CAN THE SAME GENERATOR BE ITS OWN CORRECTOR WITHOUT TEXT FEEDBACK?

Anonymous authors
Paper under double-blind review

ABSTRACT

Text-to-image generators have achieved remarkable visual quality, yet a single generation can still violate the prompt through incorrect counting, attribute binding, or spatial relations. Correcting such errors usually takes more models: a VLM or LLM diagnoses the image and writes an edit instruction, and a separate editor carries it out. An underexplored question is whether the image generator itself can take over this role: rather than treating an imperfect output as terminal, can the same generator interpret it as an intermediate state and move it toward a more prompt-consistent result, without any text feedback? We introduce *SelfFix*, which learns a prompt-conditioned transition from an initial generation to a selectively corrected one inside the generator itself. SelfFix keeps the generator frozen and adds a lightweight adapter of 37.5M parameters; switching the adapter off recovers the original generator bit for bit, so generation is lossless. Given only the original prompt and the current image, the corrector learns to recognize prompt-image inconsistencies, and a Content-Aware Correction (CAC) module decides for every token whether to preserve the input content or revise it. This correction behavior requires no external LLM or VLM analysis, textual error description, or editing instruction at inference. On GenEval, a single correction raises SANA-1.0-1.6B from 0.66 to 0.77, and with a budget of 20 generator calls SelfFix reaches 0.91, surpassing fine-tuned and VLM-guided reflection baselines under the same budget as well as SANA-1.5-4.8B with Best-of-2048. Training only 2.3% of the parameters, SelfFix also outperforms full fine-tuning of the generator. On Emu Edit, it aligns the image with the target caption better than re-generation, on both Emu Edit and PIE-Bench it preserves far more of the source, and on real photographs it revises only the mismatched object, indicating that the correction transfers beyond the training templates.

1 INTRODUCTION

Modern text-to-image models generate increasingly realistic images, yet they still make systematic mistakes in object counting, attribute binding, and spatial relations, even for unambiguous prompts (Peebles & Xie, 2023; Esser et al., 2024; Labs, 2024; Podell et al., 2023). These failures reveal a gap between visual synthesis and prompt-consistent generation: the model may produce many correct elements while missing a single relation, object, or attribute that is crucial to the user's intent. A natural remedy is to correct the image after it has been generated. The central question of this work is whether the generator itself can perform this correction.

A generated image carries substantially more information than a binary success or failure signal. Even when it does not fully satisfy the prompt, it may already establish a suitable composition, appearance, and set of objects. This suggests a useful form of correction: instead of discarding the current visual state, the model can condition on it and generate along a correction direction that retains valid content while resolving the prompt-image mismatch. Such a transition concentrates generation around a promising region of the conditional distribution, turning a near-miss into a satisfactory result.

However, realizing this behavior is not straightforward. A conventional generator is trained to synthesize an image from text and noise, rather than to examine its own output, determine whether it agrees with the prompt, and decide which content should be changed or preserved. Existing reflection systems often obtain this diagnosis from an additional multimodal understanding module

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

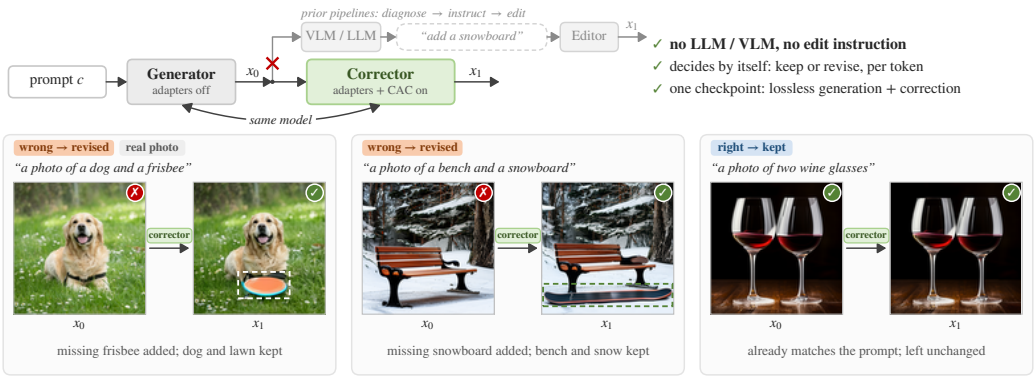

Figure 1: **The same generator as its own corrector.** Prior pipelines diagnose a generated image with a VLM or LLM, write an edit instruction, and pass it to a separate editor. SelfFix uses one model: with its adapters off it is the original generator, and with the adapters and Content-Aware Correction (CAC) switched on, it corrects its own output using only the original prompt. Bottom: a missing frisbee is added to a real photograph and a missing snowboard to a generated image while the rest is kept, and an image that already matches its prompt is left unchanged.

Table 1: **One model instead of three.** Models that must be loaded to generate and correct an image with SANA-1.0-1.6B, and the parameters they add to the generator. Our corrector is the generator itself with a lightweight adapter switched on; with the adapter off it is the original generator bit for bit, so generation is lossless.

Pipeline	Models	Text for the correction	Added parameters
VLM diagnosis + instruction editor (Qwen2.5-VL-7B + FLUX.1 Kontext [dev])	3	VLM-written instruction	+25.16 B
Reflect-DiT (Qwen2.5-VL-3B judge + SigLIP + context module)	3	VLM-written feedback	≈ +4.17 B
Ours (generator = corrector)	**1**	**original prompt only**	**+0.038 B**

that produces textual feedback (Li et al., 2025; Zhuo et al., 2025; Wu et al., 2024; Wang et al., 2024), and act on it with a separate editor or an extended generator. As Tab. 1 shows, such pipelines load three models and add billions of parameters to the generator. We instead ask whether the same image generator can be its own corrector without any text feedback, acquiring the joint prompt-image reasoning needed for correction, so that the generation process becomes both the source of the image and the mechanism that improves it.

