiSE-9-DCT uses #18 setting.**

#	Rank	FT	Shared	N_q	p_g	Router	AvgI	AvgC	Avg
#1		Full-parameter Fine-tune					0.541	0.211	0.353
#2	1	A + M	✓	10	0	Offline	0.411	0.423	0.418
#3	1	A + M	✓	20	0	Offline	0.539	0.416	0.469
#4	1	A + M		1	0	Offline	0.637	0.535	0.579
#5	1	A + M	✓	1	0	MoE†	0.682	0.435	0.541
#6	1	A + M	✓	1	0	MoE	0.686	0.540	0.603
#7	1	A	✓	1	0	Offline	0.613	0.573	0.590
#8	1	M	✓	1	0	Offline	0.654	0.580	0.612
#9	16*	A + M	✓	1	0	Offline	0.658	0.596	0.623
#10	8*	A + M	✓	1	0	Offline	0.663	0.587	0.620
#11	4*	A + M	✓	1	0	Offline	0.659	0.589	0.619
#12	1*	A + M	✓	1	0	Offline	0.656	0.587	0.617
#13	16	A + M	✓	1	0	Offline	0.666	0.587	0.621
#14	8	A + M	✓	1	0	Offline	0.665	0.580	0.616
#15	4	A + M	✓	1	0	Offline	0.665	0.593	0.624
#16	1	A + M	✓	1	0	Offline	0.663	0.596	0.625
#17	1	A + M	✓	1	1	Offline	0.648	0.519	0.574
#18	1	A + M	✓	1	0.5	Offline	0.690	0.581	0.628

RiSE design. Table 3 ablates the main RiSE-9 design choices using 5,000 training steps. Among the non-DCT settings ($p_g = 0$), #16 performs the best, achieving an Avg. F1 of 0.625, and we use #16 as the default setting of RiSE-9. Compared with full-parameter fine-tuning (#1), #16 improves Avg. F1 by 77.1%, showing the superior performance of residual subspace adaptation. Increasing the number of queries to 10 or 20 (#2–#3) reduces Avg. F1 by 33.1% and 25.0%, respectively. Making the mask module expert-specific (#4) causes a 7.4% drop, indicating that stronger parameter sharing is beneficial. Replacing offline assignment with standard MoE (#6) or two-stage MoE (#5), where experts are trained first and the router is trained afterward, lowers Avg. F1 by 3.5% and 13.4%, respectively. Using only attention adaptation (#7) or only MLP adaptation (#8) reduces Avg. F1 by 5.6% and 2.1%, suggesting that both adaptations help, with MLP being slightly more important. The residual-rank study further supports a lightweight design. Increasing the residual rank to 4, 8, and 16 (#15, #14, #13) reduces Avg. F1 by 0.2%, 1.4%, and

0.6%, while the LoRA-style variants (#9–#12) are also consistently worse, with drops of 0.3%, 0.8%, 1.0%, and 1.3%. After introducing DCT traces, always enabling fusion during training ($p_g = 1$, #17) reduces AvgI from 0.663 to 0.648 and AvgC from 0.596 to 0.519, a 12.9% drop relative to #16. In contrast, random-gated fusion with $p_g = 0.5$ (#18) boost the AvgI to 0.690 and the AvgC to 0.581, with a Avg. F1 of 0.628 which is 0.5% higher than #16.

Table 4. Ablation on confidence estimation for RiSE-9. F&R: Flip-and-Rotation TTA; GP: Latent Gaussian Perturbation; AGP: Analytic Gaussian Propagation.

Method	*T-SROIE*	*OSTF*	*TPIC-13*	*RTM*	AvgC
Mean Expert	0.688	0.282	0.311	0.094	0.344
MSP	0.789	0.548	0.698	0.093	0.532
SDC	0.790	0.551	0.698	0.096	0.534
TTA-SDC (F&R)	0.789	0.679	0.785	0.159	0.603
LPC (F&R)	0.748	0.674	0.780	0.158	0.590
LPC (GP)	0.784	0.689	0.796	0.175	0.611
LPC (AGP)	0.785	0.692	0.795	0.175	0.611
Oracle	**0.859**	**0.757**	**0.843**	**0.277**	**0.684**

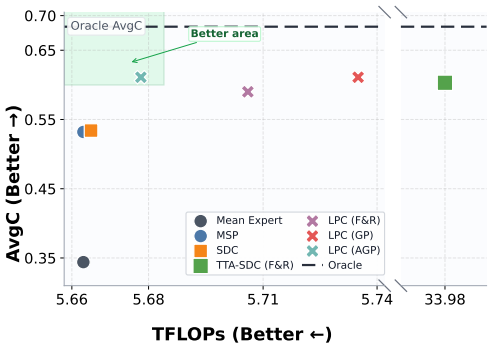


Figure 5. Cross-domain F1 versus inference TFLOPs.

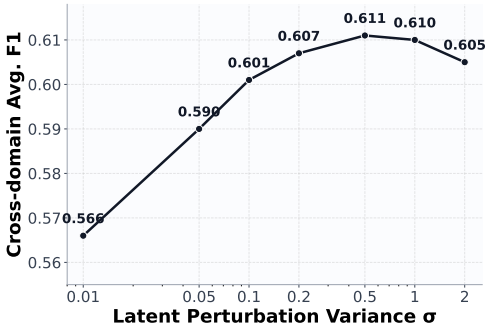


Figure 6. Effect of the latent perturbation variance on cross-domain performance.

Expert confidence estimation. Table 4 compares expert-selection strategies for RiSE-9 on the 4 cross-domain test sets, while Fig. 5 summarizes the corresponding Avg. F1 and TFLOPs comparison. A simple mean over expert predictions performs poorly with 0.344 F1, while MSP and SDC improve F1 to 0.532 and 0.534, respectively. Input-space TTA-SDC with five Flip-and-Rotation (F&R) augmentations (horizontal flip, vertical flip, and clockwise rotations by 90°, 180°, and 270°) boosts Avg. F1 to 0.603. This supports the view that TTA stabilizes confidence estimation, but it requires $6\times$ total computation because the model is evaluated on the original input plus 5 augmented samples, as shown in Fig. 5. Our proposed LPC (F&R) achieves a similar Avg. F1 of 0.590 with the same F&R augmentations, but it is more efficient because it applies perturbations in the latent space and does not require repeated VFM forwards. LPC (GP) with 10 latent Gaussian perturbations reaches the same Avg. F1 of 0.611 as our analytic Gaussian propagation LPC (AGP), but requires more computation. LPC (AGP) is especially attractive because it preserves single-pass inference and closes more of the gap to the oracle selector without incurring the compute growth of TTA. Fig. 6 further examines the effect of the latent perturbation variance (σ^2) on cross-domain performance. The trend shows that a moderate perturbation variance ($\sigma^2 = 0.5$) gives the best confidence estimation, whereas too little noise provides limited stability information and too much noise degrades the localization signal.

5 Conclusion

We presented RiSE, a residual subspace expert model for text-centric image forgery localization. Motivated by the observation that FPFT causes low-rank feature collapse and poor cross-domain generalization, RiSE freezes the dominant VFM subspace, adapts only a small residual component, and scales capacity through independently trained forgery-aware experts. We also introduced LPC, a confidence estimator that selects experts by measuring prediction stability under latent perturbations via a single-pass approximation. Experiments show that RiSE consistently improves cross-domain localization as the number of experts increases. In particular, relative to ADCD-Net, RiSE-1 improves AvgC by 20.4%, and scaling to RiSE-9 increases the gain to 35.8%. RiSE-9-DCT obtains an in-domain average F1 of 0.767, a cross-domain average F1 of 0.616, and an overall average F1 of 0.681. These results suggest that preserving transferable VFM priors while scaling expertized residual capacity is a promising direction for robust and efficient TFL.

Acknowledgements

This work was supported in part by Macau Science and Technology Development Fund under 001/2024/SKL, 0119/2024/RIB2 and 0110/2025/R1B2; in part by Research Committee at University of Macau under MYRG-CRG2025-00031-FST and MYRG-GRG2025-00086-FST; in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2024A1515012536.

References

Bruno Laboissiere Camargos Borges, Bruno Machado Pacheco, and Danilo Silva. Soft dice confidence: A near-optimal confidence estimator for selective prediction in semantic segmentation. *arXiv preprint arXiv:2402.10665*, 2024.

