PoetryDiffusion: Towards Joint Semantic and Metrical Manipulation in Poetry Generation

Zhiyuan Hu[1]*, Chumin Liu[2], Yue Feng[3], Anh Tuan Luu[2], Bryan Hooi[1]

[1]National University of Singapore (NUS), Singapore
[2] Nanyang Technological University (NTU), Singapore
[3] University College London (UCL), UK

Abstract

Controllable text generation is a challenging and meaningful field in natural language generation (NLG). Especially, poetry generation is a typical one with well-defined and strict conditions for text generation which is an ideal playground for the assessment of current methodologies. While prior works succeeded in controlling either semantic or metrical aspects of poetry generation, simultaneously addressing both remains a challenge. In this paper, we pioneer the use of the Diffusion model for generating sonnets and Chinese SongCi poetry to tackle such challenges. In terms of semantics, our PoetryDiffusion model, built upon the Diffusion model, generates entire sentences or poetry by comprehensively considering the entirety of sentence information. This approach enhances semantic expression, distinguishing it from autoregressive and large language models (LLMs). For metrical control, its constraint control module which can be trained individually enables us to flexibly incorporate a novel metrical controller to manipulate and evaluate metrics (format and rhythm). The denoising process in PoetryDiffusion allows for the gradual enhancement of semantics and flexible integration of the metrical controller which can calculate and impose penalties on states that stray significantly from the target control distribution. Experimental results on two datasets demonstrate that our model outperforms existing models in terms of automatic evaluation of semantic, metrical, and overall performance as well as human evaluation. Codes are released to https://github.com/ChorlingLau/PoetryDiffusion/.

Introduction

Deep learning has greatly influenced natural language generation (NLG). Models like Seq2Seq (Sutskever, Vinyals, and Le 2014), GAN (Goodfellow et al. 2020), VAE (Kingma and Welling 2013), pre-trained language models, and LLMs have led NLG advancements. Among these, controllable text generation (CTG) is an emerging area within NLG and it is important and practical to consider specific constraints. Poetry generation stands out as a distinct domain with its unique characteristics, demanding not just coherent semantics but also strict adherence to metrical rules tied to format and pronunciation. These intricate requirements present the dual challenge of mastering semantics and metrics in tandem. In comparison to other coarse-grained CTG tasks like attribute-based generation (involving topics, emotions, and keywords), dialogue generation, and storytelling, poetry generation holds a unique position. Particularly evident in forms like sonnets and Songci, it necessitates adherence to well-defined and demanding metrics. Such specificity makes metrical poetry an ideal testing ground to validate the potency of the latest methodologies. Moreover, the available poetry data resource is of unparalleled quality in NLP, providing a strong foundation for our future work.

Sonnet and SongCi are two classical and famous forms of poetry, which share two major characteristics: (1) The

Sonnet

Time, the old humourist, has a trick to day A
Of moving landmarks and of levelling down B
Till into town the suburbs edge their way A
And in the suburbs you may scent the town B

With mount St. thus approaching muswell hill C
And clapham common marching with the mile D
You get a hammersmith that fills the bill C
A hampstead with a serious sense of style D

**Metrical Features:*
· Selected excerpts from a Sonnet with 14 lines originally.
· Words marked in red with the same letter notation rhyme.

SongCi

上林春慢 (CiPai)

Word 帽落宫花，衣惹御香，凤辇晚来初过(Guo)。
Tone − − + + 0 + − − + 0 − − − + + − 0

The hat adorned with palace flowers. The clothes infused with imperial fragrance. The phoenix carriage arrives late, just passing by.

Word 鹤降诏飞，龙擎烛戏，端门万枝灯火(Huo)。
Tone − − − + 0 + + + − 0 + + − + + − 0

Cranes descend as decrees fly. Dragons lift torches and play. Countless lanterns and lights at every palace gate.

Word 满城车马，对明月有谁闲坐(Zuo)。
Tone − + + − 0 − + − − + + − 0

Carriages and horses fill the city. Who sits leisurely in the face of the bright moon?

**Metrical Features:*
· Format is decided by punctuations "，" or "。".
· Words marked in red are positioned for rhyme, and their corresponding Mandarin Pinyin is provided in parentheses.
· Each word has its own tone, categorized as "+", "-", or "0".

Figure 1: Examples of Sonnet and SongCi

*Corresponding to Zhiyuan Hu: zhiyuan_hu@u.nus.edu

poems must adhere to special **format restrictions**. Sonnets must have 14 lines; similarly, the number of lines as well as the length of each line in SongCi is prescribed by the corresponding CiPai (Title of SongCi) (2) The chosen words must be consistent with specific **rhythm rules**. The last word of each line in Sonnets, as shown in Figure 1, should follow the rhyme scheme ”ABAB CDCD EFEF GG”. In SongCi, the rhyme rule of the last word in every line is also set by its CiPai. In this example, the phonetic transcription of each word is ”Guo”, ”Huo” and ”Zuo”, corresponding to one of 16 rhyme rules, ”o”. In addition, every word in SongCi must comply with the tone rule (Ping, Ze), which dictates pronunciation requirements. Level and oblique tones can be symbolized as ”+”, ”-”, and ”0” (without a tone requirement).

In previous CTG tasks, GAN and VAE models have been widely adopted as popular frameworks. However, Bond-Taylor et al. (2022) points out their limitations. Several key challenges associated with GANs encompass slow convergence, instability, vanishing gradients, mode collapse, and catastrophic forgetting. VAEs also suffer from posterior collapse, where the model ignores the latent variable and generates less diverse samples. Moreover, even the powerful LLMs, especially ChatGPT(OpenAI 2021), surprise us with excellent generation capability, yet adhering strictly to specific instructions remains a challenge. Pu and Demberg (2023) conduct empirical studies that demonstrate ChatGPT’s superiority over some previous SOTA models according to automated metrics. Despite this, notable discrepancies persist between ChatGPT’s output and human-authored content. Our experiments also highlight this issue, revealing that ChatGPT’s BLEU, ROUGE, and other semantic scores, which gauge the quality of generated poetry, fall short of our proposed method’s scores. Additionally, the performance on metrics employed to evaluate ChatGPT’s ability to adhere to metrical instructions and generate accurate metrical structures exhibits subpar performance, particularly in the SongCi dataset. It is worth noting that LLMs are trained on extensive corpora and derive their capabilities from instruction tuning. However, their generalization extent is uncertain, and further instruction tuning is resource-intensive and might compromise their original text generation quality. Although we might consider adopting efficient parameter-tuning techniques, such as Lora (Hu et al. 2021), to mitigate this challenge, fine-tuning LLMs, especially those Billion-level models, remain more complex than fine-tuning our model, which has only 87 Million parameters.

