From Denoising to Refining: A Corrective Framework for Vision-Language Diffusion Model

Yatai Ji[1], **Teng Wang**[2,*], **Yuying Ge**[2,*]
Zhiheng Liu[1], **Sidi Yang**[1], **Ying Shan**[2], **Ping Luo**[1,*]
[1]The University of Hong Kong [2]ARC Lab, Tencent PCG

Abstract

Discrete diffusion models have emerged as a promising direction for vision-language tasks, offering bidirectional context modeling and theoretical parallelization. However, their practical application is severely hindered by a train-inference discrepancy, which leads to catastrophic error cascades: initial token errors during parallel decoding pollute the generation context, triggering a chain reaction of compounding errors and leading to syntactic errors and semantic hallucinations. To address this fundamental challenge, we reframe the generation process from passive denoising to active refining. We introduce **ReDiff**, a **re**fining-enhanced **diff**usion framework that teaches the model to identify and correct its own errors. Our approach features a two-stage training process: first, we instill a foundational revision capability by training the model to revise synthetic errors; second, we implement a novel online self-correction loop where the model is explicitly trained to revise its own flawed drafts by learning from an expert's corrections. This mistake-driven learning endows the model with the crucial ability to revisit and refine its already generated output, effectively breaking the error cascade. Extensive experiments demonstrate that ReDiff significantly improves the coherence and factual accuracy of generated content, enabling stable and efficient parallel generation far superior to traditional denoising methods.

1 Introduction

Discrete diffusion models have recently emerged as a promising alternative to the dominant autoregressive (AR) paradigm for vision-language models (VLMs) [40, 38, 17, 34, 28, 18, 41]. Unlike AR models, which generate text token-by-token in a fixed unidirectional manner, diffusion models conceptualize generation as an iterative denoising process. This approach allows for bidirectional context modeling, granting them greater flexibility in controlling the generation process and a theoretical potential for massive parallelization, promising significant gains in inference efficiency [22, 39, 26, 37].

However, a significant gap exists between the theoretical promise and the practical reality of these models. Existing discrete diffusion models [22, 40, 17] are often plagued by incoherent and hallucinated artifacts (e.g., formatting errors like sequential commas or visually misaligned text) when parallel generation, frequently defaulting to one-token-per-step decoding process. We argue that these shortcomings are symptoms of a deeper, more fundamental problem: the error cascade driven by a training-inference discrepancy. Models are trained exclusively on clean, ground-truth data but are required at inference to generate from their own noisy, intermediate outputs. In a parallel decoding scenario, this discrepancy becomes catastrophic. As illustrated in Figure 1 (a), an error in a few tokens instantly pollutes the context for all other tokens being generated in parallel, initiating a cycle of compounding errors, which produces a detailed yet entirely fabricated description of the input image.

* Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

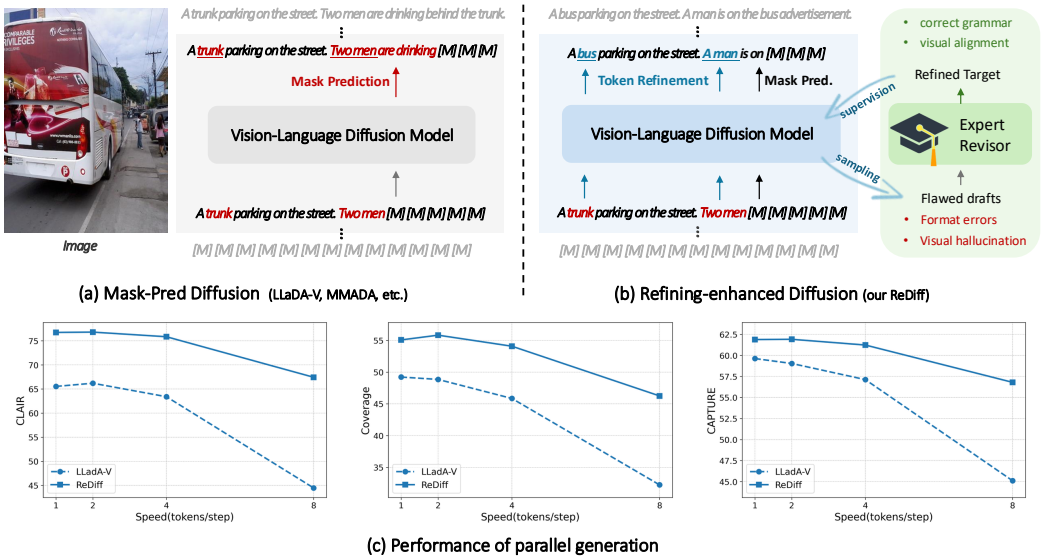


Figure 1: Comparison between standard vision-language diffusion models and our proposed refining-enhanced approach. (a) *Mask-pred diffusion* is trained for passive denoising (mask recovering under fixed context). An initial error, such as misidentifying the “bus" as a “trunk", triggers an error cascade. The model cannot correct this mistake and proceeds to hallucinate further details based on the flawed context (e.g., “Two men are drinking”), leading to a factually incorrect description. (b) *Refining-enhanced diffusion* introduces a paradigm of active refining, teaching the model not only to predict masked tokens but also to perform token refinement. Our ReDiff learns to self-correct through an online loop where its own “flawed drafts" are revised by an expert revisor. As a result, the model can identify and correct its initial mistakes (revising “trunk" to “bus", “Two men" to “A man"), breaking the error cascade and generating a factually grounded response. (c) Performance comparison between LLaDA-V [40] and ReDiff under different inference speeds. “CLAIR” and “Coverage” are detailed caption metrics on CapMAS [15], and “CAPTURE” is on DetailCaps-4870 [11]. Our model delivers superior generation quality and achieves more stable results when using fewer inference steps.

To break this vicious cycle, we propose a paradigm shift: from passive denoising (mask recovering under fixed context) to active refining. We introduce a corrective framework for vision-language diffusion models, called ReDiff, which systematically teaches the model to identify and correct its own errors during denoising. Unlike previous models that merely fill masked tokens, ReDiff actively refines the entire context to guide the generation process. Our approach consists of a two-stage training process. First, we instill a foundational revision capability by training the model to correct synthetic errors, such as random token corruptions and injected hallucinations, moving beyond simple denoising to build a general capacity for revision. Second, we introduce an online self-correction loop where the model is forced to confront and learn from its own mistakes. By capturing its flawed“drafts" during training and learning to predict an expert’s revision, the model directly mitigates the training-inference discrepancy. As shown in Figure 1 (b), our refinement-based model successfully identifies and revises an initial error, leading to a more factually grounded and accurate generation.

This mistake-driven learning endows the model with a crucial, previously absent capability: the ability to revisit and refine its own outputs, including previously unmasked tokens. By learning to self-correct, our model develops robustness to its own imperfections, effectively breaking the error cascade and enabling robust parallel generation. As presented in Figure 1 (c), ReDiff retains stable performance as parallel generation speed increases in detailed image caption task. Our contributions are threefold:

1) We propose a new perspective that reframes the generation process of diffusion models from passive denoising to active, iterative refining to address the core challenge of error cascades.

2) We design and implement a two-stage training framework, featuring a core online self-correction loop that enables the model to learn to fix its own intrinsic errors.

3) Extensive experiments demonstrate that our method significantly improves the coherence and factual accuracy of generated content, exhibiting stability far superior to traditional denoising methods, especially in challenging few-step parallel generation scenarios, thereby greatly enhancing inference efficiency.

2 Related Work

2.1 Large Language Diffusion Models

Discrete diffusion models [2, 21, 12, 1, 27, 25] represent a class of generative models tailored for discrete data like text. In contrast to image diffusion models, which corrupt data by adding Gaussian noise towards a standard Gaussian prior, text diffusion models typically operate by replacing original tokens to degrade semantic content. Early approaches, such as D3PM [2], employed discrete Markov chains where a transition matrix is progressively applied to the input, corrupting it towards a uniform distribution (i.e., any token becomes any other with equal probability) or an absorbing state (e.g., a [MASK] token). More recently, mask-and-pred diffusion models have demonstrated significant empirical success. For instance, LLaDA [22] achieves performance comparable to autoregressive large language models by generating sentences from a fully masked sequence, progressively unmasking tokens with the highest confidence. Similarly, Dream [39] has shown strong results by initializing its parameters from a pre-trained autoregressive model.

