Efficient Document Tampering Localization with Multi-Level Discrepancy Features and Unified DCT–Quantization Embedding

Mohamed Dhouib[1], Ye Zhu[1], Sonia Vanier[1], and Aymen Shabou[2]

[1] LIX, École Polytechnique, IP Paris, France
[2] DataLab Groupe, Crédit Agricole S.A., France
mohamed.dhouib@polytechnique.edu

Abstract. Localizing document tampering is extremely challenging, as manipulations are crafted to appear visually consistent and often leave only subtle traces that are nearly invisible to the human eye. In prior work, evaluation has been largely dominated by synthetic benchmarks that closely match the training distribution, and methods have shown steady progress under this setting. However, these gains often translate poorly to human-made forgeries and to cross-domain evaluation, where both the source documents and the tampering pipeline can change, leading to a distribution shift. In addition, since the introduction of the Frequency Perception Head for the discrete cosine transform (DCT) modality, it has become a standard choice, and subsequent work has largely focused on downstream modules and fusion strategies rather than revisiting the backbone itself. To help close this gap in cross-domain performance and improve the DCT backbone design, we propose **DiffNet**, a relatively simple yet effective RGB–DCT early-fusion architecture driven by two key design choices. First, to ensure that the decoder aggregates multi-scale inconsistency evidence rather than operating on raw, content-heavy activations, we apply a lightweight multi-level discrepancy transformation at the output of each backbone stage, replacing features with magnitude-only responses to learned zero-sum filters. Second, we design an efficient DCT-domain backbone that relies on a lightweight frequency-index-aware DCT–quantization joint embedding. Our approach achieves state-of-the-art performance on cross-domain and human-made document tampering localization, outperforming prior methods by around 30%, with up to 7× higher throughput than the previous best model.

Keywords: Document Tampering · Document forensics · Forgery detection · Image manipulation localization · DCT and quantization tables

1 Introduction

Document images are central to financial, administrative, and identity-related workflows, making them attractive targets for tampering. Such manipulations, if undetected, can enable fraud and identity theft, undermine legal evidence,

and trigger costly downstream errors in automated decision-making systems. Yet state-of-the-art models [6, 18, 33] still generalize poorly to cross-domain and human-made forgeries, and evaluation has mainly been conducted on synthetic tampering benchmarks [6, 9, 24, 33] that often remain close to the training distribution.

Because annotated human-made tampering data are scarce and costly to collect, prior work typically resorts to large-scale synthetic pretraining as a practical necessity. However, this setting encourages shortcut learning. Synthetic generation pipelines can introduce systematic signatures. Inserted text is generally rendered from a limited font set or a fixed generator family, while coverage often reuses a small set of algorithms that leave characteristic boundary or texture artifacts. Similar repetitive signatures arise in copy-move, splicing, and blending operations, enabling models to rely on pipeline-specific fingerprints instead of the underlying *inconsistencies* that manipulations create and that are more likely to transfer across domains and forgery tools.

Moreover, synthetic datasets frequently come from a narrow set of source domains and acquisition pipelines, and for underrepresented document types such as payslips, bank statements, and tax forms, they often generate many manipulated variants from the same source image. For instance, [7] draws a substantial portion of its source documents from IIT-CDIP [15], a collection originating from tobacco litigation materials which represents a highly specific provenance. As a result, models can reduce training loss in hard cases by exploiting domain-specific cues rather than focusing on the inconsistencies the tampering introduces. These issues can persist even in human-made document tampering datasets, which are often small [1, 30] and produced by few experts.

This motivates us to tackle the task from a different direction: rather than further increasing architectural complexity, we constrain the feature pyramid so the decoder becomes mainly driven by inconsistency cues. Specifically, we introduce a lightweight *multi-level discrepancy transformation* applied at the output of every backbone stage before multi-level fusion and prediction. It converts stage features into magnitude-only responses of learned zero-sum filters, yielding a sign-invariant discrepancy representation across the entire feature pyramid. As a result, multi-level fusion aggregates inconsistency cues across stages rather than operating on raw content-heavy activations.

Beyond RGB cues, recent work has emphasized frequency evidence, in particular DCT coefficients, and has developed several RGB–DCT fusion architectures [6, 13, 18, 24, 33]. The Frequency Perception Head [24] (FPH) has emerged as a standard choice in several recent state-of-the-art document tampering architectures [6, 25, 33], which have mainly emphasized stronger fusion strategies and auxiliary enhancement modules, while leaving the encoder backbone mostly unchanged. In contrast, we propose a more efficient alternative to the Frequency Perception Head. Since both quantized DCT coefficients and quantization-table entries are discrete integers, we leverage the expressiveness of embedding layers to map each coefficient–quantization pair to a compact joint embedding, explic-

itly injecting the frequency index while avoiding early high-dimensional feature expansions and auxiliary blocks.

Together, these two design choices yield **DiffNet**, a simple and efficient architecture that improves both cross-domain accuracy and throughput. On cross-domain evaluation, our architecture improves upon the previous best by around 30% while increasing throughput by up to 7×. Our contributions are as follows:

- We introduce a multi-level discrepancy transformation applied at every backbone stage output, converting features into magnitude-only responses of learned zero-sum filters and yielding a sign-invariant discrepancy representation across the feature pyramid.
- We design an efficient DCT-domain backbone based on a frequency-index-aware joint embedding of quantized DCT coefficients and quantization tables, providing a strong and lightweight frequency branch for early fusion.
- We run extensive ablations on both components and their main design choices, confirming the contribution of each part and supporting the design decisions within each component.
- Combining these components in a simple RGB–DCT early-fusion architecture yields a new state-of-the-art on human-made document tampering detection and localization, improving over the previous best by around 30% while substantially increasing throughput. The code and pretrained weights are available at `https://github.com/Mohamed-Dhouib/DiffNet`.