Except for the aforementioned drawbacks about semantic performance and generating correct metrics based on the instructions, most previous poetry generation works (Zhang and Lapata 2014; Ghazvininejad et al. 2016; Benhardt et al. 2018; Van de Cruys 2020; Tian and Peng 2022), solely concentrated on modeling the semantics, do not explicitly enforce metrical constraints despite evaluating the metrical performance. Only two works SongNet(Li et al. 2020) and MRCG(Zhang et al. 2019) directly incorporate the metrical rules representation into the generative model. However, both works utilize similar methods of encoding metrical rules into continuous representations and concatenating them with word embeddings, making it difficult to achieve satisfactory semantic performance when metrical features are combined in the modeling phase.

To address the challenges mentioned above, we propose the PoetryDiffusion model, which combines a Diffusion model for semantics with a metrical controller for metrics. Unlike other generative models, our diffusion approach introduces controlled noise through diffusion steps and then learns the reverse process to generate desired data. This design enhances training stability and generation quality. Moreover, this generative module and constraint separation increases adaptability for different generation tasks.

Specifically, the Diffusion model utilizes a noising process to transfer poetry representation into a normal distribution and samples from it as the input for the denoising phase, which reduces the noise and reverts it to the original poetry. The noising process is similar to gradually ”masking” tokens, phrases, or certain dimensions of the representation. On the other hand, the denoising phase aims to ”predict” the masked information and evaluate the success rate in each step. This mechanism ensures that the model captures all information of poetry rather than continuing to predict words based on wrong words in an autoregressive generation. The metrical controller employs classifier guidance, which offers notably higher precision and stability than other generative models, particularly LLMs. This approach adeptly incorporates metrical rules into a representation while also assessing the validity of the encoded rules. This allows for individual training and flexible integration, enabling efficient manipulation and assessment of metrics. Furthermore, when combining these two components to generate poetry, the modules for each step are updated based on feedback from Diffusion and controller in the previous step which indicates the accuracy of prediction for masked semantics and metrics.

To summarize, our contributions are as follows:

- We propose the PoetryDiffusion model, which employs the Diffusion model to optimize the poetry semantic performance, leverages the metrical controller to model the metrical rules, and combines them flexibly and effectively, for the first time in poetry generation.
- Comprehensive experiments through automatic semantic tests, metrical evaluations, case studies, and human evaluation on Sonnet and SongCi datasets demonstrate the effectiveness of our model.
- The visualization and analysis of the stepwise process reveals how the PoetryDiffusion model integrates the semantics and metrics gradually.

Related Work

Controllable text generation refers to the task of generating text according to the given controlled element. Hu et al. (2017) used differentiable approximation to discrete text samples, explicit constraints on independent attribute controls, and efficient collaborative learning of generators and discriminators to generate realistic sentences with desired attributes. Betti, Ramponi, and Piccardi (2020) introduced Controlled TExt generation Relational Memory GAN which utilizes an external input to influence the coherence of sentence generation. Furthermore, Li et al. (2022) proposed the

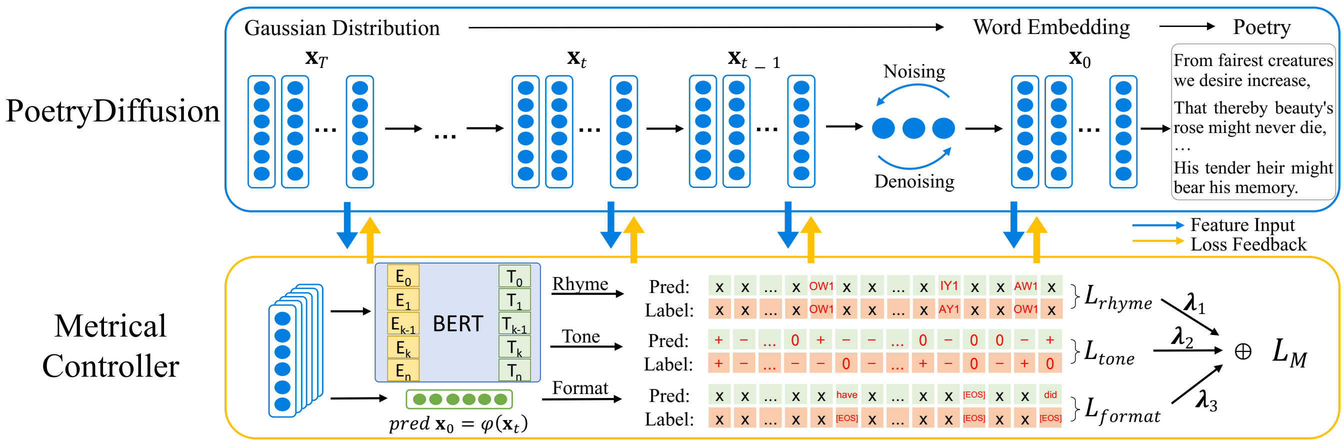

Figure 2: Model Architecture. PoetryDiffusion denoises $\mathbf{x}_T$ to poetry $\mathbf{w}$ based on joint loss $\mathcal{L}$ of each step.

Diffusion-LM to achieve several fine-grained controls. Chen and Yang (2023) incorporated different levels of conversation structures via Diffusion models to directly edit the prototype conversations.

Additionally, researchers also conducted some exploration based on the pretraining language model and large language model. Zhang and Song (2022) introduced a method incorporating attribute knowledge into control prompts to steer a frozen casual language model to produce attribute-specific texts. Sheng et al. (2021) utilized the masked sequence to sequence pre-training and attention-based alignment modeling for lyric-to-melody and melody-to-lyric generation. Zhang, Liu, and Zhang (2023) employs multiple LLM as different roles in text generation to iteratively refine the generation results. Zhou et al. (2023) conducted extra instruction tuning for lexical, syntax, semantic, style, and length constraints based on the LLM.

