Revisiting Image Manipulation Localization under Realistic Manipulation Scenarios Supplementary Material

Xuekang Zhu[1,2†] Ji-Zhe Zhou[1†‡] Kaiwen Feng[1] Chenfan Qu[3,2] Xiwen Wang[1]
Yunfei Wang[1] Liting Zhou[1] Jian Liu[2‡]
[1]Sichuan University [2]Ant Group [3]South China University of Technology

1. Current Status of Manipulation Datasets

Most existing IML benchmarks are annotated in a one-shot manner, providing only the final binary manipulation mask. However, the underlying manipulation workflows that give rise to these datasets are far from single-step. Early benchmarks such as Columbia (2006) [7] and CASIA v1.0 (2013) [3] largely consist of pure copy–paste operations. CASIA v2.0 (2013) [3] already introduced additional post-processing steps such as geometric transforms and boundary smoothing.

Subsequent datasets demonstrate increasingly multi-stage editing procedures. NIST16 (2019) [5] and IMD20 (2020) [13] were constructed through professional forensic pipelines that involve object removal, inpainting, color adjustment, and blending — inherently multi-step sequences. The most recent datasets, including COCO-GLID [6]and AutoSplice (2023) [8], are based on diffusion models whose generative dynamics are iterative by design, making their manipulation process intrinsically multi-step.

Thus, although these datasets appear as one-shot benchmarks in their released annotations, the manipulations they contain are products of multi-stage generation processes. This structural inconsistency further motivates our sequence-prediction perspective, which aligns directly with the progressive nature of real-world manipulations.

2. Data Distribution

2.1. Traditional IML Training Protocols

Following the standard settings in IMDL-BenCo [12], our experiments in the Traditional one-shot IML scenario adopt both the MVSS and CAT protocols, consistent with Section 4.3 of the main paper. The MVSS protocol uses CASIA v2.0 [3] as the sole training set and evaluates on CASIA v1.0 [3], Columbia [7], NIST16 [5], IMD20 [13], COCO-GLID [6], and AutoSplice [8], providing a benchmark for cross-source and cross-manipulation generalization. The CAT protocol constructs a balanced multi-source training set from CASIA v2.0, FantasticReality v1 [9], IMD20, and the four tampCOCO [10] subsets (sp/cm/bcm/bcmc), sampling 1840 images from each training subset per epoch [12] to ensure distributional consistency. Its evaluation set matches the MVSS protocol but excludes IMD20, enabling a fair assessment of unified learning under heterogeneous training distributions.

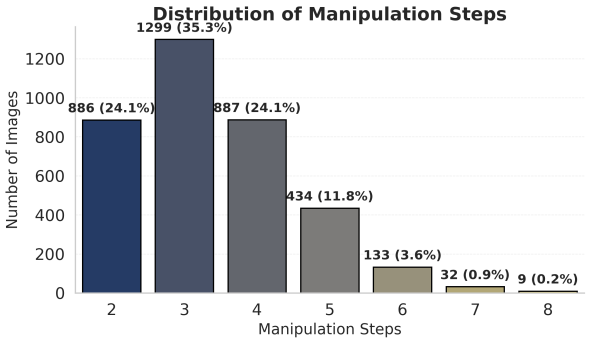

Figure 1. Distribution of manipulation step counts in the synthetic multi-step dataset.

2.2. Synthetic Multi-Step Dataset

To support the proposed sequence prediction paradigm (Section 3.1.1 of the main paper), we construct a synthetic multi-step dataset based on the manipulated images of CASIA v2.0 using our tree-structured reverse sampling algorithm. The resulting dataset contains 3680 samples, each exhibiting a multi-step editing trajectory with highly non-uniform step counts. The statistical distribution, visualized in Figure 1, reveals a long-tailed structure that closely reflects realistic manipulation processes. This dataset compensates for the limitations of existing IML benchmarks, which provide only the final manipulation mask and lack supervision over intermediate editing stages, thereby mitigating the dimensional collapse phenomenon associated with one-shot learning.

†Equal contribution. Corresponding authors: Jian Liu (rex.lj@antgroup.com) and Jizhe Zhou (jzzhou@scu.edu.cn).

2.3. Hierarchical Sequence Structure of the Test Set

The HSIM dataset serves as the test set in our sequence prediction experiments. Each image is constructed through a hierarchical and stepwise editing workflow: manipulation paths are first designed using GPT-4o and then realized via a progressive pixel-level refinement process assisted by GPT-Image-1. This results in multi-path samples with explicit temporal ordering and hierarchical dependency structures, fully aligned with the sequence prediction formulation in Section 4.4 of the main paper.

