GlyphDiffusion: Text Generation as Image Generation

Anonymous EMNLP submission

Abstract

Diffusion models have become a new generative paradigm for text generation. Considering the discrete nature of text, in this paper, we propose GLYPHDIFFUSION, a novel diffusion approach for text generation via text-guided image generation. Our key idea is to render the target text as a *glyph image* containing visual language content. In this way, conditional text generation can be cast as a text-guided glyph image generation task, and it is then natural to apply continuous diffusion models to discrete texts. Specially, we utilize a cascaded architecture (*i.e.,* a base and a super-resolution diffusion model) to generate high-fidelity glyph images based on the input text. Finally, we design a text grounding module to transform and refine the visual language content from generated glyph images into the final texts. In experiments over four conditional text generation tasks and two classes of metrics (*i.e.,* quality and diversity), GLYPHDIFFUSION can achieve comparable or even better results than several baselines, including pretrained language models. Our model also makes significant improvements compared to the recent diffusion model.

1 Introduction

Diffusion models (Sohl-Dickstein et al., 2015) are a class of generative models that have recently shown to be powerful in synthesizing high-quality image (Saharia et al., 2022), audio (Kong et al., 2021) and video (Ho et al., 2022a). They are trained to gradually transform random noise drawn from a Gaussian distribution into a sample from the target distribution portrayed by a collection of samples. Compared to existing generative models such as GANs (Goodfellow et al., 2014), VAE (Kingma and Welling, 2014), and flow-based models (Dinh et al., 2017), diffusion models present several useful properties, *e.g.,* distribution coverage, a stationary training objective, and easy scalability (Dhariwal and Nichol, 2021). It has been shown that diffusion models are theoretically underpinned by non-equilibrium thermodynamics and score-based generative models (Nichol and Dhariwal, 2021).

Although diffusion models have made great success in the vision and audio domains (Kong et al., 2021; Saharia et al., 2022; Ramesh et al., 2022), it remains an open challenge to extend diffusion models to natural language due to the inherently discrete nature of texts. Consequently, prior work has focused on developing approaches based on discrete diffusion by introducing transition matrices between tokens to corrupt and recover texts (Austin et al., 2021; He et al., 2022; Reid et al., 2022). However, these methods cannot benefit from the improvements made on continuous diffusion models. Another line of work considers continuous text representations (*e.g.,* word embedding or hidden states) as training target, and learns diffusion models in the corresponding semantic space (Li et al., 2022; Gong et al., 2022; Strudel et al., 2022; Lin et al., 2022). However, unlike the target is usually fixed for continuous data (*e.g.,* image and audio), such training targets need to be learned from scratch for discrete texts, and they also correspond to different representation space depending on the pre-trained models. Thus, it might cause the collapse of the denoising loss function and bring instability to the training process (Gao et al., 2022).

In this paper, we propose GLYPHDIFFUSION, a novel diffusion approach for text generation via text-guided image generation. The key idea is that we render a target text as an image containing visual language content (called *glyph image*). In this way, the conditional text generation task can be cast as a text-guided glyph image generation task, where the glyph image is expected to contain the generated content in a visual form. Prior research (Ma et al., 2023; Liu et al., 2022b) has shown that the pixel representation of text can capture the spatial structure of characters for generating precise texts. More important, our method can naturally leverage continuous diffusion models and the fixed target (*i.e.,* glyph image) can avoid simultaneous changes in model predictions and ground truth to solve the collapse of the denoising loss.

Specifically, GlyphDiffusion introduces a cascaded architecture that integrates base and super-resolution diffusion models for glyph image generation. We conduct the image generation based on the input text semantics captured by a frozen T5 language model (Raffel et al., 2020). Since our goal is to produce high-quality text output that satisfies the need of the input text, we employ classifier-free guidance (Ho and Salimans, 2022) to enhance the content fidelity of a generated glyph image. Further, to improve the quality of the text output, we design a text grounding component to refine and transform the visual language content from generated images into the final generation results.

To the best of our knowledge, we are the first that adapts continuous diffusion models to discrete text generation via generating glyph images. While conceptually and intuitively simple, our model yields surprisingly strong results. Compared to AR and NAR models, GlyphDiffusion obtains over 50% improvements in metrics such as BLEU and ROUGE-L. Our model outperforms prior diffusion models *w.r.t.* quality and diversity (*e.g.,* $+2.54$ BLEU in Quasar-T and $+2.24$ Diverse-4 in GYAFC).

2 Background

Diffusion Models. Diffusion models are a class of generative models that convert Gaussian noise into samples via an iterative denoising process (Sohl-Dickstein et al., 2015; Ho et al., 2020). Given a sample from the target data distribution $\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0)$, the *forward process* of diffusion models produces a Markov chain of latent variables $\boldsymbol{x}_1, ..., \boldsymbol{x}_T$ by adding Gaussian noise to the sample:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t \boldsymbol{I}), \quad (1)$$

where $\beta_1, ..., \beta_T$ are small enough noise levels that make $\boldsymbol{x}_T$ well approximated by $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. We can further compute the posterior $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$ using Bayes theorem:

$$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_{t-1}; \tilde{\mu}_t(\boldsymbol{x}_t, \boldsymbol{x}_0), \tilde{\beta}_t \boldsymbol{I}), \quad (2)$$

where $\tilde{\mu}_t = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\boldsymbol{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{x}_t$. For generation, diffusion models are trained to reverse this forward process. The *reverse process* starts from a Gaussian noise $\boldsymbol{x}_T \sim \mathcal{N}(0, \boldsymbol{I})$ and gradually denoise $\boldsymbol{x}_t$ with learned Gaussian transition:

$$p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t), \boldsymbol{\Sigma}_\theta(\boldsymbol{x}_t, t)). \quad (3)$$

The reverse process is to match the joint distribution of the forward process by optimizing the variational lower bound (VLB). The VLB objective can be estimated using the posterior $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$ in Eq. 2 and the prior $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ in Eq. 3. To parameterize $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$, the most straight method is to predict $\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t)$ with a neural network. However, (Ho et al., 2020) have shown that predicting the noise $\boldsymbol{\epsilon}$ works much better. So the final objective can be simplified as follows:

$$L_{\text{simple}}(\theta) = \mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon},t}(\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\|_2^2). \quad (4)$$

This objective is equal to optimizing a reweighted VLB and has a connection to generative score matching (Song and Ermon, 2019; Song et al., 2020). To compute this surrogate objective, we generate samples $\boldsymbol{x}_t \sim q(\boldsymbol{x}_t|\boldsymbol{x}_0)$ by applying Gaussian noise $\boldsymbol{\epsilon}$ to $\boldsymbol{x}_0$ then train a model $\boldsymbol{\epsilon}_\theta$ to predict the added noise using Eq. 4.