Discrete diffusion models offer some advantages over traditional autoregressive models [30, 29, 5, 32, 23]. Their bidirectional context modeling enables flexible and controllable generation, while their inherent parallelism promises significant acceleration in sampling speed. However, this potential for parallel generation remains largely untapped. Current models often suffer from output degradation—such as repetition and grammatical errors—when attempting to predict multiple tokens per step. Our work directly addresses this by enhancing the stability of parallel decoding. This aligns with a recent line of work exploring the correction of generated content. For example, SEED-Diffusion [26] introduced an “edit-based forward process" for code generation, which adds edit-specific noise in the final 20% of steps to allow for revisions. Likewise, FUDOKI [34], a multimodal model based on discrete flow matching, progressively revises a random sentence, where each word is uniformly sampled from the vocabulary, to the correct answer. Our method is distinct in that it treats revision not as another form of noise, but as a high-level refinement process. Specifically, our framework trains the model to learn from and correct its own characteristic errors, moving beyond simple noise reversal.

2.2 Large Vision Language Models

Large vision language models (LVLMs) [20, 10, 16, 3, 13, 36] have achieved remarkable success in vision understanding and have been applied to a myriad of real-world scenarios [14, 42, 8]. The dominant architecture connects a pre-trained vision encoder [24, 31] to an autoregressive language model via a lightweight module like an MLP or Q-Former. These models first realize cross-model alignment with pre-training and then conduct visual instruction tuning to handle a wide range of vision-centric tasks.

Despite their success, a persistent challenge in LVLMs is the phenomenon of hallucination [4], where the model generates text that is factually inconsistent with the visual input. In autoregressive models, this issue is exacerbated by error propagation; an incorrectly generated token can irreversibly misguide the subsequent generation path. Notably, current multimodal discrete diffusion models, such as LLaDA-V [40], LaViDa [17], and MMaDA [38], also adhere to this limitation, fixing tokens in place once they are unmasked. Our ReDiff, however, leverages the bidirectional attention mechanism inherent to the diffusion paradigm. This allows our model to revisit and optimize already-generated content, enabling a progressive refinement process that directly mitigates hallucination.

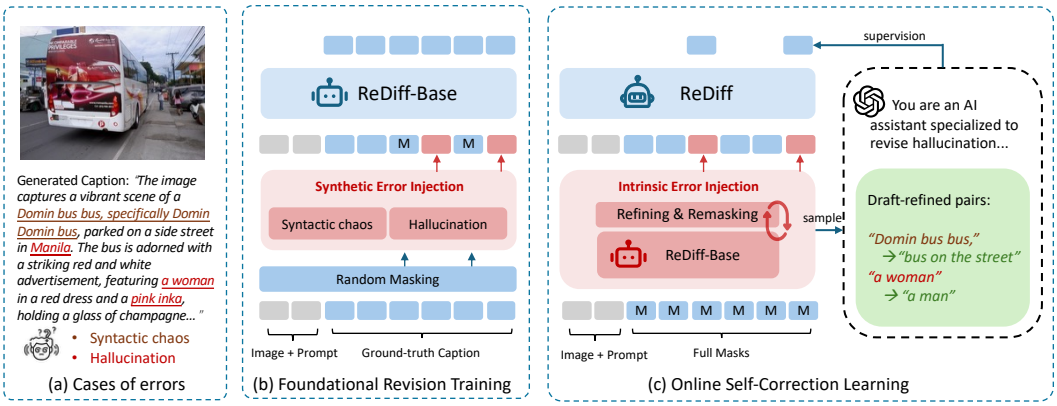


Figure 2: Overview of our proposed two-stage training framework for corrective refining. (a) We illustrate common failure modes in standard vision-language diffusion models, which are prone to generating syntactic errors (e.g., "Domin bus bus") and factual hallucinations (e.g., "a woman"). (b) In the foundational revision training stage, we instill a general corrective capability by training a base model (ReDiff-Base) to revise synthetic errors that are intentionally injected into ground-truth captions. (c) For the second stage, i.e., online self-correction learning, the model generates its own flawed "drafts". These drafts, containing the model's intrinsic errors, are then revised by an expert AI assistant. The resulting "draft-refined pairs" provide strong supervision, teaching our final model (ReDiff) to identify and correct its own characteristic mistakes, thus breaking the error cascade.

3 Methodology

In this section, we introduce our refining-enhanced diffusion framework, ReDiff, designed to enhance the generation accuracy and stability of vision-language diffusion models. In contrast to traditional approaches that focus on recovering text from all [MASK] noise, our work emphasizes the high-level refinement of generated text. Guided by an expert model, our framework enables the model to learn from its own generation errors. This fosters a self-correction capability during inference, allowing it to simultaneously unmask new tokens while refining previously generated ones, thereby mitigating the problem of error cascades in parallel generation.

3.1 Preliminaries of Discrete Diffusion Models

A discrete diffusion model formalizes text generation through a forward and a reverse process. The forward process gradually corrupts a clean text sequence x_0 into a noisy state x_t over a series of timesteps $t \in [0, 1]$. In mask-pred models, this is achieved by replacing tokens with a [MASK] token based on a noise schedule γ_t, culminating in a fully masked sequence as a prior distribution.

The reverse process aims to reverse this corruption. Starting from a fully masked sequence, the model iteratively predicts the original tokens. At each step, it predicts probabilities for all masked positions, unmasks a few high-confidence tokens, re-masks the rest, and feeds the updated sequence back into the model for the next iteration.

The model, a parametric mask predictor, is trained to predict all masked tokens (denoted by a set $\mathbf{M}$) simultaneously. The training objective is a cross-entropy loss computed only on the masked tokens:

$$\mathcal{L}_{\text{CE}}(\theta) = -\mathbb{E}_{t,v,p_0,r_0,r_t}\left[\frac{1}{t}\sum_{i=1}^{L_{r_0}} \mathbf{1}[r_t^i = \mathbf{M}] \log p_\theta(r_0^i | v, p_0, r_t)\right], \tag{1}$$

where v and p_0 denote visual content and prompt, r_0 is the correct response, t is sampled uniformly, and r_t is sampled from the forward process.

A key advantage of discrete diffusion models is their potential for parallel generation, where multiple tokens are unmasked in a single step, significantly reducing the number of required iterations. However, existing model [22, 39, 40] treat already-unmasked tokens as fixed conditions for future predictions. If an incorrect token is generated, it can derail subsequent steps, leading to an error

cascade. Yet, unlike the unidirectional attention in AR models, the bidirectional attention mechanism inherent to these models provides the architectural foundation for updating previously generated tokens, a potential we exploit in our framework.

3.2 Stage I: Foundational Revision Training

Observations of existing vision-language diffusion models, especially in few-step generation scenarios, reveal two predominant error types: syntactic chaos (e.g., incoherence, repetition, grammatical errors) and semantic hallucinations (content that contradicts the visual input), as shown in Figure 2 (a). In this first training stage, we teach the model to correct these two types of errors, extending its capability from simple denoising to foundational text revision.

We use two data construction ways. For syntactic errors, we corrupt the text from ground-truth image-text pairs by randomly replacing a fraction of tokens with other tokens from the vocabulary, creating syntactically chaotic inputs. For hallucination errors, we leverage pairs of correct captions and human-corrputed captions with factual errors (e.g., incorrect objects, attributes, or counts), which directly provide examples of visually inconsistent text.