2 Related Work

2.1 Document tampering detection and localization datasets

Progress in document tampering localization has long been constrained by the limited availability of public, annotated data. Early works [4, 14, 27] were therefore developed and evaluated on private collections. [32] introduced T-SROIE, a small benchmark centered on character-level replacements produced by SRNet [35]. While useful for evaluation, such synthetic edits can exhibit systematic visual artifacts and may overestimate real-world performance. Later human-made datasets such as FindIt [1], FindItAgain [30], and RTM [18] were introduced, providing human-made manipulations that better reflect real-world scenarios, but remain comparatively limited in scale relative to synthetic corpora, making them more suitable for evaluation and fine-tuning than for pretraining. To enable large-scale training, DocTamper [24] constructed a substantially larger dataset via a rule-based generation pipeline. The authors also introduce two test splits intended for cross-domain evaluation. However, the source documents visual characteristics remain similar to those used to generate the training set, and the tampering is still produced by the same synthesis pipeline. Recently, [7] proposed a learning-guided generation pipeline that trains auxiliary networks to produce higher-quality and more diverse tampered documents. Although the quality has improved compared to previous synthetic datasets, the pipeline used still introduces pipeline-specific traces that models can use as tampering cues,

rather than relying on the inconsistencies they introduce. Nonetheless, large-scale pretraining on synthetic data is necessary given the scarcity of human-made forgeries. We therefore propose to constrain the feature pyramid, reducing its reliance on pipeline-specific artifacts.

2.2 Document tampering detection and localization architectures

DCT-domain frequency evidence has proven particularly effective for exposing manipulation traces in document images. DTD [24] introduced a dual RGB and DCT architecture whose Frequency Perception Head (FPH) first processes the quantized DCT coefficients before injecting quantization-table information. Concretely, each discrete coefficient is passed through a dilated convolution that expands the signal to a high-dimensional feature map of 64 channels at the initial image resolution. The result is then projected and modulated by an embedded quantization table. Next, the modulated and unmodulated features are concatenated, augmented with explicit coordinate channels, and fed to a strided convolution that expands the representation to 256 channels, followed by several MBConv blocks [29]. This sequence materializes large intermediate activations early, contributing significant overhead in the DCT branch. FPH was reused by state-of-the-art models such as FFDN [6] and ADCD-Net [33]. FFDN [6] further strengthens spatial–frequency modeling and trace sensitivity with additional enhancement and fusion components. Other approaches further add specialized components [8, 16, 20], often at the cost of increased architectural complexity. More recently, ADCD-Net [33] improves robustness under practical post-processing that can disrupt JPEG block alignment. Overall, while these methods steadily improve performance, their initial evaluations and ablations of proposed components are often conducted on synthetic documents generated with the same pipeline as the training set, and their benefits have yet to be demonstrated for realistic human-made tampering and cross-domain settings, where both the source images and the manipulation pipeline can differ drastically. We therefore start from a standard early-fusion architecture and introduce a small set of well-motivated architectural changes aimed at improving generalization and providing a more efficient DCT backbone.

2.3 Residual features

In addition to leveraging frequency features, another common strategy in image forensics is to introduce residual-like operators as an additional modality alongside RGB. Hand-crafted residual features such as SRM [10] and Bayar convolution layers [2,3] are used to produce high-pass representations that are fused with semantic features early on or processed in a parallel branch to improve sensitivity to subtle manipulation cues. While such auxiliary modalities can improve within-domain performance, recent unified evaluations suggest that they do not improve generalization [9]. In our work, we take a different direction and instead constrain the feature pyramid, rather than adding another modality.

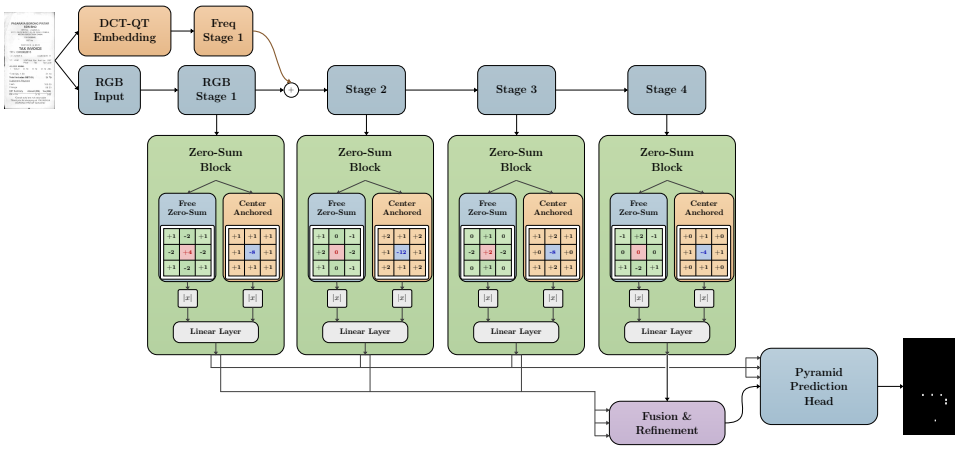

Fig. 1: Overview of the proposed **DiffNet** architecture.

3 Method

3.1 Overview

Our model follows a two-branch early-fusion design. The RGB branch processes the input image with a four-stage ConvNeXt-V2 [34] backbone, producing a multi-scale feature pyramid. In parallel, the DCT branch encodes the quantized DCT coefficients together with the quantization table to produce a frequency feature map. We fuse the DCT features into the RGB stream after stage 1. At the output of each backbone stage, we apply our *multi-level discrepancy transformation*, yielding a discrepancy feature pyramid, which is then aggregated by a multi-scale fusion module to predict the tampering mask. An overview of the architecture is shown in Fig. 1.

3.2 Backbone Design

RGB branch. Given an image $I \in \mathbb{R}^{H\times W\times 3}$, the RGB branch uses a ConvNeXt-V2 backbone to compute four stages of features $\{F_1, F_2, F_3, F_4\}$, with $F_s \in \mathbb{R}^{C_s\times \frac{H}{2^{s+1}}\times \frac{W}{2^{s+1}}}$ for $s \in \{1, 2, 3, 4\}$.

