LaDiC: Are Diffusion Models Really Inferior to Autoregressive Counterparts for Image-to-Text Generation?

Yuchi Wang[1*], **Shuhuai Ren**[1*], **Rundong Gao**[1], **Linli Yao**[1], **Qingyan Guo**[2], **Kaikai An**[1], **Jianhong Bai**[3], **Xu Sun**[1†]
[1] National Key Laboratory for Multimedia Information Processing, Peking University
[2]Tsinghua University [3]Zhejiang University
{wangyuchi,shuhuai_ren}@stu.pku.edu.cn xusun@pku.edu.cn

Abstract

Diffusion models have exhibited remarkable capabilities in text-to-image generation. However, their performance in image-to-text generation, specifically image captioning, has lagged behind Auto-Regressive (AR) models, casting doubt on their applicability for such tasks. In this work, we revisit diffusion models, highlighting their capacity for holistic context modeling and parallel decoding. With these benefits, diffusion models can alleviate the inherent limitations of AR methods, including their slow inference speed, error propagation, and unidirectional constraints. Furthermore, we identify the prior underperformance of diffusion models stemming from the absence of an effective latent space for image-text alignment, and the discrepancy between continuous diffusion processes and discrete textual data. In response, we introduce a novel architecture, LaDiC, which utilizes a split BERT to create a dedicated latent space for captions and integrates a regularization module to manage varying text lengths. Our framework also includes a diffuser for semantic image-to-text conversion and a Back&Refine technique to enhance token interactivity during inference. LaDiC achieves state-of-the-art performance for diffusion-based methods on the MS COCO dataset with 38.2 BLEU@4 and 126.2 CIDEr, demonstrating exceptional performance without pre-training or ancillary modules. This indicates strong competitiveness with AR models, revealing the previously untapped potential of diffusion models in image-to-text generation.[1]

1 Introduction

In recent years, there has been a surge of impressive applications of diffusion models in text-to-image generation tasks (OpenAI, 2023; Podell et al., 2023; Dai et al., 2023). However, the inverse process of image-to-text generation remains less explored. Some pioneering efforts (Li et al., 2022b; Yuan et al., 2022) have attempted to integrate diffusion models into text generation tasks. They mainly followed the traditional Encoder-Decoder framework, utilizing the diffusion model as a text decoder. Subsequent research (He et al., 2023b; Liu et al., 2023a) introduces visual capability into this paradigm by treating visual inputs as special tokens or encoded hidden states, thereby extending the research scope to the realm of multi-modal tasks, such as image-to-text generation. However, their performance has consistently lagged behind that of Auto-Regressive (AR) models (Li et al., 2022a; Zhang et al., 2021; Wang et al., 2022). Only through intricate architecture (Luo et al., 2022) or external data (Zhu et al., 2022) can they barely achieve comparable results,

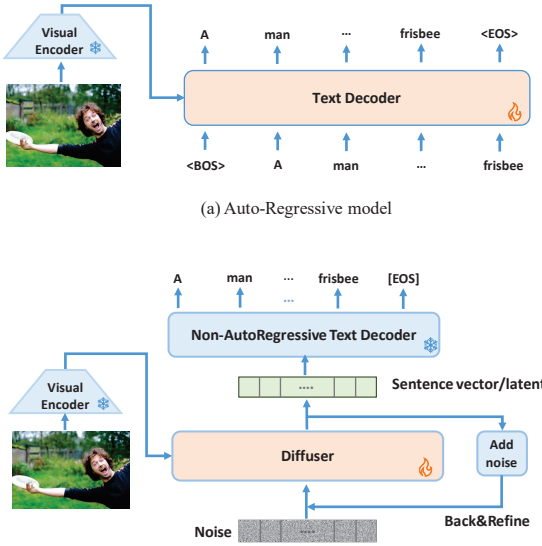
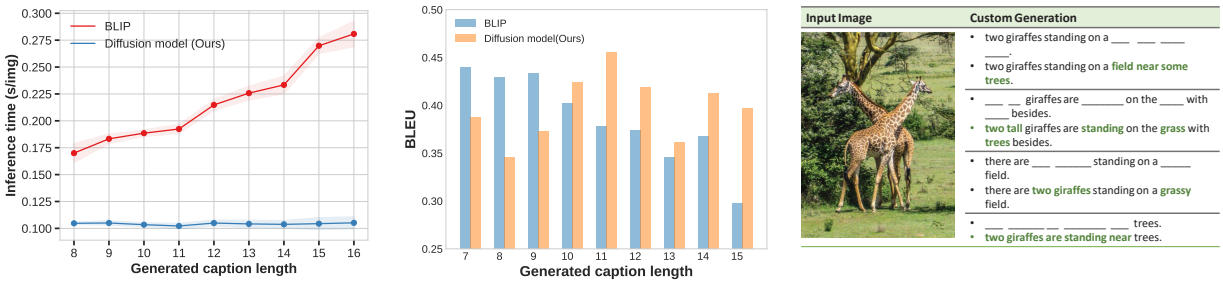


Figure 1: Inference process for image captioning. (a) Token-by-token generation manner of AR-based model. (b) Gradually denoising generation manner of the diffusion-based model (Ours).

[*] Equal contribution.
[†] Corresponding author.
[1]Code released at https://github.com/wangyuchi369/LaDiC

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 6699–6715
June 16-21, 2024



(a) (b) (c)

Figure 2: (a) Inference time of AR model (BLIP) and our diffusion model (LaDiC) as generated caption length increases. (b) BLEU score of BLIP and LaDiC with increasing generated caption length. (c) LaDiC's ability of custom generation.

raising doubts about whether diffusion models have inherent limitations, potentially making them less suitable for the image-to-text task.

In this study, we aim to dispel this doubt by deeply reexamining the diffusion-based image-to-text paradigm and unveiling its distinct benefits. Unlike conventional AR approaches that sequentially generate captions token by token (Fig. 1a), diffusion-based models take Gaussian noise as input and iteratively denoise it under image guidance to simultaneously produce the entire caption (Fig. 1b). This Non-AutoRegressive (NAR) diffusion-based model exhibits three key advantages: **(1) Parallel Decoding:** Diffusion-based models emit all tokens in parallel, significantly reducing inference time for lengthy target captions. As illustrated in Fig. 2a, the inference time of AR models like BLIP (Li et al., 2022a) proliferates as text length increases, while our model can emit all tokens concurrently, ensuring stable inference time regardless of the length increase. For instance, when the caption length reaches 16, our model is approximately 3× faster than BLIP. **(2) Holistic Context Consideration:** Unlike the uni-directional information flow of AR models (left to right), diffusion-based models can consider more holistic contexts, mitigating error accumulation (He et al., 2023b). As depicted in Fig. 2b, the BLEU metric of BLIP-generated captions declines rapidly with increasing text length, whereas our diffusion-based model maintains performance. **(3) Flexible Generation:** AR models adhere to a fixed unidirectional generation manner, whereas our model demonstrates much greater flexibility. We can custom generate captions based on tokens in nearly any position, as shown in Fig. 2c, a capability challenging for AR image captioning models.

Despite the above benefits, the underperformance of diffusion models on image-to-text tasks hinders their popularity. Upon examining prior diffusion-based models, we deduce that their unsatisfactory performance primarily stems from two factors: **(I)** Two significant gaps in translating between images and text, i.e., 1) the gap between visual information and textual representation, and 2) the gap between high-level text semantics and specific words. Simultaneously addressing both gaps within the previous paradigm as shown in Fig. 3, proves to be a challenging task for diffusion models. **(II)** Substantial discrepancies exist between text and other modalities with continuous inputs (e.g., images). For instance, classical continuous diffusion models naturally align with the pixel space but struggle to transition directly to the discrete text space. Additionally, generated images have a fixed size, while caption lengths vary, presenting another challenge for diffusion models in determining the boundaries of generated captions. Given these considerations, we meticulously design a novel architecture LaDiC, a **La**tent **Di**ffusion-based **C**aptioner, for further amplifying the capability of diffusion models in image-to-text generation. As depicted in Fig. 1b and Fig. 3, rather than directly generating discrete text from image representation, we treat the diffuser as an interface translating image information to high-level text representation (sentence latent). This approach alleviates the diffusion model's burden, enabling it to leverage its powerful generation capabilities in high-level semantic spaces (Ramesh et al., 2022), while the Non-AutoRegressive (NAR) text decoder retains its ability to generate discrete tokens from latent space. During training, a text encoder is employed to generate ground-truth text latent codes, and during inference, it can be discarded.