Xinru Chen, Chengbo Dong, Jiaqi Ji, Juan Cao, and Xirong Li. Image manipulation detection by multi-view multi-scale supervision. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 14185–14193, 2021.

Zhongxi Chen, Shen Chen, Taiping Yao, Ke Sun, Shouhong Ding, Xianming Lin, Liujuan Cao, and Rongrong Ji. Enhancing tampered text detection through frequency feature fusion and decomposition. In *Proc. Eur. Conf. Comput. Vis.*, pages 200–217, 2024.

Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proc. IEEE Comput. Vis. Pattern Recogn.*, pages 1290–1299, 2022.

Li Dong, Weipeng Liang, and Rangding Wang. Robust text image tampering localization via forgery traces enhancement and multiscale attention. *IEEE Trans. Consum. Electron.*, pages 3495–3507, 2024.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. Int. Conf. Learn. Representat.*, 2021.

Bo Du, Xuekang Zhu, Xiaochen Ma, Chenfan Qu, Kaiwen Feng, Zhe Yang, Chi-Man Pun, Jian Liu, and Ji-Zhe Zhou. Forensichub: A unified benchmark & codebase for all-domain fake image detection and localization. In *Adv. Neural Inf. Process. Syst.*, 2025.

Jochen Gast and Stefan Roth. Lightweight probabilistic deep networks. In *Proc. IEEE Comput. Vis. Pattern Recogn.*, pages 3369–3378, 2018.

Fabrizio Guillaro, Davide Cozzolino, Avneesh Sud, Nicholas Dufour, and Luisa Verdoliva. Trufor: Leveraging all-round clues for trustworthy image forgery detection and localization. In *Proc. IEEE Comput. Vis. Pattern Recogn.*, pages 20606–20615, 2023.

Aleksandr Gushchin et al. NTIRE 2026 challenge on robust AI-generated image detection in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1895–1913, 2026.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *Proc. Int. Conf. Learn. Representat.*, 2017.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *Proc. Int. Conf. Learn. Representat.*, 2023.

Tommie Kerssies, Niccolo Cavagnero, Alexander Hermans, Narges Norouzi, Giuseppe Averta, Bastian Leibe, Gijs Dubbelman, and Daan De Geus. Your vit is secretly an image segmentation model. In *Proc. IEEE Comput. Vis. Pattern Recogn.*, pages 25303–25313, 2025.

Chenqi Kong, Anwei Luo, Peijun Bao, Yi Yu, Haoliang Li, Zengwei Zheng, Shiqi Wang, and Alex C Kot. Moe-ffd: Mixture of experts for generalized and parameter-efficient face forgery detection. *IEEE Trans. Depend. Sec. Comput.*, 2025.

Pavel Korshunov, Vidit Vidit, Amir Mohammadi, Christophe Ecabert, Nevena Shamoska, Sébastien Marcel, Zeqin Yu, Ye Tian, Jiangqun Ni, Lazar Lazarevic, Renat Khizbullin, Anastasiia Evteeva, Alexey Tochin, Aleksei Grishin, Anjith George, Daniel Dealcala, Tamas Endrei, Javier Muñoz-Haro, Ruben Tolosana, Ruben Vera-Rodriguez, Aythami Morales, Julian Fierrez, György Cserey, Hardik Sharma, Sachin Chaudhary, Akshay Dudhane, Praful Hambarde, Amit Shukla, Prateek Shaily, Jayant Kumar, Ajinkya Hase, Satish Maurya, Mridul Sharma, and Pallav Dwivedi. DeepID challenge of detecting synthetic manipulations in ID documents. In *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, pages 521–530, 2025.

Myung-Joon Kwon, Seung-Hun Nam, In-Jae Yu, Heung-Kyu Lee, and Changick Kim. Learning jpeg compression artifacts for image manipulation detection and localization. *Int. J. Comput. Vis.*, pages 1875–1895, 2022.

Songze Li, Yunfei Guo, Shen Chen, Bin Li, Kaiqing Lin, Changsheng Chen, Haodong Li, Taiping Yao, and Shouhong Ding. Ditl2: Dual-stage invariance transfer learning for generalizable document image tampering localization. In *Proc. ACM Int. Conf. Multimedia*, pages 82–91, 2025.

Wenxi Liu, Hao Zhang, Xinyang Lin, Qing Zhang, Qi Li, Xiaoxiang Liu, and Ying Cao. Attentive and contrastive image manipulation localization with boundary guidance. *IEEE Trans. Inf. Forensics Secur.*, pages 6764–6778, 2024.

Xiaohong Liu, Yaojie Liu, Jun Chen, and Xiaoming Liu. Pscc-net: Progressive spatio-channel correlation network for image manipulation detection and localization. *IEEE Trans. Circuit Syst. Video Technol.*, pages 7505–7517, 2022.

Dongliang Luo, Yuliang Liu, Rui Yang, Xianjin Liu, Jishen Zeng, Yu Zhou, and Xiang Bai. Toward real text manipulation detection: New dataset and new solution. *Pattern Recognition*, page 110828, 2025.

Xiaochen Ma, Bo Du, Zhuohang Jiang, Ahmed Y Al Hammadi, and Jizhe Zhou. Iml-vit: Benchmarking image manipulation localization by vision transformer. *arXiv preprint arXiv:2307.14863*, 2023.

Daniel Marczak, Simone Magistri, Sebastian Cygert, Bartłomiej Twardowski, Andrew D. Bagdanov, and Joost van de Weijer. No task left behind: Isotropic model merging with common and task-specific subspaces. In *Proc. Int. Conf. Mach. Learn.*, 2025.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Trans. Mach. Learn. Res.*, 2024. ISSN 2835-8856.

Chenfan Qu, Chongyu Liu, Yuliang Liu, Xinhong Chen, Dezhi Peng, Fengjun Guo, and Lianwen Jin. Towards robust tampered text detection in document image: New dataset and new solution. In *Proc. IEEE Comput. Vis. Pattern Recogn.*, pages 5937–5946, 2023.

Chenfan Qu, Yiwu Zhong, Fengjun Guo, and Lianwen Jin. Revisiting tampered scene text detection in the era of generative ai. In *Proc. AAAI Conf. Arti. Intell.*, pages 694–702, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proc. Int. Conf. Mach. Learn.*, pages 8748–8763, 2021.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *Proc. Int. Conf. Learn. Representat.*, 2017.

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.

Yalin Song, Wenbin Jiang, Xiuli Chai, Zhihua Gan, Mengyuan Zhou, and Lei Chen. Cross-attention based two-branch networks for document image forgery localization in the metaverse. *ACM Trans. Multimedia Comput. Commun. Appl.*, pages 1–24, 2025.

George Stoica, Pratik Ramesh, Boglarka Ecsedi, Leshem Choshen, and Judy Hoffman. Model merging with SVD to tie the KnOTS. In *Proc. Int. Conf. Learn. Representat.*, 2025.

Zhilin Tu, Kemou Li, Fengpeng Li, Jianwei Fei, Jiamin Zhang, and Haiwei Wu. FeatDistill: A feature distillation enhanced multi-expert ensemble framework for robust AI-generated image detection. *arXiv preprint arXiv:2603.21939*, 2026.

Yuxin Wang, Hongtao Xie, Mengting Xing, Jing Wang, Shenggao Zhu, and Yongdong Zhang. Detecting tampered scene text in the wild. In *Proc. Eur. Conf. Comput. Vis.*, pages 215–232, 2022a.

Yuxin Wang, Boqiang Zhang, Hongtao Xie, and Yongdong Zhang. Tampered text detection via rgb and frequency relationship modeling. *Chin. J. Netw. Inf. Secur.*, pages 29–40, 2022b.

Ka Him Wong, Jicheng Zhou, Jiantao Zhou, and Yain-Whar Si. An end-to-end model for logits-based large language models watermarking. In *Proc. Int. Conf. Mach. Learn.*, pages 66971–66991, 2025a.

Kahim Wong, Jicheng Zhou, Kemou Li, Yain-Whar Si, Xiaowei Wu, and Jiantao Zhou. Fontguard: A robust font watermarking approach leveraging deep font knowledge. *IEEE Trans. Multimedia*, pages 7876–7890, 2025b.

Kahim Wong, Jicheng Zhou, Haiwei Wu, Yain-Whar Si, and Jiantao Zhou. Adcd-net: Robust document image forgery localization via adaptive dct feature and hierarchical content disentanglement. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 19280–19289, 2025c.