As shown in Figure 2 (b), we task the model with restoring a "polluted" response r_t to its original, correct version r_0. We first apply the standard masking process to r_0 according to a sampled noise level t. Then, on the remaining unmasked tokens, we inject the synthetic errors described above. This corrupted sequence serves as the model's input. The model is trained to predict the entire original text r_0. The loss is computed not only on the [MASK] tokens but also on the syntactically corrupted tokens ($\mathcal{L}_{\text{syntax}}$) and hallucinated tokens ($\mathcal{L}_{\text{hallucination}}$). We also include a loss on the uncorrupted tokens ($\mathcal{L}_{\text{clean}}$) to encourage the model to preserve correct content. The loss of each type is calculated as follows:

$$\mathcal{L}_{\mathrm{T}}(\theta) = -\mathbb{E}_{t,v,p_0,r_0,r_t}\left[\frac{1}{t}\frac{1}{N_{\mathrm{T}}}\sum_{i=1}^{L_{r_0}}\mathbf{1}[r_t^i \in \mathrm{T}]\log p_\theta(r_0^i|v,p_0,r_t)\right], \tag{2}$$

where $\mathrm{T} \in \{\mathrm{mask}, \mathrm{syntax}, \mathrm{hallucination}, \mathrm{clean}\}$. Each loss component is normalized by the number of its corresponding tokens N_{T} to balance their contributions. The total loss is:

$$\mathcal{L}_{\text{revision}} = \mathcal{L}_{\text{mask}} + \mathcal{L}_{\text{syntax}} + \mathcal{L}_{\text{hallucination}} + \mathcal{L}_{\text{clean}}. \tag{3}$$

This objective remains consistent with discrete diffusion: the standard absorbing-mask process is a special case of a more general forward corruption kernel [2, 25]. In Stage I, we extend the corruption space from only [MASK] to [MASK], token substitution, and hallucination perturbation. Here, q denotes the fixed corruption process that produces r_t from the clean response r_0, while p_θ denotes the learned reverse denoising model. For any fixed valid corruption kernel, minimizing token-wise cross-entropy makes the optimal model distribution $p_\theta^\star$ match the clean-token posterior induced by that corruption process, i.e., $q(r_0^i \mid v, p_0, r_t)$. We provide the formal statement and proof in Appendix A.

After Stage I, we obtain ReDiff-Base, a model equipped with the foundational capability to correct both syntactic errors and factual hallucinations. However, this stage has a limitation: the errors are synthetic and may not reflect the characteristic mistakes the model itself is prone to making.

3.3 Stage II: Online Self-Correction Learning

To teach the model to fix its own idiosyncratic errors, we introduce an online self-correction learning framework. The process, illustrated in Figure 2 (c), proceeds as follows: (1) Generating drafts: We use ReDiff-Base to generate a response for an image, denoted as r_{draft}. We typically use decoding results of different generation steps to cover more mistakes. (2) Expert revision: The image I, the generated draft r_{draft}, and the ground truth are fed to a powerful external "expert model". With a carefully designed prompt, the VLM identifies and corrects both grammatical and hallucinatory errors in r_{draft}, producing a refined version, r_{refined}. We specifically extract the pairs of erroneous and corrected segments. (3) Learning to refine: We form a new training instance $<I, r_{\text{draft}}, r_{\text{refined}}>$ and fine-tune our model on these data. Note that the training loss is computed only on the segments that the expert model identified and corrected. This targeted learning prevents the model from being

Table 1: Performance comparison with state-of-the-art models on three detailed image caption benchmarks. The best scores of vision-language diffusion models are in **bold**.

Model	CapMAS			CapArena	DetailCaps-4870
	CLAIR	Coverage	Factuality	CapArena-Auto	CAPTURE
AR model					
LLaVA-1.5-7B [19]	62.10	34.30	52.80	-94.00	51.08
InternVL-2.5-7B [7]	78.37	52.57	78.69	-29.83	57.80
Qwen2.5-VL-7B [3]	80.48	57.32	82.73	-16.83	60.61
Discrete diffusion model					
MMaDA [38]	35.45	14.33	57.98	-97.00	19.55
FUDOKI [34]	51.94	39.18	46.04	-98.83	57.92
LaViDa [17]	56.22	44.18	53.57	-90.00	57.28
LLaDA-V [40]	65.54	49.22	61.06	-77.17	59.62
ReDiff (ours)	**76.74**	**55.07**	**63.29**	**-51.50**	**61.88**

Table 2: Performance comparison of different inference steps on CapMas benchmark. "Mask-pred training" indicates training with the traditional mask-pred objective following the same settings: identical datasets for Stage I data, as well as the same number of expert-generated captions used in Stage II.

Metrics	CLAIR				Coverage				Factuality			
Speed (token/step)	1	2	4	8	1	2	4	8	1	2	4	8
LLada-V	65.54	66.20	63.40	44.47	49.22	48.85	45.85	32.24	61.06	**61.10**	60.69	64.97
Mask-pred training	74.53	73.57	69.23	46.38	54.15	54.11	47.60	29.69	59.68	58.43	59.66	**67.79**
ReDiff	**76.74**	**76.81**	**75.85**	**67.44**	**55.07**	**55.82**	**54.08**	**46.25**	**63.29**	60.95	**60.87**	65.14

penalized for other potential errors in the draft that the expert may have missed. The training loss is:

$$\mathcal{L}_{\text{refine}}(\theta) = -\mathbb{E}_{t,v,p_0,r_{\text{draft}},r_{\text{refined}}}\left[\frac{1}{N_{\text{mistake}}}\sum_{i=1}^{L_{r_0}}\mathbf{1}[r^i_{\text{draft}} \in \text{mistake}]\log p_\theta(r^i_{\text{refined}}|v,p_0,r_{\text{draft}})\right]. \quad (4)$$

To maintain the foundational capabilities learned in the first stage, we mix in a small amount of the Stage I data during this phase. This entire cycle can be iterated: the refined model from one round can be used to generate new drafts for the next round of expert revision and fine-tuning, progressively enhancing its self-correction ability. The key advantage here is that the model learns from its own mistakes, which is a more targeted and efficient way to improve its robustness and the stability of parallel generation.

3.4 Inference Process

Our inference process differs from that of traditional discrete diffusion models by integrating refinement into each generation step. Specifically, the process starts with a fully masked sequence. At each step, the model computes the output probability distribution over the entire vocabulary for all token positions. For masked positions, if the inference speed is n tokens per step, we select the top-n most confident tokens and unmask them. For previously unmasked positions, we replace the existing tokens with the newly predicted ones. This allows for the simultaneous unmasking of new content and refining of existing content. As more context is generated, previously generated tokens are iteratively updated to be more coherent and factually accurate, effectively reducing the occurrence of syntactic chaos and hallucinations. Note that our method does not bring additional inference latency compared to standard discrete diffusion models, because logits of all tokens are calculated in one forward pass.

4 Experiments

4.1 Experiment Settings

Training Setup. Our primary focus is on enhancing the generative capabilities of vision-language diffusion models. We select detailed image captioning as the representative task, although the

Table 3: Performance comparison of different inference steps on CapArena and CAPTURE metrics.

Metrics	CapArena-Auto				CAPTURE			
Speed (token/step)	1	2	4	8	1	2	4	8
LLada-V	-77.17	-84.00	-90.50	-99.00	59.62	59.04	57.12	45.11
mask training	-56.00	-70.50	-90.33	-98.33	59.98	59.61	56.99	45.12
ReDiff	**-51.50**	**-56.83**	**-72.67**	**-91.67**	**61.88**	**61.91**	**61.23**	**56.80**



The image captures a vibrant scene of a Domin bus bus, specifically Domin Domin bus, parked on a side street in Manila. The bus is adorned with a striking red and white advertisement, featuring a woman in a red dress and a pink inka, holding a glass of champagne. The text "UNCOMPARABLE PRIVIVIVILIVES RWM EXCLUSIVIVES" is prominently displayed the the top of the advertisement. The bus is parked on a busy street, with other cars and buildings visible in the background. A few people can be seen walking on the sidewalk, adding to the lively atmosphere of the scene.

LLaDA-V

The image captures a bustling street scene with a large white double-decker bus prominently in the foreground. The bus features a striking advertisement on the back, reading "Uncompareable Privileges, Nothing Compares," with a picture of a man in a red dress holding a glass of champagne. The website "www.rwmanila.com" and a telephone number are also displayed on the back of the bus. The bus is parked on the side of the street, with several other cars visible in the background. Two people can be seen walking on the sidewalk, adding to the lively atmosphere of the scene.

ReDiff



The image captures a small, cozy bathroom bathed in soft light. The room is characterized by a white door, inviting ajar, leading to a pristine bathroom. Theacent, a the bathroom, complete with a white sink and sink, and a mirror, the the focal side of the room, a the toilet, a the toilet, a the centerpiece of the room. On the wall, On the wall, of the room, a the picture on the wall, a the picture, adding to the overall ambiance of the room. The wooden floor floor reflects the light, creating a sense of calm and tranquility in this intimate space.