Technically, we leverage BERT (Devlin et al.,

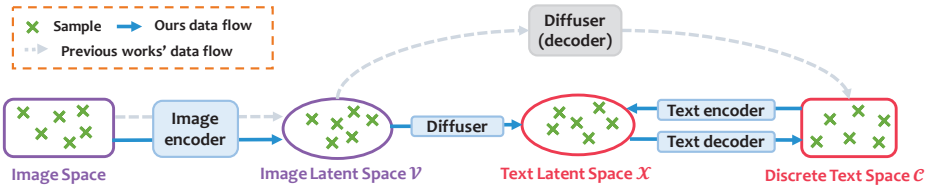


Figure 3: Comparison of the pipeline between our LaDiC and that of previous diffusion-based models. We introduce text latent space to alleviate the burden on the diffuser.

2019) to generate the text latent space and propose a post-processing submodule including normalization and reassignment procedures for addressing problems like variable length of text. Furthermore, the diffuser serves as a bridge between image and text, aiming to fit the distribution of the text latent space conditioned on the image, wherein we utilize a cross-attention mechanism for better modality fusion. Lastly, during inference, we propose the Back&Refine technique to provide more interaction between tokens. At a high level, we intuitively summarize our model in Fig. 1(b). The generation process commences with Gaussian noise, and at each step, we subtract a certain amount of noise conditioned on the image. As the number of steps increases, the denoised text latent, or sentence feature of BERT in our model, converges toward the ground-truth sentence feature. Ultimately, the well-denoised sentence feature is passed through the NAR frozen text decoder to concurrently generate discrete words.

We conducted experiments mainly on the COCO dataset (Lin et al., 2014) to validate our model's capabilities. Remarkably, without pretraining or external modules, our model achieves 38.2 BLEU@4 and 126.2 CIDEr, significantly surpassing both diffusion-based methods and traditional NAR models. In addition to the unique advantages discussed earlier, our model also matches the performance of well-established pretrained AR models and outperforms BLIP in image paragraph captioning. These results underscore the potent generative ability and immense potential of diffusion models in image-to-text generation. We aspire that our work offers a fresh perspective, fostering future research on diffusion models for image-to-text generation or even other text-centered multimodal generation tasks.

2 Related Works

2.1 Diffusion Models and their Applications

Diffusion models have recently emerged as powerful generative models, with representative foundational architectures such as DDPM (Ho et al., 2020b) and DDIM (Song et al., 2020). These methods gradually transform samples into Gaussian noise and train a model to recover them, presenting a simple and stable learning objective for addressing issues like posterior and mode collapse that challenge prior models like VAE (Kingma and Welling, 2013) and GAN (Goodfellow et al., 2014).

The impressive generative capabilities of diffusion models have led to their application across a spectrum of fields, including image (Ramesh et al., 2022; Dai et al., 2023), audio (Liu et al., 2023b), video (Blattmann et al., 2023; Bai et al., 2024), 3D (Poole et al., 2022), and human avatar (He et al., 2023a; Hu et al., 2023), among others. Yet, their application to text is still in its initial state. How to adapt discrete tokens into a diffusion model is an ongoing challenge. Existing approaches for tackling this problem generally fall into two categories: **(1) Discrete Text Diffusion Models** (Austin et al., 2021; Reid et al., 2022; He et al., 2022), which mimic the diffusion process on the discrete space by directly corrupting text with [MASK] tokens. **(2) Continuous Text Diffusion Models** (Li et al., 2022b; Gong et al., 2022; Dieleman et al., 2022; Yuan et al., 2022; Lin et al., 2022), which use continuous embeddings to represent each token and then perform the classical diffusion process. While these approaches demonstrate the feasibility of applying diffusion models to text generation and show comparability with AR methods, they are limited to unimodal representations and may overlook high-level overall semantics to some extent. Furthermore, we notice the work (Lovelace et al., 2023), which explores the concept of a text latent space. Yet its diffusion model, designed for predicting BART's (Lewis et al., 2019) hidden states, still relies on an AR generation mechanism, which suffers from its issues like low inference efficiency.

2.2 Image-to-text Generation

Image-to-text generation, especially the image captioning task, aims to describe the content of an

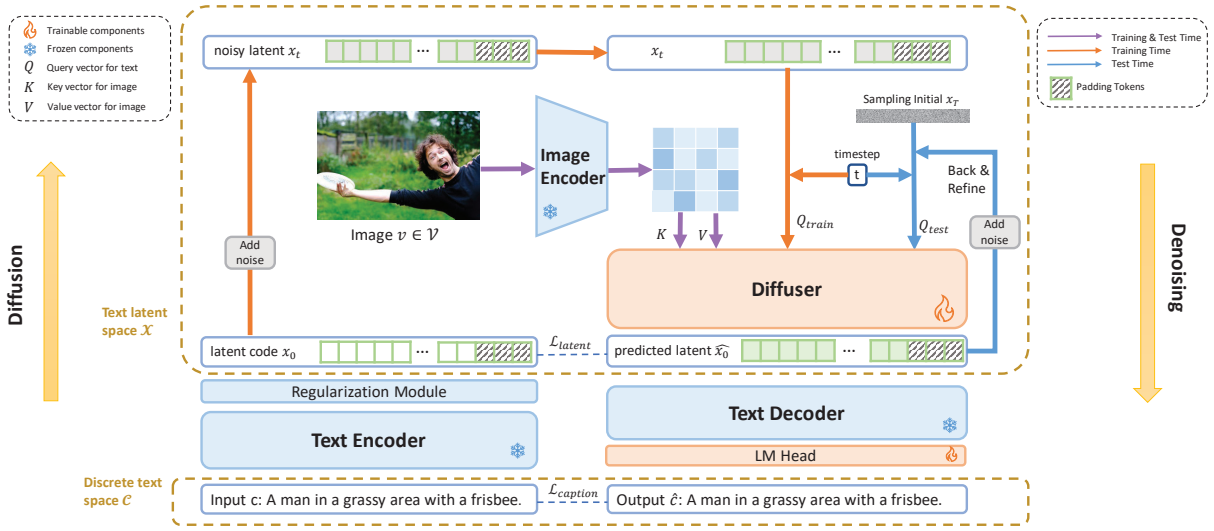


Figure 4: An overview of our LaDiC model. It mainly consists of Text Encoder, Diffuser, and Text Decoder. On the left is the diffusion process, and on the right is the denoising process. Initially, the sentence c is converted into a text latent space $\mathcal{X}$ by the text encoder. Subsequently, diffusion is performed on the text latent space, wherein a diffuser is trained to denoise noisy text latent representations x_t. Finally, the predicted text latent representation $\hat{x}_0$ without noise is passed through a NAR text decoder to generate the final sentence $\hat{c}$.

image in natural language. Other task variants include dense captioning, which illustrates each object in the picture (Johnson et al., 2016), and paragraph captioning, which generates a detailed, lengthy paragraph (Krause et al., 2016) and so on. Early AR approaches for captioning (Karpathy and Fei-Fei, 2017; Vinyals et al., 2014) employed an encoder-decoder architecture with a CNN (Convolutional Neural Network) to encode images and an RNN (Recurrent Neural Network) to generate captions. With the advent of Transformer (Vaswani et al., 2017) and large-scale pretraining methods, pretrained vision-language models (Li et al., 2022a; Zhang et al., 2021; Li et al., 2020; Ren et al., 2023, 2021; Zhao et al., 2023; Liu et al., 2023c) emerged and achieved high performance.

In contrast to the unidirectional generation of AR models, NAR models generate entire captions in parallel. MNIC (Gao et al., 2019) introduced the mask token strategy, and NAIC (Guo et al., 2020) employed reinforcement learning in NAR caption generation. A special class of NAR methods, diffusion-based models has recently emerged. Most diffusion-based models (Xu, 2022; He et al., 2023b; Liu et al., 2023a) follow the paradigm utilized in continuous diffusion models mentioned above. Additionally, Bit Diffusion (Chen et al., 2022a) encodes captions into binary bits, and DDCap (Zhu et al., 2022) applies a discrete diffusion model to captioning. SCD-Net (Luo et al., 2022) is the state-of-the-art diffusion-based model with a semantic-conditional diffusion process. However, its cascaded architecture is relatively complex and requires an external retrieval module, limiting its further extension. Our work reexamines the diffusion-based paradigm and proposes a novel, compact architecture with improved performance.

3 Methodology

In this section, we introduce our diffusion-based image captioning model, LaDiC. In § 3.1, we present the overall architecture of LaDiC, including its training and inference pipeline. Subsequently, from § 3.2 to § 3.4, we offer a detailed illustration.

