Instructions for *ACL Proceedings

Anonymous ACL submission

Abstract

Attribute-based Controlled Text Generation (CTG) aims to generate texts that contains desirable attributes. Previous work have demonstrated remarkable language generation capabilities, yet they often suffer performance degradation when the length of generations increases. To tackle this challenge, we propose Reinforced-Decoding, a novel lightweight decoding framework for CTG, whose main idea is strategically enhancing the controllability of prefixes on target attributes to construct better attribute distributions. Specifically, We train prefixes by prefix-tuning to obtain Class-conditional language models'(CC-LMs) next-token distributions. Then We leverage a reinforcement learning approach to explore the optimal policy which decides whether to insert prefixs to enhance their influence towards CC-LMs' next-token distribution, and reconstruct attribute distributions at each time step to guide LM to generate texts with desired attributes, effectively mitigating the issue of degrading performance when the length of generations increases. Extensive experiments on a range of CTG tasks demonstrate that Reinforced-Decoding outperforms existing strong baselines with improvements of 1%-4% in Acc and maintains high fluency across a wide range of length settings.

1 Introduction

Controllable Text Generation (CTG) aims to generate text that conforms to predefined constraints. These constraints can encompass a wide range of factors, from broad semantic features like sentiment, topic relevance, and toxicity control(Gu et al., 2023; Kangaslahti and Alvarez-Melis, 2024; Shi et al., 2024). To more specific content requirements, such as the inclusion of particular concepts or key elements like style, kyewords and length (Ashok and Poczos, 2024; Zhou et al., 2023; Sun et al., 2023). Recent advancements in neural text generation, driven by pre-trained language models(PLMs), have achieved unprecedented text quality. However, attribute-based controllable text generation (Zhang et al., 2024a) — involving attributes like sentiment, topics, and detoxification — continues to pose significant challenges. While some CTG method (Zhong et al., 2023; Feng et al., 2024) have demonstrated considerable success in achieving attribute control,it remains challenging to control the generated text to simultaneously satisfy certain attributes and maintain a reasonable coherence and diversity in long text generation.This capability plays a pivotal role in various real-world applications, including writing support and imaginative story creation.

Current approaches for CTG can be categorized into three paradigms based on their intervention mechanisms (Zhang et al., 2023): The first one involves retraining the entire parameters of PLMs to achieve attribute control (Keskar et al., 2019; Wang et al., 2021). While these methods can achieve impressive control, they suffer from high computational costs as the scale of PLMs grows and becomes increasingly impractical to develop and deploy a separate model for each attribute. The second approach focuses on prefix/prompt tuning to control PLMs generation. Techniques such as Attribute Alignment (Yu et al., 2021), Contrastive Prefixes Qian et al. (2022) and Tailor (Yang et al., 2023) optimize lightweight prefixes to steer generation. Though efficient, these methods exhibit limited generalization due to overfitting to training corpus patterns.

The third approach is based on adjusting the output probability distribution of the model during the inference. Most methods incorporate an additional well-trained attribute classifier to steer PLMs through gradient backpropagations (Dathathri et al., 2019) or by weighting the output logits(Krause et al., 2021; Yang and Klein, 2021; Liu et al., 2021). (Gu et al., 2023) which leverages normalizing flows

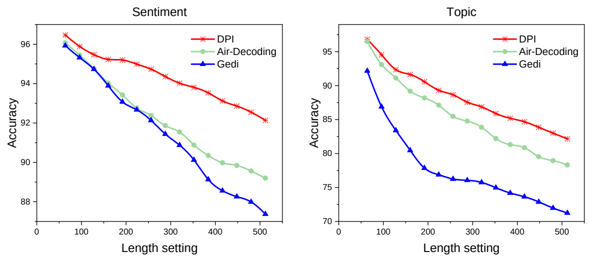

Figure 1: The phenomenon of performance degradation based on sentiment and topic control tasks.The accuracy is an directly proportional metric to attribute relevance.

to transform complex, high-dimensional distributions into tractable Gaussian. (Kangaslahti and Alvarez-Melis, 2024) uses a continuous linear interpolation between fine-tuned models to achieve fine-grained control. This approach typically demonstrates strong attribute relevance, yet suffers from increased inference latency and reduced fluency.

Previous SOTA (Zhong et al., 2023) introduces the phenomenon of Attribute Collapse in decoding-time approaches and leverages a distribution reconstruction method to achieve a more balanced attribute distribution, thereby maintaining stable perplexity. However, these methods suffer from accuracy decrease when the length of generations increases, as shown in 1. Our analysis reveals that this decrease is primarily due to two factors: attention dilution, which stems from the progressive weakening of control signals through standard attention process, and error accumulation, which arises from compounding misalignments in autoregressive generation. As a result, the final generation direction deviates from the desired target.

To overcome the aforementioned limitations,we propose **Reinforced-Decoding**, a lightweight approach that bridges the efficiency of prefix-tuning and the precision of decoding-time control. This method employs dynamic interventions that adapt to the generation context. Specifically, we construct class-conditional language models (CC-LMs) through a prefix-based approach, and model the next-token distribution of the PLM via Bayes' rule. During generation, we insert attribute-specific prefixes into the CC-LMs' keys and values at regular intervals, strengthening attribute control. Furthermore, a policy network autonomously determines whether to insert an additional prefix at the current timestep, optimizing attribute control. This enables more precise steering of the model's output, ensuring that it aligns with the intended direction. Our main contributions are as follows:

- We first identify the phenomenon of performance degradation in CTG tasks and propose a novel lightweight framework named Reinforced-Decoding to mitigate this issue. The framework is trained through a policy gradient approach and autonomously determines when to insert an additional prefix vector to CC-LMs, effectively promotes the attribute alignment during generation across various length settings while maintaining stable perplexity.

- We combine the strengths of prefix-based control and decoding-time control by applying Bayes' rule to create a more efficient and precise generation process. Then Reinforced-Decoding naturally unifies the efficiency of prefix-tuning with the precision of weighted decoding.

- We demonstrate the effectiveness of Reinforced-Decoding across three typical control tasks: Sentiment, Topic, and Detoxification. The evaluation results shows Reinforced-Decoding's superior performance compared to existing baselines both in short and long text generations, advancing the previous SOTA results with 1%–4% improvement on accuracy, and effectively mitigates performance degradation typically observed on longer texts.

2 Related Work

Prefix/Prompt Learning. With the emergence of billion-parameter language models, lightweight fine-tuning methods such as prefix-tuning such as prefix-tuning(Li and Liang, 2021) and prompt-tuning (Lester et al., 2021) have gained increasing attention. ConPrefixes(Qian et al., 2022) introduce contrastive prefixes, which account for inter-prefix relationships during training. Discup (Zhang and Song, 2022) integrate an attribute discriminator with unlikelihood training to refine prompt learning but at the cost of training efficiency. Tailor (Yang et al., 2023) deploy a set of plug-and-play attribute-specific soft prompts to guides the generation suffers from limited control strength. PPP (Ajwani et al., 2024) leverages the gradients from an external discriminator model to adjust the prompt parameters, transforming them into control commands that guide the language model's generation. DATG(Liang et al., 2024) introduces dynamic attribute graphs that modulates occurrence of key words, either aligned with or opposed to the target attribute.

Decoding-time The decoding-time methods do not alter the PLMs but adjust the output probability distribution of PLMs during generation, offering fine-grained control without modifying model parameters. PPLM (Dathathri et al., 2019) iteratively updates hidden states using gradients from an attribute classifier, guiding them toward generating text with the specified attributes. On the other hand, FUDGE (Yang and Klein, 2021) directly uses an attribute classifier to compute the relevance of the next token's attributes and reweights the output probabilities of PLMs. (Zhang et al., 2024b) proposed a Residual Memory Transformer that performs late fusion with a frozen PLMs, enabling non-invasive steering of the generation process.(Yu et al., 2024) alters the output context throughout the generation process of a base language model. The FreeCtrl (Feng et al., 2024) employ real-time intervention by analyzing the sensitivity of feedforward layer vectors in PLMs, dynamically adjusting their weights to steer generation trajectories. (Dekoninck et al., 2024) propsed a model arithmetic that express prior CTG techniques as simple formulas. Energy-based approaches (Mireshghallah et al., 2022; Son and Lee, 2024) utilizes a set of blackbox expert models and combine their energy values to enforce desired property such as fluency and attribute alignment. While this method may suffers from weaker controllability or inefficient inference. Similar to (Liu et al., 2021), Proxy-tuning (Liu et al., 2024) uses two extral small tuned experts to guide PLMs' generation. However, this method may not be well-suited for multi-category attribute-controlled generation due to it requires exponential model variants. GeDi (Krause et al., 2021) uses class-conditional language models as generative discriminators to steer text generation.Despite achieving impressive attribute alignment, its fluency degrades significantly. To solve this issue, (Zhong et al., 2023) introduces a distribution reconstruction method to achieve a more balanced attribute distribution.

Reinforcement Learning RL was first proposed in the context of language generation as an auxiliary algorithm to mitigate exposure bias in the teacherforcing training of sequences.(Kim et al., 2022) utilize a Actor-Critic framework to adjust the PLM's output distributions. (Lu et al., 2022) employ a coarse-grained feedback to train PLM, while (Li et al., 2024) design a fine-grained feedback to provide precise guidance for PLM. In contrast, our method trains only a lightweight policy network that dynamically determines when to insert prefixes, making it more efficient and scalable.