This goal is also distinct from general-purpose image editing. An editing model is typically given an explicit instruction such as "make the car blue" or "move the object to the right." The instruction already specifies the intended transformation and is usually written by a user or produced by an upstream VLM or agent after analyzing the image. Self-correction begins with only the original generation prompt and the current image. The model must infer whether the image violates the prompt, determine what should change, and preserve everything that already satisfies the specification. The desired capability is therefore autonomous, conservative correction rather than open-ended editing.

We propose *SelfFix*, which elicits this capability within the generator itself (Fig. 1). SelfFix is trained on aligned negative-positive image pairs so that, given an initial generation and its original prompt, it learns a conditional transition toward a more prompt-consistent image. At inference, the model receives no detector output, VLM-generated error description, or editing instruction. Rather than fine-tuning the whole generator, SelfFix freezes it and adds a lightweight adapter: LoRA (Hu et al., 2022) and a Content-Aware Correction (CAC) module that decides, for every token, whether to keep the content of the current image or to revise it (Fig. 2). Switching the adapter off returns the original generator bit for bit, so the same checkpoint serves both as the generator and as its corrector. The model uses the current image as a structured visual state, identifies the discrepancy through its learned

prompt-image representation, selectively revises the inconsistent regions, and retains the layout and appearance that are already correct.

On GenEval, a single correction by SelfFix raises SANA-1.0-1.6B from 0.66 to 0.77, above SD3 and Playground v3 (Tab. 2). With a budget of 20 generator calls, SelfFix reaches 0.91, outperforming SFT and Reflect-DiT under the same budget as well as SANA-1.5-4.8B with Best-of-2048. SelfFix trains only 37.5M parameters, yet it outperforms full fine-tuning of the generator, which trains 1.60B parameters, gives up the original generator, and redraws content that was already correct; plain LoRA training of this kind diverged in our attempts. On instruction-free editing, SelfFix aligns the image with the target caption better than re-generating it from that caption while preserving much more of the source, and the per-token decisions of CAC concentrate the changes on the content that violates the prompt (Fig. 3).

Overall, this work makes four contributions:

- **Lossless generation.** SelfFix keeps the generator frozen; switching the adapter off recovers the original generator bit for bit, so generation is unaffected and no second model has to be stored.
- **A lightweight adapter.** Only a lightweight adapter (LoRA, a condition AdaLN table, and CAC) is trained: 37.5M parameters, about 2.3% of the base DiT.
- **No LLM or VLM.** The corrector sees only the current image and the original prompt; it needs no external diagnosis, feedback text, or edit instruction at inference.
- **Per-token keep or revise.** The Content-Aware Correction module decides for every token whether to keep the input content or revise it, which makes the correction both local and interpretable.

2 Related Work

Generative models. Research on generative models spans multiple paradigms. GANs (Brock et al., 2018; Goodfellow et al., 2014; Karras et al., 2019) pioneered high-fidelity image synthesis but suffer from unstable training. VAEs (Kingma & Welling, 2013) improve stability but often generate blurry outputs. Autoregressive models (Tian et al., 2024; Sun et al., 2024) capture rich dependencies by sequentially predicting tokens, though their autoregressive nature incurs prohibitive costs for high-resolution images. Diffusion models (Ho et al., 2020; Sohl-Dickstein et al., 2015; Song et al., 2020b) have recently become the leading framework, offering both stability and high quality. A number of extensions (Song et al., 2020a; Nichol et al., 2021; Ye et al., 2023; Zhang et al., 2023) further enhance them by improving controllability and sampling efficiency.

Image modification. Image modification has been studied through training-free feature manipulation and instruction-based editing. Methods such as SDEdit (Meng et al., 2021), Prompt-to-Prompt (Hertz et al., 2022), MasaCtrl (Cao et al., 2023), and Plug-and-Play (Tumanyan et al., 2023) alter a source image by controlling the denoising trajectory or internal features. Instruction-based models (Brooks et al., 2023; Labs et al., 2025; Wu et al., 2025) instead learn to follow an explicit edit request. In both cases, the desired modification is assumed to have already been specified, either by a user or by an upstream VLM or agent that diagnoses the image. SelfFix addresses a different problem: it receives only the original generation prompt and the current image, and must autonomously infer whether and where the image violates that prompt. Its objective is not arbitrary editing, but minimal prompt-consistent correction with preservation of already correct content.

Test-time scaling. Test-time scaling improves generative performance by allocating additional inference computation through expanded sampling, search, or iterative refinement (Ma et al., 2025; Singhal et al., 2025; Xie et al., 2025; Li et al., 2025). Recent refinement-based methods further condition later generation on previous outputs, sometimes using VLM-generated analyses, explicit error descriptions, or user guidance (Li et al., 2025; Zhuo et al., 2025; Wu et al., 2024; Wang et al., 2024; Yu et al., 2023). SelfFix explores a complementary axis: it trains the image generator to use a previous output as a structured intermediate state and to produce a more prompt-consistent successor directly. Because the learned transition is conditioned only on the original prompt and the current image, it needs no text feedback.

3 METHOD

We formulate self-correction as learning a correction transition conditioned on the original prompt and a previously generated image. Training consists of two stages: (1) constructing positive-negative pairs that map prompt-inconsistent images to corrected counterparts (Section 3.1, Fig. 5 in Appendix A); and (2) training a lightweight adapter on the frozen generator to predict the corrected image from the current image and original prompt (Section 3.2, Fig. 2). External detectors, GPT-4o, and an editing model are used only offline to curate supervision. At inference, SelfFix receives no VLM-generated diagnosis or editing instruction and performs correction in one generative forward process. Although the pairs are synthetic, the learned transition transfers to real images and broader prompts.