In terms of poetry generation, Yu et al. (2017) proposed SeqGAN, Lin et al. (2017) introduced RankGAN, and Che et al. (2017) came up with MaliGAN for poem generation. (Chen et al. 2019) proposed the semi-supervised VAE model for sentiment control in poetry generation. Yi et al. (2020) leveraged the MixPoet to enhance the diversity and quality of the poem. Deng et al. (2020) utilized a Quality-Aware Masked Language Model to polish the draft poetry generated by the encoder-decoder model.

Methodology

Overview

As shown in Figure 2, the proposed method is divided into two parts. PoetryDiffusion is a Diffusion based framework. It converts poetry input into continuous word representation, then encodes it as a Gaussian distribution by noising. The denoising step samples an initial state from Gaussian distribution and reverts it into poetry. Metrical Controller evaluates metrics and transmits the loss to denoising steps, guiding the poetry to approach the control objectives.

Diffusion Based Framework

Intuition As mentioned earlier, poetry is a well-structured literary form that demands thematic consistency and clarity, emphasizing coherence between its sub-sentences. Autoregressive models, which generate text word-by-word, have limitations that may lead to the accumulation of errors, resulting in off-topic or thematically inconsistent poetry. Additionally, generating only one token at a time makes it harder to conform to strict structural forms, which may require longer-term global context from multiple lines, or going back to revise earlier written content like a human would do. Therefore, considering the intricate control of poetic rhythm and the high demands placed on it, we opt for the Diffusion model (Sohl-Dickstein et al. 2015) as the semantic framework for generating poetry. It allows for comprehensive consideration of information from the entire poem during each iterative generation step and provides ample manipulative space for text controlling, especially format and rhyme scheme in poetry, throughout the iterations, and avoids getting restricted based on earlier generated tokens.

To formulate the Diffusion model's principle, we define $q(\cdot)$ as its forward propagation distribution, while $p_\theta(\cdot)$ is the trainable backward one.

Given a poem with N words, we represent it as a sequence $\mathbf{w}$ $(w_1, w_2, ..., w_N)$, where w_i stands for the i-th word of the poem. We adopt the methodology introduced by (Li et al. 2022), wherein we tailor the continuous Diffusion model to our specific task and incorporate a word embedding function

$$E(\mathbf{w}) = [E(w_1), E(w_2), ..., E(w_N)] \in \mathbb{R}^{N\times d}, \quad (1)$$

where the embedding $E(\cdot)$ comes from BERT, to map words into continuous representations, instead of operating discrete input directly. So the transformational step $\mathbf{w} \rightarrow \mathbf{x}_0$ can be described as

$$q(\mathbf{x}_0|\mathbf{w}) = E(\mathbf{w}), \quad (2)$$

where $\mathbf{x}_0 \in \mathbb{R}^{\mathrm{d}}$ is the initial representation of continuous Diffusion. Inversely, the trainable function

$$p_\theta(\mathbf{w}|\mathbf{x}_0) = \prod_{i=1}^{n} p_\theta(w_i|x_i) \quad (3)$$

is utilized to transfer continuous vectors into words. Among them, x_i is the representation of i-th word in $\mathbf{x}_0$ and $p_\theta(w_i|x_i)$ is an MLP network with softmax, mapping a high-dimensional x_i to a specific token w_i.

The model uses a Markov Chain $\{\mathbf{x}_0,\ \mathbf{x}_1,\ ..,\ \mathbf{x}_t,\ ..,\ \mathbf{x}_T\}$ to model the "noising" step and generate a Gaussian distribution $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$. The forward noising process is parameterized by

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}), \quad (4)$$

where β_t is the amount of noise added in the t-th step of noising phase. $\mathbf{x}_T$ is sampled as the initial state at the beginning of the reverse process, and the backward denoising can be formulated as

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_\theta(\mathbf{x}_t, t)), \quad (5)$$

where functions μ_θ and σ_θ are learnable and trained in the reverse phase.

Based on the forward noising process (Eq.4) and reparameterizing trick, $\mathbf{x}_t$ can be expressed by $\mathbf{x}_0$:

$$\begin{aligned}\mathbf{x}_t &= \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\mathbf{z}_{t-1} \\ &= \sqrt{\tilde{\alpha}}\mathbf{x}_0 + \sqrt{1-\tilde{\alpha}}\tilde{\mathbf{z}},\end{aligned} \quad (6)$$

where $\alpha_t = 1 - \beta_t$ and $\tilde{\alpha} = \prod_{t=1}^{T} \alpha_t$. In addition, noise added is defined by $\mathbf{z}_t \sim \mathcal{N}(0, \mathbf{I})$ and $\tilde{\mathbf{z}}$ is the Gaussian superposition of $\{\mathbf{z}_0, \mathbf{z}_1, ..., \mathbf{z}_t\}$.

Therefore, the training goal of the Diffusion model, which is also regarded as the semantic loss function $\mathcal{L}_\mathrm{S}$, is to estimate the distribution of p_θ in which the VLB (Variational Lower-Bound) is used as a computable lower-bound:

$$\begin{aligned}-\mathbb{E}_{q(\mathbf{x}_0)}[\log p_\theta(\mathbf{x}_0)] &\leq \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\right] \\ &= \mathbb{E}_{q(\mathbf{x}_{0:T})}\Bigg[\tfrac{1}{2\sigma^2}||\hat{\mu}(\mathbf{x}_T, \mathbf{x}_0)||^2 \\ &\quad + \textstyle\sum_{t=2}^{T} \tfrac{1}{2\sigma^2}||\mu_\theta(\mathbf{x}_t, t) - \hat{\mu}(\mathbf{x}_t, \mathbf{x}_0)||^2 \\ &\quad - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)\Bigg],\end{aligned} \quad (7)$$

where μ_θ is the mean of $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ and $\hat{\mu}$ is the mean of the posterior $q(\mathbf{x}_{t-1}|\mathbf{x}_0, \mathbf{x}_t)$. Removing the constant terms and the coefficient $\frac{1}{2\sigma^2}$, the loss function is simplified as:

$$\mathcal{L}_\mathrm{S}(\mathbf{x}_0) = \textstyle\sum_{t=1}^{T} \mathbb{E}||\mu_\theta(\mathbf{x}_t, t) - \hat{\mu}(\mathbf{x}_t, \mathbf{x}_0)||^2. \quad (8)$$

