Proceedings of CONF-CDS 2026 Symposium: Computer Vision-Based Multimodal Cognitive Load Estimation for Adaptive Media Communication
DOI: 10.54254/2755-2721/2026.GL34635

PC-Refine: A Lightweight Residual Refinement Module for Parameter-Efficient Latent Diffusion Inpainting

Xiaotian Tian

School of Mathematical Sciences, Nankai University, Tianjin, China
13707579071@163.com

Abstract. Diffusion models have emerged as a powerful tool for high-quality image synthesis and are increasingly favored for restoration tasks such as image inpainting. This work investigates face inpainting within the latent diffusion framework, where denoising is performed by a UNet operating in a VAE-compressed latent space. We introduce PC-Refine, a lightweight residual refinement module attached to the mid-block of the U Net, and employ a parameter-efficient training strategy (pc_only) that freezes the original UNet while optimizing only the newly added parameters. Using DDIM-based sampling on the CelebAMask-HQ dataset, we evaluate performance with mask-only MAE and RMSE metrics focused on unknown regions. A controlled multi-seed evaluation demonstrates that PC-Refine consistently improves upon the baseline, showing that a single mid-block refinement yields practical, stable gains with minimal additional trainable parameters.

Keywords: Face Inpainting, Latent Diffusion, Parameter-Efficient, UNet, Refinement Module

1. Introduction

Image inpainting synthesizes plausible content for missing regions conditioned on visible context, and face inpainting additionally requires identity-consistent structure and realistic fine-scale detail. Recent progress in deep generative modeling supports semantically coherent completions, and diffusion models provide strong fidelity through iterative denoising [1]. Latent diffusion improves computational efficiency by performing diffusion in a compact VAE latent space [2].

Despite these advances, task adaptation for inpainting commonly relies on training or fine-tuning large denoising UNets, which incurs substantial compute, storage, and checkpoint- management overhead. Parameter-efficient adaptation methods, such as adapters [3] and low-rank updates [4], reduce training cost across several domains. Still, lightweight residual refinement within dif-fusion UNets remains underexplored for inpainting, especially for insertion choices that preserve pretrained compatibility and stable optimization.

This study investigates parameter-efficient latent diffusion inpainting for faces by introducing PC-Refine, a lightweight residual refinement module attached to the UNet mid-block, which aggregates global semantic information. The insertion location is motivated by the interaction between global semantics and bidirectional information flow across the U-shaped architecture. The

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.GL34635

design aims to decouple task adaptation from backbone retraining while maintaining compatibility with pretrained weights.

A mask-aware c onditioning formulation is adopted in latent space by concatenating the noisy latent, the visible-region latent, and a downsampled mask. Training follows a PC-only protocol that freezes the original UNet and updates only PC-Refine parameters. Masked-region error metrics define evaluation to isolate synthesis quality on unknown pixels under aligned sampling conditions.

This formulation is expected to reduce adaptation costs and improve reproducibility under limited resources, and to provide a basis for future extensions, such as iterative refinement.

schedules, finer-grained insertion into residual blocks, and perceptual criteria for quality assessment.

2. Related work

Image inpainting has progressed from classical patch-based methods, which often fail in complex scenes, to deep learning approaches that improve semantic plausibility via encoder–decoder architectures and masked convolutions. Among generative models, denoising diffusion probabilistic models (DDPMs) and their efficient samplers (e.g., DDIM [5]) have become particularly effective for conditional inpainting. In contrast, latent diffusion reduces computational cost by operating in a compact VAE-learned latent space.

Adapting such large diffusion models efficiently is a key practical concern. Parameter-efficient techniques such as adapters, LoRA, and lightweight residual branches have been widely adopted to fine-tune pretrained networks with minimal trainable parameters. In the context of diffusion-based inpainting, this paradigm enables injecting task-specific refinements into a frozen UNet backbone, achieving strong adaptation without retraining the entire model.