Two distributional properties of HSIM are particularly relevant to evaluating sequence-level generalization. First, the number of complete manipulation paths per image exhibits a clear long-tailed structure: 79% of the images contain 1–10 valid paths, 19% contain 10–20 paths, and only 2% exceed 20 paths. This reflects realistic variability in manipulation-path complexity. Second, despite this heterogeneity in path counts, the editing depth within each path is highly consistent: every valid path in HSIM contains exactly 4–5 manipulation steps. This combination of "diverse path counts but compact step depth" makes HSIM a controlled yet representative test environment, enabling a precise assessment of the model's cross-depth and cross-path generalization capabilities and its ability to capture the underlying temporal dependencies of manipulation sequences.

3. Evaluation on the Synthetic Multi-Step Dataset

3.1. Quantitative Experiments

This experiment aims to verify whether RITA can still maintain leading performance when trained on the Synthetic Multi-Step Dataset, a dataset that contains multi-step manipulation trajectories. To ensure a fair comparison, all models are trained and evaluated under identical data settings, with the only difference lying in the form of supervision they receive.

RITA is trained with full multi-step supervision: the model learns to predict each intermediate manipulation mask in sequence and performs multi-step rollout during inference. The final-step prediction is treated as the final detection result, while any pixel predicted as manipulated at any intermediate step is unified as 1 to match the binary evaluation format used by baseline models.

Baseline models (MVSS [1], CAT-Net [10], PSCC [11], TruFor [6], Mesorch [20], etc.) are trained on the same images but can only access the final binary mask (single-step final mask) and do not observe any intermediate editing states. As shown in Table 1, RITA consistently outperforms all single-step models across all six cross-source datasets in the MVSS protocol, achieving the highest overall and cross-source averages. Moreover, when we remove multi-step supervision and constrain RITA to single-step prediction (w/o multi-step), its Overall Avg drops from 0.486 to approximately 0.298, a decrease of nearly 20 percentage points. This further highlights the importance of maintaining the full sequence prediction framework in RITA.

Algorithm 1: MonotonicF1Match(P, M)

Input: Predicted sequence $P = \{P_1, \ldots, P_{T_p}\}$
Ground-truth sequence $M = \{M_1, \ldots, M_{T_g}\}$
Output: Average F1 score of the optimal monotonic alignment

```
// 1. Compute pairwise similarity matrix
```
Initialize matrix $F \in \mathbb{R}^{T_p \times T_g}$;
for $i = 1, \ldots, T_p$ **do**
  for $j = 1, \ldots, T_g$ **do**
    $F[i, j] \leftarrow$ F1_score(P_i, M_j);

```
// 2. Find the optimal cumulative score via Dynamic Programming
```
Initialize DP table $D \in \mathbb{R}^{T_p \times T_g}$ to store maximum cumulative scores;
for $i = 1, \ldots, T_p$ **do**
  for $j = 1, \ldots, T_g$ **do**
    if $i = 1$ **then**
      $D[i, j] \leftarrow F[i, j]$; // Base case: first predicted step
    else
      // Find max score from any valid previous alignment
      $max\_prev\_score \leftarrow \max_{1 \le k \le j} D[i-1, k]$;
      $D[i, j] \leftarrow F[i, j] + max\_prev\_score$;

```
// 3. Extract and normalize the final score
```
$max\_cumulative\_score \leftarrow \max_{1 \le j \le T_g} D[T_p, j]$;
// Find the best path's total score
if $T_p > 0$ **then**
  $F1_{match} \leftarrow$ max_cumulative_score$/T_p$;
else
  $F1_{match} \leftarrow 0$;
return $F1_{match}$;

3.2. Qualitative Experiments

We further conduct a qualitative analysis to examine how RITA behaves when trained on multi-step synthetic data but evaluated on a standard one-step test set. As shown in Figure 2, the model does not collapse into a single-shot predictor; instead, it retains a clear internal multi-stage reasoning process throughout inference. We consistently observe

Table 1. Fair comparison between multi-step and single-step training methods. All models use the same Synthetic Multi-Step Dataset; RITA is trained on intermediate masks, while baselines are trained only on the final mask. Results are evaluated under the MVSS protocol. Column-wise best scores are in red, second-best results are underlined.