3.1 GENERATING A TRAINING DATASET

To promote self-correction, it is essential to construct high-quality training data in which incorrect images serve as negatives and are paired with corrected positives (Fig. 5 in Appendix A). We leveraged the 78.5k image–feedback pairs released in Reflect-DiT (Li et al., 2025), built from approximately 3.9k prompts constructed from GenEval templates (with those appearing in the test set filtered out), each associated with 20 generated images and object-detector feedback (Ghosh et al., 2023). For each negative image, GPT-4o (Hurst et al., 2024) generates an editing prompt from the image's original prompt and detector feedback, following an instruction selected for its error type, and FLUX.1-Kontext-dev (Labs et al., 2025) generates the corresponding corrected image; keeping the successful corrections and filtering them with image quality metrics yields 14.8k aligned positive–negative pairs. This deliberately elaborate pipeline is used only offline: our goal is to distill it into a single generation-only model, so that diagnosis and correction no longer require separate modules at test time. Negative-positive pairs teach the model *how* to move an incorrect sample toward a corrected one, but correction also requires learning when no intervention is needed. We therefore augment the correction pairs with pure-positive identity pairs at a 0.3 ratio, in which an already satisfactory image should be preserved unchanged. This asymmetric behavior—targeted revision for errors and preservation otherwise—is what distinguishes self-correction from unconditional re-generation or open-ended image editing.

3.2 SELF-CORRECTION MODEL

We use the constructed pairs to train SelfFix. Following Reflect-DiT (Li et al., 2025), we use SANA-1.0-1.6B (Xie et al., 2024) as the base generator because its relatively small size and fast sampling make it suitable for both generation and correction.

3.2.1 FULLY PRESERVE IMAGE INFORMATION

Flow-based methods (Lipman et al., 2022; Liu et al., 2022; Xie et al., 2024) regard the denoising process as probability density flow, modeling the vector field $u_t(x)$ with a neural network v_θ:

$$\mathcal{L}_{FM} := \mathbb{E}_{t,p_t(x)}\left[\|v_\theta(x,t) - u_t(x)\|_2^2\right] \tag{1}$$

where $p_t(x)$ represents the probability density path, $x \sim p_t(x)$ and $t \sim \mathbb{U}[0,1]$. In text-to-image generation scenarios, the model with parameters θ receives the time step t, text prompt C_T, and noisy image features X as input, and outputs the corresponding velocity $v_\theta(\cdot)$ at that moment.

Subsequently, many methods introduced image conditions C_I into generative models, aiming to provide finer-grained spatial guidance (Zhang et al., 2023; Ye et al., 2023; Brooks et al., 2023; Tan et al., 2024; 2025; Labs et al., 2025) or to enable the model to perform test-time reflection based on previously generated images (Li et al., 2025). Among these methods, a common strategy is to use a separate pre-trained image encoder, such as CLIP (Radford et al., 2021) or SigLIP (Zhai et al., 2023), to extract representations from the image condition C_I. However, these representations may compress fine-grained spatial details and are not naturally aligned with the noisy latent features. Since our approach requires the model to reason over and partially rectify the input image while ensuring consistency across other regions, it is crucial to maximize the preservation of image condition information. We therefore process both the image condition C_I and the noisy image input X with the same VAE encoder, mapping them into a shared latent space (Fig. 2), where they are represented as

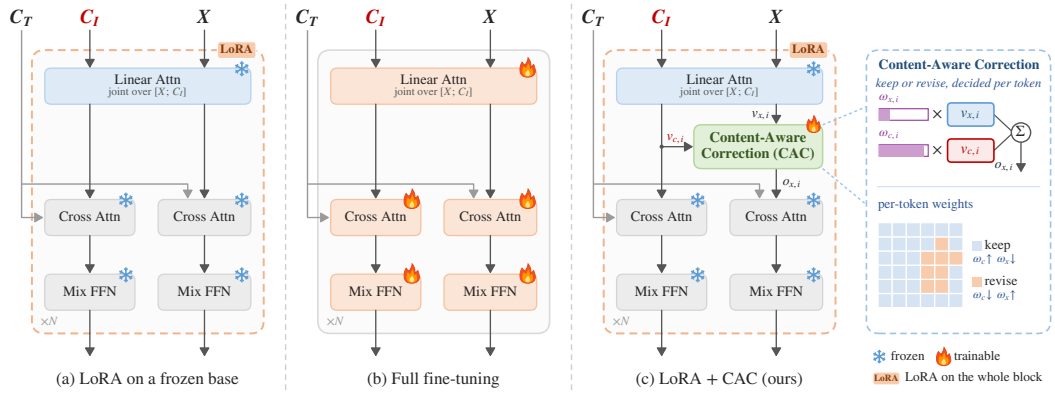

Figure 2: **Three ways to turn SANA into its own corrector.** X: noisy latent of the target image; C_I: latent of the image to be corrected; C_T: text prompt. All three share one backbone, in which the two image streams interact only in the joint linear attention. (a) LoRA on a frozen base. (b) Full fine-tuning. (c) Ours: LoRA with Content-Aware Correction (CAC), which weighs the output of each noisy token ($v_{x,i}$) against that of the condition token at the same position ($v_{c,i}$) and thereby keeps ($\omega_c\uparrow, \omega_x\downarrow$) or revises ($\omega_c\downarrow, \omega_x\uparrow$) every token. In (a) and (c), LoRA stands for two adapters (image tokens and condition tokens) and a condition AdaLN table.

$C_I^h \in \mathbb{R}^{N\times d}$ and $X^h \in \mathbb{R}^{N\times d}$. Similarly, the original text prompt is fed through the text model to extract its feature embedding $C_T^h \in \mathbb{R}^{M\times d}$. Here, d denotes the embedding dimension, while N and M represent the number of image and text tokens, respectively.

3.2.2 Incorporating Image Condition

We adjusted the model architecture to better incorporate image condition, as depicted in Fig. 2(c). The original SANA model is composed of stacked Linear-DiT blocks, each including linear-attention, cross-attention, and a feed-forward network. To align the features of the image condition with the noisy image input, both are passed through the Linear-DiT blocks.