3.1 Overview

As illustrated in Fig. 3, we utilize a text encoder to transform the discrete text space $\mathcal{C}$ into a continuous text latent space $\mathcal{X}$. Subsequently, a diffuser is trained to establish a connection between the image representation space $\mathcal{V}$ and the text space $\mathcal{X}$, and finally, a text decoder maps the text latent codes back to the discrete text space $\mathcal{C}$. Specifically, in Fig. 4, given an image $v \in \mathcal{V}$ and its corresponding caption $c \in \mathcal{C}$, we encode the caption c into the latent space, yielding its latent code $x_0 \in \mathcal{X}$. Then, we employ the diffusion model's diffusion-denoising procedure. Initially, various levels of noise (introduced by different timesteps t) are added to x_0 to generate a noisy version x_t (left panel). Subse-

quently, the diffuser acts as a denoiser, recovering x_0 conditioned on the images v (right panel). Once the diffuser is trained, a function $f : x_t \xrightarrow{v} x_0$ is established, connecting the image space $\mathcal{V}$ and the latent space $\mathcal{X}$. During inference, given an image v^*, x_t ($t \to \infty$) is replaced with pure Gaussian noise $x_\infty \sim N(\mathbf{0}, \mathbf{I})$ and iteratively denoised by f, resulting in $x_\infty \xrightarrow{v^*} \hat{x}_0^*$, where $\hat{x}_0^*$ represents the predicted text latent code. Finally, the decoder converts the acquired latent code back into discrete text $\hat{c}^* \in \mathcal{C}$.

3.2 Latent Space Tailored for Text

As discussed in § 1, the text latent space $\mathcal{X}$ serves as a bridge between image space $\mathcal{V}$ and discrete text space $\mathcal{C}$, significantly alleviating the burden on diffusion models. Therefore, careful design of the text latent space is essential, aiming to incorporate rich text semantic information and facilitate the integration of image information by the diffuser. Basically, we leverage BERT (Devlin et al., 2019) to construct such a high-level semantic latent space and meanwhile harness abundant inherent knowledge from the pre-trained corpus. Moreover, we highlight the pivotal importance of selecting a latent space with appropriate information density aligned with that of the image, facilitating the fusion of visual information. On the one hand, prior studies (He et al., 2023b; Liu et al., 2023a) have typically converted discrete text into a continuous form using a simple embedding layer. If we regard this word embedding space as a form of shallow text latent space, we notice a lack of interaction between tokens and overall semantic modeling, which poses a challenge in aligning images with these independent token embeddings. On the other hand, it is noteworthy that the information density of vision is lower than that of text (He et al., 2021), with large portions of image pixels often containing redundancy, while natural language tokens typically convey rich semantic information. To address this discrepancy, instead of employing the entire BERT model as a text encoder, we opt to divide the BERT model into two parts, utilizing the lower portion as the text encoder and the upper portion as the decoder. Setting the text latent space based on the middle layer of BERT yields improved alignment between vision and language, thereby enhancing performance. In addition, to improve the decoder's ability to reconstruct the text space, we make the parameters in the language model head trainable.

However, this BERT sentence feature space still exhibits drawbacks. To achieve a more standardized sentence feature space conducive to noise addition, we employ normalization. We gather a subset of all captions from the dataset and compute the mean and standard deviation of their corresponding latent codes, $\hat{\mu}(x)$ and $\hat{\sigma}(x)$. During training, these statistics are utilized to regularize the feature space of BERT by operating on each sample as follows: $\mathrm{norm}(x) = [x - \hat{\mu}(x)]/[\hat{\sigma}(x) + \epsilon]$. During inference, an unnorm module is applied to the predicted $\hat{x}_0$ before feeding it to the decoder. Moreover, a discrepancy between applying the diffusion model to text and image is the variable length of text, which forces the model to implicitly learn this supervised signal. In LaDiC, we extract all positions of special tokens like `[CLS]`, `[SEP]`, `[PAD]`, whose representations will be messy in contextual embeddings, forming a set $\mathcal{S}$. We then reassign what we call an empty token to the latent code in these locations, namely pasting vectors with all 0s, as demonstrated in Equ. 1. Here, x_i^{final} represents the i-th position of the final latent codes.

$$x_i^{final} = \begin{cases} [\mathrm{norm}(x)]_i & i \notin \mathcal{S} \\ \mathbf{0}, & i \in \mathcal{S} \end{cases} \tag{1}$$

Through this technique, for short captions with pad tokens at the end, the diffuser can quickly identify this repeated pattern and easily recover these unified zero vectors, implicitly learning sentence boundaries. This approach avoids the need for an additional module for predicting sentence length, as seen in DDCap (Zhu et al., 2022). Furthermore, despite a fixed length given during inference, the token forecasted as a pad will be mapped to the empty token defined above, and can be easily erased by postprocessing. Through these two regularization modules, we regularize the sentence feature space of BERT and finally obtain a latent space $\mathcal{X}$ tailored for captions.

3.3 Diffuser Mapping Image to Text

The caption diffuser serves as an interface transforming the vision space $\mathcal{V}$ into the text latent space $\mathcal{X}$. To fit the distribution of space $\mathcal{X}$ by diffusion models, firstly we sample x_t, the noisy version of the latent code $x_0 \in \mathcal{X}$, as $x_t|x_0 \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, \sqrt{1-\bar{\alpha}_t}\mathbf{I})$, where $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i = \prod_{i=1}^{t}(1-\beta_i)$ and $\beta_t \in (0,1)$ is the variance schedule. A notable property of this setting is that as $t \to \infty$, x_t is equivalent to an isotropic Gaussian distribution, aligning with the starting state

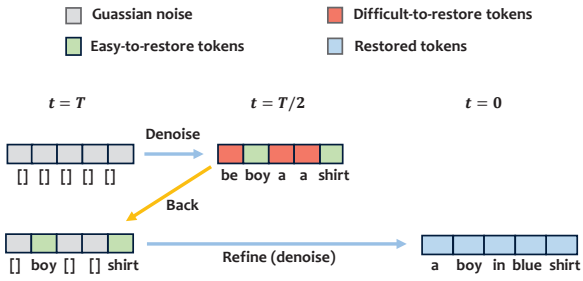


Figure 5: Illustration of Back&Refine technique.

of inference. Then for the denoising process, we use a Transformer encoder and predict the original x_0 based on the image directly, denoted as $\hat{x}_0 = f_\phi(x_t, v, t)$, where ϕ represents the parameters of the diffuser. A rigorous mathematical explanation of the diffusion model can be found in App. D if necessary.

In contrast to some previous approaches that inject image information by appending the `[CLS]` token of the vision encoder to text (Xu, 2022; He et al., 2023b), our LaDiC model adopts the cross-attention mechanism, treating text as the query to extract information from related image patches. We hypothesize that this approach will inject vision information more effectively and verify it in the ablation study. Additionally, we adapt classifier-free guidance (Ho and Salimans, 2022) to this task by randomly zeroing out some images and feeding them into the model together with normal training samples. During inference, a linear combination of the conditional estimate $f_\phi(x_t, v, t)$ and unconditional one $f_\phi(x_t, \emptyset, t)$ is performed: $\hat{x}_0 = (1+w)f_\phi(x_t, v, t) - wf_\phi(x_t, \emptyset, t)$ where w is a predefined hyperparameter.

We use a two-fold loss to train the caption diffuser in LaDiC. The first one is the loss $\mathcal{L}_{\text{latent}}$, operating within the text latent space. This loss calculates the Mean Squared Error (MSE) between the predicted text latent $\hat{x}_0 = f_\phi(x_t, v, t)$ and the original text latent x_0. The second one is the loss $\mathcal{L}_{\text{caption}}$ in the discrete caption space. Specifically, let $\hat{x}_0^i$ be the i-th position of the text latent and w_i be the i-th word in the ground-truth caption. The probability of correctly predicting w_i in the vocabulary given $\hat{x}_0^i$ is $p_\theta(w^i|\hat{x}_0^i)$, where θ represents the parameters of the diffuser and language model head in text decoder. This loss makes the output of the caption diffuser shrink faster, sharing the same intuition with XE loss in (Luo et al., 2022) and anchor loss in (Gao et al., 2022). Meanwhile, it also helps train the language model head in the decoder. In summary, the final loss $\mathcal{L}$ is:

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_{\text{latent}} + \lambda\mathcal{L}_{\text{caption}} \\ &= \|f_\phi(x_t, v, t) - x_0\| - \lambda \prod_{i=1}^{n} p_\theta(w^i|\hat{x}_0^i),\end{aligned} \tag{2}$$

where λ is a hyper-parameter.