Model	Source-Aligned	Cross-Source							Overall Avg
	CASIAv1	Coverage	Columbia	NIST16	IMD2020	CocoGlide	Autosplice	**Cross-Source Avg**	
MVSS	0.534	0.259	0.386	0.246	0.279	0.291	0.294	0.292	0.327
CAT-Net	0.581	0.296	0.584	0.269	0.273	0.290	0.354	0.344	0.378
PSCC	0.381	0.286	0.573	0.185	0.251	0.399	0.487	0.363	0.366
Trufor	0.434	0.331	0.689	0.257	0.258	0.399	0.439	0.395	0.401
Mesorch	0.639	0.319	0.744	0.321	0.350	0.317	0.356	0.401	0.435
Ours	0.422	0.357	0.793	0.306	0.360	0.503	0.661	0.497	0.486

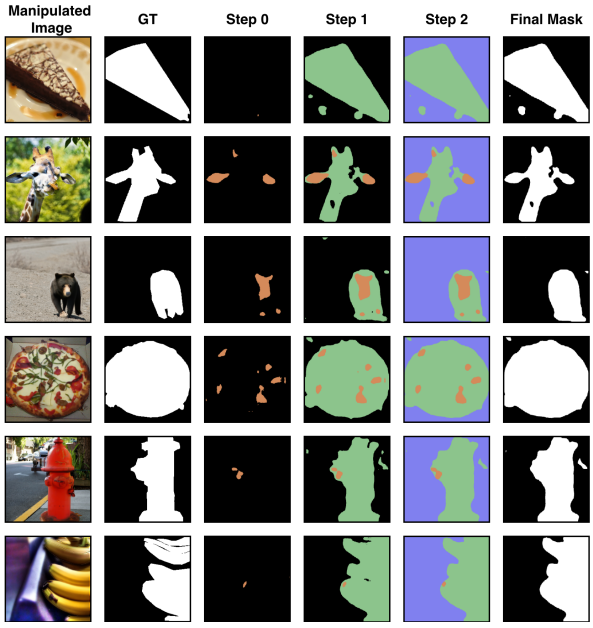

Figure 2. Qualitative results of RITA on a one-shot test dataset. Although trained on a multi-step dataset, the model still exhibits a multi-stage refinement process: early steps emphasize salient manipulation cues, while later steps progressively correct errors and consolidate true manipulated regions, resulting in clean and coherent final masks.

two characteristic behaviors: **(1) Salient-Cue First Activation**: in the first step, the model preferentially highlights the most prominent manipulation cues—such as boundary inconsistencies, structural discontinuities, or conspicuous texture anomalies—thus providing a coarse but meaningful initialization of manipulated regions; **(2) Progressive Error-Correction Refinement**: in the following steps, the model iteratively expands, suppresses, or reshapes these initial activations, correcting false positives and consolidating fragmented predictions into coherent manipulated regions.

Across the entire test dataset, RITA follows a stable three-step prediction schedule, and this autoregressive pattern is consistently observed: the first step identifies salient manipulation traces, the second step further discovers additional related regions by conditioning on earlier predictions, and the third step revises and corrects errors introduced in previous steps to produce a clean and coherent final mask. These observations demonstrate that even in a one-step evaluation dataset

4. Structure Matching with Dynamic Programming

This section provides the full algorithmic details of MonotonicMatch(Algorithm 1), which is referenced in the main paper (Sec. 3.1.3, Hierarchical Sequential Score) and deferred to the Supplementary Material.

A core challenge in evaluating the predicted path is that its length (T_p) may differ from the ground-truth path's length (T_g). A simple frame-by-frame comparison is therefore inadequate. To address this, we introduce **MonotonicMatch**, a dynamic programming algorithm designed to find the optimal alignment between the two sequences.

The goal of MonotonicMatch is to identify a monotonic, non-decreasing mapping between the steps of the predicted sequence and the ground-truth sequence. This mapping is "optimal" in that it maximizes the cumulative stepwise F1 score along the alignment path. First, we compute a pairwise F1 score matrix $F \in \mathbb{R}^{T_p \times T_g}$, where each element $F[i, j]$ represents the F1 score between the i-th predicted mask and the j-th ground-truth mask. Then, a dynamic programming table is populated to find the path that yields the highest average F1 score, enforcing the monotonic constraint. The detailed procedure is outlined in Algorithm 1.

5. Robustness

This robustness analysis belongs to the main paper, Section 4.3.1 (Performance Comparison), and complements the quantitative comparison by evaluating model stability under realistic perturbations.

