CorrFill: Enhancing Faithfulness in Reference-based Inpainting with Correspondence Guidance in Diffusion Models – Supplementary Materials

Kuan-Hung Liu[1] Cheng-Kun Yang[2*] Min-Hung Chen[3] Yu-Lun Liu[1] Yen-Yu Lin[1]
[1]National Yang Ming Chiao Tung University [2]National Taiwan University [3]NVIDIA
https://corrfill.github.io/

In this supplementary material, we provide implementation details, showcase examples to support our proposed approaches, and present advanced analyses of the proposed method.

1. Implementation Details

1.1. Implementation Details of Baselines

For comparisons, we implement all baseline methods using the Python library Diffusers [3]. For Paint-by-Example [4], we utilize their publicly released model weights. Side-by-side [1] is essentially an inpainting base model, and we directly utilize Stable Diffusion v2 Inpainting model[1] as its implementation. In the case of LeftRefill [1], we use the same model of Side-by-side and incorporate LeftRefill's learned prompt embedding. We integrate IP-Adapter-Plus [5] module into a Stable Diffusion Inpainting model using their released pre-trained weights.

1.2. Implementation Details of CorrFill

CorrFill modify baseline models by substituting the attention processing function across all self-attention layers. Correspondence estimation and attention masking are then carried out in the substituted function. We also collect the attention maps used to optimize input latent tensor z_t in the attention processing function, and the gradients are computed in the denoising main loop of the diffusion models. Since optimizing z_t requires additional memory, a gradient accumulation strategy can be employed to trade off inference time for lower memory requirements. We conduct the experiments using an NVIDIA RTX A5000 GPU with 24GB of memory.

1.3. Details of Dataset Sampling

RealEstate10K is a video dataset comprising approximately 80,000 clips sourced from YouTube. Given that the clips are recorded by cameras with stable trajectories, adjacent frames tend to exhibit high similarity. Therefore, when selecting image pairs from RealEstate10K, we specifically consider frames that are separated by 30 frames during the sampling process.

1.4. Choices of Parameters

The parameters used in the comparisons presented in the main papers are reported in Table 1. Step$_a$ and Step$_o$ represent the number of steps guided by attention masking and latent tensor optimization, respectively, out of a total of 50 sampling steps. Win$_a$ is the radius that determines the neighborhood of a token used in the creation of attention masks, and Win$_s$ is the radius that determines the neighborhood for the weighted average used in attention smoothing. Str$_a$ and Str$_o$ indicate the value v added to the attention mask and the weight for controlling the guidance strength of latent tensor optimization, respectively.

We selected the parameters by evaluating the subsets of our datasets. During this evaluation, we tested various parameter settings and observed their responses in the results of different baseline methods and datasets. The general strategy is to increase the influence of guidance for the combinations that can significantly benefit from enhanced faithfulness.

2. Effectiveness of Proposed Components

2.1. Attention Smoothing

In the quantitative ablation study presented in the main paper, the performance gains from attention smoothing are not particularly significant. However, we provide one example demonstrating how attention smoothing serves as a crucial component in achieving accurate inpainting results in Figure 1.

2.2. Correspondence Update Policies

In this section, we demonstrate the effectiveness of two policies including cyclic enhancement and accumulation of attention maps over timesteps. We conduct a comparison of the correctness of the estimated correspondences against two counterparts excluding the two policies on

[*] Now at MediaTek Inc., Taiwan.

[1] https://huggingface.co/stabilityai/stable-diffusion-2-inpainting

Parameter	Paint-by-Example		IP-Adapter-Plus		Side-by-side		LeftRefill	
	RealEstate10K	MegaDepth	RealEstate10K	MegaDepth	RealEstate10K	MegaDepth	RealEstate10K	MegaDepth
Step$_a$	50	25	25	25	50	25	5	5
Step$_o$	50	50	50	50	50	50	50	5
Win$_a$	4(t)	4(t)	0.3(i)	5(t)	2(t)	3(t)	0.3(i)	2(t)
Win$_s$	0.4	0.4	0.2	0.2	0.2	0.2	0.05	0.2
Str$_a$	1	1	1	1	1	1	1	0
Str$_o$	2	0.5	2	0.5	2	0.5	0.5	0.5

Table 1. **List of parameters.** The comparisons presented in the main papers are conducted using these parameters. (t) indicates that the value refers to the number of tokens, and (i) denotes that the value is the ratio to the size of encoded images, *i.e.*, h'. For Win$_s$, all the values are the ratios to the size of encoded images.