Combined with the step $\mathbf{w} \to \mathbf{x}_0$ and $\mathbf{x}_0 \to \mathbf{w}$, the loss function can be rewritten as:

$$\begin{aligned}\mathcal{L}_\mathrm{S}(\mathbf{w}) &= \mathbb{E}[\mathcal{L}_\mathrm{S}(\mathbf{x}_0) + \log q(\mathbf{x}_0|\mathbf{w}) - \log p_\theta(\mathbf{w}|\mathbf{x}_0)] \\ &= \mathbb{E}[\mathcal{L}_\mathrm{S}(\mathbf{x}_0) + ||E(\mathbf{w}) - \mu_\theta(\mathbf{x}_1, 1)||^2 \\ &\quad - \log p_\theta(\mathbf{w}|\mathbf{x}_0)].\end{aligned} \quad (9)$$

Metrical Controller

Motivation To ensure the primary model focuses more on the generation of text content itself, we devise a separately trained Metrical Controller to achieve format and rhyme control. In this way, PoetryDiffusion does not need to concatenate controlling encoding onto content-encoding as many previous methods do, which could scatter the model's semantic attention, thus minimizing the potential weakening of semantic representation caused by metrical control. Furthermore, the modular controller design enables our method to be easily adapted to other CTG tasks, significantly enhancing the practicality and versatility of our approach.

We employ deep neural network-based classifiers as the metrical controller due to two key advantages. Firstly, they adeptly model intricate distributions of specific attributes, thereby offering precise guidance during the diffusion process. Secondly, these classifiers enhance stability by easily calculating and imposing penalties on states that stray significantly from the target distribution. This not only addresses the instability often associated with the diffusion process but also ensures reliable samples.

Format The chosen poetic forms, Sonnet and SongCi, exhibit considerable flexibility in terms of sentence length. For instance, SongCi's under different CiPai's feature distinct theatrical formats, which differ from the fixed 5-character and 7-character poetic structures. Furthermore, while end signals of line or sentence are present in the original data, these signals encapsulate a significant amount of control information, encompassing not only line count and sentence length but also implicit positional cues for rhyme words, which must be at the end of lines or sentences. Therefore, explicit encoding of format information is essential to enhance format control and emphasize other associated details.

We define a sequence of format metrics, denoted by S (with the same length as $\mathbf{w}$), to indicate the target locations of ending signals. $S = (m, ..., m, \langle eos\rangle, m, ...)$, where "$\langle eos\rangle$" represents the end of each Sonnet line and m is a mask symbol meaning that its corresponding word has no specific format rule. In SongCi, the punctuation characters "," and "." will replace "$\langle eos\rangle$" and act as the ending signals of each sentence. Similarly, S can be generalized to other sentences with requirements of the signal's location. The format loss is calculated using MSL (Mean-Squared Loss) between target sequence S and predicted sequence $\mathbf{x}_0$ based on the Diffusion feature representation $\mathbf{x}_t$. The formula for the format loss is

$$\mathcal{L}_\mathrm{format} = MSL(S, \varphi(\mathbf{x}_t)), \quad (10)$$

where $\varphi(\cdot)$ is an MLP network with softmax.

Rhyme Regarding rhyme control, we construct a rhyme categories space, whose representation is a vector in $\mathbb{R}^{6219}$ in Sonnet and $\mathbb{R}^{17}$ in SongCi. The last (l-th) word of each (n-th) line, w_{n_l}, is chosen as the input for the word-level classifier based on BERT (Devlin et al. 2018). This classifier aims to provide the convincing rhyme category of w_{n_l} as its output. Additionally, the tone rule constraints of all words, w_m, in SongCi should also be considered, with a tone categories space of $\mathbb{R}^3$ ("+", "-", "0"). After acquiring the representation of w_{n_l} or w_m, we can readily compute the probability distribution of rhyme or tone rules by applying an MLP network with softmax. Hence, the loss can be formulated as

$$\mathcal{L}_{\text{rhyme}} = \rho_{n_l} \log(BERT(w_{n_l}; \rho_{n_l})), \tag{11}$$

$$\mathcal{L}_{\text{tone}} = \tau_m \log(BERT(w_m; \tau_m)), \tag{12}$$

where ρ_{n_l} and τ_m are rhyme and tone ground truth labels of the target word.

Notably, the Controller is employed throughout the denoising process, rather than solely in the final step, thereby achieving concurrent augmentation of semantics and metrics. Consequently, the format and rhyme of the poetry being generated will progressively enhance amid semantic refinement, avoiding any detriment to the meticulously crafted semantics in the end.

Joint Manipulation

The controllable decoding process is similar to the process of training the Diffusion model. While, the distinction lies in the use of a trained p_θ as the initial denoising distribution and we conduct sampling from $\mathbf{x}_T$ to $\mathbf{x}_0$.

When we combine PoetryDiffusion and Metrical Controller to generate poetry, the feature representation from each step of Diffusion would be adopted to act as the input of Metrical Controller. In step $i-1$, sample $\mathbf{x}_{i-1}$ through p_θ (Eq.5), input $\mathbf{x}_{i-1}$ and condition ρ into the well-trained BERT-based Metrical Controller to obtain the metric loss (Eq.10-12). Then, the Controller transmits the Metrical loss $\mathcal{L}_{\text{M}}$ to PoetryDiffusion, which can be written as

$$\mathcal{L}_{\text{M}} = \lambda_1 \mathcal{L}_{\text{format}} + \lambda_2 \mathcal{L}_{\text{tone}} + \lambda_3 \mathcal{L}_{\text{rhyme}}, \tag{13}$$

where λ_i $(i = 1, 2, 3)$ are the hyperparameters selected by the scale of semantic loss and losses for each metric, ensuring that metrical losses can impact the performance evenly without overshadowing semantic aspects. The term related to $\mathcal{L}_{\text{tone}}$ should be omitted when dealing with sonnets.