To comprehensively assess the stability of different models under realistic conditions, we consider three representative perturbation families that widely occur in practical

Table 2. **Robustness under common image perturbations.** Entries are *mean Binary F1* on the test set computed under the CAT-Net evaluation protocol. Perturbations include Gaussian noise (standard deviation), Gaussian blur (kernel size), and JPEG compression (quality factor). The rightmost *Average* is the arithmetic mean of the per–condition means within each block (including "None").

Pertubation	Model	Sandard Deviations							Average
		None	3	7	11	15	19	23	
GaussNoise	MVSS	0.495	0.502	0.500	0.492	0.493	0.489	0.489	0.494
	CAT-Net	0.533	0.512	0.500	0.484	0.473	0.462	0.454	0.488
	PSCC	0.555	0.539	0.531	0.522	0.521	0.518	0.512	0.528
	Trufor	0.531	0.450	0.418	0.398	0.381	0.366	0.372	0.417
	Mesorch	0.593	0.563	0.543	0.529	0.521	0.517	0.507	0.539
	Ours	0.643	0.622	0.609	0.602	0.598	0.592	0.590	0.608
		Kernel Size							Average
		None	3	7	11	15	19	23	
GaussBlur	MVSS	0.495	0.422	0.349	0.310	0.273	0.244	0.225	0.331
	CAT-Net	0.533	0.487	0.458	0.429	0.417	0.402	0.392	0.445
	PSCC	0.555	0.509	0.454	0.414	0.377	0.343	0.310	0.423
	Trufor	0.531	0.422	0.367	0.317	0.254	0.191	0.147	0.318
	Mesorch	0.593	0.526	0.471	0.430	0.387	0.340	0.292	0.434
	Ours	0.643	0.577	0.523	0.505	0.485	0.467	0.456	0.522
		Quality Factors							Average
		None	100	90	80	70	60	50	
JpegCompression	MVSS	0.495	0.493	0.462	0.434	0.408	0.392	0.369	0.436
	CAT-Net	0.533	0.547	0.522	0.495	0.484	0.480	0.474	0.505
	PSCC	0.555	0.534	0.478	0.449	0.435	0.423	0.392	0.467
	Trufor	0.531	0.481	0.437	0.408	0.387	0.366	0.318	0.418
	Mesorch	0.593	0.577	0.527	0.508	0.506	0.495	0.465	0.524
	Ours	0.643	0.639	0.604	0.586	0.575	0.565	0.539	0.593

imaging scenarios: (i) *Gaussian noise*, which emulates sensor-level corruption and thermal noise during acquisition; (ii) *Gaussian blur*, which mimics defocus and motion blur frequently introduced by imperfect optics or camera shake; and (iii) *JPEG compression*, which reflects storage and transmission artifacts due to lossy coding. For each perturbation family, we progressively increase the perturbation intensity, thereby creating a spectrum of degradation levels that range from mild to severe.

Following the CAT-Net evaluation protocol, we compute the mean Binary F1 score for every condition, and summarize the overall performance with a block-wise *Average*, defined as the arithmetic mean of the per–condition means (including the "None" case). This evaluation protocol ensures fairness across methods and allows us to capture the sensitivity of each model to different perturbation strengths.

Results are reported in Table 2. Across all perturbation families, our method consistently achieves the highest averages, obtaining **0.608** under Gaussian noise, **0.522** under Gaussian blur, and **0.593** under JPEG compression. These results represent substantial margins over the strongest baselines (e.g., Mesorch: 0.539/0.434/0.524; CAT-Net: 0.488/0.445/0.505), highlighting the robustness of our approach. More importantly, we observe that prior methods degrade significantly when perturbation severity increases—for example, Trufor exhibits rapid performance drops under blur and noise, and PSCC becomes unstable under low-quality JPEG factors. By contrast, our method maintains comparatively stable scores even at the most extreme settings, such as large noise standard deviations, large blur kernels, and very low JPEG quality.

Taken together, these experiments demonstrate that our model generalizes robustly to realistic degradations and is thus more reliable for deployment in unconstrained environments.

6. Quantitative Experiment

6.1. Results on Traditional Datasets

This qualitative analysis corresponds to the main paper, Section 4.3 (Existing Manipulation Scenarios), and complements the quantitative comparisons with visual evidence.

In addition to quantitative benchmarks, we further provide qualitative comparisons in Figure 3. We randomly selected two non-semantically manipulated images and five semantically manipulated images according to their proportions in the dataset, covering diverse object categories, background contexts, and manipulation types. This setup ensures a representative evaluation across both manipulations with clear semantic meaning (e.g., replacing or altering salient objects) and manipulations that operate at a lower, often background or texture level without explicit semantic cues.