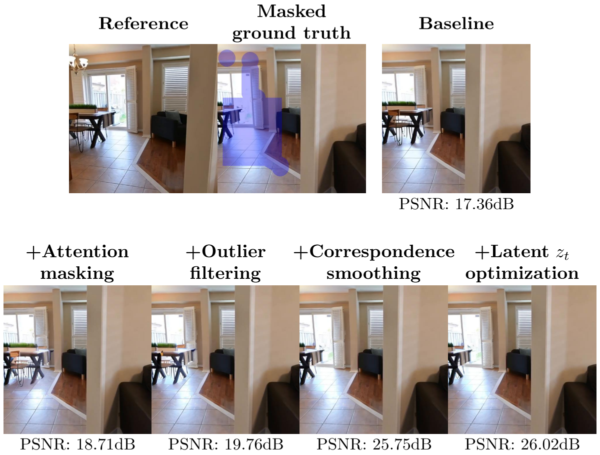

Figure 1. **Importance of Smoothing.** An example where correspondence smoothing is the pivotal component for correcting the incorrect geometry in the result of the baseline [1].

RealEstate10K. To estimate the correctness of the correspondences, we generate pseudo-ground truth correspondences using an image matching method [2]. We define a correspondence with an error within the size of one token as a correct correspondence. The counterpart that does not accumulate attention scores over time utilizes the most recently produced correspondences for guidance. The counterpart without cyclic enhancement is guided by the correspondences computed in the first step. The average numbers of correct correspondences during different stages of the inpainting process are illustrated in Figure 2. The counterpart "No acc" fails to achieve stability, while "No cyc." relies on the correspondence produced in the first step for guidance, resulting in inferior results. The PSNR performance results for "Ours", "No acc.", and "No cyc." are 27.39dB, 27.34dB, and 27.25dB, respectively.

3. Further Analysis

3.1. Time Efficiency

We analyze the average execution time for the inpainting of a single input with different key components enabled, following the experimental settings described earlier. The average execution times with LeftRefill as the baseline are reported in Table 2, which indicates that the latent input optimization contributes the most additional execution time within the proposed method. The increase in execution time is primarily attributed to the necessity of gradient calculation during each denoising iteration.

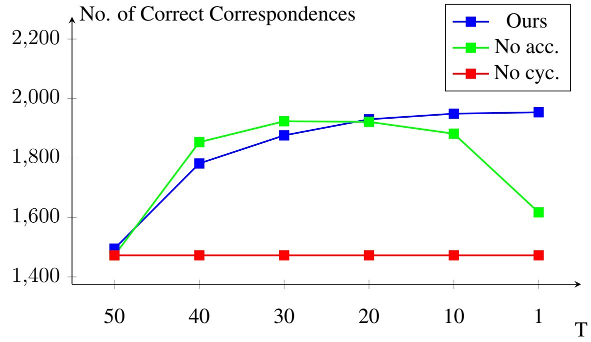

Figure 2. **Numbers of correct correspondences.** The graph illustrates the numbers of correct correspondences for three versions of CorrFill. "T" denotes the timesteps of the reverse process, where inpainting progresses from T = 50 to 0. "Ours" represents our proposed method, which utilizes cyclic enhancement and estimates correspondences using aggregated attention scores across different timesteps. "No acc." and "No cyc." are the counterparts that exclude the accumulation of attention maps and cyclic enhancement, respectively.

Method	Execution Time(s)	Change(s)
Baseline	6.69	-
+ Attention Masking	13.77	+7.08
+ Outlier Filtering	14.97	+1.20
+ Correspondence Smoothing	15.76	+0.79
+ Latent z_t Optimization	66.52	+50.76

Table 2. **Time analysis of key components of CorrFill.** The execution times for the inpainting of an input were measured while incrementally enabling the key components. The baseline used in the analysis is LeftRefill.

3.2. Extreme Case

Since CorrFill is an improvement method designed to enhance faithfulness, it encounters certain extreme cases

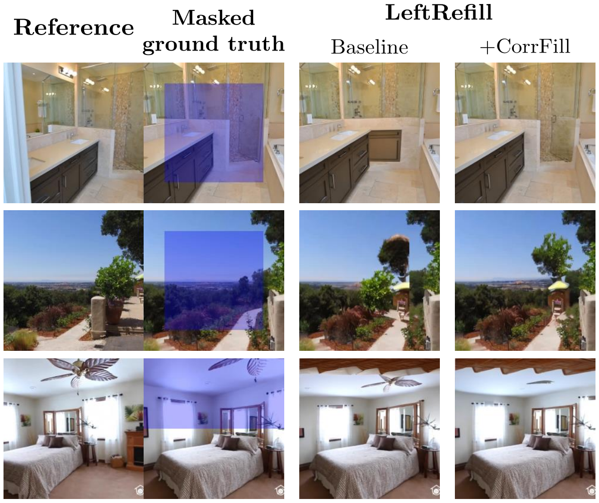

Figure 3. **Results with large masks.** The inpainting and outpainting results for the baseline method and CorrFill are presented. The first two rows depict the inpainting results, while the last row illustrates the outpainting results. All masks cover 50% of the target images. CorrFill cannot consistently enhance the results due to the significant degradation in the inpainting performance of the baseline method.