Before entering the linear attention, the two features C_I^h and X^h are concatenated, enabling interaction through joint self-attention. This joint attention is the only place where the two image streams interact: each stream then attends to the text prompt in its own cross-attention and passes through its own Mix-FFN, and no additional image branch is added to the cross-attention. Since prior studies (Islam et al., 2020; Xie et al., 2021), both theoretical and empirical, suggest that applying 3×3 convolutions with zero padding implicitly embeds positional information, SANA removes Positional Embedding and utilizes the positional bias inherent in convolution to convey position information. To maintain this positional inductive bias within the Mix-FFN, we do not incorporate interaction inside the convolutional layer.

Content-Aware Correction. In each Linear-DiT block, the joint attention produces an output $v_{x,i}$ for the i-th token of the noisy image and an output $v_{c,i}$ for the condition token at the same spatial position. A standard residual update adds only the token's own output to its representation:

$$x_i \leftarrow x_i + v_{x,i}, \tag{2}$$

so the content of the current image at that position reaches the token only implicitly, through the attention mixing, and with the same unit coefficient for every token. Such a **fixed** and **uniform** update gives the model no mechanism to decide, token by token, whether the existing content should be kept or regenerated. Content-Aware Correction (CAC) replaces it with a **content-aware** aggregation over the two streams (Fig. 2, right). For each position i, we define the candidate set

$$\mathcal{V}_i = \{\, v_{x,i},\ v_{c,i} \,\}, \tag{3}$$

where $v_{x,i}$ carries the new generation (*revise*) and $v_{c,i}$ carries the content of the current image (*keep*). Each candidate is scored from the normalized feature of its own token with a learnable routing vector:

$$s_{x,i} = q_x^\top \mathrm{LN}(h_{x,i}) + b_x, \qquad s_{c,i} = q_c^\top \mathrm{LN}(h_{c,i}) + b_c, \tag{4}$$

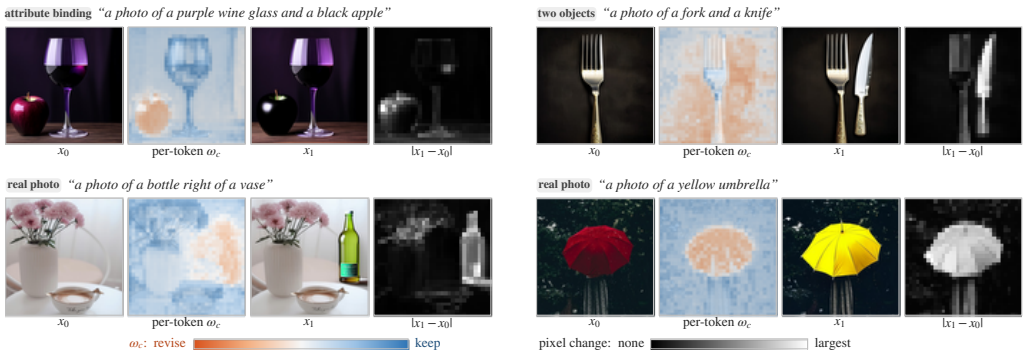

Figure 3: **Content-Aware Correction decides per token.** For each example: the input x_0, the per-token weight ω_c (orange: revise, blue: keep), the corrected image x_1, and the pixel change $|x_1 - x_0|$ (black: none, white: largest). Changes concentrate on the tokens that CAC revises.

where $h_{x,i}$ and $h_{c,i}$ are the token features entering the attention and LN is layer normalization. The scores are turned into aggregation weights,

$$\omega_{c,i} = \sigma(s_{c,i}), \qquad \omega_{x,i} = 1 - \alpha\,\sigma(s_{x,i}), \tag{5}$$

where σ is the sigmoid function and α is a learnable scale; the two weights are computed independently, so a token can take in the input content without giving up its own output. The residual update then becomes

$$o_{x,i} = \omega_{x,i}\, v_{x,i} + \omega_{c,i}\, v_{c,i}, \qquad x_i \leftarrow x_i + o_{x,i}. \tag{6}$$

With $\omega_{x,i} = 1$ and $\omega_{c,i} = 0$, Eq. (6) reduces to the fixed update of Eq. (2), so CAC generalizes the standard residual rather than replacing the block. A high ω_c with a low ω_x copies the content of the current image at that token (*keep*); a low ω_c with a high ω_x lets the token follow the new generation (*revise*). Because the weights are computed from each token's own content, CAC keeps what already agrees with the prompt and concentrates the revision where it is needed. CAC is inserted in every block and adds only 0.09M parameters.

Lightweight adapter and lossless generation. Fig. 2 contrasts this design (c) with LoRA on a frozen base (a), which has no explicit way to decide what to keep, and with full fine-tuning (b), which updates all 1.60B parameters of the DiT. Our model is initialized with the released weights of SANA-1.0-1.6B, which we keep frozen, thereby preserving its native generative capacity. We train two LoRAs (Hu et al., 2022) of rank 16, one for the noisy-image tokens and one for the condition tokens, a timestep-modulation (AdaLN) table for the condition tokens, and CAC: 37.5M parameters in total, about 2.3% of the DiT. As illustrated in ControlNet (Zhang et al., 2023), zero-initialization allows for the gradual injection of image condition information. Following this idea, the LoRAs start from zero and CAC is initialized as the identity ($\omega_x = 1$, $\omega_c \approx 0$). Because the base weights are never updated, switching the adapter off returns the original SANA-1.0-1.6B bit for bit: a single checkpoint serves as the unchanged generator and as its own corrector, and generation is lossless. The parameters of the image and text encoders remain frozen. We adopt the SANA training objective, with a 0.1 probability of dropping the image condition during training.

4 Experiments

4.1 Setup

Implementation details. We build our method upon SANA-1.0-1.6B (Xie et al., 2024), which is a high-efficiency flow-based model for image generation. We train our model with a batch size of 16 and employ the Prodigy optimizer (Mishchenko & Defazio, 2024) with safeguard warmup and bias correction enabled, and a weight decay of 0.01. The model is trained for 60,000 iterations on RTX 4090 GPUs.