3.4 Back&Refine Technique during Inference

We observe that diffusion models exhibit a certain degree of independence in both spatial and temporal dimensions during the inference process. In terms of the temporal aspect, in the typical inference process of diffusion models like DDIM (Song et al., 2020), which gradually denoise with time progress, the model initially predicts $\hat{x}_0$ from x_t, then obtains a less noisy version x_{t-1} from a distribution of $p(x_{t-1}|x_t, \hat{x}_0)$. However, (Chen et al., 2022a) found that in this manner, the previously estimated $\hat{x}_0$ is simply discarded in a new inference step. In response, they propose a self-conditioning technique, utilizing the previously generated result to improve sample quality. However, there is little exploration in the spatial dimension, i.e., the positions of each word in a sentence. In contrast to AR models with explicit sequential dependencies across tokens, the diffusion model emits all tokens in parallel. Undoubtedly, this approach boosts the inference speed but partially loses the information flow between tokens. Considering that some tokens are easily recovered, such as the main objects in the picture, adding the same scale of noise to these well-restored tokens as the others is somewhat unreasonable and wasteful. On the contrary, we should leverage these informative tokens. Therefore, we propose a technique named Back&Refine. As illustrated in Fig. 5, let's say we want to predict a sentence with a sequence length L and a sampling step T. Then at time $T/2$, several tokens are considered good enough, measured by the confidence scores of our model. We rank these scores and label tokens that fall in the lagging half. For these $L/2$ tokens that the model is not currently confident about, we try to reproduce them by noising them with complete Gaussian noise, while the others remain unchanged as information. Then we set the current $t = T$ and start a brand new denoising procedure. We will demonstrate the tuning of this technique in the experiments and offer another example for deeper understanding in App. A.5.

Model	# Images	BLEU@4	CIDEr	METEOR	SPICE	ROUGE-L	CLIP-Score	BERT-Score
Autoregressive								
Show and Tell (Vinyals et al., 2014)	-	31.4	97.2	25.0	18.1	53.1	69.7	93.4
CLIPCap (Mokady, 2021)	-	33.5	113.1	27.5	21.1	-	-	-
OSCAR† (Li et al., 2020)	7M	36.5	123.7	30.3	23.1	-	-	-
ViTCap† (Fang et al., 2021)	4M	36.3	125.2	29.3	22.6	58.1	-	-
VinVL† (Zhang et al., 2021)	6M	38.2	129.3	30.3	23.6	60.9	76.6	88.5
BLIP† (Li et al., 2022a)	129M	39.7	133.3	-	-	-	77.4	94.4
GIT† (Wang et al., 2022)	4M	40.4	131.4	30.0	23.0	-	-	-
Traditional Non-autoregressive								
NAIC_{KD} (Guo et al., 2020)	0.1M	28.5	98.2	23.6	18.5	52.3	-	-
MNIC (Gao et al., 2019)	0.1M	31.5	108.5	**27.5**	**21.1**	**55.6**	-	-
FNIC (Fei, 2019)	0.1M	**36.2**	**115.7**	27.1	20.2	55.3	-	-
Diffusion model based								
DiffCap (He et al., 2023b)	0.1M	31.6	104.3	26.5	19.6	55.1	73.6*	92.2*
Bit Diffusion (Chen et al., 2022b)	0.1M	34.7	115.0	-	-	58.0	-	-
DDCap (Zhu et al., 2022)	0.1M	35.0	117.8	28.2	21.7	57.4	74.1*	93.4*
SCD Net (Luo et al., 2022)	0.1M	37.3	118.0	28.1	21.6	58.0	74.5*	93.4*
LaDiC (ours, 5 steps)	0.1M	35.1	115.2	27.4	21.3	56.7	77.1	93.8
LaDiC (ours, 30 steps)	0.1M	**38.2**	**126.2**	**29.5**	**22.4**	**58.7**	**77.3**	**94.4**

Table 1: Model performance on COCO dataset. † indicates pretrained models, and we gray them out since they use much more training data. * represents the results of models reproduced by ourselves. Our model achieves state-of-the-art performance across various metrics for both diffusion-based and traditional NAR models, and exhibits comparable performance with some well-established pretraining auto-regressive frameworks, despite being trained on significantly less data. The inference time measured on an A100 GPU for 5 steps is 0.020 s/img and 30 steps is 0.105 s/img.

4 Experiments

4.1 Experimental Settings

Dataset and Metrics We conduct our experiments on MS COCO Karpathy split (Lin et al., 2014; Karpathy and Fei-Fei, 2014), which comprises 113,287 training images, 5,000 validation images, and 5,000 test images. Each image is associated with 5 reference captions. For evaluating model performance, following the common practice of image captioning community, we use several metrics including BLEU@4 (Papineni et al., 2002), CIDEr-D (Vedantam et al., 2014), METEOR (Banerjee and Lavie, 2005), ROUGE-L (Lin, 2004), and SPICE (Anderson et al., 2016). Additionally, we employ two model-based metrics: CLIP-Score (Hessel et al., 2021) to assess semantic alignment between generated captions and images, and BERT-Score (Zhang et al., 2019) to evaluate text quality.

Implementation Details In our LaDiC model, the encoder and decoder are frozen, except for the LM-head. The weights of the encoder and decoder are initialized from the bottom 6 layers and top 6 layers of $\text{BERT}_{\text{base}}$, respectively. The rationale for selecting such a latent space is explained in App. A.4. For the diffusion forward process, we employ the widely used cosine β schedule and adopt the noise factor (Gao et al., 2022). The diffuser consists of 12 transformer encoder blocks with additional cross-attention layers in each block, and the weights are randomly initialized. To extract image features, we use the pretrained image encoder from $\text{BLIP}_{\text{base}}$ (Li et al., 2022a), which employs ViT-B/16, for a fair comparison with BLIP. The model is trained on 8×V100 GPUs for 60 epochs with a peak learning rate of 5e-5 and a warmup ratio of 0.1. Further details can be found in App. C.

4.2 Quantitative Analysis

We benchmark our LaDiC model against prior baselines, encompassing auto-regressive, traditional non-autoregressive, and diffusion-based models, leveraging the COCO dataset (refer to Tab. 1). Our model achieves state-of-the-art performance across various metrics for both diffusion-based and traditional NAR models. Specifically, LaDiC achieves 38.2 BLEU@4 and 126.2 CIDEr, marking improvements of 0.9 and 8.2, respectively, compared to the previous state-of-the-art method, SCD-Net. Remarkably, a variant of our model, utilizing only 5 inference steps, even outperforms all prior diffusion-based models in both CLIP-Score and BERT-Score. Moreover, in addition to its distinctive advantages over AR models, it is noteworthy

BLIP: a baseball player holding a bat next to home plate.
Ours 1: a batter getting ready to swing at a baseball game.
Ours 2: a baseball player standing near an umpire at home plate.
Ours 3: a baseball player is swinging a bat at a game.
GT 1: a baseball player is going to swing a bat.
GT 2: a man is at bat at a baseball game with a crowd watching.

Figure 6: An example generated by our model.

Finetuned BLIP:
a man playing tennis. the man is wearing a white shirt and black shorts. the man is holding a tennis racket in his hand. the man is wearing a white shirt and black shorts.

Ours:
a man playing tennis is standing on a tennis court. there is a green tennis ball above him. he is wearing white shirt, and blank shorts. there is a white line on the court.

Figure 7: An example generated by fine-tuned BLIP model and ours in image paragraph captioning.

Model	SCD-Net	BLIP	Ours
Fluency	2.8	**4.9**	4.5
Accuracy	3.3	4.2	**4.4**
Conciseness	3.4	4.4	**4.7**

Table 2: Results of human evaluation.

that LaDiC exhibits comparable performance with well-established pretraining auto-regressive frameworks such as ViTCap and VinVL, despite being trained on significantly less data.

To evaluate our model's capacity for considering holistic context, we tackle the task of image paragraph captioning (Krause et al., 2016) to generate a multi-sentence description of an image. Our model seamlessly adapts to paragraph captioning by extending the predefined length without additional special designs. Training our model on the dataset from (Krause et al., 2016) yields a BLEU@4 score of 7.3, surpassing finetuned BLIP's 6.1 and highlighting our model's advantage in mitigating error accumulation (refer to App. B.1 for more details). All these quantitative indicators above substantiate the accuracy and high quality of the captions generated by our model.

4.3 Case Studies and Human Evaluation

We conduct a case study to illustrate the faithfulness and diversity of the captions generated by LaDiC. As depicted in Fig. 6, the generated captions are not only reasonable and fluent but also exhibit inherent diversity due to the varied sampling noises introduced at the start of inference. Additional examples can be found in App. A.1. In the context of image paragraph captioning generation, Fig. 7 reveals a notable difference. While each sentence in the captions generated by BLIP demonstrates good quality, they tend to appear somewhat independent of each other, with many initiating with "the man" and occasionally featuring repetitions. Conversely, by leveraging a broader context, our model produces sentences with a more cohesive logical relationship.

We conduct user studies to evaluate the generated captions of LaDiC, inviting volunteers to rate captions on a five-point scale (1-5) for accuracy, conciseness and fluency. The results, presented in Tab. 2, demonstrate that our model surpasses the previous diffusion-based state-of-the-art SCD-Net in both aspects and achieves comparable results with BLIP. Details can be found in App. B.2.