Diffusion Models for Conditional Generation. In conditional generation, the data $\boldsymbol{x}_0$ is associated with a condition $\boldsymbol{c}$, such as a label in class-condition generation (Ho et al., 2022b), a low-resolution image for super-resolution (Saharia et al., 2021), or a text prompt in text-guided generation (Ramesh et al., 2022). The goal is to learn a conditional diffusion model $p_\theta(\boldsymbol{x}_0|\boldsymbol{c})$. Thus, the condition $\boldsymbol{c}$ is included into the reverse process in Eq. 3 as $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{c})$ for deriving a new objective:

$$L_{\text{simple}}(\theta) = \mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon},t}(\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t, \boldsymbol{c})\|_2^2). \quad (5)$$

During training, the data $\boldsymbol{x}_0$ and the condition $\boldsymbol{c}$ are sampled jointly from the data distribution $q(\boldsymbol{x}_0, \boldsymbol{c})$, and the forward process $q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$ remains unchanged. The only change required is to add the condition $\boldsymbol{c}$ as an extra input to the neural network in the reverse process $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{c})$.

3 GlyphDiffusion

In this section, we present GlyphDiffusion that casts conditional text generation as *text-guided image generation*, by establishing the semantic map from text condition to visual language content based on diffusion models. The overall sketch of our approach is shown in Figure 1.

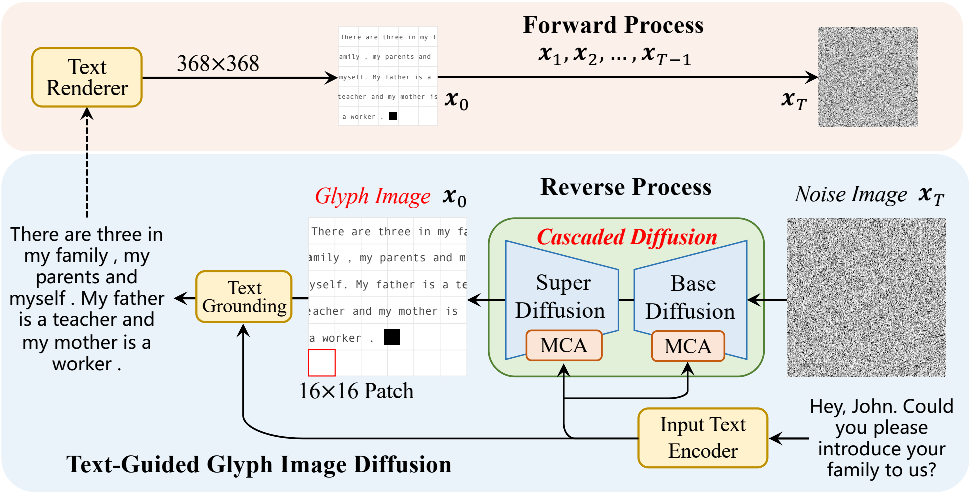
Forward Process
$x_1, x_2, \ldots, x_{T-1}$
Text Renderer
368×368
x_0
x_T
Reverse Process
Glyph Image x_0
Noise Image x_T
Cascaded Diffusion
Super Diffusion
Base Diffusion
MCA
MCA
There are three in my family , my parents and myself . My father is a teacher and my mother is a worker .
Text Grounding
16×16 Patch
Input Text Encoder
Hey, John. Could you please introduce your family to us?
Text-Guided Glyph Image Diffusion

Figure 1: Overview of our GlyphDiffusion model. "MCA" denotes multi-head cross attention.

3.1 Overview

To adapt diffusion models to text generation, existing work typically reconstructs *continuous targets*, *e.g.,* word embeddings (Li et al., 2022; Gong et al., 2022) and hidden states (Lovelace et al., 2022). Since these targets need to be learned beforehand during training, it is likely to cause the collapse of the denoising loss function (Gao et al., 2022). Different from previous work, we introduce a novel diffusion approach for conditional text generation by directly learning to map *a text condition* into *an image containing the generated text content*.

Task Formulation. Formally, given an input text (*a.k.a.,* a condition) $\boldsymbol{c}$, the conditional text generation task aims to generate an output text $\boldsymbol{w} = \{w_1, \ldots, w_n\}$ that consists of a sequence of words. In our approach, we propose to represent the output text with *glyph image* $\boldsymbol{x}$, which is taken as the training target of a text-guided image diffusion model $f(\cdot)$. Our focus lies in training a capable glyph image diffusion model, so as to generate high-quality language content in the visual form. Furthermore, we use a lightweight and disentangled text grounding model $g(\cdot)$ to refine and transform the visual content (glyph image) into the final text output $\hat{\boldsymbol{w}}$.

Text Rendering. To train our diffusion model, we need to prepare condition-image pairs $\langle \boldsymbol{c}, \boldsymbol{x} \rangle$ to replace condition-text pairs $\langle \boldsymbol{c}, \boldsymbol{w} \rangle$. For this purpose, we follow Rust et al. (2022) to design a text renderer that can convert one or more pieces of text (*i.e.,* a target text in text generation datasets) into an RGB image $\boldsymbol{x} \in \mathbb{R}^{H \times W \times C}$ (taken as the *target output* of diffusion models). We set the height $H = 16$, the width $W = 8464$, and select $C = 3$ RGB input channels. In this setting, the rendered glyph image is equal to a sequence of 529 image patches of size 16×16 pixels, and can be equally converted into a square image with a 368×368 resolution (see Figure 1 for an example of text rendering). For those texts longer than the maximum length, we truncate them as in discrete case. In this way, we can readily transform any existing text generation dataset to fit our setting.

3.2 Glyph Image Diffusion

In this section, we first introduce condition encoding, then present text-guided glyph image diffusion, and finally describe text grounding that maps images into text output.

3.2.1 Text Condition Encoding

In general text-to-image diffusion models, the input texts are encoded by text encoders which can be trained on specific datasets (Nichol et al., 2021) or pretrained on large-scale image-text data (Radford et al., 2021a). Since they focus on natural images for generation, the goal of text encoder is to encode visually meaningful and relevant semantics from input texts. By contrast, in our approach, the image to be generated is a rendering image only containing glyph features. Therefore, without considering visual features, we adopt pretrained text language models (*e.g.,* BERT (Devlin et al., 2019) and T5 (Raffel et al., 2020)) as text encoder to capture the semantics from the condition.

Compared to pre-trained image-text models (Jia et al., 2021; Radford et al., 2021b), language models are pretrained on text corpus substantially larger than paired image-text data, thus being exposed to very rich and diverse distribution of text and having a strong ability of deep textual understanding. In this paper, we use T5-Base model as our frozen text encoder, which can achieve decent performance in our experiments. We leave scaling the text encoder size for an improvement as future work.

3.2.2 Text-Guided Glyph Image Diffusion

Since we consider glyph image containing visual language content, it is infeasible to reuse or fine-tune prior general text-to-image models (Nichol et al., 2021; Ramesh et al., 2022) for glyph image. In order to generate high-fidelity images containing clear glyphs, we adopt a cascaded architecture (Ho et al., 2022b) to model the reverse process $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{c})$ for glyph image diffusion.