DCT branch. DCT coefficients together with quantization tables, which capture block-level frequency content and the per-frequency quantization strength, have proven to be an effective modality for document tampering detection [6, 33]. Quantization and rounding introduce characteristic artifacts, often termed *codec traces*, that have been shown to be important cues for tampering detection and are most naturally described by the pair of quantized DCT coefficients and the associated quantization table [12,13,22,23]. We therefore operate in the *quantized* DCT domain. DCT maps each non-overlapping 8×8 image block from the pixel

domain to 64 frequency coefficients in a cosine basis. Concretely, for a block $B \in \mathbb{R}^{8\times 8}$ with samples $B_{x,y}$, the unquantized 2D DCT coefficient at frequency index $(u, v) \in \{0, \dots, 7\}^2$ is:

$$C_{u,v} = \alpha(u)\alpha(v) \sum_{x=0}^{7} \sum_{y=0}^{7} B_{x,y} \cos\left(\frac{(2x+1)u\pi}{16}\right) \cos\left(\frac{(2y+1)v\pi}{16}\right), \quad (1)$$

where $\alpha(0) = \frac{1}{\sqrt{8}}$ and $\alpha(t) = \sqrt{\frac{2}{8}}$ for $t \geq 1$. For each 8×8 block and frequency index (u, v), the raw quantized DCT coefficient is computed as:

$$\tilde{C}_{u,v} = \text{round}\left(\frac{C_{u,v}}{Q_{u,v}}\right), \quad (2)$$

where $C_{u,v}$ is the unquantized coefficient and $Q_{u,v}$ is the corresponding quantization table entry. Let $\hat{C} \in \{0, \dots, 20\}^{\frac{H}{8}\times\frac{W}{8}\times 64}$ denote the stacked clipped absolute quantized DCT coefficients over all 8×8 blocks, obtained by taking the absolute value of each raw quantized DCT coefficient $\tilde{C}_{p,k}$ and clipping it to $[0, 20]$, and let $Q \in \{1, \dots, 255\}^{64}$ denote the quantization table. We index blocks by p and frequencies by $k \in \{1, \dots, 64\}$, and denote the quantization entry at frequency k as Q_k. Since both $\hat{C}$ and Q take discrete integer values, we use a learnable *frequency-index-aware joint embedding* that maps each pair $(\hat{C}_{p,k}, Q_k)$ to a vector in $\mathbb{R}^{d_{\text{dct}}}$, with a small embedding size d_{dct}. Concretely, we embed the local quantized coefficient value into $v_{p,k} \in \mathbb{R}^{d_{\text{dct}}}$ and we embed the corresponding quantization entry Q_k and transform it into per-frequency modulation parameters $\gamma_k, \beta_k \in \mathbb{R}^{d_{\text{dct}}}$, which modulate $v_{p,k}$ in a FiLM-like manner [21]. We additionally inject two complementary signals: (i) a learned frequency-index embedding $f_k \in \mathbb{R}^{d_{\text{dct}}}$ that identifies which DCT position k the token comes from, and (ii) a global table bias $t \in \mathbb{R}^{d_{\text{dct}}}$ computed from the full quantization table Q, using an MLP, to capture quantization patterns beyond the single entry Q_k. The resulting per-frequency joint embedding is:

$$e_{p,k} = (1 + \gamma_k) \odot v_{p,k} + \beta_k + f_k + t, \quad (3)$$

where $\odot$ denotes element-wise multiplication. Stacking these embeddings for all frequencies yields a per-block tensor of shape $64 \times d_{\text{dct}}$. We then obtain a single block-level representation by a simple reshape that concatenates the 64 frequency embeddings along the channel dimension, producing a vector in $\mathbb{R}^{64\, d_{\text{dct}}}$ per block. This produces an embedded tensor $E \in \mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times C_{\text{dct}}}$ where $C_{\text{dct}} = 64\, d_{\text{dct}}$. We then process E with a stack of standard ConvNeXt-V2 blocks [34], producing a compact frequency feature map $F_{\text{dct}} \in \mathbb{R}^{C_{\text{dct}}\times\frac{H}{8}\times\frac{W}{8}}$.

Early fusion. We inject the DCT features into the RGB stream after the first stage by concatenation followed by a lightweight projection back to the RGB channel dimension. The remaining stages operate on the fused representation.

3.3 Multi-level discrepancy transformation

As discussed in the introduction, synthetic document tampering datasets used for pretraining can encourage models to treat domain-specific cues and pipeline-specific artifacts as evidence of tampering, rather than focusing on the *inconsistencies* introduced by the manipulation. To reduce reliance on such cues and instead emphasize inconsistency evidence, we introduce a lightweight *multi-level discrepancy transformation* applied at the output of every backbone stage before multi-level fusion and prediction. It converts stage features into magnitude-only responses of learned zero-sum filters, yielding a sign-invariant discrepancy representation across the feature pyramid.

Channel-wise zero-sum discrepancy filters. Let $F \in \mathbb{R}^{C\times H_i\times W_i}$ be a feature map at stage i. For each channel c independently, we produce M responses using $K \times K$ depthwise filters:

$$u_{c,m}(p) = \sum_{\Delta\in\{-\frac{K-1}{2},\dots,\frac{K-1}{2}\}^2} k_{c,m}(\Delta)\, F_c(p+\Delta), \qquad m = 1,\dots,M. \tag{4}$$

We use zero-sum filters, i.e., $\sum_\Delta k_{c,m}(\Delta) = 0$, which suppresses channel-wise bias components and emphasizes discrepancy responses rather than absolute feature intensity. We then take the absolute response, $v_{c,m}(p) = |u_{c,m}(p)|$, so the representation captures the *strength* of the mismatch independently of its sign. We can split the kernel into positive and negative parts $k = k^+ - k^-$ with $k^+(\Delta) = \max(k(\Delta), 0)$ and $k^-(\Delta) = \max(-k(\Delta), 0)$. Let $\mathcal{I} := \{-\frac{K-1}{2},\dots,\frac{K-1}{2}\}^2$ denote the set of spatial offsets. Then we have:

$$\begin{aligned} v_{c,m}(p) = \Big|\sum_{\Delta\in\mathcal{I}} k^+_{c,m}(\Delta)\, F_c(p+\Delta) - \sum_{\Delta\in\mathcal{I}} k^-_{c,m}(\Delta)\, F_c(p+\Delta)\Big|, \\ \text{with } \sum_{\Delta\in\mathcal{I}} k^+_{c,m}(\Delta) = \sum_{\Delta\in\mathcal{I}} k^-_{c,m}(\Delta). \end{aligned} \tag{5}$$

Since both terms have the same total weight, they form two competing weighted sums, and $v_{c,m}(p)$ captures their difference in magnitude, producing a discrepancy-strength signal.