4.4 Unleashing the Speed of Diffusion Model

Despite their powerful generative capabilities, diffusion models are notorious for their slow inference speed. Most previous works require more than 50 inference steps, significantly slower than traditional NAR methods, which typically involve around 10 refinement procedures. However, as shown in Tab. 1, our model achieves remarkable performance even with just 5 steps. We attribute this surprising convergence speed to specific techniques employed in our LaDiC model. Firstly, the direct prediction of x_0 and the definition of caption loss enable the model to rapidly learn the distribution of discrete caption text, akin to the consistency model (Song et al., 2023). Secondly, the carefully selected noise schedule and noise factor significantly enhance the learning process of diffusion models. Regarding observed latency, the results in Tab. 3 (measured on a single A40 GPU with a batch size of 256) and Fig. 2a demonstrate that our model showcases a rapid inference speed, excelling not only in the domain of diffusion-based models but also when compared to auto-regressive models.

4.5 Customizing the Generation Process

In contrast to the unidirectional generation manner of AR models, our LaDiC model adeptly fills in empty words at almost any position within a sentence, harnessing its capability to capture more holistic information, as demonstrated in Fig. 2c. Technically, when provided with a caption containing blanks, we extract contextual embeddings of

Model	DiffCap	DDCap	Ours
Inference latency(s/img)	0.625	0.113	**0.049**

Table 3: Inference latency of diffusion-based models.

the given tokens and mask the blank tokens with Gaussian noise. The standard denoising process is then applied, with the exception of reinserting the embeddings of predefined tokens back to their respective positions after each inference step, ensuring that the given information is retained. Through this method, our model functions as a customized generator based on the provided tokens. Additional results can be found in App. A.2.

4.6 Analysis for the Back&Refine Technique

Regarding the Back&Refine technique, as discussed in § 3.4, we specify that when predicting a sentence with a sequence length of L and a sampling step of T, we opt to backtrack at time $t = T/2$ ($T \rightarrow 0$) and discard $l = L/2$ tokens. As for the tuning of these two hyperparameters, we present an experiment conducted with Back&Refine applied once in Tab. 4. Analysis of these results reveals that choosing an early backtracking time ($t = 0.8T$) leads to inadequate recovery of well-stored tokens, failing to provide sufficient information and resulting in performance just similar to scenarios without Back&Refine. Conversely, backtracking at a late time ($t = 0.2T$) does not yield significant improvement, as easy-to-restore tokens are typically recovered quickly, and the additional steps introduce an undesirable drop in inference speed. Similarly, dropping the majority of tokens in the Back procedure ($0.8L$) results in a situation akin to scenarios without Back&Refine. Dropping too few tokens ($0.2L$) may introduce many mistaken tokens, adversely affecting performance. Therefore, we deliberately choose the setting $t = T/2$ and $l = L/2$ for our final version. Furthermore, an alternative option is to establish a confidence score threshold instead of directly dropping the last half. However, in the majority of our early experiments, these two settings exhibit a negligible performance gap. Consequently, we opt for the simpler second method in our final version.

4.7 Ablation study

To validate the effectiveness of our core designs, we conduct ablation studies on the COCO dataset. Owing to the extensive time required for the ablation study, we opted for a subset of the dataset and trained all models for 40 epochs, (at which point the validation loss has already converged). For the inference phase, we performed 5 steps. We begin with a simple baseline that appends only the `[CLS]` token of the image feature to the end of text embeddings and then trains the diffuser to recover them. Subsequently, we progressively incorporate our proposed techniques to evaluate their performance. As depicted in Tab. 5, all modules exhibit performance gains. The use of PLM (BERT) and regularization in this space significantly enhances performance, emphasizing the importance of a refined latent space. Techniques aimed at better capturing visual information, such as cross-attention and splitting the BERT, also play pivotal roles in improving performance.

	BLEU-4	CIDER
w/o Back & Refine	37.3	121.5
$t = 0.5T; l = 0.5L$ (Our final version)	**38.2**	<u>126.2</u>
$t = 0.8T; l = 0.5L$	<u>38.1</u>	**126.6**
$t = 0.2T; l = 0.5L$	37.5	122.9
$t = 0.5T; l = 0.8L$	37.6	123.5
$t = 0.5T; l = 0.2L$	37.5	122.3

Table 4: Results of different settings of Back&Refine Technique.

#Row	Cross-attention	Caption loss	PLM	Norm-Reass	Split	B&R	B@4	C
a							15.4	46.3
b	✓						20.3	59.1
c	✓	✓					22.8	76.3
d	✓	✓	✓				26.9	91.8
e	✓	✓	✓	✓			31.6	103.5
f	✓	✓	✓	✓	✓		33.4	110.0
g	✓	✓	✓	✓	✓	✓	**34.1**	**113.4**

Table 5: Ablation on COCO dataset.

5 Conclusion

In this paper, we reexamine the diffusion-based image-to-text paradigm and introduce a novel architecture, denoted as LaDiC. Our model attains state-of-the-art performance among diffusion-based methods and demonstrates comparable capabilities with some pre-trained AR models. Moreover, our extensive experiments reveal the exciting advantages of diffusion models over AR models in considering more holistic contexts and emitting all tokens in parallel. Consequently, we posit that diffusion models hold substantial potential for image-to-text generation and we anticipate that our work will open new possibilities in this field.

Limitations

For simplicity and focus, this paper concentrates on the main research topic of image-to-text generation. Nevertheless, we observe that our model can be readily adapted to other modalities or even pure text generation with minimal modifications. We leave these potential extensions for future work, and meanwhile, we hope this paper will inspire confidence among researchers engaging in text-centered multimodal generation tasks with diffusion models and look forward to exciting future works in this area. Furthermore, due to resource constraints, the model parameters and datasets employed in our study are not extensive. Considering the remarkable emergent abilities demonstrated by scaling up autoregressive models like GPT, it becomes an intriguing and worthwhile exploration to investigate whether our model or general diffusion models, can exhibit similar scalability.

Risk Consideration: As a generative model, our model may inadvertently produce results that are challenging to distinguish from human-written content, raising concerns about potential misuse. Employing text watermark techniques could be beneficial in mitigating this issue. Additionally, diffusion models typically demand substantial computational resources for training, leading to increased carbon dioxide emissions and environmental impact.

References

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. *ArXiv*, abs/1607.08822.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. 2021. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*.

Jianhong Bai, Tianyu He, Yuchi Wang, Junliang Guo, Haoji Hu, Zuozhu Liu, and Jiang Bian. 2024. Uniedit: A unified tuning-free framework for video motion and appearance editing. *ArXiv*, abs/2402.13185.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *IEEvaluation@ACL*.

A. Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, and Dominik Lorenz. 2023. Stable video diffusion: Scaling latent video diffusion models to large datasets. *ArXiv*, abs/2311.15127.

Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. 2022a. Analog bits: Generating discrete data using diffusion models with self-conditioning. *arXiv preprint arXiv:2208.04202*.

Ting Chen, Ruixiang Zhang, and Geoffrey E. Hinton. 2022b. Analog bits: Generating discrete data using diffusion models with self-conditioning. *ArXiv*, abs/2208.04202.

Xiaoliang Dai, Ji Hou, Chih-Yao Ma, Sam Tsai, Jialiang Wang, Rui Wang, Peizhao Zhang, Simon Vandenhende, Xiaofang Wang, Abhimanyu Dubey, Matthew Yu, Abhishek Kadian, Filip Radenovic, Dhruv Mahajan, Kunpeng Li, Yue Zhao, Vladan Petrovic, Mitesh Kumar Singh, Simran Motwani, Yi Wen, Yiwen Song, Roshan Sumbaly, Vignesh Ramanathan, Zijian He, Peter Vajda, and Devi Parikh. 2023. Emu: Enhancing image generation models using photogenic needles in a haystack.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805.

Sander Dieleman, Laurent Sartran, Arman Roshannai, Nikolay Savinov, Yaroslav Ganin, Pierre H. Richemond, A. Doucet, Robin Strudel, Chris Dyer, Conor Durkan, Curtis Hawthorne, Rémi Leblond, Will Grathwohl, and Jonas Adler. 2022. Continuous diffusion for categorical data. *ArXiv*, abs/2211.15089.

Zhiyuan Fang, Jianfeng Wang, Xiaowei Hu, Lin Liang, Zhe Gan, Lijuan Wang, Yezhou Yang, and Zicheng Liu. 2021. Injecting semantic concepts into end-to-end image captioning. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17988–17998.

Zhengcong Fei. 2019. Fast image caption generation with position alignment. *ArXiv*, abs/1912.06365.

Junlong Gao, Xi Meng, Shiqi Wang, Xia Li, Shanshe Wang, Siwei Ma, and Wen Gao. 2019. Masked non-autoregressive image captioning. *ArXiv*, abs/1906.00717.