LLaDA-V

The image depicts an interior bathroom with a white open door. The bathroom features a white toilet with a closed lid, a white sink, and a white countertop. The white countertop is complemented by a black cabinet beneath it. The walls are painted in a light shade, and the floor is a light wood. A white bath mat is placed on the floor, and a small mirror is mounted on the wall to the right of the toilet. Additionally, there is a white hand towel mounted on the wall to the left of the toilet. The open door spans the left width of the image, from the doorway to the center of the room.

ReDiff

Figure 3: Cases comparison between LLaDA-V and our ReDiff under 4 tokens/step inference speed. ReDiff demonstrates superior fluency and accuracy in its generated captions.



[mask] [mask] [mask] to the the colors of the rocks
a [mask] contrast to the vibrant colors of the rocks.

Step 16

a collection of rocks painted rocks. [mask] [mask]
a collection of colorful painted rocks. These rocks

Step 20

These rocks, painted in shades of red, blue, and blue, are [mask] [mask]
These rocks, painted in shades of red, blue, and gray, are arranged in

Step 24

[mask] the [mask], a tall green plant stands out among the rocks
To the right, a small green plant stands out among the rocks

Step 28



[mask],, to the [mask] [mask] of the scene
[mask], adding to the overall beauty of the scene.

Step 4

The image captures a serene landscape with a cityscape in the foreground
The image captures a serene landscape with a city building in the foreground
The image captures a serene landscape with an apartment building in the foreground

Step 7-8

and is surrounded by a dense of of trees [mask]
and is surrounded by a dense cluster of trees. The

Step 14

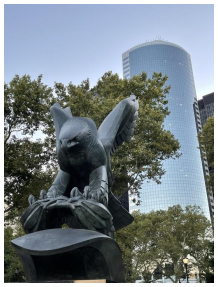
with a warm orange hue near the top, then middle a [mask] yellow
with a warm orange hue near the top, then to a soft yellow

Step 21

Figure 4: Refinement process of ReDiff at different inference step. Red tokens indicate the errors produced during generation, while green tokens mean the corresponding refined results.

methodology is generalizable to other generative tasks (see Appendix C). Our model is built upon the existing LLaDA-V model, leveraging its foundational mask prediction capabilities while endowing it with the ability to refine. The training data comprises caption datasets from LLaVA-1.5 [20], ShareGPT4V [6], and the ViCrit dataset [35]. For Stage I (foundational revision training), we use a total of 260k image-text pairs, with a random token replacement probability of 0.1 for creating syntactic chaos. For Stage II (online self-correction learning), we generate approximately 10k draft-refined caption pairs in each round. The drafts are generated with 128, 32, and 16 inference steps, and o4-mini serves as the expert model for revisions. The system prompt and data quality control are detailed in Appendix D.

Benchmarks and Evaluation Setup. We evaluate our model on three recent benchmarks for detailed image caption: CapMAS [15] uses three metrics evaluated by GPT-4o: CLAIR for overall caption quality, Coverage for the comprehensiveness of the description, and Factuality for the accuracy



The image depicts a striking statue of a bird perched on a pedestal, situated in a city park. The bird, possibly a a eagle, is depicted in a dynamic pose, with its wings spread wide and if about to take flight... The bird's gaze is directed on the viewer, creating a sense of connection and presence.. The statue is surrounded by lush green trees,, adding a natural contrast to the scene. In the background, the building stands tall, a reflective glass glass building, typical of a the U.S. skyline. The backdrop is a clear blue sky, adding to the overall ambiance of the scene.

Inference w/o Refinement

The image depicts a striking statue of an eagle perched on a pedestal, situated in a city park. The bald eagle, rendered in gray, is poised with its wings spread wide, as if about to take flight. The statue is mounted on a square pedestal, which is a dark gray with a subtle gray section at the base. The statue is surrounded by lush green trees, adding a touch of nature to the scene. In the background, a tall, very-rise office building with reflective windows stands prominently, reminiscent of a typical U.S. skyline. The backdrop is a clear blue sky, adding to the overall ambiance of the scene.

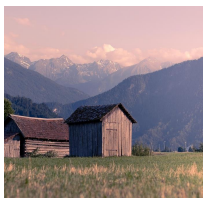
Inference with Refinement

Figure 5: Generation results of ReDiff with or w/o refinement during inference.



Input: a the to computer teach
Output: In the image, a baseball game is in progress with a batter, a catcher, and an umpire at home plate. The batter, dressed in a gray uniform with a red helmet, is poised to swing at the incoming pitch. The catcher, clad in a blue and white uniform with a blue helmet, is crouched behind home plate. The umpire, dressed in a black shirt and gray pants, stands just behind the catcher, observing the play. The stadium is filled with spectators, both seated and standing, dressed in various colors including red, blue, white, and black, adding to the lively atmosphere of the game.

Input: Some people are playing basketball
Output: Three people are playing baseball in a stadium. The batter, dressed in a gray uniform with a red helmet, is poised to swing at a pitch. The catcher, wearing a blue and white uniform with a blue helmet, is crouched behind the batter. The umpire, dressed in a black shirt and gray pants, is standing behind the catcher, observing the play. The stadium is filled with spectators, many of whom are wearing red and blue shirts, watching the game intently. A chain-link fence is visible along the perimeter of the field. Advertisements are visible in the stands, adding to the lively atmosphere of the game.



Input: house the to teach
Output: In the foreground of image, a grassy field stretches out, dotted with tall, dry grass. In the distance, two rustic wooden barns stand, their weathered wood adding a touch of rustic charm to the scene. The barns are characterized by their brown roofs, and one of them features a sturdy wooden door. In the background, a majestic range of mountains stretches across the horizon, their slopes adorned with a mix of snow and evergreen trees, creating a striking contrast against the clear sky. The mountains, with their snow-dusted peaks, glisten under a pale pink sky, adding to the serene ambiance of the scene.

Input: There are some people
Output: There are no people visible in the image. The image captures a serene rural landscape with a large, open grassy field in the foreground. In the field, there are two rustic wooden barns, one larger than the other, with a single wooden door and a dark, moss-covered roof. Small trees and bushes can be seen near the edge of the field, adding a touch of life to the scene. The background features a range of majestic mountains, some capped with snow, under a clear pink sky, enhancing the tranquil atmosphere of the scene. More trees can be seen in the distance, adding to the natural beauty of the scene.

Figure 6: ReDiff can revise wrong input answers.

of the content. CapArena [9] employs a pairwise comparison methodology where the outputs of the test model are compared against those of three baseline models with GPT-4o. A final score is calculated based on these win ratio. DetailCaps-4870 [11] uses the CAPTURE metric, which scores the generated caption by comparing its scene graph to that of the ground-truth description. We compare ReDiff against several vision-language diffusion models, including LLaDA-V, LaViDa, MMaDA, and FUDOKI. We also include results from some typical AR-based VLMs. At inference, the maximum generation length is 128. An inference process of 128 steps corresponds to a speed of 1 token/step, while 32 steps correspond to 4 tokens/step.

4.2 Main Results

As shown in Table 1, our ReDiff achieves state-of-the-art results among all diffusion-based models across each metric. On CapMas, our model's CLAIR score shows a remarkable 11.2 point improvement over the LLaDA-V, reaching a comparable level to InternVL-2.5. The Coverage and Factuality scores also increase by 5.85 and 2.23 points, respectively, indicating that our captions are not only richer in content but also more accurate. On CapArena, our model outperforms LLaDA-V by 25.67 points. Furthermore, we achieve a CAPTURE score of 61.88, surpassing the powerful Qwen2.5-VL. These results demonstrate that our refining-enhanced diffusion method effectively improves fluency and mitigates hallucinations, leading to a substantial enhancement in overall caption quality.

In Tables 2 and 3, we compare models trained with the traditional mask-pred objective versus our refinement framework, using identical datasets and expert supervision. Our model consistently outperforms the mask-trained baseline at every step count. Crucially, as the generation speed increases (i.e., fewer steps), our model's performance degrades much more gracefully, demonstrating superior stability in parallel generation. For instance, on the CLAIR metric, as the speed increases from 1 token/step to 8 tokens/step, the mask-trained model's score plummets from 74.53 to 46.38, whereas our model's score only decreases from 76.74 to 67.44. Notably, our model's performance at 4 tokens/step is higher than that of both LLaDA-V and the mask-trained baseline at 1 token/step. A similar trend is observed for Coverage. The trend for Factuality is less pronounced, as the baseline's score does not drop significantly at fewer steps. This is because the metric relies on extracting valid

Table 4: Effect of each training stage in the refining-enhanced diffusion paradigm.