Table 2: **GenEval results.** SelfFix corrects a SANA-1.0-1.6B generation using only the image and its original prompt, with the generator itself and 37.5M adapter parameters (0.66 → 0.77). Best-of-20 rows use at most 20 generator calls (Algorithm 1). † models are fine-tuned on our constructed data; Reflect-DiT without † and ‡ SANA-1.5 results are reported from the original papers.

Generator	Params	Overall	Single	Two	Counting	Color	Position	Attribution
SDXL (Podell et al., 2023)	2.6B	0.55	0.98	0.74	0.39	0.85	0.15	0.23
DALLE 3 (Betker et al., 2023)	-	0.67	0.96	0.87	0.47	0.83	0.43	0.45
SD3 (Esser et al., 2024)	8B	0.74	0.99	0.94	0.72	0.89	0.33	0.60
Flux.1-Dev (Labs, 2024)	12B	0.68	0.99	0.85	0.74	0.79	0.21	0.48
Playground v3 (Liu et al., 2024)	-	0.76	0.99	0.95	0.72	0.82	0.50	0.54
SANA-1.5-4.8B Pre (Xie et al., 2025) ‡	4.8B	0.72	0.99	0.85	0.77	0.87	0.34	0.54
+ Best-of-2048 ‡	+ 0B	0.80	0.99	0.88	0.77	0.90	0.47	0.74
SANA-1.0-1.6B (Xie et al., 2024)	1.6B	0.66	0.99	0.77	0.62	0.88	0.21	0.47
+ Ours	+ 0.038B	**0.77**	1.00	0.87	0.67	0.90	0.56	0.62
+ SFT (Best-of-20) †	+ 0B	0.87	1.00	0.98	0.83	0.91	0.81	0.70
+ Reflect-DiT (N=20)	+ 0.1B	0.81	0.98	0.96	0.80	0.88	0.66	0.60
+ Reflect-DiT (N=20) †	+ 0.1B	0.78	0.99	0.91	0.74	0.88	0.66	0.55
+ Ours (Best-of-20)	+ 0.038B	**0.91**	**1.00**	**0.99**	**0.86**	**0.95**	**0.90**	**0.78**

Evaluation protocols and baselines. We evaluate whether a generated image can be improved using only that image and the original prompt; each image is corrected once. On GenEval (Ghosh et al., 2023), SANA-1.0-1.6B produces the initial image and SelfFix corrects it. For a fair comparison, we retrain Reflect-DiT on our constructed dataset (marked by †) and fine-tune the original SANA model on the positive images of that dataset to obtain the SFT baseline. In the Best-of-20 setting, all methods based on SANA-1.0-1.6B use at most 20 generator calls. To test correction beyond the training templates, we further use two image editing benchmarks without their instructions, Emu Edit (Sheynin et al., 2024) (add, color, local, and remove; 200 images) and PIE-Bench (Ju et al., 2024) (700 images); in both, the corrector receives only the source image and the target caption, as it does for real photographs. We compare the three structures in Fig. 2: (a) LoRA on a frozen base, (b) full fine-tuning of the DiT without the image cross-attention branch (1.60B trainable parameters), and (c) SelfFix.

4.2 Main Results

We evaluate whether SelfFix turns a previous generation into a more prompt-consistent one, and analyze the corrections produced from previously generated images.

4.2.1 Per-token Keep or Revise

CAC makes the choice between keeping and revising explicit for every token. Fig. 3 visualizes the per-token weight ω_c next to the pixel change $|x_1 - x_0|$. The image changes on the tokens that CAC revises (low ω_c), while the tokens it keeps stay untouched: SelfFix turns the apple black and leaves the purple wine glass as it is, adds a knife beside the fork, places a bottle to the right of the vase in a real photograph, and recolors the umbrella while keeping its surroundings. CAC therefore not only localizes the correction but also exposes it: its map shows which content the corrector keeps and which it rewrites.

4.2.2 Quantitative Comparison

Tab. 2 summarizes the GenEval results. A single correction by SelfFix raises SANA-1.0-1.6B from 0.66 to 0.77, surpassing much larger generators such as SD3 and SANA-1.5-4.8B, while the generator itself stays unchanged. The gains come mainly from position and attribute binding, which require relating the prompt to the current image rather than improving generic image quality. With a budget of 20 generator calls, SelfFix reaches 0.91, the best result in Tab. 2: it outperforms SFT and Reflect-DiT under the same budget and even SANA-1.5-4.8B with Best-of-2048. Algorithm 1 spends half of the budget on initial images from the original generator (adapter off) and the other half on correcting them (adapter on), so one checkpoint serves both roles; the same selector ranks the candidates of



Figure 4: **Left: the same input corrected by the three structures. Right: real photographs corrected by SelfFix.** (a) LoRA on a frozen base: LoRA training of this kind diverged (NaN) in our attempts. (b) Full fine-tuning without the image cross-attention branch (1.60B trainable parameters). (c) SelfFix (37.5M). Full fine-tuning also redraws the orange object into a new oven, whereas SelfFix keeps it. On the right, each photograph is corrected with a prompt that it does not match; SelfFix revises the mismatched object and keeps the rest of the scene.

every Best-of-N method. Conditioning generation on an existing image thus lets the model keep its successful visual decisions and spend its capacity on the remaining inconsistency.

Generalization and content preservation. Because the correction pairs are constructed from GenEval-style prompts, a potential concern is that SelfFix may overfit to their object-centric templates. Tab. 4 evaluates it on instruction-free editing, where it receives only the source image and the target caption, never the edit instruction. On Emu Edit, SelfFix aligns the image with the target caption better than re-generating it from that caption (CLIP-T 0.28 vs. 0.25) while staying much closer to the source (DINO 0.67 vs. 0.46). On PIE-Bench, it roughly halves the structure distance of re-generation and keeps the unedited background far better (PSNR 15.80 vs. 11.08). SelfFix therefore improves alignment and preservation at the same time rather than trading one for the other. The right half of Fig. 4 shows the same behavior on real photographs with deliberately mismatched prompts: SelfFix revises the object that contradicts the prompt while most of each photograph is kept. These results suggest that SelfFix learns a conservative prompt-conditioned transition rather than memorizing GenEval templates or re-generating the entire image.