Zhujin Gao, Junliang Guo, Xuejiao Tan, Yongxin Zhu, Fang Zhang, Jiang Bian, and Linli Xu. 2022. Difformer: Empowering diffusion model on embedding space for text generation. *ArXiv*, abs/2212.09412.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and LingPeng Kong. 2022. Diffuseq: Sequence to sequence text generation with diffusion models.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *NIPS*.

Longteng Guo, Jing Liu, Xinxin Zhu, Xingjian He, Jie Jiang, and Hanqing Lu. 2020. Non-autoregressive

image captioning with counterfactuals-critical multi-agent learning. In *International Joint Conference on Artificial Intelligence*.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Doll'ar, and Ross B. Girshick. 2021. Masked autoencoders are scalable vision learners. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15979–15988.

Tianyu He, Junliang Guo, Runyi Yu, Yuchi Wang, Jialiang Zhu, Kaikai An, Leyi Li, Xu Tan, Chunyu Wang, Han Hu, HsiangTao Wu, Sheng Zhao, and Jiang Bian. 2023a. Gaia: Zero-shot talking avatar generation.

Yufeng He, Zefan Cai, Xu Gan, and Baobao Chang. 2023b. Diffcap: Exploring continuous diffusion on image captioning. *ArXiv*, abs/2305.12144.

Zhengfu He, Tianxiang Sun, Kuan Wang, Xuanjing Huang, and Xipeng Qiu. 2022. Diffusionbert: Improving generative masked language models with diffusion models. In *Annual Meeting of the Association for Computational Linguistics*.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. 2021. Clipscore: A reference-free evaluation metric for image captioning. *ArXiv*, abs/2104.08718.

Jonathan Ho, Ajay Jain, and P. Abbeel. 2020a. Denoising diffusion probabilistic models. *ArXiv*, abs/2006.11239.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020b. Denoising diffusion probabilistic models. *Neural Information Processing Systems,Neural Information Processing Systems*.

Jonathan Ho and Tim Salimans. 2022. Classifier-free diffusion guidance.

Li Hu, Xin Gao, Peng Zhang, Ke Sun, Bang Zhang, and Liefeng Bo. 2023. Animate anyone: Consistent and controllable image-to-video synthesis for character animation. *arXiv preprint arXiv:2311.17117*.

Justin Johnson, Andrej Karpathy, and Li Fei-Fei. 2016. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Andrej Karpathy and Li Fei-Fei. 2014. Deep visual-semantic alignments for generating image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:664–676.

Andrej Karpathy and Li Fei-Fei. 2017. Deep visual-semantic alignments for generating image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 664–676.

Diederik P. Kingma and Max Welling. 2013. Auto-encoding variational bayes. *CoRR*, abs/1312.6114.

Jonathan Krause, Justin Johnson, Ranjay Krishna, and Li Fei-Fei. 2016. A hierarchical approach for generating descriptive image paragraphs. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3337–3345.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Fei-Fei Li. 2016. Visual genome: Connecting language and vision using crowdsourced dense image annotations.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdel rahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Annual Meeting of the Association for Computational Linguistics*.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022a. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation.

XiangLisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and TatsunoriB. Hashimoto. 2022b. Diffusion-lm improves controllable text generation.

Xiujun Li, Xi Yin, Chunyuan Li, Xiaowei Hu, Pengchuan Zhang, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Annual Meeting of the Association for Computational Linguistics*.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*.

Zheng-Wen Lin, Yeyun Gong, Yelong Shen, Tong Wu, Zhihao Fan, Chen Lin, Nan Duan, and Weizhu Chen. 2022. Text generation with diffusion language models: A pre-training approach with continuous paragraph denoise. In *International Conference on Machine Learning*.

Guisheng Liu, Yi Li, Zhengcong Fei, Haiyan Fu, Xiangyang Luo, and Yanqing Guo. 2023a. Prefix-diffusion: A lightweight diffusion model for diverse image captioning. *ArXiv*, abs/2309.04965.

Haohe Liu, Zehua Chen, Yiitan Yuan, Xinhao Mei, Xubo Liu, Danilo P. Mandic, Wenwu Wang, and MarkD . Plumbley. 2023b. Audioldm: Text-to-audio generation with latent diffusion models. *ArXiv*, abs/2301.12503.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023c. Visual instruction tuning. *ArXiv*, abs/2304.08485.

Justin Lovelace, Varsha Kishore, Chao Wan, Eliot Shekhtman, and Kilian Q. Weinberger. 2023. Latent diffusion for language generation.

Jianjie Luo, Yehao Li, Yingwei Pan, Ting Yao, Jianlin Feng, Hongyang Chao, and Tao Mei. 2022. Semantic-conditional diffusion networks for image captioning.

Ron Mokady. 2021. Clipcap: Clip prefix for image captioning. *ArXiv*, abs/2111.09734.

OpenAI. 2023.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

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

Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. 2022. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *ArXiv*, abs/2204.06125.

Machel Reid, VincentJ. Hellendoorn, and Graham Neubig. 2022. Diffuser: Discrete diffusion via edit-based reconstruction.

Shuhuai Ren, Sishuo Chen, Shicheng Li, Xu Sun, and Lu Hou. 2023. TESTA: Temporal-spatial token aggregation for long-form video-language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics.

Shuhuai Ren, Junyang Lin, Guangxiang Zhao, Rui Men, An Yang, Jingren Zhou, Xu Sun, and Hongxia Yang. 2021. Learning relation alignment for calibrated cross-modal retrieval. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. *ArXiv*, abs/1505.04597.

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

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. 2023. Consistency models. *ArXiv*, abs/2303.01469.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2014. Cider: Consensus-based image description evaluation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4566–4575.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and D. Erhan. 2014. Show and tell: A neural image caption generator. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3156–3164.

Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. 2022. Git: A generative image-to-text transformer for vision and language. *ArXiv*, abs/2205.14100.

Shitong Xu. 2022. Clip-diffusion-lm: Apply diffusion model on image captioning.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Fei Huang, and Songfang Huang. 2022. Seqdiffuseq: Text diffusion with encoder-decoder transformers. *ArXiv*, abs/2212.10325.

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. 2021. Vinvl: Revisiting visual representations in vision-language models. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tianyi Zhang, Varsha Kishore, Felix Wu, KilianQ. Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *Cornell University - arXiv,Learning*.

Haozhe Zhao, Zefan Cai, Shuzheng Si, Xiaojian Ma, Kaikai An, Liang Chen, Zixuan Liu, Sheng Wang, Wenjuan Han, and Baobao Chang. 2023. Mmicl: Empowering vision-language model with multi-modal in-context learning. *ArXiv*, abs/2309.07915.

Zixin Zhu, Yixuan Wei, Jianfeng Wang, Zhe Gan, Zheng Zhang, Le Wang, Gang Hua, Lijuan Wang, Zicheng Liu, and Han Hu. 2022. Exploring discrete diffusion models for image captioning.

A Additional Results

A.1 Generated Samples from COCO Dataset

Additional examples generated by our LaDiC model are presented in Fig. 10. It is shown that our model adeptly captures the main objects and their relationships in the depicted images. Simultaneously, the generated captions exhibit a high level of fluency.

A.2 Custom Generation

Utilizing the partially adding noise technique described in § 4.5, we observed that, unlike the unidirectional generation approach of AR models, our LaDiC model can effectively insert words into almost any position within a sentence. Fig. 11 offers additional examples to illustrate the generalization ability of this method.

A.3 Gradual Denosing Procedure during Inference

As a generative model, the diffusion model is capable of modeling the distribution of any space by being trained to progressively transform random noise into a ground truth sample. In our model, we opt to apply diffusion to the latent space of text, i.e., the sentence feature space of BERT. As illustrated in § 1, our process begins with Gaussian noise. At each step, we subtract a certain amount of noise. As the number of steps increases, the denoised sentence feature converges towards the ground-truth sentence feature. Mathematically, our diffuser will model such a distribution: $P(T_{i+1}|T_i, I)$. Here, I represents image features, T_i denotes the sentence feature at the last step i, and T_{i+1} signifies the new sentence feature with reduced noise.

As an illustrative example, refer to a specific case in Fig. 9. With an increase in steps, the Mean Squared Error (MSE) distance between the current sentence vector and the ground-truth sentence vector diminishes, and the sentences generated by the predicted sentence vector at each step become progressively more fluent. These findings collectively demonstrate the capability of our diffusion model to gradually steer noised sentence vectors towards ground-truth sentence vectors and generate high-quality samples. When we carefully check the generated captions, notably, the main objects initially emerge, and subsequently, more details are incrementally added, resulting in increasingly fluent sentences. This characteristic also serves as inspiration for our Back&Refine Technique, as discussed in § 3.4.