Compared with state-of-the-art baselines, our method produces more precise and coherent masks. Specifically, the predicted regions not only align with the manipulated object layout but also preserve fine-grained boundaries, even in challenging cases involving subtle splicing, occlusion, or background-level editing. For semantically manipulated examples, our model accurately captures the global structure of manipulated objects while suppressing false positives in unaltered areas, which is crucial for practical scenarios where semantic consistency is essential. In contrast, existing baselines often either over-segment (producing large false-positive regions) or under-segment (missing critical manipulated parts), leading to incomplete or noisy masks.

For non-semantically manipulated cases, where manipulations are less visually salient and often manifest as texture inconsistencies or geometric misalignments, our autoregressive paradigm demonstrates robustness by sequentially refining predictions and yielding compact, accurate masks that isolate the true tampered regions. Unlike Traditional feed-forward architectures that may overlook weak or ambiguous traces, the autoregressive design enables iterative reasoning across spatial contexts, progressively consolidating local evidence into globally consistent predictions. Such a mechanism is particularly important in real-world images, where manipulations may be subtle and interwoven with natural variations.

Overall, these qualitative results confirm that our design generalizes well across manipulation types, successfully handling both semantically meaningful and background-level manipulations. They further illustrate that the autoregressive paradigm provides a principled way to enhance manipulation localization, leading to high-fidelity results and making our method better suited for deployment in diverse and unconstrained environments compared with prior approaches.

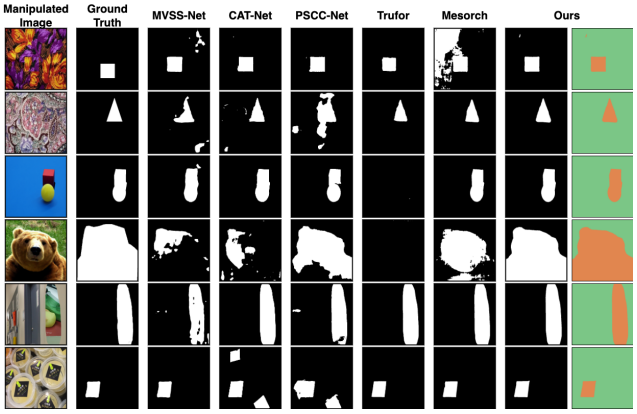

Figure 3. Qualitative analysis of SOTA models on Traditional datasets. We randomly selected and compared two semantically manipulated images and five non-semantically manipulated images according to their proportions in the dataset. The first two rows show non-semantically manipulated examples, while the last four rows correspond to semantically manipulated cases. The rightmost column presents our two-step reasoning results: the orange region indicates Step 1, and the green region Step 2. The second-to-rightmost column shows the corresponding 0/1 mask outputs.

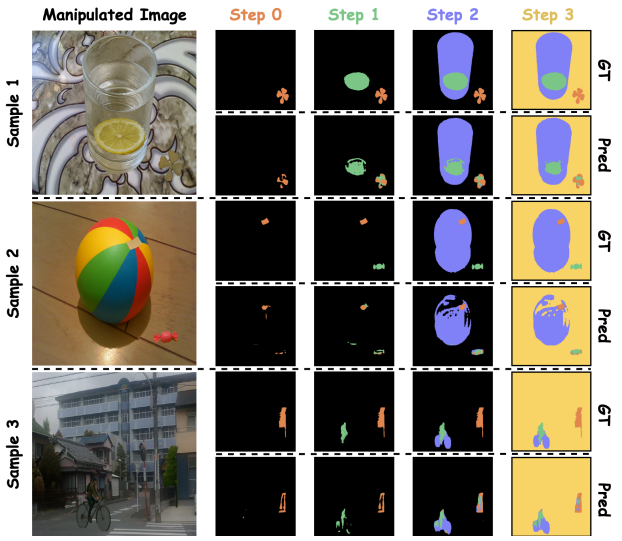

Figure 4. Qualitative results on the proposed HSIM dataset. Each row corresponds to one sample, where columns show sequential tampering steps (Step 0–Step 3). GT denotes the ground-truth mask at each stage, while Pred indicates our predictions. Our method successfully tracks manipulation evolution across steps, aligning predictions with the progressive nature of multi-step edits.