The final loss of each step which would affect the denoising process, updating p_θ, is:

$$\mathcal{L} = \mathcal{L}_{\text{S}} + \mathcal{L}_{\text{M}}. \tag{14}$$

Then $\mathbf{x}_i$ would be sampled based on new p_θ.

Consequently, the feature representation would be determined by PoetryDiffusion and its Metrical Controller through the loss $\mathcal{L}$ in each step. This process continues until we sample $\mathbf{x}_0$ and decode it to $\mathbf{w}$ through Eq.3.

Experiments

Dataset and Evaluation

We train our model on two datasets, Sonnet and SongCi. Sonnet consists of 3,355 sonnets collected by (Lau et al. 2018). SongCi comprises 82,724 SongCi's, curated by (Zhang et al. 2019).

To evaluate semantic and metrical performance together, we propose a simple average evaluation score:

$$S^{\text{Sonnet}}_{\text{overall}} = 0.5 \times avg(S_{\text{BLEU}}, S_{\text{ROUGE}}, S_{\text{Distinct}}, 100 - S_{\text{PPL}}) + 0.5 \times avg(S_{\text{format}}, S_{\text{rhyme}}),$$

where the previously settings for BLEU (Papineni et al. 2002), ROUGE (Lin 2004), Distinct (Li et al. 2015) and Perplexity (PPL) are utilized. BLEU and ROUGE are scored by comparing generated poems, which are segmented into lines or subphases, to a reference database of sub-sentences with poetic phrases. PPL is computed by output, using the language-specific BERT. In addition, the tone accuracy in SongCi would be considered:

$$S^{\text{SongCi}}_{\text{overall}} = 0.5 \times avg(S_{\text{BLEU}}, S_{\text{ROUGE}}, S_{\text{Distinct}}, 100 - S_{\text{PPL}}) + 0.5 \times avg(S_{\text{format}}, S_{\text{tone}}, S_{\text{rhyme}}).$$

Moreover, more detailed methods of calculating metrical scores are described as follows.

Format. For Sonnet, the accuracy score (%) is formulated as:

$$S_{\text{format}} = 1 - |N - 14|/14,$$

where *N* stands for the number of lines in generated poetry and 14 is the fixed number of lines for sonnets. For SongCi, the formula is expressed as:

$$S_{\text{format}} = Ts/L,$$

where *Ts* stands for the number of symbols with the true type (ending marks or meaningful words) compared with the original poetry, and *L* is the whole length of the poetry.

Rhyme. For Sonnet, we try to match the rhyme scheme of each generated poetry with 5 types of classic sonnets (Table 3) and report the highest accuracy score. The selected words for evaluation are the last words of each line. For SongCi, since not all the last words of sentences which end with "," or "." satisfy the same type of rhyme, we select the rhyme appearing most in the target original poetry and record their locations for evaluation. Words with the same rhyme on selected locations are regarded as true. The accuracy score (%) of rhyme can be written as:

$$S_{\text{rhyme}} = Tr/Ls,$$

where *Tr* means the number of words with true rhyme within locations selected, and Ls means the number of locations selected. Likewise, the accuracy score (%) of tone can be expressed as:

$$S_{\text{tone}} = Tt/L,$$

where *Tt* means the number of words with the true tone, and *L* remains consistent with the previous statement.

Training Details

This section shows the optimal hyperparameters of our PoetryDiffusion model. The number of decoding or encoding steps T is set to be 2000 steps. In addition, we rescale the diffusion steps into 200 to accumulate the poetry generation process based on DDIM (Song, Meng, and Ermon 2020). The dimension of word embedding is chosen to be 16. The method of organizing batches differs between the two datasets. For Sonnet, pad each piece of poetry to the same length and then concatenate the number of sequences corresponding to batch size. While for SongCi, firstly concatenate all sequences of text and then cut into blocks with appropriate shapes. The number of training iterations is set to 150K. It takes approximately 4 hours to train PoetryDiffusion and Metrical Controller on an NVIDIA A100 GPU monopolized by one job.

Model	Semantics					Metrics			Overall ↑
	BLEU ↑	ROUGE ↑	Distinct ↑	PPL ↓	Avg ↑	Format ↑	Rhyme ↑	Avg ↑	
SeqGAN	26.56	27.61	82.24	32.93	50.87	97.13	35.41	66.27	56.00
MRCG	28.18	23.63	55.14	13.04	48.48	**100.00**	37.59	68.80	55.25
SongNet	25.09	37.78	77.20	12.50	56.89	99.95	29.79	64.87	59.55
GPT3	26.59	32.70	59.01	11.72	51.65	75.63	35.55	55.59	52.96
ChatGPT	30.91	42.78	81.64	9.52	61.45	89.55	50.45	70.00	64.30
Llama2-70B-chat	32.20	41.45	**87.30**	8.72	63.06	96.96	**54.64**	75.80	67.31
PoetryDiffusion(w/o C)	30.18	38.67	86.43	**8.48**	61.70	96.00	23.68	59.84	61.08
PoetryDiffusion	**32.94**	**44.75**	87.15	10.44	**63.60**	**100.00**	52.28	**76.14**	**67.78**

Table 1: Performance on Sonnet obtained by the testing methods. The best results are in bold.

Model	Semantics					Metrics				Overall ↑
	BLEU ↑	ROUGE ↑	Distinct ↑	PPL ↓	Avg ↑	Format ↑	Tone ↑	Rhyme ↑	Avg ↑	
SeqGAN	24.49	15.45	90.06	10.79	55.30	79.58	65.68	53.77	66.27	60.78
MRCG	22.90	14.78	90.06	10.32	54.42	99.35	**93.71**	**98.28**	**97.02**	75.72
SongNet	21.23	14.04	86.82	11.48	52.65	99.42	76.22	80.01	85.22	68.93
GPT3	25.17	16.17	71.88	9.77	50.86	71.80	50.13	29.64	50.52	50.69
ChatGPT	18.29	11.96	91.36	8.79	53.20	84.58	70.23	51.55	68.79	61.00
Llama2-70B-chat	23.79	15.41	90.70	10.28	54.91	80.03	65.11	51.98	65.71	59.53
PoetryDiffusion(w/o C)	25.59	16.86	92.06	9.14	56.35	80.44	64.33	50.94	65.24	60.79
PoetryDiffusion	**28.98**	**17.11**	**92.07**	**8.76**	**57.35**	**99.51**	91.64	95.37	95.51	**76.43**

Table 2: Performance on SongCi obtained by the testing methods. The best results are in bold.

Type	Rhyme Scheme
Shakespearean Sonnets	ABAB CDCD EFEF GG
Spenserian Sonnets	ABAB BCBC CDCD EE
Italian or Petrarchan Sonnets (1)	ABBA ABBA CDC CDC
Italian or Petrarchan Sonnets (2)	ABBA ABBA CDE CDE
Terza Rima Sonnet	ABA BCB CDC DED EE

Table 3: Five types of sonnets and relevant rhyme schemes

Compared Prior Art

We conduct a comparative analysis between our proposed method and established state-of-the-art (SOTA) techniques. To ensure a fair comparison, datasets in two languages are partitioned into train/valid/test in the same way as used in previous work. Details of the realization are listed below.

SeqGAN (Yu et al. 2017) employs a GAN framework, treating the generator as a stochastic policy in reinforcement learning. We utilize the inherent approach in its unaltered form to accomplish the task while substituting our dataset.

MRCG (Zhang et al. 2019) introduces a CVAE framework to generate SongCi while adhering to metric constraints. When generating SongCi, we simply follow its method and settings. And we migrate the model to the Sonnet dataset by changing the Chinese rhyme rules into English and removing the restriction of tone.

SongNet (Li et al. 2020) integrates metrical symbols into continuous representations and combines them with a Transformer-based autoregressive language model. We directly employ the original method to complete the task, with our dataset replaced.

GPT3 (Brown et al. 2020) is fine-tuned on SongCi and Sonnet respectively. For the SongCi dataset, CiPai, which can be regarded as the title of SongCi, acts as the prompt to generate a whole poem. However, due to the lack of titles in the Sonnet dataset, GPT3 receives the first line content in Sonnet as its prompt to generate the rest of the poetry.

ChatGPT (OpenAI 2021) (GPT-3.5-Turbo) is asked to generate a new Sonnet or SongCi under one instruction example in the test set. Prompts used are as Figure 3.

Sonnet:
Please write a sonnet with exactly the same number of lines and rhymes as the example, where '<eos>' is the end flag of each line. Example: [***Example***]

SongCi:
请写一首以“[***CiPai***]”为词牌名的宋词，注意仿照后面例子的句长、平仄和押韵。例子：[***CiPai***] | [***Example***]
*(Please compose a SongCi titled "[**CiPai**]", paying attention to emulate the sentence length, tonal pattern, and rhyme scheme of the provided example. Example: [**CiPai**] | [**Example**])*

Figure 3: Prompts for ChatGPT to generate poetry

Llama2-70B-chat (Touvron et al. 2023) is the open source LLM developed by meta. We utilize all the SongCi and Sonnet training data to finetune Llama2 based on the LoRA technique (Hu et al. 2021). During finetuning, we employ similar format in Figure 3, which use the poetry generation instruction and examples as the input, and then enable Llama2 to generate the poetry conforming to the corresponding format.

Model	Fluency		Coherence		Meaningfulness		Poeticness		Average	
	Sonnet	SongCi	Sonnet	SongCi	Sonnet	SongCi	Sonnet	SongCi	Sonnet	SongCi
SongNet	2.67	3.39	2.72	3.29	2.55	3.40	3.04	3.49	2.75	3.39
ChatGPT	**3.45**	**3.46**	3.35	3.29	**3.60**	**3.52**	3.19	3.20	3.40	3.37
PoetryDiffusion	3.40	3.43	**3.43**	**3.44**	3.32	3.47	**3.62**	**3.52**	**3.43**	**3.46**

Table 4: Mean ratings elicited by humans on generated poetry. Best rates except for Human Poet are in bold.

Example

With all our science we are slaves of fate
What is to come we know not, cannot know
Grief, suffering, death,— all touch us soon or late
The master question, how to meet the blow

Grant me, ye gods, through life a steadfast eye
And then, with equanimity, to die

ChatGPT

Oh, how I wish this wisdom could reside,
In letters radiant, before my gaze,
To face life's trials with calm, upright stride,
Unswayed by praise or calumny's dark haze.

Our science fails to break fate's binding chain,
The future's veiled, its secrets unrevealed,
Yet still we strive against inevitable pain,
To meet the blows with courage unconcealed.

Gods, grant me strength to gaze with steady sight,
And meet life's end with equanimity's light.

PoetryDiffusion

Like Rome, Fiona stands alone, unique
None can compare, none can make her stumble
The gods have decreed, none should be so mystique
Her power and courage, to the heavens humble

In her, all virtues, do truly combine
She is synonymous with citie treasure of kind

Figure 4: Generated Sonnets (excerpt). Color-coded word pairs in the generated sonnets indicate accurate rhymes and a shared rhyme scheme followed by both models, while ChatGPT generate too many lines.

Example
Word: 玉貌香腮天赋与，清姿不假铅华。素芳寻在五陵家，欲知春信息，庚岭一枝斜。
Tone: - + + + + - - 0 + + + - + + 0 - + + - - + + 0 + + + - - 0 + - + + + 0
Rhyme: · · · · · · · · · · · · hua · · · · · · jia · · · · · · · · · · xia

SongNet
Word: 忆昔西园携手去，春风吹遍繁华。而今景物不堪嗟。旧游如昨梦，新恨绕天涯。
Tone: - + + + + - - 0 + + + - + + 0 + + - - - + + 0 - + + + - 0 + - - + + 0
Rhyme: · · · · · · · · · · · · hua · · · · · · jie · · · · · · · · · · ya

ChatGPT
Word: 绿水长流人去后，江山依旧如画。柳岸风摇烟雨下。寻觅芳菲处，桃花一片斜。
Tone: - - + + + - - 0 + + + - + - 0 - - + + + - - 0 + - + + - 0 + + + - + 0
Rhyme: · · · · · · · · · · · · hua · · · · · · xia · · · · · · · · · · xia

Poetry-Diffusion
Word: 唤得桃花三月下，风来别是离愁。不堪谁共水如秋。长空风月里，一片水中幽。
Tone: - + + + + - - 0 + + + - + + 0 - + + - - + + 0 + + + - - 0 + - + + + 0
Rhyme: · · · · · · · · · · · · chou · · · · · · qiu · · · · · · · · · · you

Figure 5: SongCi's generated by models given the same example. Errors in Tone and Rhyme control are indicated using both red font and underlining.