Free and center-anchored zero-sum filters. We let the model learn two complementary types of zero-sum filters:

- **Free zero-sum filters** keep flexibility while enforcing $\sum_\Delta k_{c,m}(\Delta) = 0$.
- **Center-anchored zero-sum filters** further constrain the filter to behave as a neighborhood-based center predictor: neighbor weights are non-negative and the center weight is the negative sum of neighbors, so the output measures a center-to-context discrepancy.

These two families are complementary: the center-anchored family injects a structured inductive bias, while the free family preserves expressivity.

Unified parameterization. We propose a single formulation that covers both families by tying the neighbor weights to the center weight, which will allow us to define an efficient implementation at training time. Let $\mathcal{N} = \left\{-\frac{K-1}{2}, \ldots, \frac{K-1}{2}\right\}^2 \setminus \{(0,0)\}$ denote the set of $K \times K$ offsets excluding the center. For each (c, m) we learn unconstrained parameters $\theta_{c,m,\Delta}$ and define:

$$a_{c,m,\Delta} = \begin{cases} \theta_{c,m,\Delta} & \text{(free family)}, \\ |\theta_{c,m,\Delta}| & \text{(center-anchored family)}, \end{cases} \qquad \Delta \in \mathcal{N}. \tag{6}$$

The convolution kernel can be formulated as follows:

$$k_{c,m}(\Delta) = \begin{cases} a_{c,m,\Delta} & \Delta \in \mathcal{N}, \\ -\sum_{\delta \in \mathcal{N}} a_{c,m,\delta} & \Delta = (0,0). \end{cases} \tag{7}$$

which automatically enforces the zero-sum constraint and correctly defines both filter families. A direct implementation would explicitly synthesize the full $K \times K$ depthwise kernels inside the forward pass, introducing additional tensor construction overhead before the convolution. Our proposed parametrization allows us to define an efficient custom CUDA kernel, which we explain in the supplementary material. At inference time, once parameters are fixed, we materialize the resulting depthwise kernels once and use a standard depthwise convolution.

Projection. The M magnitude responses per channel are then mapped back to the backbone channel dimension with a lightweight channel-wise projection, yielding the transformed feature map $\tilde{F} = \phi(F)$.

Multi-stage application. We apply the same transformation at the output of every backbone stage before multi-level fusion and prediction. For each stage s, we first project F_s to a relatively small intermediate width and then apply ϕ in this space, producing the transformed feature map:

$$\tilde{F}_s = \phi(r(F_s)), \qquad s \in \{1, 2, 3, 4\}, \tag{8}$$

where $r(\cdot)$ is a lightweight projection to a reduced channel dimension. The decoder and fusion modules then operate on $\{\tilde{F}_1, \tilde{F}_2, \tilde{F}_3, \tilde{F}_4\}$.

Feature activations before and after zero-sum filtering. To confirm the effect of our discrepancy transformation, we visualize feature activations immediately before and after the proposed zero-sum filter bank. Concretely, at a fixed backbone stage we take the feature tensor F_4 and its counterpart just after the zero-sum filters, compute an activation intensity map, and upsample it to the input size. Fig. 2 shows that pre zero-sum activations tend to follow more the document content, whereas after applying the zero-sum filters, the activation concentrates around local inconsistencies, producing sharper evidence near the edited region while suppressing content-driven responses. This qualitative behavior matches the role of the transformation: expose tampering boundaries as discrepancy cues that are later integrated by the multi-level fusion decoder.

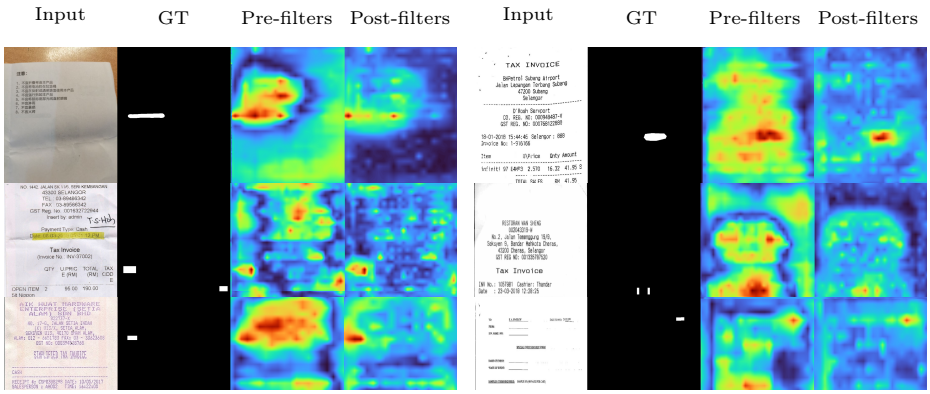

Fig. 2: Activation intensity maps before and after the proposed zero-sum filters.

Relation to constrained convolution residual modalities. Several works in image forensics use high-pass convolutions to extract a residual signal as an additional input modality. Common examples include fixed residual features such as SRM [10] and constrained convolution layers that enforce a residual form, as in Bayar convolutions [2,3]. These residuals are typically computed from the input image and then fused with RGB features or processed in a separate branch. Our approach is fundamentally different. First, our multi-level discrepancy transformation is not an extra modality. It is applied to the backbone feature maps at multiple pyramid stages, before multi-level fusion and prediction, so it directly shapes the representation used for localization rather than adding a parallel input stream. Second, we take the absolute response and treat it as a discrepancy-strength signal, while residual branches typically keep signed residuals. Third, we do not fix the center weight as in Bayar convolutions, where the center is set to -1 and neighbor weights are normalized to sum to 1 [3]. We only enforce the zero-sum property and use two filter families, center-anchored mixtures and free zero-sum mixtures. In Sec. 4.4 we test these alternatives and show that removing our pathway and adding a residual modality, or replacing our pathway with SRM kernels or Bayar-style constrained variants, reduces performance.