Metrics	CLAIR		Coverage		Factuality		CapArena-Auto	
Speed (token/step)	1	4	1	4	1	4	1	4
LLada-V	65.54	63.40	49.22	45.85	61.06	60.69	-77.17	-90.50
Base + Stage I	71.31	71.67	51.73	51.83	58.04	55.22	-69.17	-73.17
Base + Stage II	73.02	73.52	53.44	53.00	59.49	57.40	-68.00	-77.67
Stage I + Stage II	**76.74**	**75.85**	**55.07**	**54.08**	**63.29**	**60.87**	**-51.50**	**-72.67**

Table 5: Effect of different settings in the foundational revision training stage. Seed-Diffusion and REMDM denote our implementations of their methods on LLaDA-V.

Metrics	CLAIR		Coverage		Factuality		CapArena-Auto	
Speed (token/step)	1	4	1	4	1	4	1	4
LLada-V	65.54	63.40	49.22	45.85	**61.06**	**60.69**	-77.17	-90.50
Revise hallucination	69.33	67.01	51.08	46.61	59.46	57.06	-74.33	-87.67
Revise syntactic errors	69.48	70.30	**52.12**	49.96	56.57	56.15	-69.67	-88.83
Ours (ReDiff-Base)	**71.31**	**71.67**	51.73	**51.83**	58.04	55.22	**-69.17**	**-73.17**
Seed-Diffusion [26]	69.34	69.24	51.28	49.80	58.45	55.34	-70.33	-76.50
REMDM [33][1]	68.66	67.05	50.82	48.16	58.60	57.24	-73.33	-83.50

items for verification; as the baseline's output becomes more chaotic, fewer items can be extracted, artificially stabilizing the correctness ratio. On both CapArena and CAPTURE, our model also demonstrates more robust parallel generation, with the CAPTURE score dropping by only 0.65 points when accelerating from 1 to 4 tokens/step.

4.3 Qualitative Analysis

We provide qualitative examples to visually demonstrate how the refinement during inference produces more accurate and fluent results, thereby improving the stability of parallel generation.

In Figure 3, we compare the parallel-generated captions from ReDiff and LLaDA-V. The baseline's output suffers from token repetition ("bus", "the"), grammatical errors, and hallucinations (e.g., misidentifying a person on a bus advertisement as "a woman"). In contrast, our model's output is fluent, coherent, and factually grounded. In the second example, our model accurately describes all key elements in the scene, whereas the baseline's output is chaotic and omits significant details.

Figure 4 visualizes the token-level changes during a 32-step generation process. It clearly shows the model simultaneously unmasking new tokens and refining previously generated ones. For instance, in the first example, the model refines the erroneous phrase "rocks painted rocks" into "colorful painted rocks" at step 20. At step 28, it corrects "a tall green plant" to "a small green plant" to better match the visual content. Beyond correcting the model's own errors during generation, it also demonstrates a powerful, generalizable ability to revise disturbing inputs. More visualization of refinement can be found in Appendix B.

Figure 5 showcases comparison of inference with and without the refinement, showing that the refinement is critical for achieving high-quality outputs. If ReDiff inferences without the refinement, errors tend to accumulate, such as repeated words or symbols and incoherent sentences, ultimately degrading the quality of the caption. This highlights the importance of the model's refinement capability.

Beyond correcting the model's own errors during generation, ReDiff also demonstrates a powerful, generalizable ability to revise disturbing inputs. As shown in Figure 6, we provide the model with an image and a user-provided caption containing either syntactic chaos or a factual hallucination. In both cases, our model successfully corrects the initial erroneous text and proceeds to generate a coherent and accurate completion, highlighting the strong revision ability of ReDiff.

[1]REMDM requires twice the inference steps of other models at equivalent speed settings, as its remasking operation introduces additional steps to complete the full sequence generation.

Table 6: Effect of online self-correction learning rounds.

Metrics	CLAIR		Coverage		Factuality		CapArena-Auto	
Speed (token/step)	1	4	1	4	1	4	1	4
ReDiff-Base	71.31	71.67	51.73	51.83	58.04	55.22	-69.17	-73.17
Online training round 1	**76.74**	**75.85**	55.07	**54.08**	**63.29**	**60.87**	**-51.50**	**-72.67**
Online training round 2	76.10	74.99	**55.20**	**54.08**	62.24	60.46	-56.17	-72.83

4.4 Ablation Studies

Impact of Each Training Stage. Table 4 shows that both stages individually improve performance and stability over LLaDA-V, with the largest gains from their combination. Notably, Stage II alone provides a more substantial boost than Stage I, confirming that teaching the model to learn from its own intrinsic errors is a highly effective refinement strategy. After Stage I training, the model exhibits stable parallel generation performance. For example, as the speed increases from 1 to 4 tokens/step, CLAIR improves from 71.31 to 71.67, and CapArena changes from -69.17 to -73.17. The combination of the two stages yields a synergistic effect, with Stage I providing a foundational revision ability that is further amplified by Stage II, leading to large improvements in metrics like Factuality (+5.25) and CapArena (+17.67).

Analysis of Foundational Revision Training. As shown in Table 5, we investigate different settings for the stage I training. We find that revising syntactic errors primarily boosts overall quality (CLAIR) and Coverage, while also enhancing stability during parallel generation. Conversely, training on hallucination revision exhibits higher Factuality. Combining both error types allows our model (ReDiff-Base) to achieve the best overall performance. We further compare with two related methods in language diffusion models that can edit generated content: Seed-Diffusion introduces different numbers of editing operations to inject noise during training and learns to recover them, while REMDM remasks low-confidence tokens during inference and predicts them again. The results show that our Stage I training, which jointly learns semantic and syntactic correction, achieves the best overall performance.

Impact of Online Self-Correction Training Rounds. In Table 6, we examine the effect of iteration of the stage II training. The results show that while the first round of self-correction provides a substantial performance boost over the ReDiff-Base model, subsequent rounds of training on newly generated data do not yield further significant improvements across most metrics. We assume that it is primarily due to data saturation relative to the model's error space. Once the model learns to correct the dominant error patterns in the first refinement round, subsequent rounds for revising similar error types will have less effect. More ablation studies about expert model and generalization to other generative tasks can be found in Appendix C.

5 Conclusion

In this work, we addressed the critical challenge of error cascades that hampers the performance of vision-language diffusion models, particularly in efficient parallel generation scenarios. We proposed a paradigm shift from passive denoising to active refining by introducing ReDiff, a novel framework centered on a mistake-driven, online self-correction loop. This approach teaches the model to learn from its own characteristic errors, endowing it with the ability to revisit and refine its generated output. Our extensive experiments validate that this method not only achieves state-of-the-art performance but, more importantly, demonstrates far superior stability and factual accuracy in challenging few-step generation regimes. By effectively breaking the error cascade, our work presents a promising path toward more robust, efficient, and controllable generation.

6 Acknowledgements

This paper was partially supported by the General Research Fund of Hong Kong ID 17208825 and 17209324.

References

[1] Marianne Arriola, Aaron Gokaslan, Justin T. Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. In *ICLR*. OpenReview.net, 2025.

[2] Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *NeurIPS*, pages 17981–17993, 2021.

[3] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *CoRR*, abs/2308.12966, 2023.

[4] Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He, Zongbo Han, Zheng Zhang, and Mike Zheng Shou. Hallucination of multimodal large language models: A survey. *arXiv preprint arXiv:2404.18930*, 2024.

[5] Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

[6] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions. *CoRR*, abs/2311.12793, 2023.

[7] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.

[8] An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. Spatialrgpt: Grounded spatial reasoning in vision-language models. In *NeurIPS*, 2024.

[9] Kanzhi Cheng, Wenpo Song, Jiaxin Fan, Zheng Ma, Qiushi Sun, Fangzhi Xu, Chenyang Yan, Nuo Chen, Jianbing Zhang, and Jiajun Chen. Caparena: Benchmarking and analyzing detailed image captioning in the llm era. *arXiv preprint arXiv:2503.12329*, 2025.

[10] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven C. H. Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. In *NeurIPS*, 2023.