Kahim Wong, Kemou Li, Yiming Chen, Haiwei Wu, and Jiantao Zhou. SEED: Simple ViT and evolving harness for explainable text forgery detection. *arXiv preprint arXiv:2606.21138*, 2026a.

Kahim Wong, Kemou Li, Haiwei Wu, and Jiantao Zhou. k nnproxy: Efficient training-free proxy alignment for black-box zero-shot llm-generated text detection. *arXiv preprint arXiv:2604.02008*, 2026b.

Haiwei Wu, Jiantao Zhou, Jinyu Tian, Jun Liu, and Yu Qiao. Robust image forgery detection against transmission over online social networks. *IEEE Trans. Inf. Forensics Secur.*, 17:443–456, 2022.

Haiwei Wu, Yiming Chen, Jiantao Zhou, and Yuanman Li. Rethinking image forgery detection via soft contrastive learning and unsupervised clustering. *IEEE Trans. Depend. Sec. Comput.*, pages 6296–6308, 2025.

Haiwei Wu, Kemou Li, Li Yuanman, and Jiantao Zhou. Editprint: General digital image forensics via editing fingerprint with self-augmentation training. In *Proc. IEEE Comput. Vis. Pattern Recogn.*, 2026.

Zhiyuan Yan, Jiangming Wang, Peng Jin, Ke-Yue Zhang, Chengchun Liu, Shen Chen, Taiping Yao, Shouhong Ding, Baoyuan Wu, and Li Yuan. Orthogonal subspace decomposition for generalizable ai-generated image detection. In *Proc. Int. Conf. Mach. Learn.*, pages 70268–70288, 2025.

Xuanming Zhang, Sining Zhoubian, Yuxuan Chen, Tianyi Tang, An Yang, Sean Du, Chujie Zheng, Fei Huang, Dayiheng Liu, Gao Huang, and Jingren Zhou. Deeper is not always better: Mitigating the alignment tax via confident layer decoding. *arXiv preprint arXiv:2606.21906*, 2026.

Xuekang Zhu, Xiaochen Ma, Lei Su, Zhuohang Jiang, Bo Du, Xiwen Wang, Zeyu Lei, Wentao Feng, Chi-Man Pun, and Ji-Zhe Zhou. Mesoscopic insights: orchestrating multi-scale & hybrid architecture for image manipulation localization. In *Proc. AAAI Conf. Arti. Intell.*, pages 11022–11030, 2025.

A Dataset Summary

Table 5: Summary of datasets used in our experiments.

Dataset	Usage	Setting	#Sample	Description
Training Set [Qu et al., 2023]	Train	Source domain	120,000	A synthetic training set of images with copy-move, splicing, and print-based edits on contracts, receipts, and books.
Test [Qu et al., 2023]	Test	In-domain	30,000	A general synthetic test set with copy-move, splicing, and print-based edits.
FCD [Qu et al., 2023]	Test	In-domain	2,000	A synthetic test set with copy-move, splicing, and print-based edits, whose source images come from the Noisy Office Dataset.
SCD [Qu et al., 2023]	Test	In-domain	18,000	A synthetic test set with copy-move, splicing, and print-based edits, whose source images come from Huawei Cloud.
T-SROIE [Wang et al., 2022b]	Test	Cross-domain	1,579	Scanned receipts tampered using the AIGC text editing model SR-Net.
OSTF [Qu et al., 2025]	Test	Cross-domain	3,046	Natural scene text images tampered using eight different AIGC-based text editing models.
TPIC-13 [Wang et al., 2022a]	Test	Cross-domain	589	Naturally captured scene-text images tampered using the AIGC text editing model SR-Net.
RTM [Luo et al., 2025]	Test	Cross-domain	3,444	Document images with both synthetic and manual manipulations, covering copy-move, splicing, print, and erasure edits across diverse document types such as scanned forms.

Following the *Doc Protocol* in ForensicHub [Du et al., 2025], we use the fully synthetic training set from DocTamper [Qu et al., 2023] to train all models and evaluate on seven test sets covering both in-domain and cross-domain scenarios. The summary of these datasets is shown in Table 5. All samples in the in-domain *Test*, *FCD*, and *SCD* splits are also synthetic DocTamper samples, compressed once using the final JPEG quality-factor records provided by DocTamper, while all samples in the cross-domain test sets are cropped to 512×512 patches without additional compression. The training set and these three in-domain test sets share the same three forgery synthesis types: copy-move, splicing, and print-based edits. The cross-domain test sets have more diverse forgery sources, including AIGC-based text editing models and manual manipulation.

B Expert Assignment

RiSE uses different training-set partitions for different numbers of experts. RiSE-1 uses the entire training set as a single subset. RiSE-3 uses the forgery synthesis type as the subset label, which splits the training set into three subsets corresponding to copy-move (CM), splicing (SP), and generation (GE). RiSE-9 uses a finer expert-assignment label d_i for each training sample by combining two attributes, the forgery synthesis type s_i and the feature cluster index k_i. The forgery synthesis type is provided directly by the DocTamper metadata (see `https://github.com/qcf-568/DocTamper`). Each training sample belongs to one of the three synthesis categories CM, SP, or GE. The training set contains 35,000 CM samples, 45,000 SP samples, and 40,000 GE samples. We use this label as the first-level partition because these three manipulation sources exhibit different forensic traces and therefore benefit from separate expert specialization.

The feature cluster index is obtained from the single output query embedding of a trained RiSE-1 model. Specifically, RiSE sets $N_q = 1$, so for each training image the model produces one output query embedding, which we denote by $\mathbf{q}_i \in \mathbb{R}^d$. We use this single query embedding rather than spatially pooled dense features because pooled dense tokens can mix authentic and tampered regions, especially when the forged area is small, which entangles real and fake evidence in a single representation. In contrast, the output query embedding is optimized by the localization objective to summarize tampering-relevant evidence at the image level, making it a cleaner feature for clustering.

For each forgery type s, let $\mathcal{Q}_s = \{\mathbf{q}_i \mid s_i = s\}$. We first run standard K-means with N_c clusters on $\mathcal{Q}_s$ to obtain initial cluster centers $\{\boldsymbol{\mu}_{s,k}\}_{k=1}^{N_c}$. To prevent a few clusters from dominating the training

set, we then rebalance the assignments with a balanced-clustering step that approximately solves

$$\min_{\{k_i\}} \sum_{i:s_i=s} \|\mathbf{q}_i - \boldsymbol{\mu}_{s,k_i}\|_2^2 \quad \text{s.t.} \quad \big|\{i : s_i = s,\, k_i = k\}\big| \in \left\{ \left\lfloor \frac{|\mathcal{Q}_s|}{N_c} \right\rfloor, \left\lceil \frac{|\mathcal{Q}_s|}{N_c} \right\rceil \right\},\ \forall k.$$

In practice, we implement this balancing with a greedy reassignment procedure. Samples are sorted by their nearest-center distance, and each sample is assigned to the closest cluster whose capacity has not yet been filled. If all preferred clusters are already full, the sample is assigned to the currently smallest cluster. This produces nearly equal cluster sizes within each forgery type while staying close to the original K-means solution.

After obtaining both attributes, we define the final expert-assignment label as

$$d_i = (s_i - 1)N_c + k_i,$$

which yields $N_e = N_s N_c$ non-overlapping expert assignments.

C Effective Rank Computation

We compute effective rank from the empirical PCA spectrum of the saved penultimate features of DTD, FFDN, ADCD-Net and our RiSE-1 with 100 samples per dataset. For each image, we spatially average the penultimate feature map into a single feature vector, then stack the resulting vectors into a feature matrix $\mathbf{F} \in \mathbb{R}^{100\times d}$, where d is the feature dimension. After centering $\mathbf{F}$, we compute singular values $\{\alpha_a\}$ and convert them into normalized spectral weights

$$\rho_a = \frac{\eta_a}{\sum_b \eta_b}, \qquad \eta_a = \frac{\alpha_a^2}{100 - 1}.$$

Following the entropy-based definition used in low-rank analysis, the effective rank is

$$\mathrm{erank}(\mathbf{F}) = \exp\left(-\sum_a \rho_a \log \rho_a \right).$$

Intuitively, the effective rank measures how many principal directions are effectively used by the representation. If almost all variance is concentrated in one or two directions, the effective rank is close to 1, such as DTD, FFDN, ADCD-Net in Fig. 1. If variance is distributed more evenly across many directions, the effective rank increases, such as RiSE-1 in Fig. 1.