Algorithm 1 Fixed-budget correction with SelfFix (Best-of-N)

Input: prompt c; SelfFix checkpoint with the adapter off (G_θ, the original generator) and on (p_ϕ, the corrector); budget N=20; selector s
Output: selected image under the fixed generator-call budget

for $i = 1$ **to** $N/2$ **do**
  sample $x_i \sim G_\theta(\cdot \mid c)$ {adapter off}
  $x'_i \sim p_\phi(x' \mid x_i, c)$ {adapter on; no VLM, no instruction}
end for
candidates $\leftarrow \{x_1, \ldots, x_{N/2}\} \cup \{x'_1, \ldots, x'_{N/2}\}$
return $\arg\max_{x \in \text{candidates}} s(x, c)$

4.2.3 Qualitative Comparison

Fig. 1 and Fig. 4 (right) show corrections produced by SelfFix. SelfFix is explicitly trained to treat a previous image as a structured visual state and refine it according to the original prompt. This makes the correction interpretable: valid composition and appearance remain stable while a missing object, incorrect attribute, or violated relation is revised. SelfFix adds the missing frisbee and snowboard while retaining the dog, the bench, and the snowy scene, and it leaves the two wine glasses unchanged because that image already satisfies its prompt (Fig. 1). In each case the prior image supplies a stable visual context while the prompt defines the direction of improvement. Appendix B further discusses this correction direction and how self-correction differs from general-purpose editing.

4.3 Ablation Studies

Correction structure. Tab. 3 and Fig. 4 compare the three structures of Fig. 2; all variants share the same backbone and training data. LoRA on a frozen base (a) diverged (NaN) in our attempts. Full fine-tuning (b) trains stably, but even with all 1.60B parameters of the DiT it reaches only 0.69, below SelfFix (0.77), and it tends to redraw content that was already correct (Fig. 4, left). SelfFix (c) is lossless and stable and achieves the best result with 2.3% of the trainable parameters of (b).

Table 3: **Structure ablation.** (c): LoRA + CAC at the given LoRA rank; (b) fine-tunes all DiT parameters without an image cross-attention branch. *Lossless*: switching the trained parameters off recovers the base generator bit for bit. *GenEval*: base-model generations corrected by each variant.

Method	Params	Lossless	Stable	GenEval
SANA-1.0-1.6B	–	–	–	0.66
(a) LoRA, frozen base	37.4M	✓	✗	–
(b) Full fine-tuning	1.60B	✗	✓	0.69
(c) rank 8	18.9M	✓	✓	0.72
(c) rank 16 (**ours**)	37.5M	✓	✓	0.77
(c) rank 32	74.7M	✓	✓	0.76

Table 4: **Instruction-free editing.** The corrector receives only the source image and the target caption. Emu Edit: CLIP-T (image–caption) and DINO similarity to the source. PIE-Bench (official metrics): structure distance ($\times 10^3$) and PSNR on the unedited background. The first two rows are references; (c) as in the structure ablation.

	Emu Edit		PIE-Bench	
Method	CLIP-T↑	DINO↑	Struct.↓	BG PSNR↑
Input unchanged	0.24	–	–	–
Re-generation	0.25	0.46	173.78	11.08
(c) **Ours**	0.28	0.67	88.70	15.80

Role of Content-Aware Correction. CAC is what separates (a) from (c): only (c) can weigh, for every token, the output of the new generation against that of the current image. Without it, the correct content has to be reproduced implicitly through attention, which was not stable in our attempts; with it, the changes stay on the tokens that CAC revises (Fig. 3), at a cost of only 0.09M parameters.

Lossless generation. Because SelfFix never updates the generator, switching the adapter off returns SANA-1.0-1.6B bit for bit: one checkpoint serves both as the unchanged generator and as its corrector, including both halves of the Best-of-20 budget, whereas full fine-tuning (b) requires keeping a second 1.60B model.

Adapter capacity. Halving the LoRA rank lowers GenEval to 0.72, while doubling it brings no further gain (0.76), so we use rank 16 throughout. Together with (b), this shows that how the corrector combines the current image with the new generation matters more than how many parameters it trains.

Learning when to correct and when to preserve. Negative-positive pairs supply directional supervision: which parts of a candidate violate the prompt, how they should change, and which content should remain stable. They do not, however, teach the model to abstain; the pure-positive identity pairs (Section 3.1) do, so that an image that already satisfies its prompt is left essentially unchanged, as for the two wine glasses in Fig. 1. Within an image that does need revision, CAC plays the same role at the token level and keeps the regions that already agree with the prompt (Fig. 3). Self-correction therefore requires learning not only *how* to intervene, but also *when not to*.

5 Conclusion

We introduced *SelfFix*, which enables a generation-only image model to be its own corrector using only the current image and the original prompt. SelfFix autonomously identifies prompt–image inconsistencies, corrects relevant content, and preserves unrelated visual information, without relying on VLM-generated diagnoses or explicit editing instructions at inference. It keeps the generator frozen and adds a lightweight adapter with Content-Aware Correction, so that generation remains lossless and every token decides whether to keep or revise its content. With 37.5M trainable parameters, a single correction raises SANA-1.0-1.6B from 0.66 to 0.77 on GenEval, and a budget of 20 generator calls raises it to 0.91, above fine-tuned and VLM-guided reflection baselines; SelfFix also outperforms full fine-tuning of the generator. On instruction-free editing, it aligns images with the target caption better than re-generation while preserving far more of the source. Structure ablations and the per-token analysis further show that, with content-aware image–prompt interaction, the same generator can serve as its own corrector without any text feedback.