[11] Hongyuan Dong, Jiawen Li, Bohong Wu, Jiacong Wang, Yuan Zhang, and Haoyuan Guo. Benchmarking and improving detail image caption. *arXiv preprint arXiv:2405.19092*, 2024.

[12] Zemin Huang, Zhiyang Chen, Zijun Wang, Tiancheng Li, and Guo-Jun Qi. Reinforcing the diffusion chain of lateral thought with diffusion language models. *CoRR*, abs/2505.10446, 2025.

[13] Yatai Ji, Rongcheng Tu, Jie Jiang, Weijie Kong, Chengfei Cai, Wenzhe Zhao, Hongfa Wang, Yujiu Yang, and Wei Liu. Seeing what you miss: Vision-language pre-training with semantic completion learning. In *CVPR*, pages 6789–6798. IEEE, 2023.

[14] Yatai Ji, Shilong Zhang, Jie Wu, Peize Sun, Weifeng Chen, Xuefeng Xiao, Sidi Yang, Yujiu Yang, and Ping Luo. IDA-VLM: towards movie understanding via id-aware large vision-language model. In *ICLR*. OpenReview.net, 2025.

[15] Saehyung Lee, Seunghyun Yoon, Trung Bui, Jing Shi, and Sungroh Yoon. Toward robust hyper-detailed image captioning: A multiagent approach and dual evaluation metrics for factuality and coverage. *CoRR*, abs/2412.15484, 2024.

[16] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *CoRR*, abs/2408.03326, 2024.

[17] Shufan Li, Konstantinos Kallidromitis, Hritik Bansal, Akash Gokul, Yusuke Kato, Kazuki Kozuka, Jason Kuen, Zhe Lin, Kai-Wei Chang, and Aditya Grover. Lavida: A large diffusion language model for multimodal understanding. *CoRR*, abs/2505.16839, 2025.

[18] Zijie Li, Henry Li, Yichun Shi, Amir Barati Farimani, Yuval Kluger, Linjie Yang, and Peng Wang. Dual diffusion for unified image generation and understanding. In *CVPR*, pages 2779–2790. Computer Vision Foundation / IEEE, 2025.

[19] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024.

[20] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023.

[21] Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *ICML*. OpenReview.net, 2024.

[22] Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025.

[23] OpenAI. Chatgpt. *https://openai.com/blog/chatgpt/*, 2023.

[24] 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*, pages 8748–8763. PmLR, 2021.

[25] Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *arXiv preprint arXiv:2406.07524*, 2024.

[26] Yuxuan Song, Zheng Zhang, Cheng Luo, Pengyang Gao, Fan Xia, Hao Luo, Zheng Li, Yuehang Yang, Hongli Yu, Xingwei Qu, Yuwei Fu, Jing Su, Ge Zhang, Wenhao Huang, Mingxuan Wang, Lin Yan, Xiaoying Jia, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Yonghui Wu, and Hao Zhou. Seed diffusion: A large-scale diffusion language model with high-speed inference. *CoRR*, abs/2508.02193, 2025.

[27] Haoran Sun, Lijun Yu, Bo Dai, Dale Schuurmans, and Hanjun Dai. Score-based continuous-time discrete diffusion models. In *The Eleventh International Conference on Learning Representations*, 2023.

[28] Alexander Swerdlow, Mihir Prabhudesai, Siddharth Gandhi, Deepak Pathak, and Katerina Fragkiadaki. Unified multimodal discrete diffusion. *CoRR*, abs/2503.20853, 2025.

[29] Qwen Team. Qwen3: Think deeper, act faster. 2025. https://qwenlm.github.io/blog/qwen3/.

[30] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023.

[31] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

[32] vicuna. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *https://vicuna.lmsys.org/*, 2023.

[33] Guanghan Wang, Yair Schiff, Subham Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling. *arXiv preprint arXiv:2503.00307*, 2025.

[34] Jin Wang, Yao Lai, Aoxue Li, Shifeng Zhang, Jiacheng Sun, Ning Kang, Chengyue Wu, Zhenguo Li, and Ping Luo. FUDOKI: discrete flow-based unified understanding and generation via kinetic-optimal velocities. *CoRR*, abs/2505.20147, 2025.

[35] Xiyao Wang, Zhengyuan Yang, Chao Feng, Yongyuan Liang, Yuhang Zhou, Xiaoyu Liu, Ziyi Zang, Ming Li, Chung-Ching Lin, Kevin Lin, Linjie Li, Furong Huang, and Lijuan Wang. Vicrit: A verifiable reinforcement learning proxy task for visual perception in vlms. *CoRR*, abs/2506.10128, 2025.

[36] Yi Wang, Xinhao Li, Ziang Yan, Yinan He, Jiashuo Yu, Xiangyu Zeng, Chenting Wang, Changlian Ma, Haian Huang, Jianfei Gao, et al. Internvideo2. 5: Empowering video mllms with long and rich context modeling. *arXiv preprint arXiv:2501.12386*, 2025.

[37] Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. Fast-dllm: Training-free acceleration of diffusion LLM by enabling KV cache and parallel decoding. *CoRR*, abs/2505.22618, 2025.

[38] Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. Mmada: Multimodal large diffusion language models. *CoRR*, abs/2505.15809, 2025.

[39] Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025.

[40] Zebin You, Shen Nie, Xiaolu Zhang, Jun Hu, Jun Zhou, Zhiwu Lu, Ji-Rong Wen, and Chongxuan Li. Llada-v: Large language diffusion models with visual instruction tuning. *CoRR*, abs/2505.16933, 2025.

[41] Runpeng Yu, Xinyin Ma, and Xinchao Wang. Dimple: Discrete diffusion multimodal large language model with parallel decoding. *CoRR*, abs/2505.16990, 2025.

[42] Shilong Zhang, Peize Sun, Shoufa Chen, Min Xiao, Wenqi Shao, Wenwei Zhang, Kai Chen, and Ping Luo. Gpt4roi: Instruction tuning large language model on region-of-interest. *CoRR*, abs/2307.03601, 2023.

A Theoretical Justification

We provide a formal justification for applying revision losses to non-masked tokens in Stage I. The standard absorbing-mask process is a special case of discrete diffusion with a particular forward corruption kernel. Our revision training instead uses a generalized corruption kernel that includes both absorbing masking and non-absorbing token perturbations.

Let $c_i \in \{\text{mask}, \text{syntax}, \text{hallucination}, \text{clean}\}$ denote the corruption type at token position i. For a clean response r_0, define a conditional forward corruption kernel $q_\phi(r_t, c \mid r_0, v, p_0)$ with token-wise marginals:

$$q_\phi(r_t^i, c_i \mid r_0, v, p_0) = \begin{cases} \gamma_t, & c_i = \text{mask},\ r_t^i = \mathbf{M}, \\ \eta_t \pi_{\text{syn}}(r_t^i \mid r_0^i), & c_i = \text{syntax}, \\ \xi_t \pi_{\text{hall}}(r_t^i \mid r_0, v, p_0, i), & c_i = \text{hallucination}, \\ 1 - \gamma_t - \eta_t - \xi_t, & c_i = \text{clean},\ r_t^i = r_0^i, \end{cases} \tag{5}$$

where $\gamma_t, \eta_t, \xi_t \geq 0$ and $\gamma_t + \eta_t + \xi_t \leq 1$. Here, π_{syn} is the random token-replacement distribution used for synthetic syntax corruption, and π_{hall} is the empirical corruption distribution induced by hallucination pairs. Setting $\eta_t = \xi_t = 0$ recovers the conventional mask-pred process, so our Stage I training enlarges the forward corruption family rather than adding losses outside the diffusion formulation.

Proposition A.1 (Optimal denoiser under generalized corruption)**.** *For any fixed valid corruption kernel q_ϕ in Equation 5, the token-wise cross-entropy objective in Equation 2 is minimized, under unlimited model capacity, by the Bayes denoiser*

$$p_\theta^\star(a \mid v, p_0, r_t, i) = q_\phi(r_0^i = a \mid v, p_0, r_t, i). \tag{6}$$

Therefore, $\mathcal{L}_{\text{syntax}}$ and $\mathcal{L}_{\text{hallucination}}$ are standard denoising terms for non-absorbing corruptions, while $\mathcal{L}_{\text{mask}}$ is the usual absorbing-mask denoising term.