Cascaded Diffusion Achitecture. We use a pipeline of a base 64×64 model and a super-resolution model that upsamples a 64×64 base image into a 368×368 image (the target glyph image rendered in Section 3.1). For both base and super-resolution models, we adopt the U-Net model (Ronneberger et al., 2015), which is the current best architecture for image diffusion models, but change the attention layers to use multi-head attention (Vaswani et al., 2017). To adapt U-Net to text-guided glyph image diffusion, we take input text embeddings (encoded by the text condition encoder in Section 3.2.1) as input. Each step of the U-Net network can attend to the sequence of word emeddings via multi-head cross-attention. Specifically, the condition encoder τ_θ projects the input text $\boldsymbol{c}$ to a sequence of embeddings $\tau_\theta(\boldsymbol{c}) \in \mathbb{R}^{m \times d_\tau}$, where m is the number of tokens and d_τ is the embedding dimension. The text-conditional cross-attention layer is implemented as follows:

$$\text{MHA}(Q, K, V) = \text{softmax}(\frac{QK^\top}{\sqrt{d}})V, \quad (6)$$

$$Q = W_Q^{(i)} \psi_i(\boldsymbol{x}_t), K = W_K^{(i)} \tau_\theta(\boldsymbol{c}), V = W_V^{(i)} \tau_\theta(\boldsymbol{c}), \quad (7)$$

where $\psi_i(\boldsymbol{x}_t)$ denotes the flatten representation at the i-th layer, $W_Q^{(i)} \in \mathbb{R}^{d \times d_\psi}$, $W_K^{(i)}, W_V^{(i)} \in \mathbb{R}^{d \times d_\tau}$ are learnable matrices. For super-resolution model, we adopt the Efficient U-Net model (Saharia et al., 2022) for improving the memory efficiency, inference time, and convergence speed.

Enhancing the Text Guidance. Unlike general image generation, we rely on the visual content of glyph images for text generation. Thus, text semantics from the input text are particularly important to consider in our approach. To enhance the guidance of input condition on the output, classifier guidance is proposed by equipping diffusion models with a separate classifier (Dhariwal and Nichol, 2021). However, this approach strengthens the impact of input condition at the expense of output diversity. Thus, we adopt *classifier-free guidance* (Ho and Salimans, 2022) by jointly training a single diffusion model on conditional and unconditional objectives without a separate classifier as follows:

$$\hat{}_\theta(\boldsymbol{x}_t, \boldsymbol{c}) = w \cdot \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, \boldsymbol{c}) + (1 - w) \cdot \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t), \quad (8)$$

where $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, \boldsymbol{c})$ is implemented by the text-guided cascaded diffusion model, $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t)$ is realized by randomly dropping $\boldsymbol{c}$ from the diffusion model with a fixed probability (*e.g.,* 10%), and $w \geq 1$ is the guidance weight. By using classifier-free guidance, the objective in Eq. 5 can be modified and adapt to our text-guided glyph image diffusion as:

$$L_{\text{simple}} = \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{\epsilon}, t}(\|\boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t, \boldsymbol{c})\|_2^2). \quad (9)$$

3.2.3 Output Text Grounding

Once a glyph image is generated under the guidance of the text condition, we consider transforming it into an output text. A simple way is to employ some off-the-shelf toolkits such as optical character recognition for recognizing the words on the glyph image. However, such a way only focuses on word-level recognition and lacks an overall consideration of the text semantics, also suffering from potential issues such as incorrect word spelling. Therefore, we design a lightweight and disentangled text grounding module for mapping the glyph image into output text.

The text grounding module has a similar architecture to Transformer layer (Vaswani et al., 2017), while making special extensions that take a glyph image as input and condition on the input text. Specifically, it consists of three sub-layers, including multi-head self-attention (MHA), cross-attention (MCA), and feed-forward network (FFN). To feed the image as input, we flatten it into a sequence of 16×16 patches and map them to patch embeddings with dimension D:

$$\boldsymbol{h}_{inp} = [\boldsymbol{x}_p^1\mathbf{E}, ..., \boldsymbol{x}_p^j\mathbf{E}, ..., \boldsymbol{x}_p^N\mathbf{E}] + \mathbf{E}_{pos}, \quad (10)$$

where $\mathbf{E} \in \mathbb{R}^{(P^2 \cdot C) \times D}$ is a learnable matricx that projects each 2D patch $\boldsymbol{x}_p^j$ into a patch embedding, and N is the number of patches described in Section 3.1. The MHA and MCA layers use the same attention layer in Eq. 7 and attend to the input text embeddings $\tau_\theta(\boldsymbol{c})$. The final FFN layer contains two linear layers with a GELU activation and outputs a hidden state $\boldsymbol{h}_{out}$, which will be used to compute the word probability distribution over the vocabulary as follows:

$$\mathrm{Pr}(w_i|\boldsymbol{x}_0, \boldsymbol{c}) = \mathrm{softmax}(\boldsymbol{W}_v \boldsymbol{h}_{out} + \boldsymbol{b}_v). \qquad (11)$$

The text grounding model is trained to minimize the negative log-likelihood (NLL) loss as follows:

$$L_{\text{nll}} = -\sum_{i=1}^{n} \log \mathrm{Pr}(w_i|\boldsymbol{x}_0, \boldsymbol{c}). \qquad (12)$$

During optimization, the image diffusion model and the text grounding module are separately trained, and the text grounding module only introduces almost negligible parameters compared to the total parameters of the text-guided cascaded diffusion model.

3.3 Discussion and Learning

Comparison. Existing diffusion models for text generation can be categorized into two classes based on the modeling space. The first line of research, such as D3PM (Austin et al., 2021), DiffusER (Reid et al., 2022), and DiffusionBERT (He et al., 2022), proposed to model the transition between words considering the discrete categories of texts. However, these models depart from the diffusion modeling framework and lose some capabilities of diffusion models designed for continuous representations. Another line of research, such as LD4LG (Lovelace et al., 2022), DiffusionLM (Li et al., 2022), and DiffuSeq (Gong et al., 2022), focused on mapping words to continuous representations (*e.g.,* word embeddings), which need to be learned beforehand. Such a way suffers from the collapse of the denoising process and training instability. Our model is the first to map texts into glyph images, in which conditional text generation is cast as a glyph image generation task. We present a detailed comparison in Table 4.