AI USE STATEMENT

In this work, we used generative AI tools in two ways. First, as part of the offline data construction described in Section 3.1, GPT-4o writes editing prompts from the detector feedback and FLUX.1-Kontext-dev generates the corrected images of the training pairs; neither model is used at test time. Second, we used generative AI tools to help draft and polish the text of the paper, including grammar, wording, and readability, and to assist with the scripts that plot figures and format tables, which were verified by the authors. We have not used generative AI tools to propose or refine hypotheses, develop the proposed method, or design the experiments, and the remaining required-disclosure tasks (e.g., formulating or proving mathematical claims and cleaning datasets) are not applicable to this work. We have reviewed and revised all AI-assisted text, and all technical content, experimental results, and conclusions were checked by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18392–18402, 2023.

Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 22560–22570, 2023.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36: 52132–52152, 2023.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Md Amirul Islam, Sen Jia, and Neil DB Bruce. How much position information do convolutional neural networks encode? *arXiv preprint arXiv:2001.08248*, 2020.

Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. PnP inversion: Boosting diffusion-based editing with 3 lines of code. In *International Conference on Learning Representations*, 2024.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *arXiv preprint arXiv:1812.04948*, 2019.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

Shufan Li, Konstantinos Kallidromitis, Akash Gokul, Arsh Koneru, Yusuke Kato, Kazuki Kozuka, and Aditya Grover. Reflect-dit: Inference-time scaling for text-to-image diffusion transformers via in-context reflection. *arXiv preprint arXiv:2503.12271*, 2025.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Bingchen Liu, Ehsan Akhgari, Alexander Visheratin, Aleks Kamko, Linmiao Xu, Shivam Shrirao, Chase Lambert, Joao Souza, Suhail Doshi, and Daiqing Li. Playground v3: Improving text-to-image alignment with deep-fusion large language models. *arXiv preprint arXiv:2409.10695*, 2024.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, et al. Inference-time scaling for diffusion models beyond scaling denoising steps. *arXiv preprint arXiv:2501.09732*, 2025.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.

Konstantin Mishchenko and Aaron Defazio. Prodigy: An expeditiously adaptive parameter-free learner. In *Forty-first International Conference on Machine Learning*, 2024. URL `https://openreview.net/forum?id=JJpOssn0uP`.

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu edit: Precise image editing via recognition and generation tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Raghav Singhal, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and Rajesh Ranganath. A general framework for inference-time scaling and steering of diffusion models. *arXiv preprint arXiv:2501.06848*, 2025.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020a.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020b.

Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024.

Zhenxiong Tan, Songhua Liu, Xingyi Yang, Qiaochu Xue, and Xinchao Wang. Ominicontrol: Minimal and universal control for diffusion transformer. *arXiv preprint arXiv:2411.15098*, 2024.

Zhenxiong Tan, Qiaochu Xue, Xingyi Yang, Songhua Liu, and Xinchao Wang. Ominicontrol2: Efficient conditioning for diffusion transformers. *arXiv preprint arXiv:2503.08280*, 2025.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *arXiv preprint arXiv:2404.02905*, 2024.

Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1921–1930, 2023.

Zhenyu Wang, Aoxue Li, Zhenguo Li, and Xihui Liu. Genartist: Multimodal llm as an agent for unified image generation and editing. *Advances in Neural Information Processing Systems*, 37: 128374–128395, 2024.

Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, Ze Liu, Ziyi Xia, Chaofan Li, Haoge Deng, Jiahao Wang, Kun Luo, Bo Zhang, Defu Lian, Xinlong Wang, Zhongyuan Wang, Tiejun Huang, and Zheng Liu. Omnigen2: Exploration to advanced multimodal generation. *arXiv preprint arXiv:2506.18871*, 2025.

Tsung-Han Wu, Long Lian, Joseph E Gonzalez, Boyi Li, and Trevor Darrell. Self-correcting llm-controlled diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6327–6336, 2024.

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021.

Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, et al. Sana: Efficient high-resolution image synthesis with linear diffusion transformers. *arXiv preprint arXiv:2410.10629*, 2024.

Enze Xie, Junsong Chen, Yuyang Zhao, Jincheng Yu, Ligeng Zhu, Yujun Lin, Zhekai Zhang, Muyang Li, Junyu Chen, Han Cai, et al. Sana 1.5: Efficient scaling of training-time and inference-time compute in linear diffusion transformer. *arXiv preprint arXiv:2501.18427*, 2025.

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.

Tao Yu, Runseng Feng, Ruoyu Feng, Jinming Liu, Xin Jin, Wenjun Zeng, and Zhibo Chen. Inpaint anything: Segment anything meets image inpainting. *arXiv preprint arXiv:2304.06790*, 2023.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11975–11986, 2023.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3836–3847, 2023.

Le Zhuo, Liangbing Zhao, Sayak Paul, Yue Liao, Renrui Zhang, Yi Xin, Peng Gao, Mohamed Elhoseiny, and Hongsheng Li. From reflection to perfection: Scaling inference-time optimization for text-to-image diffusion models via reflection tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15329–15339, 2025.

A Training Data Construction

For reproducibility, we provide the data-construction pipeline of Section 3.1 (Fig. 5), its pseudocode (Algorithm 2), and the size of each stage (Table 5).