3.4 Deep multi-scale fusion and mask prediction

We fuse the transformed pyramid features in an FPN-style top-down pathway, progressively propagating information from fine to coarse stages. We then refine the fused representation with six ConvNeXt-V2 blocks. Finally, a lightweight segmentation head outputs the tampering mask at the input resolution. Since the transformed features operate at a relatively small width after $r(\cdot)$, these fusion and prediction modules add only minor overhead.

Table 1: Results for models trained and evaluated following the Doc Protocol. Avg. C denotes the average F1 score across all cross-domain test sets.

Method	Venue	T-SROIE	OSTF	TPIC-13	RTM	Avg. C
TruFor	CVPR'23	0.213	0.046	0.104	0.034	0.099
MVSS-Net	ICCV'21	0.187	0.037	0.113	0.027	0.091
Mesorch	AAAI'25	0.294	0.139	0.241	0.041	0.179
IML-ViT	arXiv	0.427	0.21	0.256	0.076	0.242
PSCC-Net	TCSVT'22	0.517	0.441	0.55	0.126	0.408
CAT-Net	IJCV'22	0.609	0.178	0.343	0.063	0.298
TIFDM	TCE'24	0.058	0.006	0.013	0.018	0.024
CAFTB-Net	TOMM'25	0.262	0.119	0.301	0.033	0.179
DTD	CVPR'23	0.525	0.124	0.284	0.058	0.247
FFDN	ECCV'24	0.533	0.241	0.357	0.071	0.301
ADCD-Net	ICCV'25	0.623	0.441	0.579	0.159	0.450
DiffNet (Ours)	—	**0.745**	**0.495**	**0.647**	**0.173**	**0.515**

4 Experiments

4.1 Implementation details

We set the DCT embedding dimension to $d_{\text{dct}} = 4$, the DCT backbone depth to 6, the discrepancy kernel size to $K = 7$, and the number of discrepancy responses per channel to $M = 2$, split equally across the two filter families, i.e., one free zero-sum filter and one center-anchored zero-sum filter per channel. We set the projection dimension before ϕ to 256 channels at all stages, and the decoder refinement depth to 6.

4.2 Evaluation Protocols

To thoroughly evaluate our architecture and demonstrate its superiority, we follow two training and evaluation protocols and compare against prior models under the same protocol.

Table 2: Results for models trained and evaluated under the Syn2Real-TDoc-2.8M protocol. P, R, and F1 denote precision, recall, and F1 score, respectively. Avg. F1 reports the average F1 across RTM, FindItAgain, and FindIt, separately for image-level (Im) and pixel-level (Pix) predictions.

Model	Venue	RTM						FindItAgain						FindIt						Avg. F1	
		Im			Pix			Im			Pix			Im			Pix			Im	Pix
		P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1		
PSCC-Net	TCSVT'22	0.393	0.402	0.397	0.105	0.108	0.106	0.567	0.600	0.583	0.106	0.128	0.116	0.629	0.550	0.587	0.120	0.121	0.120	0.522	0.114
CAT-Net	IJCV'22	0.400	0.350	0.374	0.092	0.110	0.100	0.480	0.342	0.400	0.110	0.056	0.074	0.515	0.425	0.466	0.130	0.105	0.117	0.413	0.097
DTD	CVPR'23	0.366	**0.733**	0.489	0.149	0.114	0.129	0.500	0.400	0.444	0.125	0.124	0.125	0.623	0.662	0.642	0.140	0.144	0.142	0.525	0.132
ASC-Former	PR'25	0.443	0.323	0.374	0.234	0.161	0.191	0.386	0.629	0.478	0.153	0.123	0.136	0.696	0.688	0.692	0.301	0.139	0.190	0.515	0.172
FFDN	ECCV'24	0.464	0.510	0.486	0.282	0.206	0.238	**0.689**	0.571	0.625	0.316	0.214	0.255	0.823	0.700	0.757	0.363	0.283	0.318	0.623	0.270
ADCD-Net	ICCV'25	0.470	0.517	0.492	0.249	0.157	0.193	0.523	**0.657**	0.582	0.136	0.149	0.142	0.828	0.662	0.736	0.191	0.179	0.185	0.604	0.174
DiffNet (Ours)	—	**0.506**	0.552	**0.528**	**0.321**	**0.212**	**0.255**	0.625	0.628	**0.626**	**0.409**	**0.246**	**0.307**	**0.833**	**0.707**	**0.765**	**0.448**	**0.287**	**0.350**	**0.640**	**0.304**

Table 3: Efficiency and localization performance for state-of-the-art models. Inference and training throughput are reported in it/s, at a resolution of 768×768. Inference throughput is reported with batch size 1. Doc Avg F1 is the average localization F1 on the cross-domain Doc Protocol test sets. TDoc Avg F1 is the average localization F1 on Syn2Real-TDoc, and Avg all is their mean. Note that ADCD-Net has 23.7M trainable parameters and 22M frozen parameters.

Params (M)	Model	Inf. (it/s)	Train (it/s)	Doc Avg F1	TDoc Avg F1	Avg all
3.7	PSCC-Net	15.4	10.0	0.408	0.114	0.261
114.0	CAT-Net	29.9	13.8	0.298	0.097	0.198
67.3	DTD	29.4	25.3	0.247	0.132	0.190
79.6	ASC-Former	29.6	27.9	—	0.172	—
45.7	ADCD-Net	5.7	4.4	0.450	0.174	0.312
140.0	FFDN	33.1	20.8	0.301	0.270	0.286
113.0	DiffNet (Ours)	**40.1**	**41.9**	**0.515**	**0.304**	**0.410**

Doc Protocol. We follow Doc Protocol's cross-domain evaluation introduced in ForensicHub [9]. Under this protocol, models are trained on DocTamper [24] and evaluated on T-SROIE [32], OSTF [26], Tampered-IC13 [31], and RTM [18].