Proof. Condition on a context $z = (v, p_0, r_t, i)$. The expected token-level cross-entropy loss is

$$\mathbb{E}_{r_0^i \sim q_\phi(\cdot \mid z)} \left[-\log p_\theta(r_0^i \mid z) \right] = -\sum_a q_\phi(r_0^i = a \mid z) \log p_\theta(a \mid z). \tag{7}$$

This is the cross entropy between the true posterior $q_\phi(r_0^i \mid z)$ and the model distribution $p_\theta(\cdot \mid z)$. It is minimized when the model distribution equals the true posterior. Summing this expected loss over timesteps and corruption types with positive weights preserves the same optimum for each observed context. □

This result gives the convergence target of Stage I: with sufficient data, optimization, and model capacity, empirical risk minimization converges to the posterior denoiser of the generalized forward process. A reverse transition can then be parameterized through this denoiser by marginalizing over predicted clean tokens in $q_\phi(r_s \mid r_t, r_0)$. If $p_\theta(r_0 \mid r_t)$ matches the true posterior, the induced reverse kernel matches the posterior reverse process, as in standard masked diffusion. The only difference is that the corruption state space also includes replace-and-revise states.

Stage II has a different role. Since drafts are sampled from ReDiff-Base, their distribution depends on the model's own inference trajectory; thus we do not claim that Stage II is itself a fixed forward diffusion ELBO. Instead, it optimizes an on-policy refinement risk:

$$\min_\theta \ \mathbb{E}_{r_{\text{draft}} \sim \text{Sampler}(\theta_{\text{base}})} \left[-\sum_{i \in \mathcal{E}(r_{\text{draft}})} \log p_\theta(r_{\text{refined}}^i \mid v, p_0, r_{\text{draft}}) \right], \tag{8}$$

where $\mathcal{E}(r_{\text{draft}})$ denotes expert-identified erroneous positions. This objective complements the diffusion-consistent Stage I denoiser by reducing the distribution shift between training states and inference states, which is the source of error cascades in few-step parallel decoding.

B More Visualization

Figure 7 and Figure 8 show the generation results of ReDiff and LLaDA-V under different inference steps. In the 2 tokens/step scenario, LLaDA-V outputs a great deal of hallucinated content, such as

“Goku" and “Vegeta" in the first case, and a mouse and keyboard in the second. This occurs because an initial hallucination can affect subsequent generation, leading to error cascades. In contrast, our ReDiff method produces captions that are consistent with the image content. In the cases of 8 tokens/step, the results of our model are more fluent and have less grammer errors.



The image captures a moment scene from the popular animated series \"Dragon Ball Super\". The central figure is Goku, a key character in the series, who is depicted in a dynamic pose. His arms are raised, suggesting a moment of action or excitement. To his right, another character, Vegeta, is seen sitting a bit relaxed, his expression conveying a sense of anticipation or contemplation. The backdrop of the image is a solid orange, providing a stark contrast to the vibrant colors of the characters. The The poses and expressions of the characters add to the overall atmosphere of the scene, making it instantly recognizable to fans of the series.

LLaDA-V

The image depicts a television screen displaying a scene featuring two toy figures. The figures are positioned on an orange couch, with a red reflection on the wall behind them. The figure on the left is a gray, dinosaur-like creature with a red eye, sharp teeth, and a claw on each hand. The figure on the right is a smaller, lighter gray creature with a black eye, sharp teeth, and no claws on each hand. The figure on the right has a more playful and less intimidating appearance compared to the one on the left. A dark gray ceiling is visible in the background, adding to the overall ambiance of the scene.

ReDiff



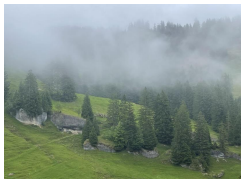
The image captures a moment on a desk, where a computer monitor, a mouse, and a keyboard are are arranged. The monitor, with its screen on, is positioned on the left side of the desk. The mouse, with its cord visible, is placed to the right of the monitor. The keyboard, with its keys black, is situated to the right of the mouse. The desk itself is a light shade of gray. The arrangement of these objects suggests an organized workspace. However, it's not clear if there is any text text in the image. The description is based on the relative positions of the objects on the desk.

LLaDA-V

The image features a close-up of a computer monitor with a prominent Google logo in the center, surrounded by colors red, yellow, green, and blue. A dinosaur sticker, resembling a T-Raurus, is positioned to the right of the center. Two blue circles with the numbers "4" and "2" are located at the bottom center of the monitor. Two small dinosaurs, both with a green skin, are positioned on the left side of the monitor, adding to the prehistoric theme of the image. Additionally, a white sticker is visible on the upper left of the monitor, adding to the overall composition of the image.

ReDiff

Figure 7: Cases comparison between LLaDA-V and our ReDiff under 2 tokens/step inference speed.

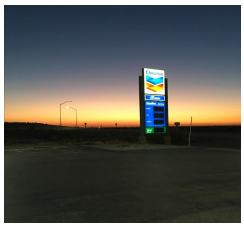


The image captures a serene scene of a mountainous landscape. Domin in the foreground, a rocky green hillside stretches out, the, leading to the the top of the.. The the hillside,, is densely with lush green trees, the, providing a natural boundary to the.. The The the envelop, the, the, the, top of the,,,, creating a to the atmosphere. The The sky is overcast,,, casting a soft light over the,,, enhancing the calmness of the.. There are no no animals or or people visible in the,, emphasizing the untouched beauty of the nature.

LLaDA-V

The image captures a serene landscape scene with a lush, green grassy field stretching in the foreground. In the field, there are several large rocks, some of which are embedded in the grass. The field and rocks, along with the trees, and fog, create a picturesque composition. In the background, slightly the green, reveals the the tops of the trees. The trees are shrouded in a light mist or fog that envelops the scene, enhancing the tranquil and serene atmosphere of the landscape. The colors in the image include shades of green, brown, white, and gray, adding to the overall beauty of the scene.

ReDiff



The image captures a serene scene of a Chevron station gas pump.. The gas,, which is a the pump,, is set against the backdrop of the sky,, which is painted in hues of blue and orange, the sky, and the the colors of the the sky, and the the colors of the the sky,, the the colors of the the sky,, the the colors of the the sky, and the the, and the the colors of the the sky,, to the the tranquility of the scene. , a and a can the visible in the background, adding to the overall composition of the image.

LLaDA-V

The image captures a serene scene of a Chevron gas station at dusk. The vibrant sky is transitioning from a deep orange to a deep blue, creating a warm backdrop for the scene. The Chevron sign, prominently prominently displayed on the right, is a striking contrast against the darkness of the sky. A few streetlights are visible in the distance, adding to the ambiance of the setting. A faint silhouette of distant hills can be seen in the background, enhancing the overall tranquility of the scene. The scene is a harmonious mix of urban, industrial, and natural elements, all illuminated by the fading warm glow of the sky.

ReDiff

Figure 8: Cases comparison between LLaDA-V and our ReDiff under 8 tokens/step inference speed.

C More Ablations

Impact of Expert Model. In the online self-correction learning stage, our framework is not strictly tied to a specific expert (e.g., o4-mini). In our design, any model can serve as an expert revisor provided it meets two essential criteria: (1) it can maintain token-number consistency between the draft and the refined version (to isolate specific error segments), and (2) it possesses sufficient knowledge to accurately identify factual and syntactic errors. To ablate how results vary with different expert models, we conduct an additional study using GPT-4o as an alternative expert revisor to generate the training data.

As shown in Table 7, the model trained with data refined by GPT-4o also outperforms the baseline (ReDiff-Base) across various inference speeds. However, the performance gain is slightly lower compared to the model trained with o4-mini. We can conclude that: (1) The effectiveness of the ReDiff framework holds regardless of the specific expert model, as long as the expert provides valid corrections. (2)The performance of the student model scales with the capability of the teacher. A stronger expert (o4-mini) provides higher-quality supervision, enabling the model to learn more nuanced correction patterns.

Extension of ReDiff to Block Diffusion Architectures. We validate the effectiveness of ReDiff on semi-autoregressive generation, such as Block Diffusion, which utilize partial masking to achieve parallel decoding within blocks while maintaining feature propagation across the entire sequence. We

Table 7: Effect of different expert models.