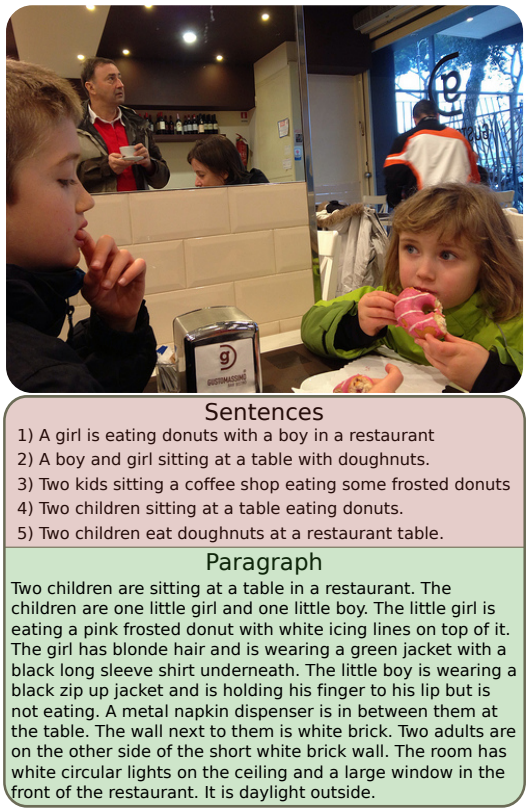


Figure 8: An example from image paragraph captioning dataset.

A.4 Exploration on the Choice of Latent Space

In § 3.2, we addressed the information density gap between vision and language by diffusing on the middle layer of the BERT model. Regarding the choice of different possible latent spaces, we conducted preliminary experiments to investigate this issue. We implemented various latent spaces extracted from different layers (specifically, the 3rd, 6th, and 9th layers of the $\text{BERT}_{\text{base}}$), with the findings presented in Tab. 6. It is important to note that in these initial tests, the model was trained for 40 epochs without incorporating caption loss and the Back&Refine technique. Our results indicate that the 6th layer outperforms the others, which is the rationale behind its selection as our final setting in the paper. Although we did not explore every layer, our preliminary experiments already provided us with a degree of confidence and suggested that layers proximal to the midpoint of BERT (while not necessarily exactly the 6th layer due to different datasets or hyperparameters) may have better align-

Timestep	MSE	Generated Captions
T=1	0.2944	a girl is a her her.
T=2	0.2191	a young girl a a cat a cat.
T=5	0.0648	young girl holding a cat holding a cat.
T = 10	0.0262	a young girl holding a small cat.

Figure 9: Gradual denosing process of diffusion models.

	BLEU-4	CIDER
Layer 3 of $\text{BERT}_{\text{base}}$	31.1	98.4
Layer 6 of $\text{BERT}_{\text{base}}$	**33.7**	**112.3**
Layer 9 of $\text{BERT}_{\text{base}}$	32.3	106.8

Table 6: Performance for different layers of BERT.

ment with image space.

A.5 The Self-Correction Ability of Back&Refine Technique

In our Back&Refine technique, we utilize the preserved easy-to-restore tokens to facilitate the generation of hard-to-restore tokens. However, it is important to emphasize that the remaining tokens in the Back procedure still have opportunities for revision in the Refine procedure rather than being fixed. On the one hand, our initial experiments find that well-restored tokens are inclined to be preserved during the Refine procedure. This observation guides our intuition to leverage these well-restored tokens to enhance the denoising process for challenging-to-restore tokens. On the other hand, it's noteworthy that, as these well-restored tokens are also required to pass through the Refine procedure, there are still chances to address errors inadvertently retained, such as grammar issues. For example, in a real-case scenario, when the Back procedure finishes, a sentence is "[] children is [] [] [] [] a pizza." where the ungrammatical word "is" is preserved. However, through the Refine procedure, the final output caption is corrected to "Two children are sitting at a table eating pizza."

B Additional Details in Experiments

B.1 Details about Experiments on Image Paragraph Captioning

The objective of image paragraph captioning is to generate comprehensive paragraphs that describe images, providing detailed and cohesive narratives. This concept was initially introduced in (Krause et al., 2016), where the authors proposed a dataset comprising 19,551 images from MS COCO (Lin et al., 2014) and Visual Genome (Krishna et al., 2016), each annotated with a paragraph description. An illustrative example is presented in Fig. 8.

To assess our model's ability to consider holistic context, we compare the performance of our model and BLIP on this task. For our model, we extend the predefined length to 60 and conduct training over 120 epochs. For BLIP, we fine-tune from $\text{BLIP}_{\text{base}}$ using the same number of epochs and an initial learning rate of 1e-5. Subsequently, we evaluate the results using BLEU on the test set. In the case of BLIP, the maximum length is set to 60, and the number of beams is 5 during inference.

B.2 Human Evaluation

As a generative task, in addition to automatic metrics, it is imperative to assess results through human subjective evaluation. To this end, we utilize MOS (Mean Opinion Score) as our metric and enlist the feedback of 20 experienced volunteers, who were tasked with rating results on a scale of 1-5. They evaluated the results from three perspectives: fluency, accuracy, and conciseness. Fluency gauges the quality of generated captions in terms of language, accuracy assesses whether the main objects and actions in the caption accurately reflect the picture, and conciseness evaluates the extent to which generative captions are informative and succinct, avoiding unnecessary details.

To ensure evaluation quality, we randomly sampled 10 pictures from the COCO dataset and generated corresponding captions for SCD-Net, BLIP[2], and our LaDiC model. Subsequently, we shuffled the three captions and required volunteers to rate them. To guarantee the reliability of the evaluation, we randomly selected 2 evaluators and calculated their correlation on each metric. This procedure

[2]For BLIP, we utilized the following page for convenient inference: https://replicate.com/salesforce/blip.

Hyperparameters	Values
Training	
Batch size	64*8(GPUs)
Epoch	60
Peak Learning rate	5e-5
Learning rate schedule	Linear
Warmup ratio	0.1
Optimizer	AdamW
β_1	0.9
β_2	0.999
Inference	
Method	DDIM
Sampling Criterion	Minimum Bayes Risk
Diffusion Process	
Diffusion steps	1000
β minimum	0.0001
β maximum	0.02
β schedule	Cosine
Classifier free probability	0.1
Classifier free weight	1
Self-conditioning probability	0.5
Loss	
λ	0.2
Loss type	l_2
Image Encoder	
Image size	224
Image Encoder	$\text{BLIP}_{\text{base}}$
Diffuser Module	
Sequence length	24
Hidden size	768
Layers	12
FFN size	3072
Attention heads	16

Table 7: More hyperparameters of our LaDiC model.

was repeated 5 times, and all results were found to be satisfactory.

As depicted in Tab. 2, our model surpasses the previous diffusion-based state-of-the-art SCD-Net in all aspects, achieving comparable results with BLIP. A slight decrease in text quality compared to BLIP may be attributed to the substantial training data used in BLIP's pretraining.

C More Hyperparameters

We list more hyperparameters for LaDiC model in Tab. 7.

D Mathematical Details for Diffusion Models

The training flow of the diffusion models is divided into two phases: the forward diffusion process and the backward denoising process. Given a data point sampled from a real data distribution $x_0 \sim q(x)$[3], we define a forward diffusion process in which Gaussian noise is incrementally added to the sample, generating a sequence of noisy samples $x_1, ..., x_T$. The noise scales are controlled by a variance schedule $\beta_t \in (0, 1)$, and the density is expressed as $q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I})$. Based on the reparameterization trick (Ho et al., 2020a), a nice property of the above process is that we can sample at any arbitrary time step in a closed form:

$$\begin{aligned}
x_t =& \sqrt{a_t}x_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1} \\
=& \sqrt{a_t}(\sqrt{a_{t-1}}x_{t-2} + \sqrt{1-\alpha_{t-1}}\epsilon_{t-2}) \\
&+ \sqrt{1-\alpha_t}\epsilon_{t-1} \\
=& \sqrt{a_t a_{t-1}}x_{t-2} + (\sqrt{a_t(1-\alpha_{t-1})}\epsilon_{t-2} \\
&+ \sqrt{1-\alpha_t}\epsilon_{t-1}) \\
=& \sqrt{a_t a_{t-1}}x_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\bar{\epsilon}_{t-2} \\
=& ... \\
=& \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon.
\end{aligned}$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. Thus:

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, \sqrt{1-\bar{\alpha}_t}\mathbf{I}), \quad (3)$$

Furthermore, from this equation, it becomes evident that as $T \to \infty$, x_T converges to an isotropic Gaussian distribution, aligning with the initial condition during inference.

However, obtaining the closed form of the reversed process $q(x_{t-1}|x_t)$ is challenging. Notably, if β_t is sufficiently small, the posterior will also be Gaussian. In this context, we can train a model $p_\theta(x_{t-1}|x_t)$ to approximate these conditional probabilities:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)),$$

where $\mu_\theta(x_t, t)$ and $\Sigma_\theta(x_t, t)$ are parameterized by a denoising network f_θ like U-Net (Ronneberger et al., 2015) or Transformer (Vaswani et al., 2017).