6.2. Results on Sequence Manipulation

This qualitative analysis is part of the main paper, Section 4.4 (Sequence Manipulation Scenario), and provides visual

illustrations of multi-step localization behavior.

To further evaluate the capability of our autoregressive framework, we conduct qualitative analysis on the proposed multi-step manipulation dataset, as shown in Figure 4. Unlike Traditional benchmarks where manipulations are applied once, these cases involve sequential tampering operations, progressively altering different regions or objects in the same image. Such a setup better reflects real-world scenarios, where images may undergo multiple edits over time.

As illustrated in Figure 4, our method is able to trace manipulation evolution across steps, from the initial local insertion (Step 0) to cumulative object-level alterations (Step 3). The predicted masks closely align with ground-truth annotations at each stage, successfully distinguishing newly introduced manipulations from previously existing edits. This progressive localization ability highlights the effectiveness of our autoregressive paradigm: rather than collapsing all manipulations into a single mask, it incrementally builds up tampering evidence in a temporally consistent manner.

For example, in Sample 1, the small inserted pattern (Step 0) is correctly identified, followed by precise delineation of the lemon slice (Step 1) and cup boundary (Step 2). In Sample 2, where both object-level (ball) and small patch manipulations are present, our model captures the structural evolution while avoiding confusion between new and old tampering. Finally, in Sample 3, the method robustly localizes subtle manipulations across different semantic categories (e.g., structural edits on poles and bicycles), demonstrating generalization across diverse manipulation styles.

These results suggest that our framework is inherently well-suited for multi-step tampering detection, offering interpretability by revealing *how* manipulations accumulate and evolve, which is not possible with Traditional one-shot segmentation baselines.

7. RITA as a Process-Guided Data Synthesis Instructor

This section provides the detailed content of the case study introduced in Section 4.5 of the main paper, illustrating how RITA guides the forgery synthesis pipeline to produce more precise and process-aligned synthetic data.

In manipulation localization, synthesizing training samples that include manipulation types unseen during training is one of the most important strategies for improving model generalization [14, 18]. Existing image and document forgery detectors typically rely on manually designed editing pipelines or randomly sampled combinations of atomic operations to create synthetic forgeries. [15, 16] These synthetic images are then fed to the base model in an attempt to improve robustness to newly emerging manipulation types. However, such heuristic strategies cannot guarantee realistic editing logic, and the generated manipulations often deviate significantly from the characteristics of these new forms of tampering.

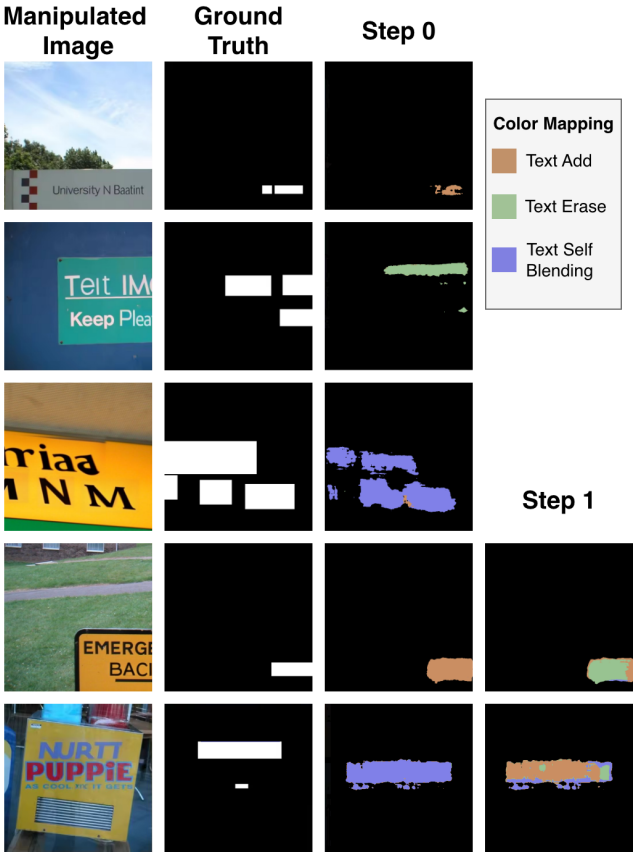

Figure 5. Examples of RITA-guided atomic-step decompositions on the OSTF dataset. RITA identifies meaningful atomic operations—such as text addition, text erasing, and text self-blending.