Experimental Performance

Automatic Evaluation With a focus on semantic performance, as shown in Table 1 and Table 2, PoetryDiffusion outperforms other models on both types of poetry, offering strong evidence of the efficacy of our model on semantic enhancement. It demonstrates the superior performance than the auto-regressive model like SongNet and LLMs such as ChatGPT and finetuned Llama2. As for metrical performance, PoetryDiffusion achieves the new SOTA results about average performance in Sonnet, surpassing the baselines with an obvious margin. It must be noted however that PoetryDiffusion's metrics are slightly worse than MRCG in SongCi. Further dataset analysis reveals SongCi demands more rigorous and intricate metrics. The rigid yet impactful final forced-word replacement technique in MRCG contributes to its favorable metrics but compromised semantics. In conclusion, the SOTA overall score proves that our model simultaneously performs well on both semantic and metrical sides compared with all kind of baseline models.

Ablation Study The semantic performance of PoetryDiffusion(w/o Controller) is among the best across baseline models, demonstrating the superiority of the Diffusion model in text generation, supporting our choice of it. Remarkably, when combining the metrical controller, we can obtain further improvement in semantic performance. Metrics capture the essence of a poem's rhythm and sound, and by incorporating metrical controllers, we maintain the authentic emulation of prosody, enhancing semantic expression through structured poetic patterns that shape the composition of the text. Meanwhile, compared to the full PoetryDiffuion, the metrical performance of the one without the Controller has a significant decrease, providing evidence that the controller is vital for augmenting metrical abilities.

Human Evaluation

Criteria We employ the assessment methodology introduced by Zhang and Lapata (2014), where human annotators rate poems using a 1-5 scale across four key dimensions:

- *Fluency*: be grammatical and syntactically well-formed
- *Coherence*: be thematically structured
- *Meaningfulness*: convey a meaningful message
- *Poeticness*: display the features of a poem

Baselines To avoid aesthetic fatigue from manually reviewing extensive poetry, we selected a few models for human evaluation. We chose SongNet, with near-top scores in both datasets, to represent autoregressive models because it's tested on both Chinese and English poetry in the original paper (Li et al. 2020). ChatGPT, known for its generative prowess, was picked to compare our approach with a leading Large Language Model.

Settings All models are provided with the same examples to produce 25 Sonnets and 25 SongCi's. A panel of 5 experts, proficient university students who have majored in English and Chinese literature respectively, assesses the generated poems, and the average of their rating scores is used as the ultimate evaluation score.

Result As shown in Table 4, our PoetryDiffusion surpasses all baseline models in overall average scores. It closely rivals the performance of ChatGPT, with ChatGPT even outperforming PoetryDiffusion in *Fluency* and *Meaningfulness*. This discrepancy can be attributed to ChatGPT having access to a significantly larger training dataset compared to ours, rendering it more adept at generating general conversational text, which places a strong emphasis on fluency and meaningfulness. Conversely, in other dimensions, *Coherence* and *Poeticness*, PoetryDiffusion excels over other models, producing text that exhibits a more distinct poetic style, aligning well with the objectives of our poetry generation task.

Case Study

In Figure 4 and 5, we compare the poetry generated by SongNet, ChatGPT, and our PoetryDiffusion to better illustrate our motivation.

In terms of Sonnet generation, SongNet could not achieve rhyme and ChatGPT exhibited deficiencies in line count control; For SongCi, both SongNet and ChatGPT exhibited slight inaccuracies in Tone and Rhyme control. In comparison, PoetryDiffusion successfully generated Sonnet and SongCi with precise control over format and rhyme[1]. Moreover, PoetryDiffusion demonstrates superior semantic attributes. Most intuitively, it exhibits enhanced diversity, in stark contrast to the consistent repetition of initial words observed in the sonnets generated by the other two models.

Visualization of Stepwise Optimization

Step 1	Impute voyager the start by of life their decke erthrew me rage allur breast well by grasshopper truth ruin both
Step 100	Days neither who on with or were < eos > Doughs 'd engage not on lieth fame < eos >
Step 500	Her on not , return 's on < eos > With not night and living the stones < eos >
Step 1000	How can mad'ning , return in < < eos > But 's night and living shall oppress < eos >
Step 1500	How can golden , return in Lyon < eos > But by night and living by oppress < eos >
Step 2000	How can golden scepter return to Lyon < eos > Day by night and living by oppress < eos >

Figure 6: Generated Sonnet in Different Steps

The denoising process may serve as a mechanism to stepwise predict the masked semantic and metrical information. To evaluate its assumption and reveal how PoetryDiffusion integrates semantics and metrics gradually, we conducted experiments focusing on poetry expression, BLEU, and metrical score stepwise.

[1]The vowels ”ou” and ”iu” in the Chinese phonetic alphabet rhyme with each other.

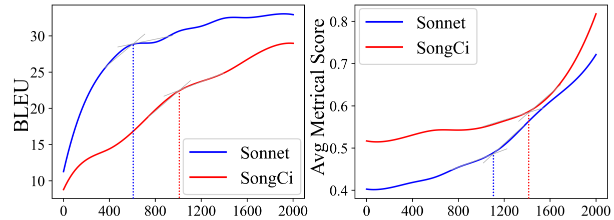

Figure 7: Stepwise evaluation scores. The intersection of the gray dashed lines highlights the point where the curve’s rate of improvement changes.

As seen in Figure 6, the denoising process leads to a clearer topic, improved fluency, and a reduction in hallucinations in the later steps. Furthermore, as depicted by the plotted curve (Figure 7), the BLEU score exhibits a consistent upward trend, reaching its peak at the end of the steps. It is noteworthy that this upward trend is discernible at the onset and subsequently moderating in the first half of the steps. In contrast, The metrical score initially rises slowly, then accelerates in the second half.