D Why TTA Averaging Stabilizes the Soft Dice Confidence

In this section, we explain why averaging predictions over Test-Time Augmentation (TTA) makes the Soft Dice Confidence (SDC) more stable than using a single prediction. Let $\mathbf{X}$ be the input image, E the expert model, $T_m(\cdot)$ the m-th augmentation, M the number of augmentations, and $N_{\mathrm{pix}} = HW$ the number of pixels. For spatial augmentations such as flips and rotations, the prediction must be mapped back to the original image coordinates before averaging, which we denote by $T_m^{-1}(\cdot)$. For non-spatial augmentations such as additive noise, T_m^{-1} is simply the identity map. The original prediction is the special case corresponding to the identity augmentation. Let

$$T_1, \ldots, T_M \overset{\text{i.i.d.}}{\sim} \mathcal{A}$$

be label-preserving augmentations drawn from an augmentation distribution $\mathcal{A}$. We denote the prediction for the m-th augmented output by $\hat{\mathbf{p}}_m$, the TTA average by $\bar{\mathbf{p}}_M$, and the ideal augmentation expectation by $\bar{\mathbf{p}}_{\mathcal{A}}$:

$$\hat{\mathbf{p}}_m := T_m^{-1}\big(E\big(T_m(\mathbf{X})\big)\big) \in [0,1]^{N_{\mathrm{pix}}}, \quad \bar{\mathbf{p}}_M := \frac{1}{M}\sum_{m=1}^{M} \hat{\mathbf{p}}_m, \quad \bar{\mathbf{p}}_{\mathcal{A}} := \mathbb{E}_{T\sim\mathcal{A}}\big[T^{-1}(E(T(\mathbf{X})))\big].$$

Variance reduction. For each pixel index $n \in \{1, \dots, N_{\text{pix}}\}$, let $\sigma^2(n) = \text{Var}_{T\sim\mathcal{A}}\big[\hat{\mathbf{p}}_m(n)\big]$. Since $\bar{\mathbf{p}}_{\mathcal{A}}(n) = \mathbb{E}[\hat{\mathbf{p}}_m(n)]$ and the $\hat{\mathbf{p}}_m$ are i.i.d., the variance of the sample mean at pixel n is

$$\mathbb{E}\big[(\bar{\mathbf{p}}_M(n) - \bar{\mathbf{p}}_{\mathcal{A}}(n))^2\big] = \text{Var}\left(\frac{1}{M}\sum_{m=1}^{M}\hat{\mathbf{p}}_m(n)\right) = \frac{1}{M^2}\sum_{m=1}^{M}\text{Var}\big[\hat{\mathbf{p}}_m(n)\big] = \frac{\sigma^2(n)}{M}.$$

To extend this pointwise result to the full prediction vector, let $\boldsymbol{\Delta} = \bar{\mathbf{p}}_M - \bar{\mathbf{p}}_{\mathcal{A}}$. Using $\|\boldsymbol{\Delta}\|_1 \le \sqrt{N_{\text{pix}}}\,\|\boldsymbol{\Delta}\|_2$ and Jensen's inequality, we obtain

$$\mathbb{E}\|\boldsymbol{\Delta}\|_1 \le \sqrt{N_{\text{pix}}}\,\mathbb{E}\|\boldsymbol{\Delta}\|_2 \le \sqrt{N_{\text{pix}}\,\mathbb{E}\|\boldsymbol{\Delta}\|_2^2} = \sqrt{N_{\text{pix}}\sum_{n=1}^{N_{\text{pix}}}\mathbb{E}[\Delta(n)^2]}.$$

Substituting the pixel-wise variance identity yields

$$\mathbb{E}\big[(\bar{\mathbf{p}}_M(n)-\bar{\mathbf{p}}_{\mathcal{A}}(n))^2\big] = \frac{\sigma^2(n)}{M}, \qquad \mathbb{E}\big\|\bar{\mathbf{p}}_M-\bar{\mathbf{p}}_{\mathcal{A}}\big\|_1 \le \sqrt{\frac{N_{\text{pix}}\sum_{n=1}^{N_{\text{pix}}}\sigma^2(n)}{M}} = \mathcal{O}\big(M^{-1/2}\big), \quad (1)$$

i.e., the TTA average concentrates around its augmentation expectation at the rate of $M^{-1/2}$, whereas a single prediction $\hat{\mathbf{p}}_m$ retains the full variance $\sigma^2(n)$.

Lipschitz continuity of SDC. Define

$$\text{SDC}(\hat{\mathbf{p}}, \hat{\mathbf{y}}) = \frac{2\hat{\mathbf{p}}^\top\hat{\mathbf{y}}}{\|\hat{\mathbf{p}}\|_1 + \|\hat{\mathbf{y}}\|_1}, \qquad \hat{\mathbf{y}} = \mathbb{1}[\hat{\mathbf{p}} > \tau].$$

Conditioned on the hard predicted mask $\hat{\mathbf{y}} \in \{0,1\}^{N_{\text{pix}}}$, we can view $\text{SDC}(\hat{\mathbf{p}})$ as a function of $\hat{\mathbf{p}}$ alone. Applying the quotient rule to the n-th entry $\hat{\mathbf{p}}(n)$ gives

$$\frac{\partial\,\text{SDC}(\hat{\mathbf{p}})}{\partial\hat{\mathbf{p}}(n)} = \frac{2\hat{y}(n)\big(\|\hat{\mathbf{p}}\|_1 + \|\hat{\mathbf{y}}\|_1\big) - 2\hat{\mathbf{p}}^\top\hat{\mathbf{y}}}{(\|\hat{\mathbf{p}}\|_1 + \|\hat{\mathbf{y}}\|_1)^2}, \qquad \left|\frac{\partial\,\text{SDC}(\hat{\mathbf{p}})}{\partial\hat{\mathbf{p}}(n)}\right| \le \frac{2}{\|\hat{\mathbf{p}}\|_1 + \|\hat{\mathbf{y}}\|_1}.$$

The inequality follows by considering the two cases $\hat{y}(n) = 0$ and $\hat{y}(n) = 1$. In both cases, the numerator has magnitude at most $2(\|\hat{\mathbf{p}}\|_1 + \|\hat{\mathbf{y}}\|_1)$ because $0 \le \hat{\mathbf{p}}^\top\hat{\mathbf{y}} \le \|\hat{\mathbf{p}}\|_1 + \|\hat{\mathbf{y}}\|_1$. Therefore, $\text{SDC}(\cdot)$ is locally L-Lipschitz in ℓ_1 along the segment between $\bar{\mathbf{p}}_M$ and $\bar{\mathbf{p}}_{\mathcal{A}}$, with $L \le 2/(\|\bar{\mathbf{p}}_{\mathcal{A}}\|_1 + \|\hat{\mathbf{y}}\|_1)$ up to local variation in the denominator. Hence,

$$\big|\text{SDC}(\bar{\mathbf{p}}_M) - \text{SDC}(\bar{\mathbf{p}}_{\mathcal{A}})\big| \;\le\; L\,\|\bar{\mathbf{p}}_M - \bar{\mathbf{p}}_{\mathcal{A}}\|_1. \quad (2)$$

Variance of SDC. Combining (1) and (2) yields

$$\mathbb{E}\big|\text{SDC}(\bar{\mathbf{p}}_M) - \text{SDC}(\bar{\mathbf{p}}_{\mathcal{A}})\big| \;\le\; \frac{2}{\|\bar{\mathbf{p}}_{\mathcal{A}}\|_1 + \|\hat{\mathbf{y}}\|_1}\sqrt{\frac{N_{\text{pix}}\sum_{n=1}^{N_{\text{pix}}}\sigma^2(n)}{M}} \;=\; \mathcal{O}\big(M^{-1/2}\big), \quad (3)$$

Equation (3) formalizes the stabilizing effect of TTA averaging on SDC. The single-pass quantity $\text{SDC}(\hat{\mathbf{p}})$ inherits the full augmentation-induced variability and can fluctuate substantially from one augmentation to another. In contrast, $\text{SDC}(\bar{\mathbf{p}}_M)$ is computed from an averaged prediction whose deviation from the augmentation expectation decreases at the rate of $M^{-1/2}$. As a result, the expert ranking induced by $\text{SDC}(\bar{\mathbf{p}}_M)$ is more stable to perturbations than the ranking induced by $\text{SDC}(\hat{\mathbf{p}})$.