Optimization. The training procedure of GlyphDiffusion can be described as: given a training pair $(\boldsymbol{c}, \boldsymbol{x}_0)$, we first obtain a low-resolution image $\boldsymbol{z}_0$ of the glyph image $\boldsymbol{x}_0$ and map the text condition $\boldsymbol{c}$ to embeddings; then, we add Gaussian noise to $\boldsymbol{z}_0$ and $\boldsymbol{x}_0$ and obtain $\boldsymbol{z}_t$ and $\boldsymbol{x}_t$ using Eq. 1; finally, a neural network $\boldsymbol{\epsilon}_\theta$ is trained to predict the Gaussian noise based on $\boldsymbol{c}$, $\boldsymbol{z}_t$, $\boldsymbol{x}_t$, and time step t with classifier-free guidance (Eq. 8). The diffusion model is optimized using L_{simple} in Eq. 9. Besides, we train the text grounding model given a training paier $(\boldsymbol{c}, \boldsymbol{x}_0, \boldsymbol{w})$, where $\boldsymbol{w}$ is the corresponding text of $\boldsymbol{x}_0$, using L_{nll} in Eq. 12. At inference time, based on the text condition, GlyphDiffusion first iteratively denoises the Gaussian noise to low-resolution glyph images, upon which the final glyph images can be generated in the same way.

4 Experiments

In this section, we detail the experimental setup and then highlight the conclusions of our results.

4.1 Experimental Setup

Tasks and Datasets. We evaluate GLYPHDIFFUSION on four conditional text generation tasks and datasets: 1) *Open-domain dialogue*: we adopt the **DailyDialogue** dataset (Li et al., 2017a), which contains $13,118$ multi-turn dialogues covering diverse daily topics; 2) *Question generation*: we use the **Quasar-T** dataset (Dhingra et al., 2017), consisting of $43,013$ open-domain trivia questions and their answers obtained from various internet sources; 3) *Style transfer*: we test on Entertainment&Music and Family&Relationship domains of the Grammarly's Yahoo Answers Formality Corpus (**GYAFC**) dataset (Rao and Tetreault, 2018), containing a total of $56,888$ informal/formal sentence pairs; and 4) *Paraphrase generation*: we adopt the Quora Question Pairs (**QQP**) dataset crawled from the community question answering forum Quora with 147K positive pairs. The detailed descriptions and statistics of these tasks and datasets are shown in Appendix A.

Baselines. Following previous work (Gong et al., 2022), we compare GLYPHDIFFUSION to four groups of baselines: 1) **GRU** with attention (Cho et al., 2014) and **Transformer** (Vaswani et al., 2017); 2) **GPT-2** (Radford et al., 2019) and **GP-VAE** (Du et al., 2022); 3) **NAR-LevT** (Gu et al., 2019); and 4) **DiffuSeq** (Gong et al., 2022) and **RDMs** (Zheng et al., 2023). We implement these models following their original papers. Other diffusion models (Lovelace et al., 2022; Yuan et al., 2022) present similar performance to DiffuSeq, so we select DiffuSeq as a representative. The details of baselines are shown in Appendix B.

Evaluation Metrics. In text generation tasks, *qual-*

Table 1: Evaluation results on four text generation tasks. The best results are denoted by **bold** fonts, and the best results without PLMs are denoted by underline fonts. “FT” means fine-tuning PLMs.

Tasks	Models	BLEU↑	ROUGE-L↑	BERTScore↑	Dist-1↑	Self-BLEU↓	Diverse-4↑	Length
Open-domain Dialogue	GRU-attention	0.0662	0.2137	0.4545	0.7889	0.8145	0.1540	10.45
	Transformer-base	0.0704	0.1990	0.4778	0.8934	0.4003	0.5777	20.01
	GPT2-base FT	0.0749	0.2176	0.5223	0.9445	0.0229	0.9654	20.23
	GPT2-large FT	0.0803	0.2434	0.5189	**0.9502**	0.0221	0.9500	20.33
	GPVAE-T5 FT	0.0843	0.2402	0.5089	0.6634	0.3677	0.5809	21.90
	NAR-LevT	0.0489	0.1054	0.4634	0.9233	0.8207	0.1453	6.43
	DiffuSeq	0.0740	0.2329	0.5794	0.9490	**0.0136**	0.9641	11.84
	GlyphDiffusion	**0.0855**	**0.2450**	**0.5844**	0.9500	0.0200	**0.9660**	13.20
Question Generation	GRU-attention	0.0651	0.2617	0.5222	0.7930	0.9999	0.3178	10.10
	Transformer-base	0.0364	0.1994	0.5334	0.8236	0.8767	0.4055	12.10
	GPT2-base FT	0.0741	0.2714	0.6052	0.9602	**0.1403**	**0.9216**	10.00
	GPT2-large FT	0.1110	0.3215	0.6346	**0.9670**	0.2910	0.8062	10.00
	GPVAE-T5 FT	0.1251	0.3390	0.6308	0.9381	0.3567	0.7282	11.40
	NAR-LevT	0.0930	0.2893	0.5491	0.8914	0.9830	0.4776	6.93
	DiffuSeq	0.1731	**0.3665**	0.6123	0.9056	0.2789	0.8103	11.50
	RDMs	0.1802	0.3550	0.6310	0.9082	-	-	-
	GlyphDiffusion	**0.1985**	0.3566	**0.6530**	0.9137	0.2005	0.8334	14.31
Style Transfer	GRU-attention	0.0502	0.2757	0.3145	0.8390	0.8290	0.3321	10.34
	Transformer-base	0.0677	0.2860	0.3232	0.8591	0.7991	0.3550	13.23
	GPT2-base FT	0.0734	0.2945	0.4360	0.9477	0.0657	0.9112	16.50
	GPT2-large FT	0.0757	0.3050	0.4143	0.9545	**0.0530**	0.9089	17.45
	GPVAE-T5 FT	0.0803	0.3048	0.4235	**0.9567**	0.0901	0.5949	19.80
	NAR-LevT	0.0538	0.2078	0.3523	0.9037	0.8343	0.3145	12.20
	DiffuSeq	0.0729	0.3046	0.4695	0.9440	0.1023	0.9120	12.35
	GlyphDiffusion	**0.0813**	**0.3088**	**0.4834**	0.9510	0.0934	**0.9344**	14.30
Paraphrase Generation	GRU-attention	0.1894	0.5129	0.7763	0.9423	0.9958	0.3287	8.30
	Transformer-base	0.0580	0.2489	0.5392	0.7889	0.7717	0.4312	5.52
	GPT2-base FT	0.1980	0.5212	0.8246	0.9798	0.5480	0.6245	9.67
	GPT2-large FT	0.2059	0.5415	0.8363	**0.9819**	0.7325	0.5020	9.53
	GPVAE-T5 FT	0.2409	0.5886	**0.8466**	0.9688	0.5604	0.6169	9.60
	NAR-LevT	0.2268	0.5795	0.8344	0.9790	0.9995	0.3329	8.85
	DiffuSeq	0.2413	0.5880	0.8365	0.9807	0.2732	0.8641	11.20
	RDMs	0.2498	0.5886	0.8466	0.9817	-	-	-
	GlyphDiffusion	**0.2503**	**0.5895**	0.8355	0.9810	**0.2344**	**0.8701**	12.32

ity and *diversity* are two key aspects for generated texts. To evaluate the quality, we adopt BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) to compute the overlapping n-grams between generated and gold texts. Since string matching based metrics can be insufficient, we use BERTScore (Zhang et al., 2020) to assess the semantic similarity between generated and gold texts at the embedding level. As for diversity, we adopt Distinct (Li et al., 2016), which computes the number of distinct n-grams in generated texts, and Diverse (Deshpande et al., 2019), which measures the ratio of distinct n-grams to the total number of generated words. Besides token-level diversity evaluation, we use self-BLEU (Zhu et al., 2018), a sentence-level metric that measures the overlapping n-grams among the generated texts. Following (Gong et al., 2022), we generate three samples for each text condition to compute the diversity metrics.