that challenge its performance, particularly when baseline models struggle to address them. While we previously discussed the issue of significant geometric variation in the main paper, another notable challenge for the baseline models involves large masks. The ratios of masked pixels for our generated pairs of inputs typically range from 10% to 40%. We find that when faced with larger masks, the inpainting results produced by LeftRefill tend to degrade to a point where CorrFill is unable to enhance faithfulness effectively. Figure 3 illustrates this limitation of CorrFill that it relies on the robustness of the baseline model. While CorrFill successfully improves the results for the first row, it does not yield similar improvements for the other cases.

References

[1] Chenjie Cao, Yunuo Cai, Qiaole Dong, Yikai Wang, and Yanwei Fu. Leftrefill: Filling right canvas based on left reference through generalized text-to-image diffusion model. In *CVPR*, 2024. 1, 2

[2] Xuelun Shen, zhipeng cai, Wei Yin, Matthias Müller, Zijun Li, Kaixuan Wang, Xiaozhi Chen, and Cheng Wang. GIM: Learning generalizable image matcher from internet videos. In *ICLR*, 2024. 2

[3] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models, 2022. 1

[4] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. In *CVPR*, 2023. 1

[5] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arxiv:2308.06721*, 2023. 1

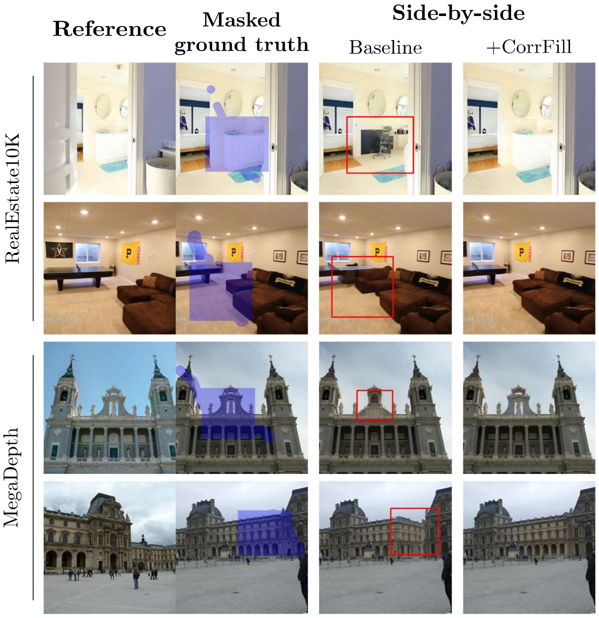

Figure 4. **Additional results with Side-by-side.** Problematic regions addressed by CorrFill are highlighted within the red boxes.

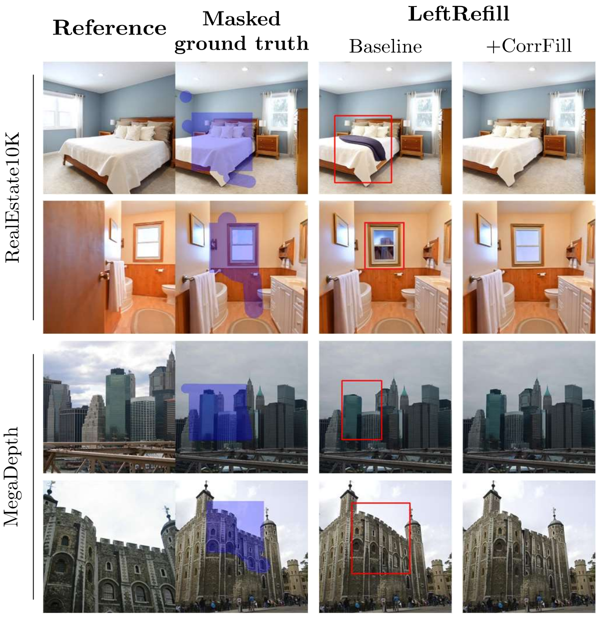

Figure 5. **Additional results with LeftRefill.** Problematic regions addressed by CorrFill are highlighted within the red boxes.