E Moment Propagation in the Mask Module

This section gives the implementation details of the proposed Analytic Gaussian Propagation (AGP) used by the Latent Perturbation Confidence (LPC). We first summarize the deterministic forward pass of the mask module with the PyTorch-style Pseudocode 1, then give the Assumed Density Filtering (ADF) rules for propagating the mean and variance under latent Gaussian perturbations.

Mask module forward. For an image, the VFM produces output queries $\mathbf{Q}_{\text{vfm}} \in \mathbb{R}^{N_q \times d}$ and dense tokens $\mathbf{Z}_{\text{vfm}} \in \mathbb{R}^{H_p W_p \times d}$. Since RiSE uses a single query, the class head ϕ_{cls} can be ignored. The mask head maps them into mask-space features $\mathbf{Q}_{\text{mask}} = \phi_{\text{mlp}}(\mathbf{Q}_{\text{vfm}})$ and $\mathbf{Z}_{\text{mask}} = \phi_{\text{up}}(\mathbf{Z}_{\text{vfm}})$, where ϕ_{mlp} is a three-layer MLP with GELU activations and ϕ_{up} is a stack of blocks consisting of transposed-convolution, GELU, depth-wise convolution, and LayerNorm2d. The transformed query embedding and dense feature map are combined by the matching operation `torch.einsum("bqc,bchw->bqhw", Q_mask, Z_mask)`, followed by the sigmoid.

Pseudocode 1: Deterministic forward pass of the mask module.

```
# Q_vfm: [B, N_q, d], Z_vfm: [B, Hp*Wp, d]
def mask_module_forward(Q_vfm, Z_vfm):
    Z_vfm = Z_vfm.transpose(1, 2).reshape(B, d, Hp, Wp)

    Q_mask = phi_mlp(Q_vfm)       # Linear-GELU-Linear-GELU-Linear
    Z_mask = phi_up(Z_vfm)        # repeated Conv-GELU-Conv-LayerNorm2d blocks
    mask_logits = torch.einsum("bqc,bchw->bqhw", Q_mask, Z_mask)
    probs = mask_logits.sigmoid()
    return probs
```

Moment Propagation. LPC perturbs only the dense feature $\mathbf{Z}_{\text{vfm}}$ with Gaussian noise and the expected latent-perturbed output is

$$\bar{\mathbf{p}}_{\text{lat}} = \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 I)} \left[\phi_{\text{mask}}(\mathbf{Q}_{\text{vfm}}, \mathbf{Z}_{\text{vfm}} + \boldsymbol{\epsilon}) \right],$$

where $\phi_{\text{mask}}(\mathbf{Q}_{\text{vfm}}, \cdot)$ is the mask module with deterministic query input $\mathbf{Q}_{\text{vfm}}$. We define ϕ_{mask} as the composition of all layers in the mask module:

$$\phi_{\text{mask}} = \phi_{\text{mask}}^{(L)} \circ \cdots \circ \phi_{\text{mask}}^{(1)}.$$

Since the nonlinear layers in ϕ_{mask} do not preserve Gaussianity, the expectation is not available in closed form. We therefore apply ADF, which propagates a Gaussian approximation through the mask module. Let $\mathbf{Z}^{(0)} = \mathbf{Z}_{\text{vfm}} + \boldsymbol{\epsilon}$ and

$$\mathbf{Z}^{(\ell)} = \phi_{\text{mask}}^{(\ell)}(\mathbf{Z}^{(\ell-1)}), \qquad \ell = 1, \ldots, L.$$

ADF maintains at each layer

$$q_\ell(\mathbf{Z}^{(\ell)}) \approx \mathcal{N}(\boldsymbol{\mu}_\ell, \operatorname{diag}(\mathbf{v}_\ell)).$$

Here $q_\ell(\mathbf{Z}^{(\ell)})$ denotes the ADF approximate distribution of the random intermediate feature $\mathbf{Z}^{(\ell)}$ at layer ℓ. Starting from $q_0(\mathbf{Z}^{(0)}) = \mathcal{N}(\mathbf{Z}_{\text{vfm}}, \sigma^2 \mathbf{I})$, each layer takes a Gaussian input, applies $\phi_{\text{mask}}^{(\ell)}$, and then re-approximates the resulting output by matching its mean and variance. The query projection $\mathbf{Q}_{\text{mask}} = \phi_{\text{mlp}}(\mathbf{Q}_{\text{vfm}})$ is deterministic, so uncertainty is propagated only through $\phi_{\text{up}}(\mathbf{Z}_{\text{vfm}} + \boldsymbol{\epsilon})$ and the final matching operation. Tables 6 and 7 summarize the corresponding mean and variance updates. In these tables, $(\boldsymbol{\mu}_x, \mathbf{v}_x)$ denotes the input mean and variance tensors, while $(\boldsymbol{\mu}_y, \mathbf{v}_y)$ denotes the output mean and variance tensors. The ADF computation is summarized with the PyTorch-style Pseudocode 2. After AGP produces the probability mean $\bar{\mathbf{p}}_{\text{lat}}$, LPC computes SDC on $\bar{\mathbf{p}}_{\text{lat}}$ for confidence. The final prediction is still taken from the original, unperturbed output.

Table 6: ADF mean propagation rules in the mask module.

Operation	Formulation	Notes
Convolution	$\boldsymbol{\mu}_y = \mathbf{W} * \boldsymbol{\mu}_x + \mathbf{b}$	$\mathbf{W}$ is the convolution kernel and $*$ denotes convolution. The matching operation `torch.einsum("bqc,bchw->bqhw", Q_mask, Z_mask)` can be viewed as query-wise 1×1 convolution with deterministic weights $\mathbf{Q}_{\text{mask}}$.
GELU	$\boldsymbol{\mu}_y \approx \frac{1}{\sqrt{\pi}} \sum_k w_k g(\sqrt{2\mathbf{v}_x}\,\xi_k + \boldsymbol{\mu}_x)$	It is a Gauss–Hermite quadrature approximation under the Gaussian input, using nodes and weights (ξ_k, w_k) with default $K = 5$ and g is GELU.
LayerNorm2d	$\boldsymbol{\mu}_{y,c} = \frac{(\boldsymbol{\mu}_{x,c} - \bar{\boldsymbol{\mu}}_x)\gamma_c}{\sqrt{\mathbf{s}^2_{\text{LN}}}} + \beta_c$	$\bar{\boldsymbol{\mu}}_x = \frac{1}{d}\sum_c \boldsymbol{\mu}_{x,c}$ is the input channel mean, $\mathbf{s}^2_{\text{LN}} \approx \text{Var}_c(\boldsymbol{\mu}_{x,c}) + (1 - \frac{1}{d})\frac{1}{d}\sum_c \mathbf{v}_{x,c}$ is the expected channel variance and γ_c, β_c are the LayerNorm affine parameters.
Sigmoid	$\boldsymbol{\mu}_y \approx \frac{1}{2}\left[\sigma(\boldsymbol{\mu}_x - \sqrt{\mathbf{v}_x}) + \sigma(\boldsymbol{\mu}_x + \sqrt{\mathbf{v}_x})\right]$	This is the 2-point Gauss–Hermite quadrature for $\mathbb{E}[\sigma(z)]$ with $z \sim \mathcal{N}(\boldsymbol{\mu}_x, \mathbf{v}_x)$: after rescaling the Gaussian variable, the standard nodes become $\boldsymbol{\mu}_x \pm \sqrt{\mathbf{v}_x}$ and both have weight $\frac{1}{2}$.

Table 7: ADF variance propagation rules in the mask module.