4.2 Main Results

Table 1 show the results of GLYPHDIFFUSION and baselines on four conditional text generation tasks.

First, compared to vanilla auto-regressive (AR) text generation models GRU and Transformer, GlyphDiffusion can achieve better results in four tasks at all quality and diversity metrics, which demonstrates the emergent capabilities of diffusion models in text generation. For the NAR baseline LevT, although it can outperform vanilla AR models in some cases, our GlyphDiffusion model can always obtain better performance with large margins (over 50% improvements on BLEU in Daily-Dialogue and ROUGE-L in GYAFC).

Second, compared to pretrained models GPT-2 and GPVAE-T5, GlyphDiffusion can outperform the base variants for most tasks and metrics, while achieving comparable performance to the large variants. It is worth noting that the large models have much more parameters than GlyphDiffusion to ensure high-quality generation results. As for the recent diffusion model DiffuSeq, our model wins 21 out of 24 competitions (4 tasks × 6 metrics), which indicates the effectiveness of our method that casts conditional text generation as a glyph image generation task.

Finally, in terms of diversity, GlyphDiffusion can generate significantly more diverse texts compared to AR, NAR, and pre-trained models, as shown by sentence-level diversity metrics (self-BLEU and Diverse-4). As for the word-level measure Distinct-1, we can observe that GlyphDiffusion is comparable with the pretrained GPT-2 models, indicating that our model has little repetition in word-by-word generation. To compare with DiffuSeq, our GlyphDiffusion model adopts a free way of generation – producing glyph images (contain visual language contents) then refining as final texts based on the condition. This approach can yield more diverse texts at both sentence and word levels.

4.3 Detailed Analysis

In this part, we conduct a series of in-depth analysis to study the effectiveness of GlyphDiffusion.

Ablation Study. In Section 3.2.2, we design a cascaded diffusion architecture to generate high-fidelity glyph images, and utilize the classifier-free guidance technique to enhance the text guidance. To examine their importance, we design two variants of our model: (1) *w/o Cascaded* removes the super-resolution model and uses the base diffusion model to generate glyph images with a 368×368 resolution; (2) *w/o Guidance* removes the unconditional objective $\epsilon_\theta(\boldsymbol{x}_t)$ from Eq. 8. Furthermore, in Section 3.2.3, we designed a text grounding module to improve the transformation from glyph images to output texts. To confirm its effectiveness, we design a counterpart: (3) *w/o Grounding* removes the text grounding module and directly recognize the content in glyph images as final output. The ablation results are shown in Table 2. We can observe that removing the cascaded pipeline suffers from a large performance drop in terms of both quality and diversity metrics. This demonstrates the effectiveness of the cascaded framework in generating high-fidelity glyph images. In addition, removing classifier-free guidance or the text grounding module results in a decreased performance, but the latter is more important. The reason might be that it may circumvent some potential issues (*e.g.,* incorrect spelling) in glyph images and improve final texts.

Table 2: Ablation study on GYAFC dataset.

Models	**BLEU**	**BERTScore**	**Dist-1**	**Diverse-4**
GlyphDiffusion	0.0813	0.4834	0.9510	0.9344
w/o Cascaded	0.0601	0.4438	0.9112	0.9011
w/o Guidance	0.0790	0.4730	0.9410	0.9219
w/o Grounding	0.0643	0.4566	0.9220	0.9090

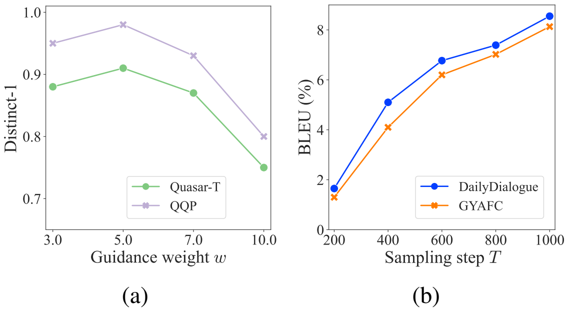

Figure 2: The Distinct-1 and BLEU scores *w.r.t.* different guidance weights w (a) and sampling steps T (b).

Sensitivity Analysis. In classifier-free guidance (Eq. 8), the weight w is an important factor affecting the guidance from the text condition. A large guidance weight can improve the image-text alignment but damage the output diversity. Here, we further examine the model performance (*i.e.,* Distinct-1) on Quasar-T and QQP datasets by varying the guidance weight in the set $\{3.0, 5.0, 7.0, 10.0\}$. As we can see from Figure 2(a), $w = 5.0$ gives the best Distinct-1 score, which is the final setting in our model. While generating using larger weights (*e.g.,* 10.0) can enhance the guidance of the condition by the super-resolution model, it gives considerably worse Distinct-1 (*e.g.,* 0.75 in Quasar-T). The sampling step T is another critical factor that significantly affects the model performance and generation speed. Here, we fix the number of diffusion steps during training while shrinking the inference steps from 1000 to 200 on DailyDialogue and GYAFC. As we can see from Figure 2(b), with the sampling step decreasing, the generated results also drop significantly (*e.g.,* from 8.55 to 1.65 BLEU in DailyDialogue). In practice, there is a trade-off between generation quality and inference speed.

4.4 Case Study

In this section, we perform qualitative analysis to show the effectiveness of our model. In Table 6, we present two examples for DailyDialogue and GYAFC datasets, and the generated outputs from three baselines (*i.e.,* GPT2-base, NAR-LevT, and

DiffuSeq) and our GlyphDiffusion model. As can be seen from Table 6, compared to NAR-LevT, our model can generate more informative and diverse texts. Since NAR-LevT adopt an iterative generation strategy, it tends to generate safe and short sentences such as "that's all right" in the dialogue task. As for GPT2-base which uses the powerful pretraining-finetuning paradigm, it can generate more fluent and richer content but sometimes going outside the topic of input texts. DiffuSeq sometimes generate irrelevant texts (*e.g.,* "drink my rests"). Since we adopt a cascaded diffusion framework, our model can generate high-quality glyph images. The text grounding module can resolve some potential issues in glyph images such as repetition (*e.g.,* "noooo") and incorrect spelling (*e.g.,* "gues"). More examples can be found in Appendix C.