Metrics	CLAIR		Coverage		Factuality		CapArena-Auto	
Speed (token/step)	1	4	1	4	1	4	1	4
ReDiff-Base	71.31	71.67	51.73	51.83	58.04	55.22	-69.17	-73.17
ReDiff (o4-mini)	**76.74**	**75.85**	55.07	**54.08**	**63.29**	**60.87**	**-51.50**	**-72.67**
DeDiff (GPT-4o)	74.70	74.42	53.16	51.94	59.27	57.83	-56.33	-79.33

Table 8: Extension of ReDiff to block diffusion generation.

	Block num	CLAIR	Coverage	Factuality	CapArena-Auto
ReDiff	1	**76.74**	55.07	63.29	-51.50
ReDiff	4	76.54	**56.55**	**63.63**	**-47.17**

experiment with a block size of B=32 for generating captions of length 128 (meaning 4 blocks). The results in Table 8 show not only sustained performance but also a tangible improvement in metrics like Coverage and CapArena-Auto, and they all outperform LLaDA-V. This confirms that ReDiff's mechanism is highly effective in hybrid generation modes that allow feature updates across tokens.

Generalization to Vision-Language Dialog Task. To demonstrate the generalization of ReDiff on other generation tasks, we conduct additional evaluations on ConvBench, a challenging multi-turn dialogue benchmark. As shown in Table 9, although ReDiff is trained specifically on caption refinement data, the learned self-correction capability effectively generalizes to multi-turn dialogue task. ReDiff achieves superior scores compared to the baseline LLaDA-V across different inference speeds. This indicates that the "active refining" mechanism is task-agnostic and robust.

D Details of Stage II Data Construction

In online self-correction learning loop, we use ReDiff-Base to generate caption drafts. Then the image, the draft, and ground truth caption are fed to o4-mini to detect and revise errors. The prompt for o4-mini is shown in Table 10.

We utilize a two-step filtering for data quality control. Firstly, we perform tokenization on all generated pairs and discard any pair where the expert's corrected segment did not match the original segment's token count. Secondly, we use a separate LLM process to filter out pairs where the correction share the same semantics with original segment or introduce inconsistency with the rest of the caption. This rigorous filtering process ensures that the 10k final pairs provide high-quality, token-consistent corrective supervision.

E Implementation Details

Both training stages are conducted on 8 NVIDIA H200 GPUs. Stage I (foundational revision training) is trained for 1 epoch with a learning rate of 1×10^{-5} and a cosine decay schedule. Stage II (online self-correction learning) is trained for 2 epochs with a learning rate of 5×10^{-6} and a cosine decay schedule.

Table 9: Generalization to vision-language dialog task.

ConvBench	Overall Score		
Speed (token/step)	1	2	4
LLaDA-V	0.69	0.45	0.18
ReDiff	**1.04**	**0.83**	**0.35**

F Limitations

While ReDiff demonstrates consistent gains over baseline discrete diffusion models, two limitations are worth noting.

Applicability to single-token generation tasks. The core mechanism of ReDiff—simultaneously unmasking new tokens and refining previously generated ones—only takes effect when multiple tokens are generated in parallel across several decoding steps. For conventional multiple-choice benchmarks (e.g., MMBench, MMStar) that evaluate a model by selecting a single answer token, every generated position is already the final output with no opportunity to revise earlier tokens. Consequently, our framework provides no improvement in these strict single-token generation scenarios, and its benefits are most pronounced in open-ended generation tasks with longer output sequences and parallel decoding.

Dependence on an expert model for Stage II data construction. The online self-correction learning stage (Stage II) requires an external expert model (e.g., o4-mini) to identify and correct errors in the drafts produced by ReDiff-Base. This introduces a reliance on a capable, and potentially proprietary, AI assistant to generate training data. Although our ablation study (Appendix C) shows that the framework is robust to the choice of expert, the quality of the learned self-correction capability is bounded by the quality of the expert's revisions. This dependency may pose a practical constraint in settings where strong external models are unavailable or prohibitively costly.

G Ethics Statement

All datasets and models used in this study are publicly available and open-source. No proprietary, private, or personally identifiable information was collected or used. The images employed are either natural scenes or normal human activities, without any violent, explicit, or otherwise harmful content. Therefore, the research meets relevant considerations regarding privacy, ethics and copyright.

Table 10: The system prompt for AI expert to revise model's drafts.

ROLE: Hallucination detect and revise Assistant

PERSONA:
You are an AI assistant specialized in hallucination revision. You integrate information from image, question and ground truth answer, to analyze and judge whether the prediction from other models is right or not. If the prediction is wrong, you need to revise the hallucination and errors in the prediction.

INPUT CONTEXT:
You will receive the following:
1. **Image:** An image.
2. **Question:** A user's question about the image.
3. **Answer:** The right answer to the question.
4. **Prediction** The answer from our model.

TASK:
Your primary task is to judge if the prediction is right according to the image and ground truth answer. If the prediction is not right, detect hallucination and wrong parts, then revise them.
* The prediction must be consistent to the image, detect all hallucination and errors.
* For the words containing hallucination, you need to replace them with right words, which have same token number with original prediction. Make the original prediction correct with as few modifications as possible.
* The answer may contain some chaos in grammar expression, such as repetition, incoherence, etc. You should also replace erroneous parts with fragments of identical token counts.

GUIDELINES & CONSTRAINTS:
1. The prediction doesn't have to be identical to the reference answer; as long as it correctly answers the question, it's acceptable. The GT (ground truth) serves merely as a reference. Focus primarily on checking whether there are hallucination issues in the prediction that contradict the image content.
2. If the prediction is correct, output only 'right'.
3. If the prediction contains hallucinations or errors, output a JSON-formatted string containing multiple pairs of phrases. Each pair should consist of the original erroneous phrase segment and its corrected counterpart.
4. Modifications should be localized to the minimal necessary extent, typically targeting short multi-word segments.
5. For each pair, ensure the tokenized length of the original and modified segments remains identical. The semantics of replacement words must be inconsistent to the original segment.
6. The original segment should be unique within the prediction to facilitate error localization by users.

OUTPUT FORMAT:
1. If the prediction is right, output only 'right'.
2. If the prediction has errors, provide the output as a single JSON object, which is a list containing multiple dictionaries with the following keys:
* 'org': (String) The hallucination or error segment in original prediction.
* 'target': (String) The right segment to replace the wrong part in prediction.

EXAMPLE:
* **Image:** [Description: A man in front of a white trunk.]
* **Question:** "What might the man in the suit be doing?"
* **Answer:** "The man dressed in business attire leaning on the white truck could be associated with the business related to the truck ..."
* **Prediction:**"The man is leaning on a pink trunk, and ..."
* **Expected Output:**
"'json ["org": "a pink truck", "target": "a white trunk"]

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: the abstract and introduction accurately reflect the paper's contributions of ReDiff

   Guidelines:

   - The answer [N/A] 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 [N/A] 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.

2. **Limitations**

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

   Answer: [Yes]

   Justification: In the appendix.

   Guidelines:

   - The answer [N/A] 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.

3. **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: [Yes]

Justification: In the appendix.

Guidelines:

- The answer [N/A] 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.

4. **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: Detailed description of experiment settings and training data.

   Guidelines:

   - The answer [N/A] 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
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) 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).
     (d) 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.

5. **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: We will release on Github.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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.

6. **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) necessary to understand the results?

   Answer: [Yes]

   Justification: We have detailed training setup.

   Guidelines:

   - The answer [N/A] 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.

7. **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: not include statistical significance, but we use multiple benchmarks for verification.

   Guidelines:

   - The answer [N/A] 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.

8. **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: In the appendix.

   Guidelines:

   - The answer [N/A] 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).

9. **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: conform with the NeurIPS Code of Ethics

   Guidelines:

   - The answer [N/A] 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).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: In the appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] 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).

11. **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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: no such risks.

    Guidelines:

    - The answer [N/A] 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.

12. **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: [N/A]

    Justification: the paper does not use existing assets.

    Guidelines:

    - The answer [N/A] 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.

13. **New assets**

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

    Answer: [N/A]

    Justification: the paper does not release new assets.

    Guidelines:

    - The answer [N/A] 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.

14. **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: [N/A]

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

    Guidelines:

    - The answer [N/A] 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.

15. **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: [N/A]

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

    Guidelines:

    - The answer [N/A] 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.

16. **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: [Yes]

Justification: The second stage uses VLM as an expert.

Guidelines:

- The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.