Predictive coding theory provides a conceptual basis for lightweight refinement, modelling perception as an iterative error correction process within a hierarchical generative system, where latent representations are continuously updated to minimise prediction errors while maintaining a stable backbone [6, 7]. This iterative refinement philosophy fits well with the goal of diffusion UNets, which is to make adaptation more efficient by using fewer parameters. These threads together suggest a way to combine the generative power of diffusion models with a simple, refinement-focused adaptation strategy that strikes a balance between semantic quality, computational efficiency, and architectural simplicity.

3. Method

3.1. Diffusion models

A diffusion model defines a forward noising process and learns a reverse denoising process [1]. In practice, the denoiser is implemented by a UNet [8] that predicts either the added noise or the clean sample at each timestep.

3.2. Latent difusion

Latent diffusion uses a VAE encoder E to map an image x to a latent z = E(x) [2, 9]. Diffusion is performed in latent space, and a decoder D reconstructs the final image = D(). This improves efficiency and allows higher-resolution training.

DOI: 10.54254/2755-2721/2026.GL34635

3.3. Task formulation

Given an input image x and a binary mask $M \in \{0, 1\}^{H \times W}$ indicating missing pixels, the goal is to reconstruct a complete image consistent with visible regions. Errors are evaluated only on the masked region to measure true inpainting quality.

3.4. Mask-aware UNet conditioning

A mask-aware conditioning scheme is adopted in the latent space. The UNet input is formed by concatenating (i) the noisy latent, (ii) the visible-region latent, and (iii) the downsampled mask, resulting in a 9-channel input tensor. This explicitly exposes known context to the denoiser.

3.5. PC-Refine module

PC-Refine is attached to the UNet mid-block as a residual refinement:

$$h' = h + f\theta(GN(h)) \tag{1}$$

where h is the mid-block feature map, GN(·) is Group Normalization, and fθ is a lightweight convolutional network. k denotes the number of refinement iterations, and nblocks denotes the depth of the refinement stack.

3.6. Parameter-efficient training (pc__only)

To reduce training costs, all original UNet parameters are frozen, and only the PC-Refine parameters θ are optimized. This yields a compact adaptation while keeping compatibility with pretrained weights.

4. Experiments

4.1. Dataset and preprocessing

Experiments are conducted on CelebAMask-HQ [10]. Images are resized to 256 × 256 using bicubic interpolation, center-cropped, and then normalized to [−1, 1] per channel. Binary masks are generated on the fly to simulate missing regions, including rectangular block masks and free-form stroke masks. The block masks are created by sampling an aspect ratio and area fraction, then placing a zero-valued rectangle at a random location. The free-form masks are drawn by a small number of random strokes with varying width and length, and a rejection step is applied to keep the missing-pixel fraction within a fixed range. This mixed mask strategy encourages robustness to both structured occlusions and irregular missing areas.

4.2. Implementation details

Both the baseline and PC-Refine variants are trained for 20k steps. For sampling, DDIM with 50 steps is used. To reduce randomness, results are aggregated over seeds 0–9, and for each seed, n = 32 images are sampled using aligned conditions.

Training is conducted in the latent diffusion setting using a pretrained VAE and UNet. The noise scheduler follows the standard DDPM formulation, and the UNet is trained to predict the added noise at uniformly sampled timesteps. Optimization uses AdamW with a constant learning rate and

DOI: 10.54254/2755-2721/2026.GL34635

fixed weight decay. Mixed precision (FP16) is enabled during training and sampling to reduce memory usage and speed up throughput. For the baseline model, the full UNet is optimized. For the parameter-efficient configuration, the pretrained UNet weights are frozen, and only PC-Refine parameters are updated, substantially reducing the number of trainable parameters and simplifying checkpoint management.

The sampling procedure is aligned between the baseline and PC-Refine models. For each seed, the same sampled image list and mask generation parameters are used, and the same DDIM steps are applied. This protocol isolates model differences from stochastic sampling effects, thereby providing a controlled comparison.

4.3. Evaluation metrics

Mask-only MAE and RMSE are reported. Let Ω = {(i, j) | Mij = 1} be the masked pixel set.