5 Related Work

Diffusion Models for Image Generation. Diffusion models (Ramesh et al., 2022; Saharia et al., 2022) have demonstrated great success in generating high-quality and realistic images. Since the emergence of denoising diffusion probabilistic models (DDPM) (Ho et al., 2020), diffusion models are formalized as a forward process that corrupts the training images using Gaussian noise and a reverse denoising process that estimates the noise in the images at each step. On top of DDPM, (Nichol and Dhariwal, 2021) observe that the linear noise schedule is sub-optimal for low resolution and propose a new method to avoid fast information destruction towards the end of the forward process. The work of (Nachmani et al., 2021) replaces the Gaussian noise distributions with two other distributions, *i.e.,* a mixture of the Gaussian and the Gamma distribution. These works focused on unconditional image generation without any supervision signals. By contrast, recent work has been devoted to studying text-conditioned image generation that relies on CLIP text encoding (Galatolo et al., 2021; Gal et al., 2022; Ramesh et al., 2022). For example, (Kim and Ye, 2021) edit images with text prompts guided by a CLIP loss between the prompt and latent vector. (Ho et al., 2022b) present cascaded diffusion models, an approach for generating high-resolution images combining multiple diffusion models. Different from prior work, our work renders the target texts as textual images and uses a diffusion model to generate visualized texts.

Diffusion Models for Text Generation. To handle discrete text, prior work has extended diffusion models by defining a discrete corruption process (Hoogeboom et al., 2021a,b). For example, (Austin et al., 2021) and (He et al., 2022) use transition matrices to enable gradual corruption and denoising on a sequence of discrete tokens. Unlike these works, more recent work has focused on continuous diffusion models for text (Li et al., 2022; Gong et al., 2022; Strudel et al., 2022). Diffusion-LM (Li et al., 2022) works on the word embeddings and uses mapping functions to connect the discrete and continuous space of texts. Similarly, DiffuSeq (Gong et al., 2022) is designed for sequence-to-sequence text generation using one single model to model the conditional probability. Furthermore, (Liu et al., 2022a) propose a new efficient approach for composable text operations in the compact, low-dimensional latent space of text. In this paper, we also focus on continuous diffusion models for text generation but differ in that texts are rendered as continuous images instead of word embeddings. The key advantage of our method is that it allows an efficient diffusion process without a need of training an embedding step and a rounding step. Therefore, rendered text images can be an effective alternative to embeddings to leverage the continuous diffucion models. To the best of our knowledge, our work is the first to explore this setting for conditional text generation.

6 Conclusion

This paper presented a diffusion model, GLYPHDIFFUSION, for conditional text generation. We render a target text onto a glyph image containing visual language content, so that conditional text generation can be cast as a glyph image generation task. It enables continuous diffusion models to be naturally leveraged in our approach. In order to generate high-fidelity glyph images, we introduce a cascaded diffusion architecture equipped with classifier-free guidance. Further, we design a text grounding module that can refine and transform the content from glyph images into final texts. Experiments on four conditional text generation tasks show the effectiveness of our model to previous AR, NAR, and diffusion models. In future work, we will consider applying our model to more kinds of tasks. This study proposes a new line of research using diffusion models for text generation and demonstrates its effectiveness.

7 Limitations

An important limitation of *GlyphDiffusion* compared with other diffusion text generation models is the requirement of glyph images. The quality of glyph images will substantially influence the quality of the final texts. Since the diffusion models has some shortcomings in text generation, such as low generation speed and relatively worse performance, compared to language models, our model will inevitably inherit these properties.

Text generation techniques has been applied to a wide range of meaningful applications for society, such as game narrative generation, news report generation, and weather report generation. However, this technique may be potentially utilized for harmful applications.

References

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. 2021. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. In *Proceedings of SSST@EMNLP 2014, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, 25 October 2014*, pages 103–111. Association for Computational Linguistics.

Aditya Deshpande, Jyoti Aneja, Liwei Wang, Alexander G Schwing, and David Forsyth. 2019. Fast, diverse and accurate image captioning guided by part-of-speech. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10695–10704.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794.

Bhuwan Dhingra, Kathryn Mazaitis, and William W Cohen. 2017. Quasar: Datasets for question answering by search and reading. *arXiv preprint arXiv:1707.03904*.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. 2017. Density estimation using real NVP. In *International Conference on Learning Representations*.

Wanyu Du, Jianqiao Zhao, Liwei Wang, and Yangfeng Ji. 2022. Diverse text generation via variational encoder-decoder models with gaussian process priors. *arXiv preprint arXiv:2204.01227*.

Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. 2017. Question generation for question answering. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 866–874.

Rinon Gal, Or Patashnik, Haggai Maron, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. 2022. Stylegan-nada: Clip-guided domain adaptation of image generators. *ACM Transactions on Graphics (TOG)*, 41(4):1–13.

Federico A Galatolo, Mario GCA Cimino, and Gigliola Vaglini. 2021. Generating images from caption and vice versa via clip-guided generative latent space search. *arXiv preprint arXiv:2102.01645*.

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

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and LingPeng Kong. 2022. Diffuseq: Sequence to sequence text generation with diffusion models. *arXiv preprint arXiv:2210.08933*.

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 *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680.

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. *Advances in Neural Information Processing Systems*, 32.

Zhengfu He, Tianxiang Sun, Kuanning Wang, Xuanjing Huang, and Xipeng Qiu. 2022. Diffusionbert: Improving generative masked language models with diffusion models. *arXiv preprint arXiv:2211.15029*.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. 2022a. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851.

Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. 2022b. Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res.*, 23:47–1.

Jonathan Ho and Tim Salimans. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*.

Emiel Hoogeboom, Alexey A Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans. 2021a. Autoregressive diffusion models. *arXiv preprint arXiv:2110.02037*.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. 2021b. Argmax flows and multinomial diffusion: Towards non-autoregressive language models.

Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. 2020. Challenges in building intelligent open-domain dialog systems. *ACM Transactions on Information Systems (TOIS)*, 38(3):1–32.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR.

Gwanghyun Kim and Jong Chul Ye. 2021. Diffusionclip: Text-guided image manipulation using diffusion models. *CoRR*, abs/2110.02711.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. 2021. Diffwave: A versatile diffusion model for audio synthesis. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 110–119. The Association for Computational Linguistics.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B Hashimoto. 2022. Diffusion-lm improves controllable text generation. *arXiv preprint arXiv:2205.14217*.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017a. Dailydialog: A manually labelled multi-turn dialogue dataset. *arXiv preprint arXiv:1710.03957*.

Zichao Li, Xin Jiang, Lifeng Shang, and Hang Li. 2017b. Paraphrase generation with deep reinforcement learning. *arXiv preprint arXiv:1711.00279*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Zhenghao Lin, Yeyun Gong, Yelong Shen, Tong Wu, Zhihao Fan, Chen Lin, Weizhu Chen, and Nan Duan. 2022. Genie: Large scale pre-training for text generation with diffusion model. *arXiv preprint arXiv:2212.11685*.

Guangyi Liu, Zeyu Feng, Yuan Gao, Zichao Yang, Xiaodan Liang, Junwei Bao, Xiaodong He, Shuguang Cui, Zhen Li, and Zhiting Hu. 2022a. Composable text controls in latent space with odes. *arXiv preprint arXiv:2208.00638*.