In contrast, this case study demonstrates that RITA acts as a reliable and structured *instructor* for data synthesis. Instead of relying on manually hypothesized operation sequences or randomly composed atomic operations, RITA extracts the latent manipulation process from an unseen sample, decomposes it into reusable atomic editing steps, and provides precise, process-aligned guidance on how synthetic data should be constructed. Importantly, RITA does *not* directly synthesize images; rather, it provides target-domain editing programs that can be applied to clean images, enabling synthetic manipulations whose editing logic faithfully matches the target domain. This produces consistent, transferable, and stylistically accurate synthetic data, improving generalization to unseen manipulation types.

7.1. Cross-Domain Setup and Baseline Performance

We adopt an $A \rightarrow B$ cross-domain setting, where domain A is TSROIE [19] and domain B is OSTF [16]. Three representative single-step document manipulation localiza-

tion models (Mesorch [20], FFDN [2], and DTD [15]) are trained on the original TSROIE training set (2747 images) and evaluated on the OSTF test set to obtain baseline cross-domain performance.

We adopt an $A \rightarrow B$ cross-domain setting, where domain A is TSOIRE [19] and domain B is OSTF [16]. These single-step models (Mesorch [20], FFDN [2], and DTD [15]) are trained using the ForensicHub [4] framework on the original TSOIRE training set (2747 images) and then evaluated on the OSTF test set to obtain baseline cross-domain performance.

7.2. Training RITA with Atomic Editing Operations

We then train a RITA model on our private document dataset, which contains 23 strictly defined atomic manipulation types, including mosaic variants (such as `mosaic_block_blur` and `mosaic_block_random_color`), text addition (`text_add`), text erasing (`text_erase`), text removal (`text_remove`), and self-blending (`text_self_blending`). Unlike the RITA in the main paper, pixel values here represent atomic operation categories directly, and the temporal order is expressed only by the index of each step in the output sequence. Each predicted mask encodes one atomic operation type, and the full sequence forms a reverse-order editing program.

7.3. Atomic Decomposition of Unseen OSTF Samples

After training, we apply RITA to manipulated samples from OSTF. As shown in Fig. 5, RITA predicts a short and plausible atomic sequence, typically:

$$\text{text\_erase} \rightarrow \text{text\_add} \rightarrow \text{text\_self\_blending}.$$

Here, `text_add` corresponds to inserting new text using the `PIL` rendering functions, `text_self_blending` is implemented by applying motion blur and median filtering to simulate blended text regions, and `text_erase` is performed by using LaMa [17] to remove the specified text region.

7.4. Constructing the Cross-Domain Augmented Set TSROIE_AUG

Using the atomic editing programs recovered from OSTF (domain B), we apply these programs to clean images from TSROIE (domain A) and construct 2551 guided synthetic samples, denoted as `TSROIE_AUG`. RITA only provides the editing instructions; the actual synthesis is performed by applying these instructions to clean images. This aligns TSROIE's manipulation logic with OSTF while preserving its visual appearance, achieving $A \rightarrow B$ manipulation-style transfer.

We retrain Mesorch, FFDN, and DTD on the combined training set (TSROIE_TRAIN + TSROIE_AUG) and evaluate them on OSTF. As shown in Table 3, all models achieve substantial improvements.

Table 3. Effect of RITA-based atomic-program augmentation on the OSFT dataset. Random augmentation is denoted by †.

Model	Before	Afte	Δ
Mesorch	0.1749	**0.3770**	+0.2021
Mesorch † (random)	—	0.0265	—
FFDN	0.2665	**0.3849**	+0.1184
FFDN † (random)	—	0.0412	—
DTD	0.1921	**0.2311**	+0.0390
DTD † (random)	—	0.0583	—

7.5. Ablation Study: Failure of Random Atomic Combinations

To rule out the possibility that the performance gains arise from generic random augmentation, we perform a controlled ablation in which both the atomic operation types and their ordering are sampled uniformly at random. This form of augmentation provides no structural guidance, ignores the editing logic commonly present in real forgeries, and produces manipulation patterns that are highly inconsistent with those observed in the target domain.

As shown in Table 3, random augmentation leads to extremely poor performance across all models. For example, the Mesorch model, which achieves 0.1749 before augmentation and 0.3770 with RITA-guided augmentation, collapses to only 0.0265 under random augmentation, losing almost all discriminative ability. Similar catastrophic degradation is observed for FFDN and DTD, confirming that random atomic combinations introduce severe distributional mismatches rather than useful diversity.

This stark contrast highlights that only structured, process-aligned atomic sequences recovered by RITA can produce effective synthetic data. Random compositions fail to approximate realistic manipulation processes and therefore provide no benefit for model generalization. RITA's guidance is thus not merely helpful but essential for generating transferable and domain-consistent training samples.