Then

$$MAE_{mask} = \frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} |\widehat{x}_{ij} - x_{ij}|, RMSE_{mask} = \sqrt{\frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} (\widehat{x}_{ij} - x_{ij})^2} \quad (2)$$

5. Results and ablations

This section summarizes quantitative comparisons and ablation studies.

5.1. PC-Refine module

To enhance the inpainting capability of a latent diffusion UNet while keeping training cost low, a lightweight residual refinement block (PC-Refine) is introduced and attached to the UNet mid-block. Let h ∈ RC×H×W denote the feature map at the UNet mid-block. PC-Refine applies a residual transformation:

$$\mathrm{h}' = \mathrm{h} + \mathrm{f}\theta(\mathrm{GN}(\mathrm{h})) \quad (3)$$

where GN(·) is Group Normalization and fθ(·) is a small convolutional network with SiLU activations. k denotes the number of refinement iterations applied sequentially; the best setting uses k = 1. During training, a parameter-efficient scheme (pc_only) is adopted, in which the original UNet weights are frozen and only PC-Refine parameters θ are optimized.

5.2. Evaluation Metrics

Reconstruction quality is evaluated only on the unknown (masked) region to avoid trivial improvements from unchanged known pixels. Let be the generated image, x the ground-truth image, and M ∈ {0, 1}H×W, the binary mask where Mij = 1 indicates masked (unknown) pixels to be synthesized. The mask-only MAE and RMSE are defined as follows:

$$MAE_{mask} = \frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} |\widehat{x}_{ij} - x_{ij}|, RMSE_{mask} = \sqrt{\frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} (\widehat{x}_{ij} - x_{ij})^2} \quad (4)$$

where Ω = {(i, j) | Mij = 1} is the set of masked pixels.

DOI: 10.54254/2755-2721/2026.GL34635

5.3. Experimental setup

Latent-space inpainting is performed using a UNet with 9-channel input formed by concatenating the noisy latent, the visible-region latent, and the downsampled mask. For sampling, DDIM with 50 steps is used. To reduce randomness, each reported number is aggregated over seeds 0... 9, and for each seed n = 32 images are sampled under aligned conditions. All quantitative results are computed only on the masked region.

5.4. Results

Table 1 shows that adding a lightweight PC-Refine block to the UNet mid-block and training it in a parameter-efficient manner (pc_only) yields consistent improvements over the baseline model trained for the same number of steps (20k). Specifically, PC-Refine achieves relative improvements of 3.35% in MAE mask and 2.55% in RMSE mask, and outperforms the baseline across all 10 evaluated random seeds. In contrast, placing refinement blocks at multiple coarse UNet locations (down/mid/up) or increasing mid-block depth (nblocks=2) does not further improve performance, suggesting that a single well-placed refinement at the mid-block provides the best trade-of between capacity and stability under the evaluated training setting.

Table 1. Ablation study on mask-only reconstruction metrics (mean±std over seeds 0–9). Lower is better

Method	Places	Train mode	MAE mask	RMSE mask
Baseline (20k)	–	full UNet	0.120475 ± 0.005665	0.187630 ± 0.013859
PC-Refine (best)	mid	pc__only	0.116414 ± 0.005272	0.182868 ± 0.014116
PC-Refine (multi-place)	down, mid, up	pc__only	0.120251 ± 0.006600	0.187884 ± 0.015211
PC-Refine (deeper mid)	mid (nblocks=2)	pc__only	0.119046 ± 0.006136	0.185605 ± 0.014676

6. Discussion and limitations

The empirical results suggest that a single refinement branch placed at the UNet mid-block can yield stable improvements under a constrained training budget. This observation is consistent with the intuition that the mid-block aggregates global semantic information and serves as a natural bottleneck for global–local interactions. At the same time, the ablations indicate that simply increasing refinement capacity (e.g., adding multiple insertion sites or stacking deeper refinement blocks) does not necessarily translate into better masked-region reconstruction. One plausible explanation is that excessive modification to the feature hierarchy may disrupt pretrained representations and degrade optimization stability under limited steps.