Rosanne Liu, Dan Garrette, Chitwan Saharia, William Chan, Adam Roberts, Sharan Narang, Irina Blok, RJ Mical, Mohammad Norouzi, and Noah Constant. 2022b. Character-aware models improve visual text rendering. *arXiv preprint arXiv:2212.10562*.

Justin Lovelace, Varsha Kishore, Chao Wan, Eliot Shekhtman, and Kilian Weinberger. 2022. Latent diffusion for language generation. *arXiv preprint arXiv:2212.09462*.

Jian Ma, Mingjun Zhao, Chen Chen, Ruichen Wang, Di Niu, Haonan Lu, and Xiaodong Lin. 2023. Glyphdraw: Learning to draw chinese characters in image synthesis models coherently. *arXiv preprint arXiv:2303.17870*.

Eliya Nachmani, Robin San Roman, and Lior Wolf. 2021. Non gaussian denoising diffusion models. *arXiv preprint arXiv:2106.07582*.

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

Alexander Quinn Nichol and Prafulla Dhariwal. 2021. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR.

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.

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

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

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

Sudha Rao and Joel Tetreault. 2018. Dear sir or madam, may i introduce the gyafc dataset: Corpus, benchmarks and metrics for formality style transfer. *arXiv preprint arXiv:1803.06535*.

Machel Reid, Vincent J Hellendoorn, and Graham Neubig. 2022. Diffuser: Discrete diffusion via edit-based reconstruction. *arXiv preprint arXiv:2210.16886*.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

Phillip Rust, Jonas F Lotz, Emanuele Bugliarello, Elizabeth Salesky, Miryam de Lhoneux, and Desmond Elliott. 2022. Language modelling with pixels. *arXiv preprint arXiv:2207.06991*.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*.

Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. 2021. Image super-resolution via iterative refinement. *arXiv preprint arXiv:2104.07636*.

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

Yang Song and Stefano Ermon. 2019. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32.

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

Robin Strudel, Corentin Tallec, Florent Altché, Yilun Du, Yaroslav Ganin, Arthur Mensch, Will Grathwohl, Nikolay Savinov, Sander Dieleman, Laurent Sifre, et al. 2022. Self-conditioned embedding diffusion for text generation. *arXiv preprint arXiv:2211.04236*.

Martina Toshevska and Sonja Gievska. 2021. A review of text style transfer using deep learning. *IEEE Transactions on Artificial Intelligence*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Jiasheng Ye, Zaixiang Zheng, Yu Bao, Lihua Qian, and Mingxuan Wang. 2023. Dinoiser: Diffused conditional sequence learning by manipulating noises. *arXiv preprint arXiv:2302.10025*.

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

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Lin Zheng, Jianbo Yuan, Lei Yu, and Lingpeng Kong. 2023. A reparameterized discrete diffusion model for text generation. *arXiv preprint arXiv:2302.05737*.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1097–1100.

Appendix

We provide some experiment-related information as supplementary materials. The appendix is organized into three sections:

- Details of each task and dataset are presented in Appendix A;
- Details of baselines and our model are presented in Appendix B;
- Generated examples by our model are presented in Appendix C.

A Details of Tasks and Datasets

We evaluate GLYPHDIFFUSION on four kinds of conditional text generation tasks and datasets. *Open-domain dialogue* requires models to generate a fluent, engaging, and meaningful natural language response given previous dialogue turns between itself and one or more other participants (Huang et al., 2020). *Question generation* aims to generate natural language questions which can be answered by the given contents (Duan et al., 2017). *Style transfer* aims to change the stylistic manner of a text while preserving its meaning (Toshevska and Gievska, 2021). *Paraphrase generation* involves rewriting a sentence with the same semantic meaning but a different syntactic or lexical form (Li et al., 2017b). The detailed information of four datasets for these tasks is listed in Table 3.

B Model Details

Baselines. Following Gong et al. (2022), we compare GLYPHDIFFUSION to four groups of baselines:

- **GRU** with attention (Cho et al., 2014) and **Transformer** (Vaswani et al., 2017). These are two popular models for conditional text generation based on the encoder-decoder architecture with the (self-)attention mechanism.
- **GPT-2** (Radford et al., 2019) and **GPVAE** (Du et al., 2022). They are two pre-trained language models, among which GPT-2 is trained with language modeling and GPVAE augments T5 (Raffel et al., 2020) with VAE.
- **NAR-LevT** (Gu et al., 2019). It is a strong iterative non-autoregressive (NAR) text generation model that adopts two operations, *i.e.,* insertion and deletion, to generate and refine sequences iteratively.
- **DiffuSeq** (Gong et al., 2022). It is the recent diffusion model specially designed for conditional text generation. It uses partially noising to model the conditional probability in a single model without a separate classifier.

We implement these models following their original papers. Other diffusion models (Lovelace et al., 2022; Yuan et al., 2022) present similar performance to DiffuSeq, so we select DiffuSeq as a representative.

Baseline Settings. We follow the same baseline settings as Gong et al. (2022) and the results on Quasar-T and QQP are also collected from their work. The settings are listed in Table 5. For GRU-attention encoder-decoder model, we do not conduct diversity search algorithms on it, leading to poor sentence-level diversity. For NAR-LevT, we set the max iteration to 9 and utilize the termination condition described in the original paper. For GPVAE-T5, we set the scalars of all tasks as 2.

GLYPHDIFFUSION Settings. For our cascaded diffusion architecture, we follow the settings as Saharia et al. (2022). For the 64 × 64 base model, we use the Adafactor optimizer with a learning rate of 1e-4 for training. The hyper-parameters are set as follows:

```
“attn_resolutions”: [32, 16, 8]
“channel_mult”: [1, 2, 4, 8]
“dropout”: 0
“embed_dim”: 128
“cond_embed_dim”: 768
“num_res_blocks”: 3
“text_cross_attn_res”: [32, 16, 8]
```

For the 64 × 64 → 368 × 368 super-resolution model, we use an Efficient U-Net architecture for this model. Besides, we use the Adam optimizer with a learning rate of 1e-4 for training. The hyper-parameters are set as follows:

```
“channel_mult”: [1, 2, 4, 8]
“embed_dim”: 128
“cond_embed_dim”: 768
“num_res_blocks”: [2, 4, 8, 8]
```

For the text grounding model, we use the Adam optimizer with a learning rate of 1e-3 for training.

Table 3: Statistics of four datasets. #Training, #Valid, and #Test denote the number of input-output pairs in the training, validation, and test sets, respectively. #Output denotes the average number of tokens in the output texts.

Task	Dataset	#Train	#Valid	#Test	#Output
Open-domain Dialogue	**DailyDialogue**	76,052	7,069	6,740	13.89
Question Generation	**Quasar-T**	116,953	2,048	10,000	10.48
Style Transfer	**GYAFC**	52,595	2,877	1,416	13.02
Paraphrase Generation	**QQP**	144,715	2,048	2,500	9.86

Table 4: Comparison of our work to existing diffusion models for text generation.