3 Methodology

3.1 Preliminary

Policy gradient is one of the most prominent approaches to solving RL problems, which directly optimize the parameters θ of the policy network, the objective of maximizing the expected return $J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T}\gamma^t r_t\right]$. This optimization is achieved by computing the gradient $\nabla_\theta J(\theta)$, which is shown to be proportional to the expected value of the gradient of the log-probability of the policy and the return G_t (Sutton and Barto, 2018):

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a_t|s_t)\cdot(G_t - b(s_t))\right] \quad (1)$$

where $G_t = \sum_{i=t}^{T}\gamma^{i-t}r_i$. High-return trajectories trigger policy updates that systematically increase the probability of selecting actions proportional to their contribution to the cumulative reward, with gradient ascent directly amplifying the likelihood of high-yielding decisions in subsequent iterations. However, prior research has shown that using G_t alone often leads to high variance. To address this, a state-dependent baseline $b(s_t)$ is subtracted, stabilizing the training process. This baseline does not affect the overall expected value due to its action-independence : $\mathbb{E}_{\pi_\theta}\left[b\nabla_\theta \log \pi_\theta(a_t|s_t)\right] = 0$.

Class-conditional language models (CC-LMs) extends the auto-regressive LMs by incorporating an explicit conditioning signal c, which represents various control attributes, such as a topic label or sentiment score. The probability distribution over the next token is modified as follows:

$$P_\theta(x_{T:N}|x_{1:T-1}, c) = \prod_{i=T}^{N} P(x_i|x_{1:i-1}, c) \quad (2)$$

where c acts as a conditioning variable guiding text generation. Models such as CTRL (Keskar et al., 2019) use predefined control codes as explicit inputs to enforce generation constraints. However, these approaches typically require large-scale pretraining or fine-tuning, which can be computationally expensive and less flexible.

To enable controlled text generation without retraining the entire model, we employ prefix-based method (Li and Liang, 2021) to obtain CC-LMs. Specifically, for a given control attribute a, we learn an attribute-specific prefix. These prefixes steer the model towards generating text with the desired sentiment while keeping the base model parameters frozen.

3.2 Decoding-Based Distribution Adjustment

Previous studies have shown that stronger control can be achieved through weighted decoding strategies. Inspired by this, we introduce a decoding-based distribution adjustment method to enhance control strength while maintaining prefix-tuning's efficiency.

Given a base GPT-2 model with next token distribution and prefix-based conditional distributions, we perform the decoding-time adjustment as follows (Yang and Klein, 2021):

$$P_\theta(x_t|x_{1:t-1}, a) \propto P_\theta(x_t|x_{1:t-1}) \cdot P(a|x_{1:t})^\omega \tag{3}$$

where $P(a|x_{1:t})$ is the probability that $x_{1:t}$ belongs to the desired attribute a, which is provided by an attribute classifier, and ω is a scaling factor that controls the influence of the attribute adjustment.

Our goal is to model $P(a|x_{1:t})$ in order to obtain the desired distribution $P_\theta(x_t|x_{1:t-1}, a)$. By applying Bayes' rule, we can further decompose $P(a|x_{1:t})$ into the following expression with CC-LMs (Krause et al., 2021):

$$P(a|x_{1:t}) = \frac{P(a)\prod_{i=1}^{t} P_{\phi_a}(x_i|x_{1:i-1}, a)}{\sum_{a' \in A}\prod_{i=1}^{t} P(a')P_{\phi_{a'}}(x_i|x_{1:i-1}, a')} \tag{4}$$

where a represents the desired attribute, and A denotes the set of possible attributes. For instance, in the case of binary sentiment control, $A = \{\text{positive}, \text{negative}\}$. The parameter $\phi_{a'}$ corresponds to the CC-LM $P_{\phi_{a'}}(x_i|x_{1:i-1}, a')$ associated with attribute a'. $P(a)$ and $P(a')$ can be omitted for uniform training data.

During text generation, we only need to compute the output distributions of the CC-LMs. Notably, the probability terms from previous steps, $P_{\phi_{a'}}(x_i|x_{1:i-1}, a')$ for $i = 2, \ldots, t-1$, have already been computed, this allows for efficient computation. To further refine control, we followed (Zhong et al., 2023) and reconstruct the attribute distributions:

$$P_{\phi_{a'}}(x_i|x_{1:i-1}, a') = -\frac{1}{ln(P_{\phi_{a'}}(x_i|x_{1:i-1}, a'))} \tag{5}$$

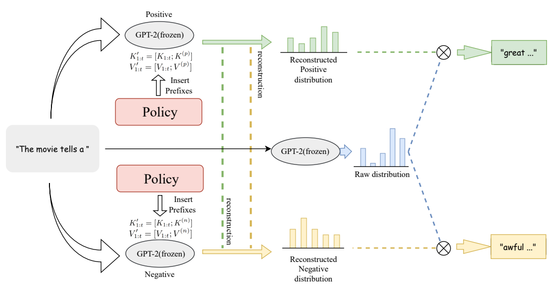

Figure 2: An illustration of Reinforced-Decoding on sentiment task.Feed the generated texts to positive CC-LMs, negative CC-LMs, and GTP-2, the policy make the decision wether to insert prefix to two CC-LMs at timestep t to enhance attribute alignment.