[3]We follow the notation and derivation process of https://lilianweng.github.io/posts/2021-07-11-diffusion-models.

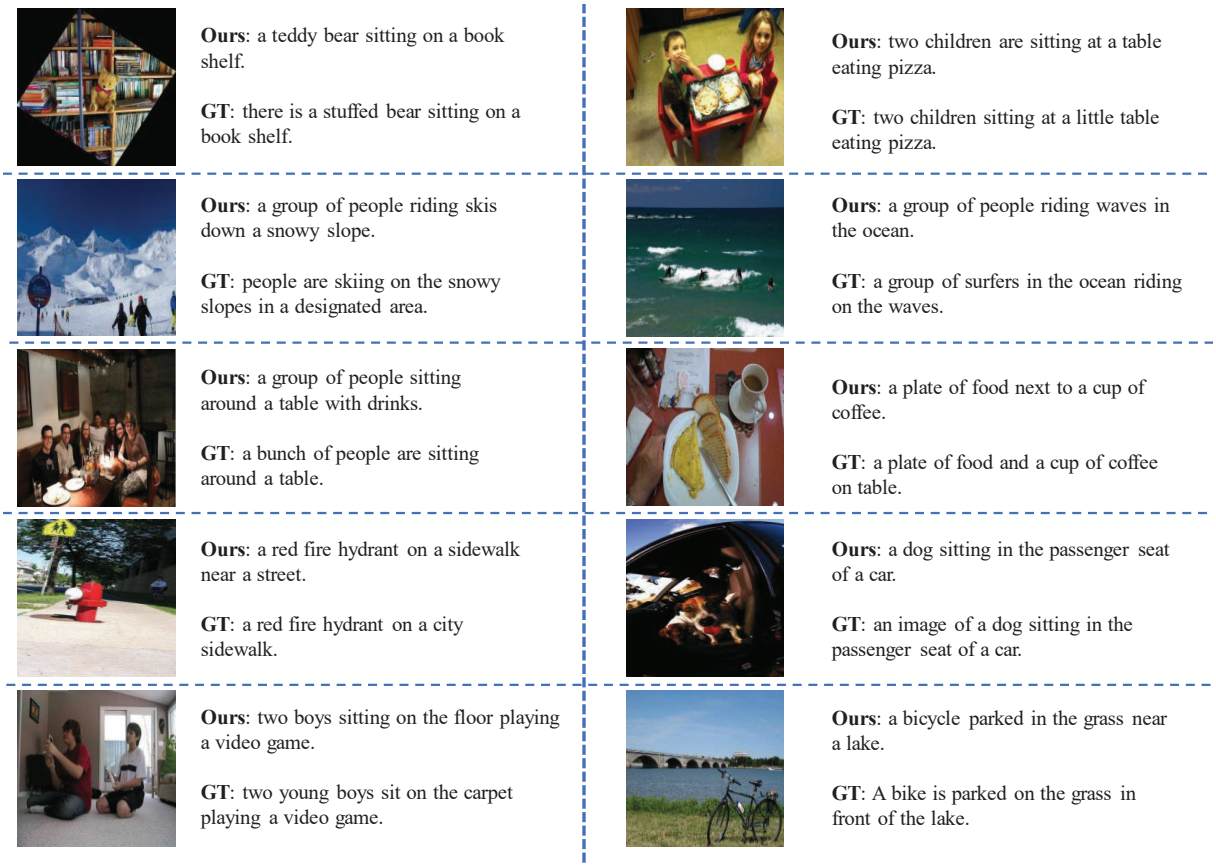


Figure 10: More examples generated by our model on COCO datasets.

Similar to VAE (Kingma and Welling, 2013), we can derive the variational lower bound to optimize the negative log-likelihood of input x_0 (Ho et al., 2020b), :

$$\mathcal{L}_{\text{vlb}} = \mathbb{E}_q[\underbrace{D_{\text{KL}}(q(x_t|x_0)||p_\theta(x_T))}_{\mathcal{L}_T}] - \underbrace{\log p_\theta(x_0|x_1)}_{\mathcal{L}_0}$$
$$+ \mathbb{E}_q[\sum_{t=2}^{T} \underbrace{D_{\text{KL}}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t))}_{\mathcal{L}_{t-1}}].$$

With an additional condition on x_0, the posterior of the forward process $q(x_{t-1}|x_t, x_0)$ can be calculated using Bayes theorem. Then in (Ho et al., 2020b) they derive:

$$\begin{aligned}
L_t &= \mathbb{E}_{x_0,\epsilon}\left[\frac{1}{2||\Sigma_\theta(x_t,t)||_2^2}||\tilde{\mu}_t(x_t,x_0) - \mu_\theta(x_t,t)||^2\right] \\
&= \mathbb{E}_{x_0,\epsilon}\left[\frac{1}{2||\Sigma_\theta(x_t,t)||_2^2}||\frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_t)\right. \\
&\quad \left. -\frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t,t))||^2\right] \\
&= \mathbb{E}_{x_0,\epsilon}\left[\frac{\beta_t^2}{2\alpha_t(1-\bar{\alpha}_t)||\Sigma_\theta||_2^2)}||\epsilon_t - \epsilon_\theta(x_t,t)||^2\right] \\
&= \mathbb{E}_{x_0,\epsilon}\left[\frac{\beta_t^2}{2\alpha_t(1-\bar{\alpha}_t)||\Sigma_\theta||_2^2)} \times \right. \\
&\quad \left. ||\epsilon_t - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t, t)||^2\right]
\end{aligned}$$

Removing the coefficients, a much more simple DDPM learning objective can be obtained:

$$\mathcal{L}_{\text{simple}} = \sum_{t=1}^{T} \mathbb{E}_q\left[||\epsilon_t(x_t,x_0) - \epsilon_\theta(x_t,t)||^2\right],$$

where ϵ_t is the noise added in original data x_0. Applied to textual data, (Li et al., 2022b) introduces an even simpler architecture to train a network to predict x_0 directly, with the loss function defined as $L = ||x_0 - f_\theta(x_t, t)||$.

Input: there is a boy [UNK] [UNK] [UNK] cows

Output: there is a boy standing by several cows

Input: [UNK] [UNK] [UNK] [UNK] [UNK] on the grass

Output: An old blue car parked on the grass

Input: a [UNK] [UNK] is holding a [UNK] in her hand

Output: a young girl is holding a cat in her hand.

Input: [UNK] [UNK] [UNK] [UNK] [UNK] in front of a computer.

Output: a cup of coffee sitting in front of a computer.

Figure 11: More examples of custom generation.

During inference, the reverse process commences by sampling noise from a Gaussian distribution $p(x_T) = \mathcal{N}(x_T; 0, \mathbf{I})$ and iteratively denoising it using $p_\theta(x_{t-1}|x_t)$ until reaching x_0. In DDIM (Song et al., 2020), a general form is derived from Equation 3.

$$\begin{aligned} x_{t-1} &= \sqrt{\overline{\alpha}_{t-1}}x_0 + \sqrt{1-\overline{a}_{t-1}}\epsilon_{t-1} \\ &= \sqrt{\overline{\alpha}_{t-1}}x_0 + \sqrt{1-\overline{\alpha}_{t-1}-\sigma_t^2}\epsilon_t \\ &\quad + \sigma_t\epsilon \\ &= \sqrt{\overline{\alpha}_{t-1}}x_0 + \sqrt{1-\overline{\alpha}_{t-1}-\sigma_t^2} \\ &\quad (\frac{x_t - \sqrt{\overline{a}_t}x_0}{\sqrt{1-\overline{\alpha}_t}}) + \sigma_t\epsilon \end{aligned}$$

$$q_\sigma(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \sqrt{\overline{a}_{t-1}}x_0 + \sqrt{1-\overline{\alpha}_{t-1}-\sigma_t^2}(\frac{x_t - \sqrt{\overline{a}_t}x_0}{\sqrt{1-\overline{\alpha}_t}}), \sigma_t^2\mathbf{I}).$$

where $\sigma_t^2 = \eta\tilde{\beta}_t = \eta\frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}\beta_t$, allowing us to adjust η as a hyperparameter to control the sampling stochasticity. The special case of $\eta = 0$ renders the sampling process deterministic. This model is referred to as the denoising diffusion implicit model (DDIM). It is noteworthy that DDIM shares the same marginal distribution as DDPM. Consequently, during generation, we can sample only a subset of diffusion steps $\tau_1, \ldots, \tau_S$, and the inference process becomes:

$$q_{\sigma,\tau}(\mathbf{x}_{\tau_{i-1}}|\mathbf{x}_{\tau_t}, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{\tau_{i-1}}; \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\frac{\mathbf{x}_{\tau_i} - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}}, \sigma_t^2\mathbf{I})$$

which, significantly reduces inference latency.