Syn2Real-TDoc. We follow the Syn2Real-TDoc protocol proposed in [7]. Under this protocol, models are trained on TDoc-2.8M and evaluated on RTM [18], FindIt [1], and FindItAgain [30].

Both pretraining datasets are synthetic, while all test datasets in the Syn2Real-TDoc protocol are human-made. In particular, RTM and FindItAgain are both high-quality and extremely challenging. On the other hand, the Doc protocol focuses on cross-domain generalization, with only RTM being human-made. We provide more details about the pretraining and testing datasets in the supplementary material. For both protocols, we follow their respective training and evaluation procedures. We note that the reported metrics are not computed in the same way: Syn2Real first aggregates precision and recall globally and then derives F1, whereas Doc Protocol averages F1 across images and excludes samples with no positive pixels. Additional details are provided in the supplementary material.

4.3 Main Results

Results on the Doc Protocol. Table 1 reports cross-domain performance under the ForensicHub Doc Protocol, comparing DiffNet against TruFor [11], MVSS-Net [5], Mesorch [36], IML-ViT [19], PSCC-Net [17], CAT-Net [13], TIFDM [8], CAFTB-Net [28], DTD [24], FFDN [6], and ADCD-Net [33]. Our method achieves the best results on all test sets and improves the cross-domain average F1 from 0.45 to 0.515.

Results on Syn2Real-TDoc. Table 2 reports performance under the Syn2Real-TDoc protocol. Our method achieves the best overall performance, reaching 0.640 average image-level F1 and 0.304 average pixel-level F1. Compared with the strongest prior baseline, this corresponds to gains of +0.017 and +0.034 respectively. The improvements are consistent across datasets and are particularly pronounced for pixel-level localization. On RTM we improve pixel-level F1 from 0.238 to 0.255, on FindItAgain from 0.255 to 0.307. Both of these datasets are human-made and extremely challenging. They contain carefully crafted manipulations, often with advanced concealment techniques designed to hide forensic traces. They also include a much higher proportion of pristine images than manipulated ones, which makes achieving high metrics difficult. Nevertheless, our method consistently improves over the previous best on both datasets.

Efficiency and robustness Table 3 summarizes throughput and the average localization accuracy across both protocols. Our model largely surpasses previous work on both protocols, while being substantially faster than previous work. It achieves 40.1 it/s at inference and 41.9 it/s during training, outperforming the fastest prior high-performing baseline FFDN and providing up to a 7× inference speedup over ADCD-Net.

4.4 Ablation Studies

We perform ablation studies to quantify the contribution of each component in our design. All reported results are averaged over three runs.
Table 4 shows ablation results of our core design choices under both protocols. Removing the zero-sum constraint (A0) lowers performance, which validates the effectiveness of the discrepancy transformation. Swapping our DCT backbone for the Frequency Perception Head at matched compute (A1) leads to a similar drop, showing that our compact DCT–quantization joint embedding provides a better representation. Finally, removing the fused-representation refinement (A2) causes a smaller but consistent decrease.
Table 5 covers several design choices of our approach. For the DCT embedding, removing the global table bias t (A3) or the frequency-index embedding f (A4) reduces performance, while removing the FiLM modulation terms γ (A5) or β (A6) causes a larger performance drop. This confirms the importance of each

Table 4: Core ablations. Doc Pix F1 and TDoc Pix F1 denote the average pixel-level F1 under the cross-domain evaluation of Doc Protocol and Syn2Real-TDoc, respectively. Avg. Pix F1 is the equal-weight mean of the two protocol averages. All values reported are averaged across three different runs.

ID	Ablation	Doc Pix F1	TDoc Pix F1	Avg. Pix F1
B	baseline (ours)	**0.515** (±0.005)	**0.303** (±0.003)	**0.409**
A0	w/o zero-sum constraint	0.496 (±0.007)	0.291 (±0.004)	0.393
A1	replace our DCT backbone with FPH (comparable params/FLOPs)	0.493 (±0.019)	0.290 (±0.006)	0.391
A2	w/o fused representation refinement (compute added to the last stage instead)	0.508 (±0.007)	0.294 (±0.006)	0.401

Table 5: Additional ablation results. All values reported are averaged across three different runs.

ID	Ablation	Doc Pix F1
B	baseline (ours)	**0.515** (±0.005)
DCT embedding design		
A3	w/o table bias t	0.513 (±0.003)
A4	w/o frequency-index embedding f	0.512 (±0.013)
A5	w/o γ	0.509 (±0.014)
A6	w/o β	0.504 (±0.017)
A7	linear instead of embedding	0.494 (±0.007)
DCT embedding dimension		
A8	$d_{\text{dct}} = 1$	0.507 (±0.008)
A9	$d_{\text{dct}} = 2$	0.511 (±0.007)
A10	$d_{\text{dct}} = 8$	0.515 (±0.011)
Discrepancy filters		
A11	zero-sum kernel size $K : 7 \to 9$	0.514 (±0.009)
A12	zero-sum kernel size $K : 7 \to 5$	0.512 (±0.006)
A13	zero-sum kernel size $K : 7 \to 3$	0.512 (±0.006)
A14	fixed SRM filters instead of learned discrepancy filters	0.511 (±0.005)
A15	no absolute value after zero-sum filters	0.510 (±0.008)
A16	tanh instead of absolute value after zero-sum filters	0.512 (±0.003)
A17	one filter family only (best single-family choice)	0.513 (±0.004)
A18	use bayar-conv	0.506 (±0.011)
A19	discrepancy filters as a modality	0.499 (±0.011)
A20	w/o zero-sum constraint + more regularization	0.497 (±0.006)

term of our joint embedding formula. Replacing the embedding with a linear mapping (A7) produces a large decrease, which confirms that learning an embedding is more efficient. Varying the embedding dimension (A8-9-10) shows that the used value of four provides the best efficiency.

The discrepancy-filter ablation results in Table 5 confirm the contribution of different used values and design choices. Varying the kernel size (A11-12-13) shows that the used kernel size of seven provides the best performance and efficiency. Using fixed SRM kernels (A14) or Bayar convolution layers (A18) reduces performance compared to our proposed learned discrepancy filters. Removing the absolute-value operator (A15) or replacing it with tanh (A16) does also degrade performance, showing that taking the absolute values of the response of these zero-sum filters is beneficial. Using only one filter family (A17) also reduces accuracy, confirming that the free and center-anchored filters complement each other. Finally, treating discrepancy filters as an additional modality (A19) rather than transforming the feature pyramid or removing the zero-sum filters and replacing them with more regularization (A20) reduces performance.