Several limitations remain. First, the evaluation focuses on a single dataset and a specific mask generation policy; performance may vary across different occlusion patterns or out-of-domain data. Second, the current analysis is driven primarily by mask-only reconstruction metrics under aligned sampling. While these metrics provide a controlled estimate of fidelity on the unknown region, they do not fully capture all aspects of visual realism. Third, the method is studied in a simplified conditional setting that uses a null text embedding. Stronger conditioning signals (e.g., text prompts or attribute guidance) may change the optimal insertion location and training protocol.

Despite these constraints, the proposed module offers a practical engineering trade-off. It allows reusing pretrained diffusion backbones and adapting them with a compact set of parameters, which is appealing when compute, storage, or iteration speed is limited.

DOI: 10.54254/2755-2721/2026.GL34635

7. Conclusion

PC-Refine is presented as a lightweight residual refinement module for parameter-efficient latent diffusion inpainting. With the PC-only training strategy, PC-Refine consistently improves mask-only MAE/RMSE relative to a baseline model across multiple seeds. Under DDIM sampling with 50 steps and aligned evaluation over seeds 0–9 with n = 32 samples per seed, the best configuration reduces mask-only MAE by 3.35% and mask-only RMSE by 2.55% relative to a baseline trained for 20k steps. Future work can proceed along multiple directions. On the modeling side, the refinement schedule can be generalized beyond a single pass (k = 1) by exploring iterative refinement with careful normalization and residual scaling, which may improve convergence without destabilizing pretrained features. On the architectural side, finer-grained insertion at specific residual blocks (rather than coarse down/mid/up placement) may yield more targeted adaptation. On the training side, different freezing strategies (e.g., unfreezing only a small subset of normalization or attention parameters) could be studied to balance stability and capacity. Finally, a broader robustness evaluation across different mask distributions and data domains would help characterize when a single mid-block refinement is sufficient and when additional adaptation is required.

References

[1] Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), Advances in Neural Information Processing Systems, 33, 6840–6851.

[2] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 10684–10695).

[3] Houlsby, N., Giurgiu, A., Jastrzębski, S., Morrone, B., de Laroussilhe, Q., Gesmundo, A., Attariyan, M., & Gelly, S. (2019). Parameter-efficient transfer learning for NLP. In Proceedings of the 36th International Conference on Machine Learning (ICML 2019) (pp. 2790–2799). PMLR.

[4] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, L., Wang, L., & Chen, W. (2022). LoRA: Low-rank adaptation of large language models. In International Conference on Learning Representations. Retrieved from https: //arxiv.org/abs/2106.09685

[5] Song, J., Meng, C., & Ermon, S. (2021). Denoising diffusion implicit models. In International Conference on Learning Representations. Retrieved from https: //openreview.net/forum?id=5GtGzU8wgH

[6] Rao, R. P. N., & Ballard, D. H. (1999). Predictive coding in the visual cortex: A functional interpretation of some extra-classical receptive-field effects. Nature Neuroscience, 2(1), 79–87. https: //doi.org/10.1038/4580

[7] Friston, K. (2010). The free-energy principle: A unified brain theory? Nature Reviews Neuroscience, 11(2), 127–138. https: //doi.org/10.1038/nrn2787

[8] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, & A. F. Frangi (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015 (Lecture Notes in Computer Science, Vol. 9351, pp. 234–241). Springer. https: //doi.org/10.1007/978-3-319-24574-4_28

[9] Kingma, D. P., & Welling, M. (2014). Auto-encoding variational Bayes. In Y. Bengio & Y. LeCun (Eds.), Proceedings of the 2nd International Conference on Learning Representations (ICLR 2014). (Conference paper). Retrieved from https: //arxiv.org/abs/1312.6114

[10] Lee, C.-H., Liu, Z., Wu, L., & Luo, P. (2020). MaskGAN: Towards diverse and accurate facial image inpainting. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops