
PixPerfect: Seamless Latent Diffusion Local Editing with Discriminative Pixel-Space Refinement

Supplementary Material

Haitian Zheng^{1*} Yuan Yao^{2*} Yongsheng Yu²
Yuqian Zhou¹ Zhe Lin¹ Jiebo Luo²

¹Adobe Research ²University of Rochester

{hazheng, yuqzhou, zlin}@adobe.com,

yynobug@gmail.com, yyu90@ur.rochester.edu, jluo@cs.rochester.edu

1 The Latent Space Spatial Disentanglement Issue

Latent diffusion models operates on a compact latent space. However, the latent space are spatially entangled and not suitable for pixel-wise tasks. In this section, we study the latent space disentanglement issue.

Latent diffusion models encode images into a compressed latent space with an autoencoder. However, this latent representation lacks spatial disentanglement, limiting its suitability for fine-grained local editing. To illustrate this issue, we design a controlled experiment shown in Fig. 1. We encode both the original image and its masked counterpart using FLUX VAE [1], then construct a hybrid latent by combining the unmasked background from the masked input with the masked region from the original. This ensures that the latent representation differs only within a small localized area.

If the VAE decoder preserved spatial locality, such a localized change would not affect the reconstruction outside the masked region. However, the decoded image exhibits global shifts in background appearance, even where latent features remain unchanged. This behavior highlights a fundamental limitation of the latent space: local modifications can induce unintended global effects due to entangled representations. These observations motivate our refinement strategy, which operates in the pixel space to preserve spatial locality and ensure coherent integration between edited and unedited regions.

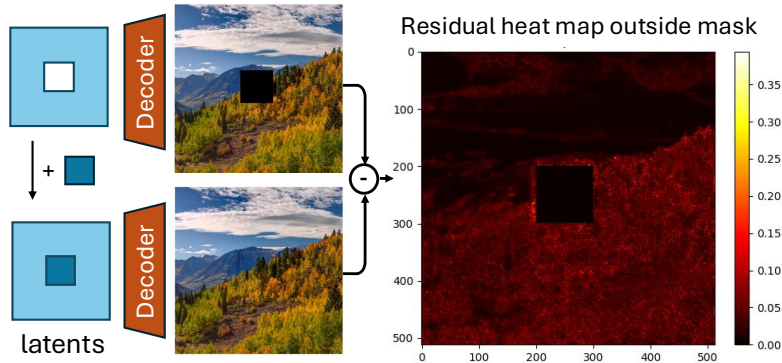


Figure 1: Replacing only the masked region in latent space leads to background drift in the decoded image, suggesting spatial entanglement in latent-based inpainting.

*Equal contribution.

2 More Experiments

2.1 Efficiency Analysis

While improving visual consistency and perceptual fidelity is the primary goal of our refinement framework, it is also critical that the added refinement stage does not significantly increase the overall runtime. To this end, we analyze the computational cost of PixPerfect in comparison to the underlying latent diffusion sampling process.

Our refiner operates as a single-stage feed-forward network in the pixel space, and introduces negligible overhead compared to the iterative denoising procedure of diffusion models. For example, when applied to a 512×512 image on a single NVIDIA A100 GPU, the diffusion sampling with FLUX-Fill [1] takes approximately 9.7 seconds, whereas our refiner adds only 2.7 seconds, accounting for only 21.8% of the total inference time.

Notably, our approach remains more efficient than additional diffusion-based refinement stage. This efficiency stems from two factors: (1) PixPerfect requires only a single forward pass without iterative sampling, and (2) its architecture is lightweight and resolution-agnostic, enabling low-latency execution. Even when inference-time pooling is enabled, the overall runtime remains within 1.3× of the baseline, while yielding measurable improvements in visual quality.

These results indicate that PixPerfect can be seamlessly integrated into existing diffusion pipelines with minimal computational cost, offering substantial perceptual gains at a fraction of the runtime.

2.2 Comparisons with Poisson Blending

In the main paper, we have presented the comparisons with decoder-based method Assymetric VQ-GAN [10] and harmonization-based method DiffHarmony [9]. In this section we will provide additional analysis about Poisson blending. Poisson blending is a classical gradient-domain technique widely used for seamless image compositing. It estimates a smooth transition between a source (edited) region and a target (background) image by solving for pixel values that minimize gradient differences while respecting boundary conditions.

However, applying Poisson blending in the context of inpainting or local editing typically requires access to a reliable gradient field within the masked region. In practice, this is often approximated using the ground truth content in the masked area to compute the desired gradients. While this produces visually smooth results, it introduces a critical ground-truth leakage issue—information that is unavailable at test time is used during blending. Consequently, Poisson blending cannot be considered a fair or deployable baseline in real-world settings.

Although Poisson blending relies on inaccessible ground-truth information, we still present some qualitative comparison results. We apply Poisson blending on the outputs of FLUX-Fill [1] using ground-truth-masked gradients to simulate its ideal behavior. Fig. 2 shows representative examples comparing our method with Poisson blending. While the latter can reduce abrupt seams at the boundary, it often introduces unnatural hue propagation and fails to correct texture inconsistencies or geometric artifacts introduced during the generation process. Furthermore, when the inpainted results differ from the original ground truth image, the Poisson blending will blend the masked part into the tone of the original ground truth and produce unnatural seams. In contrast, our method produces more coherent integration with the background, better preserves structural details, and eliminates color/texture artifacts without relying on inaccessible ground-truth information.

These results highlight that Poisson blending falls short in correcting complex local editing artifacts. Our learning-based refiner not only avoids the pitfalls of ground-truth leakage but also achieves better perceptual quality through semantically aware refinement.

2.3 More Qualitative results

To further illustrate the effectiveness and generalization of our approach, we present additional qualitative results for the two local editing tasks: object removal and object insertion. These tasks requires image editing within a masked area and keep the background unchanged.

In the object removal examples shown in Fig. 3, we present qualitative results from three representative baselines: OmniPaint [6], PowerPaint [11] and CLIPAway [3]. As indicated by the red

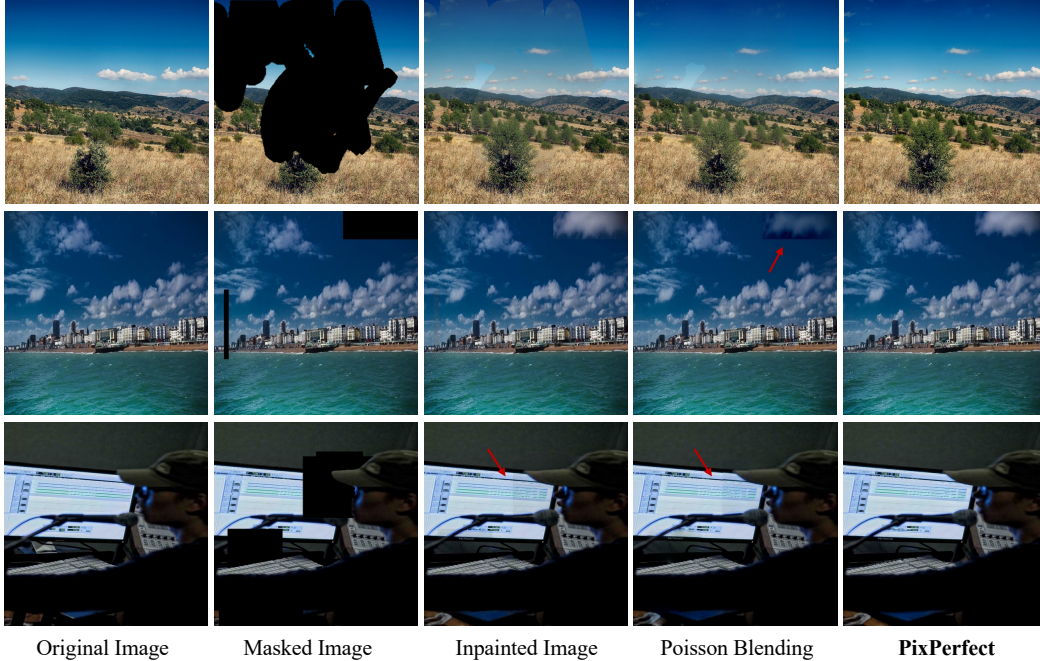


Figure 2: Qualitative comparison between our method and Poisson blending for FLUX-Fill inpainting outputs. While Poisson blending reduces edge discontinuities, it often introduces hue bleeding and fails to correct texture or structural artifacts. Further more in the cases where the inpainted results differ from the ground truth image (*e.g.* second row), poisson blending will tend to mimic the ground truth and produce unnatural results. In contrast, PixPerfect produces cleaner transitions, preserves scene structure, and avoids tone inconsistency without relying on inaccessible ground-truth information.

arrows, baseline inpainting results often exhibit low-level inconsistencies, such as chromatic shifts, particularly in regions of clean background such as floors and tables. In contrast, our method effectively eliminates these artifacts, yielding smooth and contextually coherent background completions without disrupting the surrounding scene geometry.

In the object insertion results shown in Fig. 4, we visualize our refinement performance on outputs from ObjectStitch [4], AnyDoor [2], and PBE [5]. In these cases, challenges arise from the need to harmonize inserted objects with scene textures and lighting. As highlighted in the magnified insets, baseline results often suffer from blurry transitions, scale-inconsistent textures, or unnatural object boundaries. Our method noticeably improves local consistency by refining high-frequency texture alignment, enhancing boundary sharpness, and reducing chromatic discrepancies—leading to more realistic and visually pleasing composites.

Overall, these examples demonstrate the general applicability of our method across diverse models and editing scenarios. In both insertion and removal tasks, PixPerfect consistently enhances visual quality by resolving local inconsistencies that are challenging for latent diffusion models alone. We encourage readers to examine the highlighted regions closely to appreciate the subtle yet impactful improvements brought by our approach.

3 Implementation Details

Architecture and Training. The refiner is built on the CMGAN architecture [8]. However, we replace the bottleneck fully-connected layer with a global average pooling operation, thereby making the network fully convolutional. In addition, we apply channel pruning to reduce the model size. Our final model contains 41M parameters. Training employs R1 regularization with $\gamma = 1$ and utilizes the CoModGAN mask generation scheme [7] to generate random masks on-the-fly. During an initial



Figure 3: Qualitative comparisons on object removal. Red arrows highlight residual artifacts such as color inconsistency produced by baseline diffusion models. Our method effectively eliminates such artifacts.

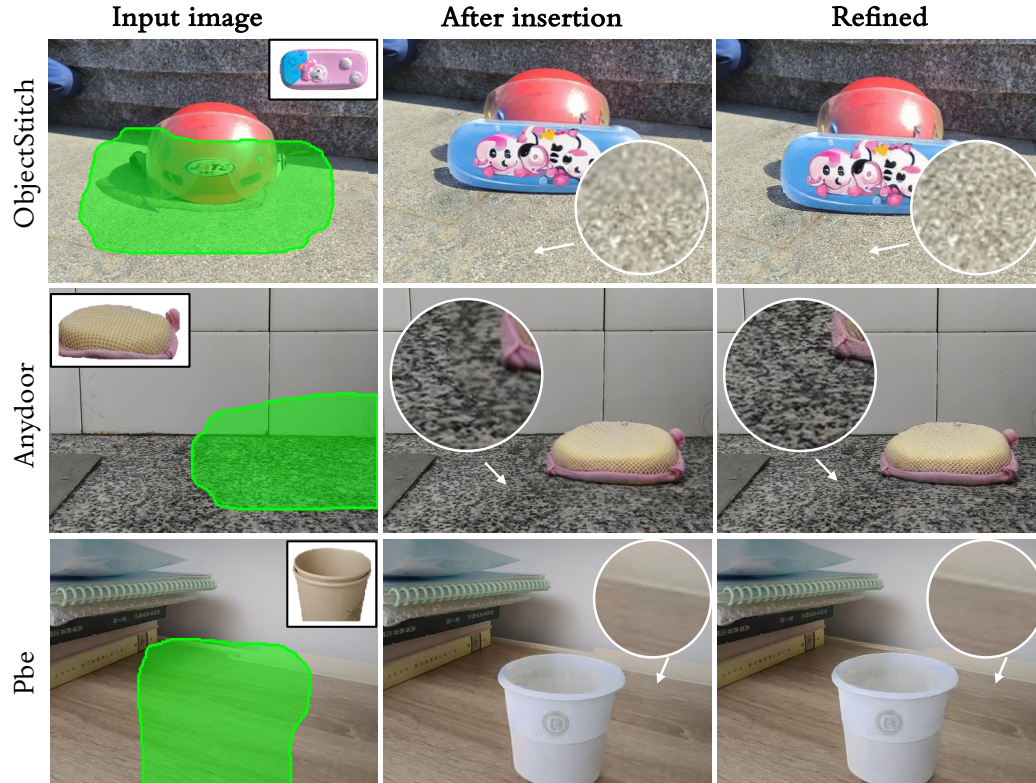


Figure 4: Qualitative comparisons on object insertion. The highlighted insets reveal artifacts in baseline results, such as blurry edges, inconsistent textures, and poor object blending. Our refinement enhances boundary sharpness, aligns local textures, and achieves more seamless visual integration.

Table 1: Summary of artifact types and their corresponding augmentation probabilities.

Artifact Type	Description	Probability
Content Discontinuity	Small misalignments / missing pixels near mask edges	0.5
Background Color Augmentation	Non-uniform hue / brightness variations applied to the background	0.8
Foreground Color Augmentation	Non-uniform / uniform / gradient color perturbations applied to the foreground region	0.8
Soft/Hard Boundary Mixing	Mixing soft / hard boundaries to mimic visual seams at compositional borders	1.0
Sensor Noise / JPEG / Blur	Injecting noise / JPEG compression / blur into foreground and/or background regions	0.5
VAE Compression Artifacts	Introducing compression artifacts simulated by a pretrained VAE to the foreground	0.5

warm-up phase, the discriminative pixel-space loss remains disabled. A constant learning rate of 5×10^{-4} is applied throughout the training.

Details on Color Shifting Augmentation. Three complementary color-shifting schemes are employed. First, *linear gradient color augmentation* constructs a mask α by projecting normalized x - y coordinate grids onto a randomly oriented unit vector and normalizing the result; the final image is obtained by alpha-blending this mask with a color-jittered version of the input. Second, *random blob color augmentation* synthesizes one or more soft ellipses per image—each defined by a random center, semi-axes sampled from a fraction of the image dimensions, and a random rotation—where pixel intensities decay smoothly from center to boundary; overlapping ellipses merge via a maximum operator to produce distinct, softly blended circular regions. Third, *uniform jitter augmentation* simulates spatially invariant color shifts by blending a uniformly color-jittered image with the original input using a fixed blending ratio. We provide an artifact generation pipeline that describes the artifact types and their corresponding augmentation probabilities in 1.

A minimal demo script for reproducing the “seam” artifacts of Flux inpainting [1] model. To facilitate reproducibility, we attached a minimal demo script that reproduces the boundary artifacts for the official FLUX-Fill model [1].

```

1 import torch
2 import numpy as np
3 from PIL import Image, ImageDraw
4 from diffusers import FluxFillPipeline
5 from diffusers.utils import load_image
6
7 # === Define input image path ===
8 input_image_path = "/your/image/path" # TODO: change to the input image path
9
10 # === Load input image ===
11 image = load_image(input_image_path).convert("RGB")
12 width, height = image.size
13
14 # === Generate irregular mask ===
15 def generate_irregular_mask(width, height, max_shapes=5):
16     mask = Image.new("L", (width, height), 0)
17     draw = ImageDraw.Draw(mask)
18
19     for _ in range(np.random.randint(1, max_shapes + 1)):
20         shape_type = np.random.choice(["ellipse", "polygon"])
21         if shape_type == "ellipse":

```



```

22         x0, y0 = np.random.randint(0, width - 50), np.random.randint(0, height
↪ - 50)
23         x1, y1 = x0 + np.random.randint(40, 120), y0 + np.random.randint(40,
↪ 120)
24         draw.ellipse([x0, y0, x1, y1], fill=255)
25         else:
26             num_points = np.random.randint(3, 8)
27             points = [(np.random.randint(0, width), np.random.randint(0, height))
↪ for _ in range(num_points)]
28             draw.polygon(points, fill=255)
29
30     return mask.convert("RGB")
31
32 mask = generate_irregular_mask(width, height)
33
34 # === Load FLUX inpainting pipeline ===
35 pipe = FluxFillPipeline.from_pretrained(
36     "black-forest-labs/FLUX.1-Fill-dev",
37     torch_dtype=torch.bfloat16
38 ).to("cuda")
39
40 # === Run FLUX-Fill ===
41 output = pipe(
42     image=image,
43     mask_image=mask,
44     prompt="",
45     height=height,
46     width=width,
47     guidance_scale=30, # The default value provided on the official huggingface page
48     num_inference_steps=50, # The default value provided on the official
↪ huggingface page
49     max_sequence_length=512 # The default value provided on the official
↪ huggingface page
50 ).images[0]
51
52 # === Composite: restore unmasked regions from original image ===
53 image_np = np.array(image)
54 output_np = np.array(output)
55 mask_np = np.array(mask.convert("L"))
56 inpainted_np = output_np.copy()
57 inpainted_np[mask_np < 128] = image_np[mask_np < 128]
58 inpainted = Image.fromarray(inpainted_np)
59
60 # === Save outputs ===
61 image.save("original.png")
62 mask.save("mask.png")
63 inpainted.save("inpainted.png")

```

Code 1: A minimal demo script for reproducing the “seam” artifacts of Flux inpainting [1] model.

References

- [1] Black Forest Labs. FLUX, 2024. URL <https://github.com/black-forest-labs/flux>. Accessed: 2025-05-15.
- [2] Xi Chen, Lianghua Huang, Yu Liu, Yujun Shen, Deli Zhao, and Hengshuang Zhao. Anydoor: Zero-shot object-level image customization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6593–6602, 2024.
- [3] Yiğit Ekin, Ahmet Burak Yildirim, Erdem Eren Çağlar, Aykut Erdem, Erkut Erdem, and Aysegül Dundar. Clipaway: Harmonizing focused embeddings for removing objects via diffusion models. *Advances in Neural Information Processing Systems*, 37:17572–17601, 2024.
- [4] Yizhi Song, Zhifei Zhang, Zhe Lin, Scott Cohen, Brian Price, Jianming Zhang, Soo Ye Kim, and Daniel Aliaga. Objectstitch: Object compositing with diffusion model. In *Proceedings of*