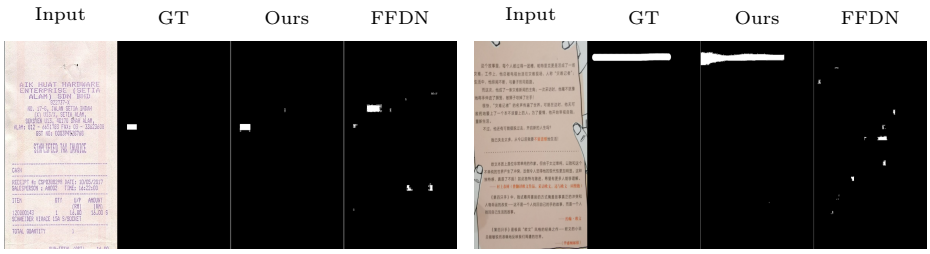

Fig. 3: Comparison on two examples of our model and the previous state-of-the-art method, FFDN, with both models pretrained on TDoc-2.8M.

4.5 Qualitative analysis

Qualitative comparison Fig. 3 shows two representative examples comparing our model against FFDN. In these examples, our predictions tend to better concentrate on the edited area and recover a more coherent manipulated region, while suppressing scattered responses on unrelated content, which is consistent with the role of our multi-level discrepancy transformation.

Hard cases. Fig. 4 illustrates hard samples where our model only partially localizes the forgery, yielding very low recall. These cases are challenging because the edits occur in areas with blank background, and are locally post-processed to conceal traces of tampering, leaving very few cues about the forgery location.

Discussion. Our method achieves a consistent boost in performance compared to prior work. However, the localization performance on human-made tampering remains low. This is due to several factors. First, these datasets contain many challenging cases where very few tampering cues remain: edits are often performed in homogeneous background regions and combined with advanced concealment techniques, making high recall difficult. In contrast, synthetic datasets typically lack concealment strategies that adapt their type and strength to each individual case. While some works attempt to conceal tampering with fixed

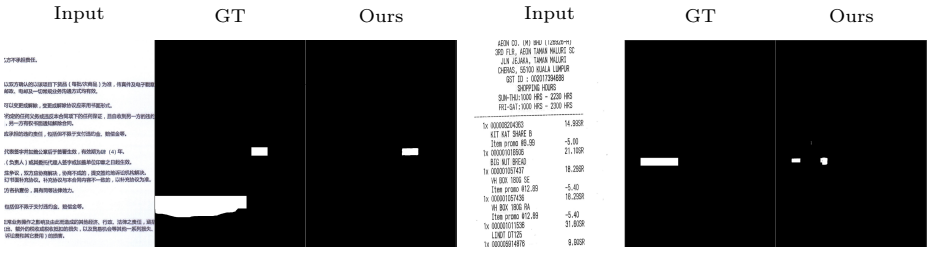

Fig. 4: Illustration of hard cases where our model struggles to recover the full manipulated region.

rules [8], this can introduce shortcuts [7] during training. Improving the realism and diversity of concealment in synthetic data is therefore a promising direction for future work. Second, these benchmarks include a large proportion of regions with no positive pixels, reflecting real-world deployment conditions and making it difficult to achieve high precision. Pixel F1 is also sensitive to label ambiguity and annotation shift: label mismatches between benchmarks and synthetic pretraining can penalize otherwise plausible predictions.

5 Conclusion

We present an efficient RGB–DCT early-fusion architecture for human-made document tampering detection and localization. Our approach is built around two complementary ideas: (i) a multi-level discrepancy transformation applied at every backbone stage to convert features into a sign-invariant discrepancy representation, and (ii) a lightweight frequency-index-aware joint embedding for quantized DCT coefficients and quantization tables, enabling an efficient DCT branch for early fusion. Across both the Doc Protocol and the Syn2Real-TDoc protocol, the proposed design consistently improves performance, while also improving training and inference throughput.

Acknowledgments

This work received financial support from Crédit Agricole S.A. through the research chair with École Polytechnique on Trustworthy and Responsible AI. This work was granted access to the HPC resources of IDRIS under the allocation 2026-AD010616244 made by GENCI.

References

1. Artaud, C., Doucet, A., Ogier, J.M., d'Andecy, V.P.: Receipt dataset for fraud detection. In: First International Workshop on Computational Document Forensics (2017)
2. Bayar, B., Stamm, M.C.: A deep learning approach to universal image manipulation detection using a new convolutional layer. In: Proceedings of the 4th ACM workshop on information hiding and multimedia security. pp. 5–10 (2016)
3. Bayar, B., Stamm, M.C.: Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection. IEEE Transactions on Information Forensics and Security **13**(11), 2691–2706 (2018)
4. Bibi, M., Hamid, A., Moetesum, M., Siddiqi, I.: Document forgery detection using printer source identification—a text-independent approach. In: 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW). vol. 8, pp. 7–12. IEEE (2019)
5. Chen, X., Dong, C., Ji, J., Cao, J., Li, X.: Image manipulation detection by multi-view multi-scale supervision. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 14185–14193 (2021)

6. Chen, Z., Chen, S., Yao, T., Sun, K., Ding, S., Lin, X., Cao, L., Ji, R.: Enhancing tampered text detection through frequency feature fusion and decomposition. In: European Conference on Computer Vision. pp. 200–217. Springer (2024)
7. Dhouib, M., Buscaldi, D., Vanier, S., Shabou, A.: Leveraging contrastive learning for a similarity-guided tampered document data generation pipeline. arXiv preprint arXiv:2602.17322 (2026)
8. Dong, L., Liang, W., Wang, R.: Robust text image tampering localization via forgery traces enhancement and multiscale attention. IEEE Transactions on Consumer Electronics **70**(1), 3495–3507 (2024)
9. Du, B., Zhu, X., Ma, X., Qu, C., Feng, K., Yang, Z., Pun, C.M., liu, J., Zhou, J.Z.: Forensichub: A unified benchmark & codebase for all-domain fake image detection and localization. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track (2025)
10. Fridrich, J., Kodovsky, J.: Rich models for steganalysis of digital images. IEEE Transactions on information Forensics and Security **7**(3), 868–882 (2012)
11. Guillaro, F., Cozzolino, D., Sud, A., Dufour, N., Verdoliva, L.: Trufor: Leveraging all-round clues for trustworthy image forgery detection and localization. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 20606–20615 (2023)
12. Huang, F., Huang, J., Shi, Y.Q.: Detecting double jpeg compression with the same quantization matrix. IEEE Transactions on Information Forensics and Security **5**(4), 848–856 (2010)
13. Kwon, M.J., Nam, S.H., Yu, I.J., Lee, H.K., Kim, C.: Learning jpeg compression artifacts for image manipulation detection and localization. International Journal of Computer Vision **130**(8), 1875–1895 (May 2022). https://doi.org/10.1007/s11263-022-01617-5
14. Lampert, C.H., Mei, L., Breuel, T.M.: Printing technique classification for document counterfeit detection. In: 2006 International Conference on Computational Intelligence and Security. vol. 1, pp. 639–644. IEEE (2006)
15. Lewis, D., Agam, G., Argamon, S., Frieder, O., Grossman, D., Heard, J.: Building a test collection for complex document information processing. In: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 665–666 (2006)
16. Li, L., Zhang, K., Lu, J., Zhang, S., Chu, N.: Multiclassification tampering detection algorithm based on spatial-frequency fusion and swin-t. IET Image Processing **19**(1), e70007 (2025)
17. Liu, X., Liu, Y., Chen, J., Liu, X.: Pscc-net: Progressive spatio-channel correlation network for image manipulation detection and localization. IEEE Transactions on Circuits and Systems for Video Technology **32**(11), 7505–7517 (2022)
18. Luo, D., Liu, Y., Yang, R., Liu, X., Zeng, J., Zhou, Y., Bai, X.: Toward real text manipulation detection: New dataset and new solution. Pattern Recognition **157**, 110828 (2025)
19. Ma, X., Du, B., Jiang, Z., Hammadi, A.Y.A., Zhou, J.: Iml-vit: Benchmarking image manipulation localization by vision transformer. arXiv preprint arXiv:2307.14863 (2023)
20. Nguyen, A.D., Kim, H.Y., Nguyen, H.N.: Taliu: A novel decoder and augmentation strategy for boosting tampered document image detection. IEEE Access (2025)
21. Perez, E., Strub, F., De Vries, H., Dumoulin, V., Courville, A.: Film: Visual reasoning with a general conditioning layer. In: Proceedings of the AAAI conference on artificial intelligence. vol. 32 (2018)

22. Pevny, T., Fridrich, J.: Detection of double-compression in jpeg images for applications in steganography. IEEE Transactions on information forensics and security **3**(2), 247–258 (2008)
23. Popescu, A.C., Farid, H.: Statistical tools for digital forensics. In: International workshop on information hiding. pp. 128–147. Springer (2004)
24. Qu, C., Liu, C., Liu, Y., Chen, X., Peng, D., Guo, F., Jin, L.: Towards robust tampered text detection in document image: New dataset and new solution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5937–5946 (2023)
25. Qu, C., Zhong, Y., Guo, F., Jin, L.: Omni-iml: towards unified image manipulation localization. arXiv preprint arXiv:2411.14823 (2024)
26. Qu, C., Zhong, Y., Guo, F., Jin, L.: Revisiting tampered scene text detection in the era of generative ai. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 694–702 (2025)
27. Shang, S., Kong, X., You, X.: Document forgery detection using distortion mutation of geometric parameters in characters. Journal of Electronic Imaging **24**(2), 023008–023008 (2015)
28. Song, Y., Jiang, W., Chai, X., Gan, Z., Zhou, M., Chen, L.: Cross-attention based two-branch networks for document image forgery localization in the metaverse. ACM Transactions on Multimedia Computing, Communications and Applications **21**(2), 1–24 (2025)
29. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International conference on machine learning. pp. 6105–6114. PMLR (2019)
30. Tornés, B.M., Taburet, T., Boros, E., Rouis, K., Doucet, A., Gomez-Krämer, P., Sidere, N., d'Andecy, V.P.: Receipt dataset for document forgery detection. In: International Conference on Document Analysis and Recognition. pp. 454–469. Springer (2023)
31. Wang, Y., Xie, H., Xing, M., Wang, J., Zhu, S., Zhang, Y.: Detecting tampered scene text in the wild. In: European Conference on Computer Vision. pp. 215–232. Springer (2022)
32. Wang, Y., Zhang, B., Xie, H., Zhang, Y.: Tampered text detection via rgb and frequency relationship modeling. Chinese Journal of Network and Information Security **8**(3), 29–39 (2022). https://doi.org/10.11959/j.issn.2096-109x.2022035
33. Wong, K., Zhou, J., Wu, H., Si, Y.W., Zhou, J.: Adcd-net: Robust document image forgery localization via adaptive dct feature and hierarchical content disentanglement. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 19280–19289 (2025)
34. Woo, S., Debnath, S., Hu, R., Chen, X., Liu, Z., Kweon, I.S., Xie, S.: Convnext v2: Co-designing and scaling convnets with masked autoencoders. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16133–16142 (2023)
35. Wu, L., Zhang, C., Liu, J., Han, J., Liu, J., Ding, E., Bai, X.: Editing text in the wild. In: Proceedings of the 27th ACM international conference on multimedia. pp. 1500–1508 (2019)
36. Zhu, X., Ma, X., Su, L., Jiang, Z., Du, B., Wang, X., Lei, Z., Feng, W., Pun, C.M., Zhou, J.Z.: Mesoscopic insights: orchestrating multi-scale & hybrid architecture for image manipulation localization. In: Proceedings of the AAAI conference on artificial intelligence. vol. 39, pp. 11022–11030 (2025)