Models	Text Condition	Learning Space	Learning Target	Target Fixed
D3PM (Austin et al., 2021), DiffusionBERT (He et al., 2022) DiffusER (Reid et al., 2022), RDMs (Zheng et al., 2023)		discrete	words	
LD4LG (Lovelace et al., 2022)		continuous	hidden states	
DiffusionLM (Li et al., 2022) SeqDiffuSeq (Yuan et al., 2022), DiffuSeq (Gong et al., 2022) DiNoiser (Ye et al., 2023)			word embeddings	
GlyphDiffusion		continuous	images	

The hyper-parameters are set as follows:

“dropout”: 0.3

“embed_dim”: 768

“ffn_dim”: 3072

“num_layer”: 2

“num_head”: 12

C Case Study

We show some qualitative examples of these four datasets in Table 7, Table 8, Table 9, and Table 10. As we can see from these tables, GlyphDiffusion tends to generate good-quality and diverse texts, but still not very fluent like pretrained models.

Table 5: The settings of different baselines. #Para. denotes the total amount of parameters.

Models	#Para.	Learning Paradigm	Diversity Method
GRU	65M	encoder-decoder	-
Transformer	80M	encoder-decoder	Temperature
GPT2-base	117M	pretrain-finetune	Hybrid strategy
GPT2-large	774M	pretrain-finetune	Hybrid strategy
GPVAE-T5	220M	pretrain+VAE	Gaussian sampling
NAR-LevT	80M	non-autoregressive	-
DiffuSeq	91M	non-autoregressive	Gaussian sampling

Table 6: Two examples of DailyDialogue and GYAFC. We present the generations from three baselines and our model. "w/o Grounding" shows the content in glyph images (omitting blanks).

	Input: good evening, saliva. what's that wonderful aroma from your kitchen? what are you doing now? [SEP] i am cooking now! [SEP] are you good at cooking? [SEP] **Gold:** i have studied its skills recently at a training school. i plan to run a restaurant, so i have to practise cooking!	**Input:** its not really a book i guess but its kind a long comic. **Gold:** it is a long comic, not a book.
GPT2-base	no, i'm just a bad. i have a little myself, regulars, programs and more	it's really a book, but it is seem it be a despite comic.
NAR-LevT	yes. that's all right.	it am not really a i book females it is
DiffuSeq	no, i don't drink my rests, and i need it crazy.	not a book, but it might seem be long comic.
Ours	no, i am not good at cooking, so i need to practise more. it is so attractive!	not really a book, but i guess it is long comic.
w/o Grounding	images/example-1.png	images/example-2.png

Table 7: Two examples for DailyDialogue. We show generated texts from three baselines and our model.

	Input: [CLS] listen, karen, i need your help. i don't know anyone here yet. [SEP] i'm glad to help you. what's wrong? [SEP] my mother - in - law just went into the hospital in l. a. hank and i will be flying there tonight. [SEP] i'm sorry to hear it. what's wrong with her? [SEP] doctors aren't sure yet. but the real problem is suzy. she has a bad cold, [SEP] **Gold:** yes, i'd ask jill, the girl i've had before, but i need someone overnight. maybe even for two nights.
GPT2-base	yes, i'd ask to her and there is girl. it's number. but i know her. she is very soon.
NAR-LevT	then have some do to side from and be an air. it its three and twenty and nothing domestic have to is is be hard.
DiffuSeq	i know. i'll know her and do an park. it's number. and nothing the soon to isn't you.
Ours	yes, i'd ask to her the girl. i've had before and i need someone but. maybe she is very tonight.
	Input: [CLS] thanks for inviting me to work out with you, joan. [SEP] don't mention it, let's go in. [SEP] yeah, this place looks great. wow, look at her, she can certainly get down, can't she? [SEP] she sure can. are you jealous, leslie? [SEP] a little, i wish i could do that. [SEP] you can! with a little practice. [SEP] look at him, he's buff. [SEP] i think he's hot too [SEP] **Gold:** that's it. i decided to turn over a new leaf. i'm going to exercise every single day.
GPT2-base	that's right. i don't want to make all of right now.
NAR-LevT	you of that for next use to have and of my left!
DiffuSeq	if you're right, it would be true. but i don't have to have to of my bad.
Ours	that's great. i decided to go there for that. i'm supposed to make all of my wife.

Table 8: Two examples for Quasar-T. We present the generations from three baselines and our model.

Input: [CLS] Numerous rocks and geological features abound around the 325 million year old volcano crater known as Arthur ’s Seat . [SEP] **Gold:** Edinburgh Castle stands on Arthur ’s Seat what was Arthur ’s seat	
GPT2-base	what was arthur ’s seat
NAR-LevT	what was castle on arthur ’s seat
DiffuSeq	what was castle on arthur ’s seat
Ours	what was edinburgh castle on arthur ’s seat
Input: [CLS] For his discovery of human blood groups he won the 1930 Nobel Prize in Physiology or Medicine . [SEP] **Gold:** Karl Landsteiner Won The Nobel Prize For Medicine In 1930 For His Discovery Of What	
GPT2-base	for what he won the 1930 nobel prize in physiology or medicine .
NAR-LevT	why he won the the 1930 physiology prize
DiffuSeq	for what he won the 1930 nobel prize in physiology or medicine .
Ours	for what he won the 1930 nobel prize in physiology or medicine .

Table 9: Two examples for GYAFC. We present the generations from three baselines and our model.

Input: [CLS] why do they try to sound british? [SEP] **Gold:** what is the appeal of sounding british?	
GPT2-base	why do they try to sound british?
NAR-LevT	what is sounding british
DiffuSeq	why do they try to sound british?
Ours	why do they attempt to sound british?
Input: [CLS] do u think 3 ppl in a band is a good amount?? [SEP] **Gold:** do you think that having three people in a band is a good amount?	
GPT2-base	do you think that three people location in a band is of amount?
NAR-LevT	do that you think 3 peoplel in a band is a amount?
DiffuSeq	do you feel three members is a good number
Ours	do you think that three people stated in a band is enjoyable positive?

Table 10: Two examples for QQP. We present the generations from three baselines and our model.

Input: [CLS] What is a good song to lyric prank your best friend? [SEP] **Gold:** What are some good lyric prank songs to send your best friends?	
GPT2-base	what songs with lyrics should you send to your best friends?
NAR-LevT	what songs will you send your friends?
DiffuSeq	what is the songs you send to your best friends?
Ours	what lyrics songs you will send to your closest friends?
Input: [CLS] What happens if dictatorship is continuing in the present days? [SEP] **Gold:** What happens if a dictatorship continues in the present day?	
GPT2-base	what would occur if a dictatorship continues in the present?
NAR-LevT	what would happen now if a dictatorship continues?
DiffuSeq	what would happen now if a dictatorship continues?
Ours	what would happen if a dictatorship continues in the present?