- the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18310–18319, 2023.
- [5] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18381–18391, 2023.
 - [6] Yongsheng Yu, Ziyun Zeng, Haitian Zheng, and Jiebo Luo. Omnipaint: Mastering object-oriented editing via disentangled insertion-removal inpainting. *arXiv preprint arXiv:2503.08677*, 2025.
 - [7] Shengyu Zhao, Jonathan Cui, Yilun Sheng, Yue Dong, Xiao Liang, Eric I Chang, and Yan Xu. Large scale image completion via co-modulated generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2021.
 - [8] Haitian Zheng, Zhe Lin, Jingwan Lu, Scott Cohen, Eli Shechtman, Connelly Barnes, Jianming Zhang, Ning Xu, Sohrab Amirghodsi, and Jiebo Luo. Image inpainting with cascaded modulation gan and object-aware training. In *European conference on computer vision*, pages 277–296. Springer, 2022.
 - [9] Pengfei Zhou, Fangxiang Feng, and Xiaojie Wang. Diffharmony: Latent diffusion model meets image harmonization. In *Proceedings of the 2024 International Conference on Multimedia Retrieval*, pages 1130–1134, 2024.
 - [10] Zixin Zhu, Xuelu Feng, Dongdong Chen, Jianmin Bao, Le Wang, Yinpeng Chen, Lu Yuan, and Gang Hua. Designing a better asymmetric vqgan for stablediffusion, 2023.
 - [11] Junhao Zhuang, Yanhong Zeng, Wenran Liu, Chun Yuan, and Kai Chen. A task is worth one word: Learning with task prompts for high-quality versatile image inpainting. In *European Conference on Computer Vision*, pages 195–211. Springer, 2024.

NeurIPS Paper Checklist

(i) Claims

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

Answer: [\[Yes\]](#)

Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

Guidelines:

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

(ii) Limitations

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

Answer: [\[NA\]](#)

Justification: The paper does not explicitly discuss its limitations. However, potential limitations might exist.

Guidelines:

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

(iii) Theory assumptions and proofs

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

Answer: [\[NA\]](#)

Justification: The paper does not include theoretical results.

Guidelines:

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

(iv) Experimental result reproducibility

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

Answer: [Yes]

Justification: We have provided all the information needed to reproduce the baseline methods and our proposed method in the supplementary.

Guidelines:

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

(v) Open access to data and code

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

Answer: [No]

Justification: The proposed method has been integrated into a commercial product and relies on proprietary code and datasets, so it will not be open-sourced at this time. However, we have provided detailed descriptions of the architecture, training pipeline, and evaluation settings in the main paper and supplementary material to support faithful reproduction of the main experimental results.

Guidelines:

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

(vi) Experimental setting/details

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

Answer: [Yes]

Justification: We have provided all these necessary information in the main paper and the supplementary material.

Guidelines:

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

(vii) Experiment statistical significance

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

Answer: [No]

Justification: We do not report error bars due to the high computational cost of running multiple full training cycles, which is prohibitive in our setting. Furthermore, our method consistently achieves substantial improvements over all baselines across multiple datasets and metrics. The performance gains are large and systematic—often exceeding typical variance ranges reported in prior work—making the observed differences highly unlikely to be attributable to random fluctuations.

Guidelines:

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

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

(viii) Experiments compute resources

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

Answer: [Yes]

Justification: We have provided all these necessary information in the main paper and the supplementary material.

Guidelines:

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

(ix) Code of ethics

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

Answer: [Yes]

Justification: We strictly follow the NeuralIPS Code of Ethics.

Guidelines:

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

(x) Broader impacts

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

Answer: [Yes]

Justification: We have discussed both potential positive societal impacts and negative societal impacts of the work performed.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

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

(xi) Safeguards

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

Answer: [NA]

Justification: Our paper have no such risk of misuse.

Guidelines:

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

(xii) Licenses for existing assets

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

Answer: [Yes]

Justification: The creators or original owners of assets used in the paper have been properly credited. The license and terms of use explicitly mentioned and properly respected.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

(xiii) New assets

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

Answer: [NA]

Justification: The paper does not introduce or release any new assets.

Guidelines:

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

(xiv) Crowdsourcing and research with human subjects

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

Answer: [NA]

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

Guidelines:

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

(xv) Institutional review board (IRB) approvals or equivalent for research with human subjects

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

Answer: [NA]

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

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

(xvi) Declaration of LLM usage

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

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.