These findings suggest that the proposed model establishes thematic semantics first, and it is only as the theme becomes relatively distinct that the influence of metrical control becomes more pronounced. This process steers the model towards imbuing metrical control into the poetry while upholding its semantic structure. These findings also elucidate the reason our PoetryDiffusion outperforms other generative models in terms of both semantics and metrics.

Discussion

In terms of the generalization of our method in other controllable text generation scenarios, we selected poetry generation for evaluation because its strict structures make it perfect for testing controlled text generation. The main challenges are ensuring the model to meet specific conditions and quantitatively evaluating the method’s controllability. Poetry, with its well-defined rules, meets these challenges better than topic control, which is often too vague. If the model excels in poetry’s stringent conditions, it likely will perform well in more relaxed contexts.

Conclusion

We proposed PoetryDiffusion which optimizes the semantic performance by stepwise denoising masked information in entire sentences and incorporating an exceptional metrical controller. By jointly utilizing these two components to generate poetry, a harmonious blend of semantic expression and syntactic control is achieved. SOTA performances in the automatic evaluation and human evaluation of PoetryDiffusion in two datasets also validate its effectiveness. Moreover, the cases study vividly showcases our model’s superiority, and the visualization of the stepwise optimization process in the Diffusion model uncovers the different modeling phases of semantic features and metrical information.

References

Benhardt, J.; Hase, P.; Zhu, L.; and Rudin, C. 2018. Shall i compare thee to a machine-written sonnet? an approach to algorithmic sonnet generation. *arXiv preprint arXiv:1811.05067*.

Betti, F.; Ramponi, G.; and Piccardi, M. 2020. Controlled Text Generation with Adversarial Learning. In *Proceedings of the 13th International Conference on Natural Language Generation*, 29–34. Dublin, Ireland: Association for Computational Linguistics.

Bond-Taylor, S.; Leach, A.; Long, Y.; and Willcocks, C. G. 2022. Deep Generative Modelling: A Comparative Review of VAEs, GANs, Normalizing Flows, Energy-Based and Autoregressive Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11): 7327–7347.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Che, T.; Li, Y.; Zhang, R.; Hjelm, R. D.; Li, W.; Song, Y.; and Bengio, Y. 2017. Maximum-likelihood augmented discrete generative adversarial networks (2017). *arXiv preprint arXiv:1702.07983*.

Chen, H.; Yi, X.; Sun, M.; Li, W.; Yang, C.; and Guo, Z. 2019. Sentiment-Controllable Chinese Poetry Generation. In *IJCAI*, 4925–4931.

Chen, J.; and Yang, D. 2023. Controllable Conversation Generation with Conversation Structures via Diffusion Models. In *Findings of the Association for Computational Linguistics: ACL 2023*, 7238–7251. Toronto, Canada: Association for Computational Linguistics.

Deng, L.; Wang, J.; Liang, H.; Chen, H.; Xie, Z.; Zhuang, B.; Wang, S.; and Xiao, J. 2020. An iterative polishing framework based on quality aware masked language model for Chinese poetry generation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 7643–7650.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ghazvininejad, M.; Shi, X.; Choi, Y.; and Knight, K. 2016. Generating topical poetry. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1183–1191.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2020. Generative adversarial networks. *Communications of the ACM*, 63(11): 139–144.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Hu, Z.; Yang, Z.; Liang, X.; Salakhutdinov, R.; and Xing, E. P. 2017. Toward controlled generation of text. In *International conference on machine learning*, 1587–1596. PMLR.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Lau, J. H.; Cohn, T.; Baldwin, T.; Brooke, J.; and Hammond, A. 2018. Deep-speare: A joint neural model of poetic language, meter and rhyme. *arXiv preprint arXiv:1807.03491*.

Li, J.; Galley, M.; Brockett, C.; Gao, J.; and Dolan, B. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

Li, P.; Zhang, H.; Liu, X.; and Shi, S. 2020. Rigid formats controlled text generation. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, 742–751.

Li, X. L.; Thickstun, J.; Gulrajani, I.; Liang, P.; and Hashimoto, T. B. 2022. Diffusion-LM Improves Controllable Text Generation. *arXiv preprint arXiv:2205.14217*.

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

Lin, K.; Li, D.; He, X.; Zhang, Z.; and Sun, M.-T. 2017. Adversarial ranking for language generation. *Advances in neural information processing systems*, 30.

OpenAI. 2021. GPT-3.5-turbo: Language Model. https://platform.openai.com/docs/guides/chatgpt.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 311–318.

Pu, D.; and Demberg, V. 2023. ChatGPT vs Human-authored Text: Insights into Controllable Text Summarization and Sentence Style Transfer. *arXiv preprint arXiv:2306.07799*.

Sheng, Z.; Song, K.; Tan, X.; Ren, Y.; Ye, W.; Zhang, S.; and Qin, T. 2021. Songmass: Automatic song writing with pre-training and alignment constraint. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 13798–13805.

Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, 2256–2265. PMLR.

Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Tian, Y.; and Peng, N. 2022. Zero-shot Sonnet Generation with Discourse-level Planning and Aesthetics Features. *arXiv preprint arXiv:2205.01821*.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Van de Cruys, T. 2020. Automatic poetry generation from prosaic text. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, 2471–2480.

Yi, X.; Li, R.; Yang, C.; Li, W.; and Sun, M. 2020. Mixpoet: Diverse poetry generation via learning controllable mixed latent space. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 9450–9457.

Yu, L.; Zhang, W.; Wang, J.; and Yu, Y. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Zhang, H.; Liu, X.; and Zhang, J. 2023. SummIt: Iterative Text Summarization via ChatGPT. *arXiv preprint arXiv:2305.14835*.

Zhang, H.; and Song, D. 2022. DisCup: Discriminator cooperative unlikelihood prompt-tuning for controllable text generation. *arXiv preprint arXiv:2210.09551*.

Zhang, R.; Liu, X.; Chen, X.; Hu, Z.; Xu, Z.; and Mao, Y. 2019. Generating Chinese CI with designated metrical structure. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 7459–7467.

Zhang, X.; and Lapata, M. 2014. Chinese poetry generation with recurrent neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 670–680.

Zhou, W.; Jiang, Y. E.; Wilcox, E.; Cotterell, R.; and Sachan, M. 2023. Controlled text generation with natural language instructions. *arXiv preprint arXiv:2304.14293*.