8. Ethics Statement

The HSIM dataset used in this paper was entirely created and manually annotated by the authors' research team. No external annotators or third-party contributors were involved in the data collection, manipulation, or labeling process. All data were generated under controlled conditions, contain no personally identifiable information, and fully comply with ethical standards for academic research.

References

[1] Xinru Chen, Chengbo Dong, Jiaqi Ji, Juan Cao, and Xirong Li. Image manipulation detection by multi-view multi-scale supervision. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, page 14165–14173, Montreal, QC, Canada, 2021. IEEE. 2

[2] 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 *European Conference on Computer Vision*, pages 200–217. Springer, 2024. 7

[3] Jing Dong, Wei Wang, and Tieniu Tan. Casia image tampering detection evaluation database. In *2013 IEEE China Summit and International Conference on Signal and Information Processing*, page 422–426, Beijing, China, 2013. IEEE. 1

[4] Bo Du, Xuekang Zhu, Xiaochen Ma, Chenfan Qu, Kaiwen Feng, Zhe Yang, Chi-Man Pun, Jian Liu, and Jizhe Zhou. Forensichub: A unified benchmark codebase for all-domain fake image detection and localization, 2025. 7

[5] Haiying Guan, Mark Kozak, Eric Robertson, Yooyoung Lee, Amy N. Yates, Andrew Delgado, Daniel Zhou, Timothee Kheyrkhah, Jeff Smith, and Jonathan Fiscus. Mfc datasets: Large-scale benchmark datasets for media forensic challenge evaluation. In *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, page 63–72, Waikoloa Village, HI, USA, 2019. IEEE. 1

[6] Fabrizio Guillaro, Davide Cozzolino, Avneesh Sud, Nicholas Dufour, and Luisa Verdoliva. 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*, pages 20606–20615, 2023. 1, 2

[7] Yu-feng Hsu and Shih-fu Chang. Detecting image splicing using geometry invariants and camera characteristics consistency. In *2006 IEEE International Conference on Multimedia and Expo*, page 549–552, Toronto, ON, Canada, 2006. IEEE. 1

[8] Shan Jia, Mingzhen Huang, Zhou Zhou, Yan Ju, Jialing Cai, and Siwei Lyu. Autosplice: A text-prompt manipulated image dataset for media forensics. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 893–903, 2023. 1

[9] Vladimir V Kniaz, Vladimir Knyaz, and Fabio Remondino. The point where reality meets fantasy: Mixed adversarial generators for image splice detection. 2019. 1

[10] 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. *International Journal of Computer Vision*, 130(8):1875–1895, 2022. 1, 2

[11] Xiaohong Liu, Yaojie Liu, Jun Chen, and Xiaoming Liu. 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. 2

[12] Xiaochen Ma, Xuekang Zhu, Lei Su, Bo Du, Zhuohang Jiang, Bingkui Tong, Zeyu Lei, Xinyu Yang, Chi-Man Pun, Jiancheng Lv, et al. Imdl-benco: A comprehensive benchmark and codebase for image manipulation detection & localization. *Advances in Neural Information Processing Systems*, 37:134591–134613, 2025. 1

[13] Adam Novozamsky, Babak Mahdian, and Stanislav Saic. Imd2020: A large-scale annotated dataset tailored for detecting manipulated images. In *2020 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, page 71–80, Snowmass Village, CO, USA, 2020. IEEE. 1

[14] Jeongsoo Park and Andrew Owens. Community forensics: Using thousands of generators to train fake image detectors. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8245–8257, 2025. 6

[15] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5937–5946, 2023. 6, 7

[16] Chenfan Qu, Yiwu Zhong, Fengjun Guo, and Lianwen Jin. Revisiting tampered scene text detection in the era of generative ai. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 694–702, 2025. 6, 7

[17] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2149–2159, 2022. 7

[18] Wenhao Wang, Longqi Cai, Taihong Xiao, Yuxiao Wang, and Ming-Hsuan Yang. Scaling laws for deepfake detection. *arXiv preprint arXiv:2510.16320*, 2025. 6

[19] Yuxin Wang, Boqiang Zhang, Hongtao Xie, and Yongdong Zhang. Tampered text detection via rgb and frequency relationship modeling. *Chinese Journal of Network and Information Security*, 8(3):29–40, 2022. 6, 7

[20] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11022–11030, 2025. 2, 7