Operation	Formulation	Notes
Convolution	$\mathbf{v}_y = (\mathbf{W} \odot \mathbf{W}) * \mathbf{v}_x$	$\odot$ denotes element-wise multiplication. The matching operation `torch.einsum("bqc,bchw->bqhw", Q_mask, Z_mask)` likewise follows this rule with squared deterministic query weights $\mathbf{Q}_{\text{mask}} \odot \mathbf{Q}_{\text{mask}}$, which yields the variance update used in Pseudocode 2.
GELU	$\mathbf{v}_y \approx \frac{1}{\sqrt{\pi}} \sum_k w_k g(\sqrt{2\mathbf{v}_x}\,\xi_k + \boldsymbol{\mu}_x)^2 - \boldsymbol{\mu}_y^2$	$\boldsymbol{\mu}_y$ is the GELU mean approximation from the mean table.
LayerNorm2d	$\mathbf{v}_{y,c} = \frac{\tilde{\mathbf{v}}_{x,c}\gamma_c^2}{\mathbf{s}^2_{\text{LN}}}$	$\tilde{\mathbf{v}}_{x,c} = (1 - \frac{2}{d})\mathbf{v}_{x,c} + \frac{1}{d^2}\sum_{c'} \mathbf{v}_{x,c'}$ is the variance after channel centering

Pseudocode 2: ADF computation in the mask module.

```
# Q_vfm: [B, N_q, d], Z_vfm: [B, Hp*Wp, d]
def mask_module_agp(Q_vfm, Z_vfm, sigma):
    Z_vfm = Z_vfm.transpose(1, 2).reshape(B, d, Hp, Wp)

    mu = Z_vfm
    var = torch.full_like(mu, sigma ** 2)

    Q_mask = phi_mlp(Q_vfm)              # deterministic query projection
    mu_u, var_u = agp_phi_up(mu, var)    # Conv-GELU-Conv-LayerNorm2d AGP rules
    mu_m = torch.einsum("bqc,bchw->bqhw", Q_mask, mu_u)
    var_m = torch.einsum("bqc,bchw->bqhw", Q_mask.square(), var_u)

    mu_p = agp_sigmoid_mean(mu_m, var_m)
    return mu_p                          # mu_p is the LPC-averaged probability map
```

F Additional Experiments and Analyses

This section examines how the learned embeddings partition forgery types, whether those partitions need type metadata, and how individual experts behave across datasets. We then study routing, transfer to natural images, robustness to image transformations, and inference latency. Unless otherwise stated, F1 is the localization metric.

F.1 Cluster Quality

We assess whether RiSE-1 query embeddings contain forgery-related structure using 3,000 DocTamper training images, with 1,000 images sampled from each of the generation (GE), copy-move (CM), and splicing (SP) categories. K-means is applied to the query embeddings, and the forgery types color the same points in the left panels of Figs. 7 and 8. The t-SNE projection is used only for visualization. Overall, the embeddings show a clear cluster between the three forgery types. With a simply K-mean

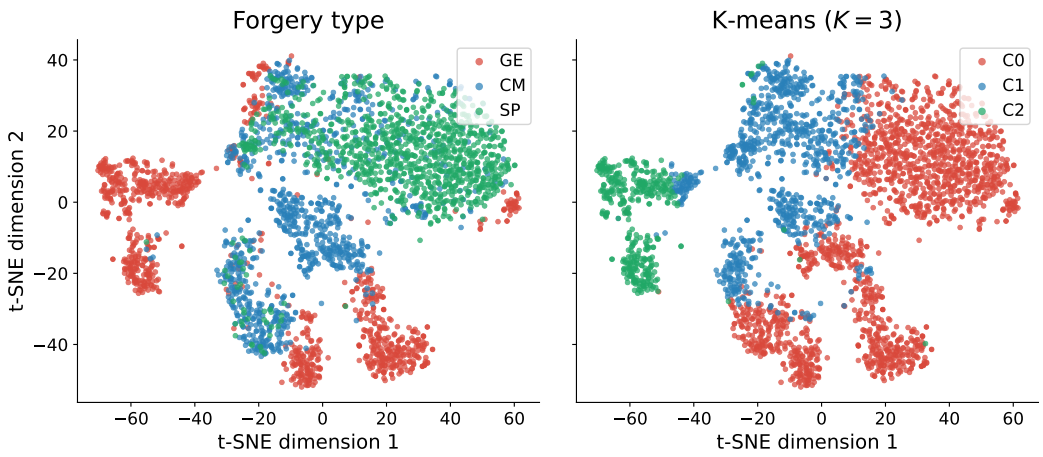


Figure 7: RiSE-1 query embeddings for 3,000 balanced training images. The same points are colored by forgery type (left) and $K = 3$ cluster (right).

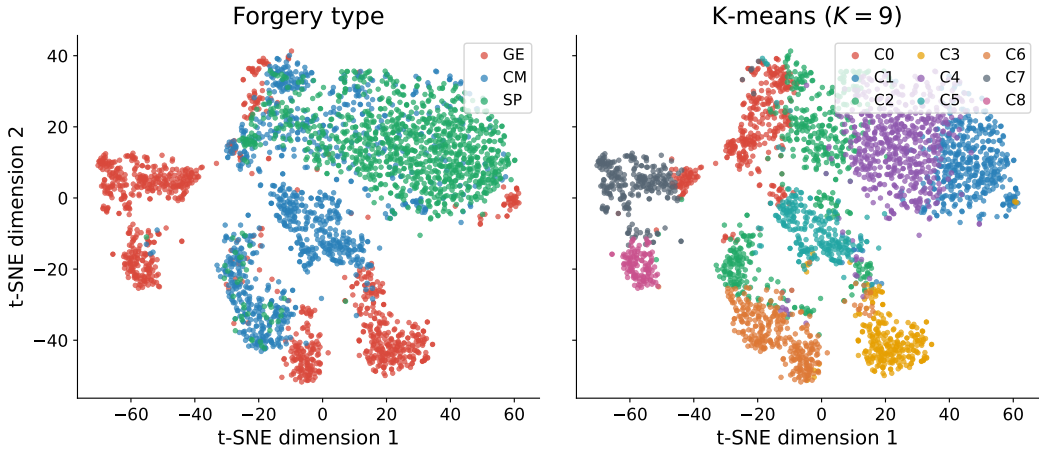


Figure 8: The same embeddings colored by forgery type (left) and $K = 9$ cluster (right).

partition, the clusters broadly align with the forgery types, but there is also overlap between clusters and forgery types. The right panels of Figs. 7 and 8 show the same embeddings colored by cluster assignment for $K = 3$ and $K = 9$, respectively. We also report silhouette scores over five random seeds to show the stability of the clustering results in Table 8.

F.2 Metadata-Free Partitioning

Since the RiSE-3 require forgery type metadata to assign experts, we investigate whether the expert partition can be learned without metadata. Table 9 compares the two partitions of RiSE-3 with both LPC selection and oracle selection. In the metadata-based partition, we use the three ground-truth forgery types, while the metadata-free partition uses the K-means ($K = 3$) clusters. The overall F1 of the two partitions is similar, with the metadata-free partition slightly outperforming the metadata-based one. The results suggest that the learned embeddings can be used to partition the training set without requiring metadata, which is useful for datasets where forgery type labels are unavailable.

F.3 Per-Expert Performance

Tables 10 and 11 show the per-expert F1 scores for RiSE-3 and RiSE-9. RiSE-3 experts E_{GE}, E_{CM}, and E_{SP} are trained on the corresponding forgery-type subsets. RiSE-9 splits each forgery subset

Table 8: Silhouette scores of the embedding clusters over five random seeds (mean $\pm$ standard deviation).

Metric	$K = 3$	$K = 9$
Silhouette score	0.1232 ± 0.0000	0.0789 ± 0.0001

Table 9: RiSE-3 F1 with metadata-based and metadata-free partitions.

Dataset	Metadata-based / LPC	Metadata-based / oracle	Metadata-free / LPC	Metadata-free / oracle
FCD	0.710	0.746	0.771	0.801
SCD	0.643	0.707	0.614	0.669
Test	0.721	0.757	0.683	0.723
AvgI	0.691	0.737	0.689	0.731
T-SROIE	0.760	0.815	0.783	0.820
OSTF	0.625	0.658	0.588	0.615
TPIC-13	0.740	0.766	0.747	0.764
RTM	0.173	0.221	0.173	0.213
AvgC	0.575	0.615	0.573	0.603
Avg. all	0.625	0.667	0.623	0.658

into three smaller clusters, yielding $E_{t,j}$ for type t and cluster j. The large performance variation among experts shows that different experts capture distinct forgery features and document semantics, which directly aligns with our expert-scaling motivation. For example, RiSE-3 E_{GE} is strongest on T-SROIE, TPIC-13, and OSTF, whereas E_{CM} and E_{SP} perform better on several in-domain sets and RTM. The RiSE average F1 exceeds that of every individual expert, reaching 0.625 for RiSE-3 and 0.650 for RiSE-9, while remaining close to the oracle scores of 0.667 and 0.723. RTM remains difficult for every expert and even for the oracle, which confirms that the main limitation is insufficient manual-forgery expertise.