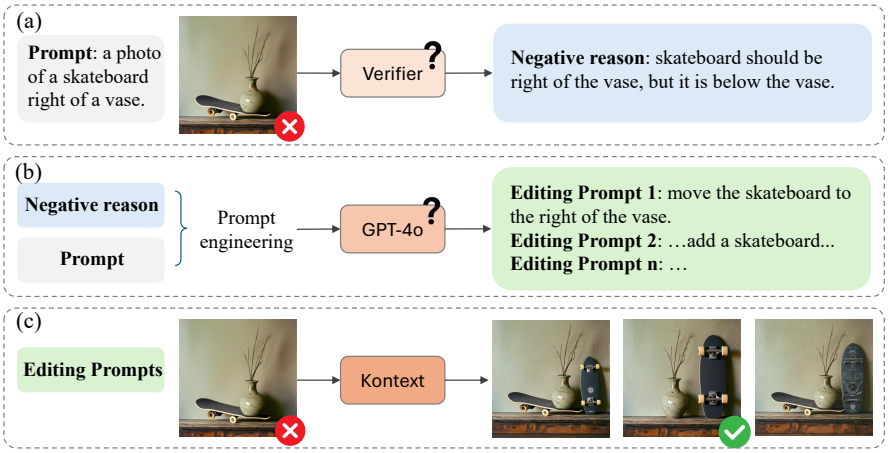

Figure 5: **Constructing positive–negative pairs.** (a) We first collect negative images and corresponding negative reasons from object detectors. (b) Then we analyze negative examples and use GPT-4o to generate editing prompts. (c) FLUX.1-Kontext-dev is then used to generate the corresponding corrected images, which are filtered further.

Algorithm 2 Offline construction of positive–negative correction pairs

Input: Reflect-DiT (Li et al., 2025) prompt–image pool: ∼3.9k GenEval-style (Ghosh et al., 2023) prompts c_j, each with 20 generated images and object-detector feedback
Output: Correction pairs $\mathcal{D}_{\text{corr}}$

$\mathcal{D}_{\text{corr}} \leftarrow \emptyset$
for each prompt c_j and each associated image $x_j^{(k)}$ **do**
  if detector feedback flags $x_j^{(k)}$ as prompt-violating (negative) **then**
    identify the error type from the detector feedback
    select the empirically most effective editing-instruction template for that error type
    query GPT-4o (Hurst et al., 2024) with (c_j, detector feedback, instruction template) to obtain an editing prompt e
    apply FLUX.1-Kontext-dev (Labs et al., 2025) with e to $x_j^{(k)}$, obtaining a candidate corrected image $\hat{x}_j^{(k)}$
    if the correction succeeds and passes the image-quality filter **then**
      $\mathcal{D}_{\text{corr}} \leftarrow \mathcal{D}_{\text{corr}} \cup \{(c_j, x_j^{(k)}, \hat{x}_j^{(k)})\}$
    end if
  end if
end for
return $\mathcal{D}_{\text{corr}}$ {14.8k pairs after filtering; see Table 5}

Algorithm 2 runs only offline; none of its components (the detector, GPT-4o, or FLUX.1-Kontext-dev) is used at test time. Table 5 summarizes the resulting training set: the yield of each stage of Algorithm 2, together with the pure-positive augmentation at the adopted ratio of 0.3.

Table 5: **Training-set composition.** The prompts and image–feedback pairs are taken from Reflect-DiT (Li et al., 2025); the 14.8k correction pairs remain after the filtering step in Algorithm 2, and adding pure-positive identity pairs at ratio 0.3 gives the adopted total of 19,240.

Stage	Count
GenEval-style prompts (Reflect-DiT pool)	3,900
Generated images with detector feedback	78,500
Curated positive–negative correction pairs	14,800
+ pure-positive identity pairs (ratio 0.3)	4,440
Total training pairs (adopted configuration)	**19,240**

B Discussion

B.1 Generating along a Meaningful Correction Direction

The central role of SelfFix is to make generation conditional on what the model has already produced. An initial image is not merely an output to be accepted or rejected; it is a structured intermediate state containing spatial layout, object identity, appearance, and partial prompt alignment. By conditioning on this state, SelfFix learns a transition

$$x' \sim p_\phi(x' \mid x, c), \tag{7}$$

where c is the original prompt and x is the current image. The transition directs the next generation toward a nearby region that better satisfies the prompt, rather than rebuilding every visual decision from scratch.

This perspective explains why correction suits compositional errors such as relative position and attribute binding. For these cases, much of the image may already be correct, and the useful next step is not a generic quality improvement but a semantically specific transformation: move one object relative to another, attach the correct attribute to the correct object, or add a missing entity while preserving the scene. SelfFix learns these transformations from negative-positive pairs and internalizes them in the generator itself. The pure-positive pairs provide the complementary signal that no transition should be applied when the current image already satisfies the prompt. Together, the two forms of supervision turn generation into an adaptive process that can either revise or preserve according to the image-prompt state, and CAC carries out this choice token by token.

B.2 Self-Correction Rather Than General-Purpose Editing

Although SelfFix produces a revised image, its objective differs from general-purpose image editing. Image editors typically follow an explicit transformation specified by a human, a VLM, or an agent, such as changing an object's color, adding an object, or moving it to a desired position. In this setting, the target modification has already been identified outside the model, and the editor mainly needs to execute the provided instruction.

In contrast, SelfFix receives only the current image and its original generation prompt, without an explicit error description or edit command. It must autonomously identify the discrepancy between the image and prompt, determine which content requires revision, and generate a better-aligned result while retaining the relevant visual context. The capability considered here is therefore prompt-conditioned self-correction rather than instruction-following image editing.

Detector feedback, GPT-4o, and FLUX.1-Kontext are used only for offline training-data construction and are not invoked during inference. At test time, correction is performed solely by the generative model based on the image–prompt pair. Because the corrector is the generator itself, the whole pipeline also loads a single model: as Tab. 1 shows, a VLM-diagnosis-plus-editor pipeline and Reflect-DiT each load three models and add billions of parameters to the generator, whereas SelfFix adds 0.038B.

C LIMITATIONS

Constrained by training resources, SelfFix is built on the comparatively small SANA-1.0-1.6B, and its correction behavior is learned entirely from the synthetic pairs constructed offline (Section 3.1); the attainable correction ability is therefore bounded by the capacity of the base model itself. Nevertheless, our results demonstrate that even a generator of this scale can acquire self-correction ability through a lightweight adapter and purely data-driven training, which offers insight into eliciting the latent potential of other generative models.