F.4 LPC Routing Accuracy

We measure LPC routing accuracy against the oracle-selected expert. For image $\mathbf{X}_i$, let j_i^* denote the oracle expert index, and let $\mathcal{R}_i^{\mathrm{LPC}}(k)$ denote the index set of k experts with the highest LPC confidence. For N images, we define

$$\text{LPC Top-}k\text{ ACC} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left[j_i^* \in \mathcal{R}_i^{\mathrm{LPC}}(k)\right].$$

The oracle uses the ground-truth mask to choose the best expert for each image, so it is an unreachable reference for deployment. Table 12 reports the routing accuracy for RiSE-3 and RiSE-9. For RiSE-9, the oracle expert is ranked within the top-3 candidates with 0.766 average accuracy and within the top-5 with 0.884 average accuracy. This indicates that LPC retains useful expert preference information despite the harder 9-expert selection problem. RTM remains the most difficult domain, with 0.221 Top-1 and 0.689 Top-5 accuracy, consistent with its manual manipulations being hard to represent by the source-domain experts. Fig. 9 compares the aggregate assignment counts; LPC and the oracle distribute images differently among experts. These counts describe routing frequency, whereas exact expert agreement and end-to-end localization performance measure different outcomes.

Table 10: Per-expert F1 for RiSE-3. Each row is one expert trained on the generation (GE), copy-move (CM), or splicing (SP) subset. Dataset order matches Table 11

Expert	Test	FCD	SCD	T-SROIE	TPIC-13	RTM	OSTF	Avg.
E_{GE}	0.393	0.600	0.445	0.743	0.763	0.113	0.631	0.527
E_{CM}	0.605	0.634	0.556	0.605	0.231	0.148	0.227	0.430
E_{SP}	0.571	0.669	0.483	0.519	0.248	0.114	0.286	0.413

Table 11: Per-expert F1 scores for RiSE-9. Each row is one expert $E_{t,j}$ for forgery type $t \in \{\mathrm{GE}, \mathrm{CM}, \mathrm{SP}\}$ and within-type cluster $j \in \{0, 1, 2\}$

Expert	Test	FCD	SCD	T-SROIE	TPIC-13	RTM	OSTF	Avg.
$E_{\mathrm{GE},0}$	0.335	0.496	0.433	0.783	0.672	0.089	0.542	0.479
$E_{\mathrm{GE},1}$	0.244	0.449	0.195	0.519	0.698	0.096	0.580	0.397
$E_{\mathrm{GE},2}$	0.407	0.594	0.494	0.749	0.798	0.122	0.683	0.550
$E_{\mathrm{CM},0}$	0.513	0.553	0.505	0.365	0.182	0.113	0.222	0.350
$E_{\mathrm{CM},1}$	0.587	0.727	0.466	0.598	0.216	0.136	0.237	0.424
$E_{\mathrm{CM},2}$	0.452	0.480	0.520	0.630	0.176	0.117	0.140	0.359
$E_{\mathrm{SP},0}$	0.513	0.512	0.451	0.487	0.126	0.080	0.153	0.332
$E_{\mathrm{SP},1}$	0.586	0.679	0.498	0.554	0.259	0.120	0.276	0.425
$E_{\mathrm{SP},2}$	0.501	0.477	0.434	0.450	0.128	0.077	0.161	0.318

Table 12: LPC routing accuracy against the oracle-selected expert.

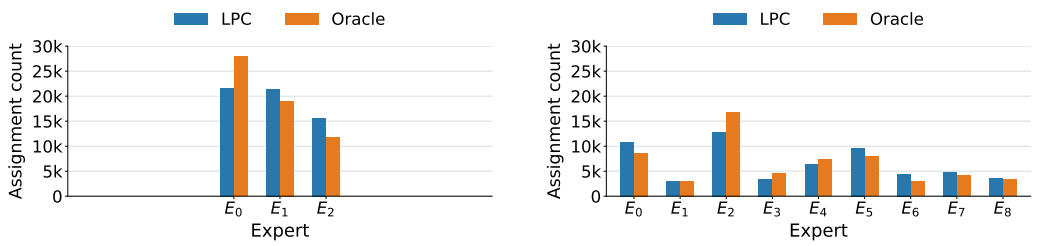
Dataset	RiSE-3 Top-1	RiSE-9 Top-1	RiSE-9 Top-3	RiSE-9 Top-5
Test	0.620	0.483	0.754	0.887
FCD	0.678	0.644	0.856	0.938
SCD	0.572	0.311	0.695	0.851
T-SROIE	0.630	0.346	0.734	0.889
TPIC-13	0.855	0.544	0.946	0.978
RTM	0.425	0.221	0.491	0.689
OSTF	0.776	0.501	0.889	0.957
Average	0.651	0.435	0.766	0.884

F.5 Zero-Shot Transfer to Natural-Image Forgeries

All methods in Table 13 are trained on DocTamper only and evaluated without target-domain adaptation on six natural-image forgery datasets. Absolute F1 is low under this substantial shift from text-on-white documents to natural scenes, people, animals, and objects. RiSE-9 achieves the highest mean F1 of 0.156, compared with 0.077 for the strongest baseline, DTD.

F.6 Robustness to JPEG Recompression and Resizing

We additionally evaluate the robustness of RiSE-9 and RiSE-9-DCT on the images of T-SROIE, TPIC-13, and OSTF under JPEG recompression and resizing against ADCD-Net. ADCD-Net is a strong robust baseline specifically designed for forgery localization under various distortions including JPEG, resize, and crop. Table 14 reports the F1 scores, and Avg. across all distortions is reported. RiSE-9 achieves the highest average F1 on all three datasets and remains robust under JPEG recompression and resizing. Relative to ADCD-Net, RiSE-9 improves the average F1 by 10.4% on T-SROIE, 44.2% on TPIC-13, and 65.2% on OSTF. RiSE-9-DCT also consistently outperforms



(a) RiSE-3 (b) RiSE-9

Figure 9: Expert selection counts across seven test sets: LPC assignments and ground-truth oracle choices for (a) RiSE-3 and (b) RiSE-9.

Table 13: Zero-shot F1 on six natural-image forgery datasets. All methods are trained only on DocTamper.

Method	NIST16	Columbia	DSO	CASIA	CASIA2	IMD2020	Avg.
DTD	0.059	0.198	0.116	0.001	0.014	0.076	0.077
FFDN	0.026	0.016	0.001	0.006	0.010	0.017	0.013
ADCD-Net	0.103	0.030	0.047	0.019	0.038	0.060	0.049
RiSE-1	0.052	0.003	0.005	0.060	0.046	0.045	0.035
RiSE-3	0.117	0.031	0.026	0.141	0.110	0.155	0.097
RiSE-9	0.216	0.072	0.030	0.280	0.148	0.190	0.156

Table 14: Robustness under JPEG recompression (J90, J80, J70: quality factors) and resizing (R90, R80, R70: retained linear scale in percent). Avg. averages the six transformed conditions; Clean is reported separately.

Dataset	Method	Clean	J90	J80	J70	R90	R80	R70	Avg.
T-SROIE	ADCD-Net	0.623	0.571	0.538	0.508	0.757	0.748	0.755	0.646
	RiSE-9-DCT	0.729	0.679	0.638	0.608	0.774	0.780	0.750	0.705
	RiSE-9	0.786	0.697	0.660	0.631	0.762	0.772	0.759	0.713
TPIC-13	ADCD-Net	0.579	0.599	0.499	0.436	0.516	0.515	0.513	0.513
	RiSE-9-DCT	0.713	0.714	0.625	0.555	0.693	0.682	0.676	0.658
	RiSE-9	0.788	0.777	0.722	0.655	0.768	0.761	0.759	0.740
OSTF	ADCD-Net	0.441	0.434	0.411	0.382	0.396	0.397	0.390	0.402
	RiSE-9-DCT	0.597	0.628	0.609	0.580	0.609	0.606	0.609	0.607
	RiSE-9	0.688	0.679	0.667	0.644	0.666	0.663	0.667	0.664

ADCD-Net, showing strong robustness when the DCT pathway is included with our DCT dropout strategy.

F.7 Single-GPU Inference Latency

To quantify the practical trade-off, we measure single-image latency for 512×512 inputs on a single RTX 4090. Table 15 reports the mean and standard deviation across 100 samples, together with the cross-domain AvgC F1. Each RiSE configuration is Pareto-efficient in the latency–AvgC plane. RiSE-1 is within 6.7 ms of DTD and 2.3 ms of FFDN, yet improves AvgC by 0.295 and 0.242, respectively. RiSE-3 strictly dominates ADCD-Net, with both lower latency (0.1165 vs. 0.1171 s) and higher AvgC (0.575 vs. 0.450). RiSE-9 is the accuracy-prioritized Pareto-efficient option. All 9 experts run in parallel, fit on one RTX 4090 using 21,691 MiB, and process a 512×512 image in 0.3571 s to achieve the highest AvgC of 0.611. This suits server-side document verification, such as banking/KYC, where a sub-second decision is practical and missing a forgery is substantially more costly than the additional latency.

Table 15: Cross-domain AvgC F1 and single-image latency on one NVIDIA RTX 4090. Latency is reported as mean $\pm$ standard deviation over 100 inputs.

Model	AvgC F1	Latency (s)
DTD	0.247	0.0407 ± 0.0006
FFDN	0.300	0.0451 ± 0.0009
RiSE-1	0.542	0.0474 ± 0.0002
ADCD-Net	0.450	0.1171 ± 0.0001
RiSE-3	0.575	0.1165 ± 0.0004
RiSE-9	0.611	0.3571 ± 0.0005

G Efficiency

Table 16 reports the measured efficiency of the RiSE variants on 4 NVIDIA RTX 4090 GPUs.

Parameter count. Relative to RiSE-1, scaling to RiSE-3 and RiSE-9 increases the total parameter count by only $1.003\times$ and $1.011\times$, while the trainable parameter count increases by $1.07\times$ and $1.29\times$. This mild growth is consistent with the RiSE design, where the dominant VFM parameters and the mask module are shared, and each additional expert contributes only a small residual subspace.

Training time. We adopt an MoE-style implementation so that multiple RiSE experts are optimized within one unified training process rather than launched as fully separate training jobs. Unlike conventional MoE, RiSE does not introduce a learnable router during training. Instead, the routing path is deterministically given by the expert-assignment label for each sample described in Section B. This fixed-routing implementation keeps the experts specialized on their assigned training subsets while reducing the overhead compared with training each expert completely in isolation. The main extra cost appears during training because more experts still require more forward and optimization runs, increasing training time by $2.66\times$ for RiSE-3 and $7.08\times$ for RiSE-9 relative to RiSE-1. Nevertheless, RiSE remains efficient in the training phase because each expert is trained for only 10k steps, which is $10\times$ fewer than the 100k steps commonly used by existing TFL detectors [Du et al., 2025]. On the same 4 RTX 4090 GPUs, the strong TFL baseline ADCD-Net requires approximately 27 training hours, so even RiSE-9 is still about $3.4\times$ faster to train.

Inference time. For inference, we convert throughput to per-sample latency in seconds. Since the experts can be executed in parallel, inference time does not grow linearly with the number of experts. Compared with RiSE-1, the latency increases by only $1.61\times$ for RiSE-3 and $4.23\times$ for RiSE-9, even though the number of experts increases by $3\times$ and $9\times$, respectively. RiSE-9 still remains below 0.1 seconds per sample on the same 4-GPU setup.

Table 16: Efficiency comparison of non-DCT RiSE variants measured on 4 NVIDIA RTX 4090 GPUs. Training time is reported in hours; inference time is converted from samples-per-second throughput to seconds per sample.

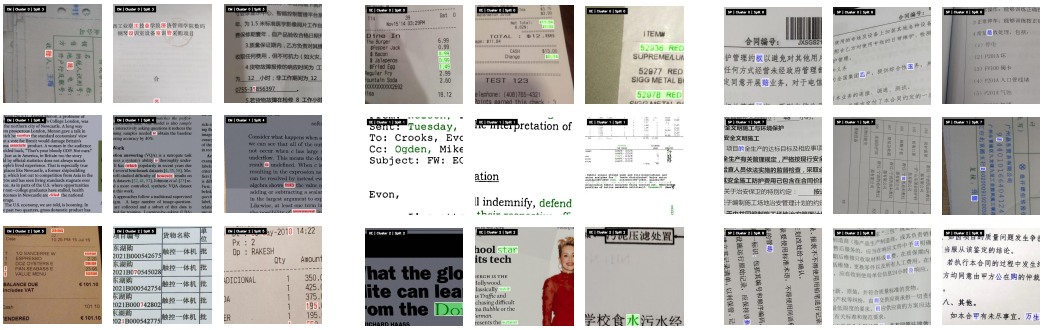
Method	#Total Params (M)	#Trainable Params (M)	Training Time (hour)	Inference Time (s/sample)
RiSE-1	315.14	12.20	1.12	0.0219
RiSE-3	316.02	13.09	2.98	0.0353
RiSE-9	318.68	15.75	7.93	0.0927

G.1 Lightweight Expert Selection

Server-side forensic analysis. Inference cost increases with expert count, but our target TFL deployment is server-side forensic analysis rather than real-time edge inference. This setting can use powerful GPU resources and does not require real-time processing. It is consistent with practical document-verification scenarios such as online banking and Know Your Customer (KYC), where images are submitted for automated forensic analysis. Recent work also frames ID-document forgery detection and localization in online authentication settings [Korshunov et al., 2025].

Dynamic early exit. A dynamic early-exit mechanism is a natural extension. EoMT could decode segmentation predictions from intermediate ViT blocks with homogeneous feature shapes, allowing LPC-based confidence evaluation at intermediate layers for early termination, analogous to confident layer decoding in LLMs [Zhang et al., 2026]. Developing and evaluating such a strategy is beyond the scope of this paper, and we leave it for future work.

Expert merging. Merging expert models into a unified model is another promising direction. Existing work on task arithmetic, common and task-specific subspaces, and SVD-based alignment for LoRA merging could provide practical starting points [Ilharco et al., 2023, Marczak et al., 2025, Stoica et al., 2025]. Adapting these methods to RiSE's residual subspace experts and evaluating the resulting accuracy–efficiency trade-off remain future work.



(a) Copy-move (b) Generation (c) Splicing

Figure 10: Visualization of training samples and expert assignments. Each forgery synthesis type is further divided into three feature clusters, producing nine expert assignments in total. Within each subplot, each row shows samples from a single feature cluster.

H Visualization

We visualize the samples for each subset of the training set in Fig. 10 and the corresponding expert assignments. The training set contains three forgery synthesis types: copy-move, splicing, and generation. Each type is further clustered into three feature clusters based on the query embedding of a trained RiSE-1 model, resulting in a total of nine expert assignments. The visualization shows that samples from different forgery types and feature clusters exhibit distinct visual characteristics, forgery traces, capturing conditions, and document semantics, which motivates the design of RiSE to learn specialized experts for each cluster.

I Limitation

RiSE has several limitations. First, scaling the number of residual experts improves cross-domain F1 but requires evaluating all experts at inference. In the single-GPU measurement, RiSE-9 uses 21,691 MiB of memory and takes 0.3571 seconds per 512×512 image; this limits its use in resource-constrained settings. Early-exit expert pruning and residual-expert merging are promising directions for reducing this cost, but we have not evaluated them here. Second, LPC still leaves a gap to the oracle selector, and exact top-1 routing agreement is modest for RiSE-9. Third, RTM remains difficult because the source training data provide limited coverage of manual erasure, physical edits, and other handcrafted manipulations. Our added corruption experiments cover JPEG recompression and resizing; dedicated print-and-scan robustness evaluation remains future work.

J Broader Impacts

The proliferation of document forgery poses significant risks to information integrity, financial security, and legal processes. By improving the robustness of document forgery detection across diverse manipulation types and domains, our work contributes to safeguarding against these threats. However, it is important to consider potential misuse of such detection technologies, such as infringing on privacy or being used for surveillance. We advocate for responsible deployment of forgery detection systems with appropriate safeguards to balance security benefits with ethical considerations.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims match theoretical and experimental results.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a dedicated section discussing the limitations of the proposed approach.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The paper provides all necessary assumptions and complete proofs for the theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper provides all necessary details for reproducing the main experimental results, including dataset descriptions, model architectures, and training procedures.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The paper provides all necessary instructions and resources to reproduce the main experimental results, including access to code and model checkpoints.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper provides all necessary details for reproducing the main experimental results, including dataset descriptions, model architectures, and training procedures.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper evaluates the proposed method on large-scale test sets.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provides the hardware specifications and computational requirements for reproducing the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper adheres to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses both potential positive and negative societal impacts of the work performed.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper properly credits all creators and original owners of assets used, and the licenses and terms of use are explicitly mentioned and respected.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper provides detailed documentation for all new assets introduced, including datasets, code, and models.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The paper does not use LLMs as an important, original, or non-standard component.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.