3.3 Reinforced-Decoding

Although the approach has achieved promising results in controlled text generation, However, We observe that the influence of the prefix on the generated sequence gradually diminishes as the sequence length increases.To mitigate this issue, we intuitively re-inserting the prefix at appropriate intervals during the generation process,prepending learnable parameter vectors to both Key and Value matrices in self-attention layers. Formally, for each transformer layer at generation step t, we concatenate the prefix parameters $K^{(p)} \in \mathbb{R}^{lp \times d}$ and $V^{(p)} \in \mathbb{R}^{lp \times d}$ with the contextual Key/Value representations:

$$K'_{1:t} = [K_{1:t}; K^{(p)}], \quad V'_{1:t} = [V_{1:t}; V^{(p)}] \tag{6}$$

where $K^{(p)}$ represents the prefix-generated key, and $K_{1:t-1}$ denotes the sequence of keys from earlier tokens, l_p denotes the prefix length and d represent dimensions. The CC-LMs then compute the self-attention output as follows:

$$\text{Attention}(Q, K', V') = \text{softmax}\left(\frac{QK'^{\top}}{\sqrt{d_k}}\right)V' \tag{7}$$

where Q, K', and V' represent the query, key, and value matrices, respectively, and d_k is the scaling factor.

For example, consider sentiment control, the probability of a positive token under the positive-conditioned distribution $P_{\phi_a}(x_i|x_{1:i-1}, a)$ at time step t increase from 0.1 to 0.15, while its probability under the negative-conditioned distribution $P_{\phi_{\bar{a}}}(x_i|x_{1:i-1}, \bar{a})$ decrease from 0.05 to 0.03. Consequently, the overall probability of the desired attribute $P(a|x_{1:t})$ is reinforced by Eq. 5,rising from

0.1/(0.1+0.05)=0.667 to 0.15/(0.15+0.03)=0.83, thereby ensuring stronger control over the generated sentiment.

Nevertheless, simply inserting the prefixes at fixed intervals may not always be the optimal choice. For instance, some texts that already exhibit strong alignment with the desired attribute do not require additional prefixes insertion. Moreover, arbitrary insertion of the prefix may sacrifice fluency in order to better align with the target attribute. To overcome the aforementioned problems, we propose a lightweight policy network, a small feedforward neural network trained by policy gradient in an offline manner, to determine whether to insert an additional prefixes after a certain number of tokens are generated by the language model. The overall framework is illustrated in Figure 2. Specifically, after each b generation steps are completed at time step t during generation, the policy network considers the generated content so far as the state and computes the action distribution:

$$a_t = Policy(h_{1:t}^{last}) \tag{8}$$

Based on the action distributiona_t, we sample a discrete action m_t to determine whether to insert prefixes into the generation sequence. The action space consists of two possible decisions:

$$m_t \in \{0, 1\}, \quad \text{where} \begin{cases} 0 : \text{Continue generation} \\ 1 : \text{Insert prefix } p_k \end{cases} \tag{9}$$

A full generation process may involve multiple decision points. Since the influence of each decision is uncertain, after the sample s_1 is completed, we compare it with a naturally generated sample s_2 (i.e., one without any prefix insertion) to derive the reward signal. The reward signal derives from their differential performance, combining both perplexity computed by GPT-2$_{\text{Medium}}$, and an attribute score computed by a classifier to evaluate the generated text's attribute alignment. The final reward is defined as follows:

$$r(s) = \alpha \cdot r_{attribute}(s_1, s_2) + \beta \cdot r_{ppl}(s_1, s_2) \tag{10}$$

where $r_{attribute}$ evaluates the difference in attribute scores between the sentences s_1 and s_2, and r_{ppl} measures the difference in perplexity between s_1 and s_2. α and β is hyperparameters that control the relative importance of the two terms in the reward. The policy is trained by maximizing the expected cumulative reward over all generated trajectories:

$$J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T} \gamma^t r(s)\right] \tag{11}$$

By optimizing this reward, the RL model learns to insert the prefix at the most effective points, ensuring both high-quality generation and strong attribute control. Our approach is designed to maintain control over longer sequences, enabling better performance on more complex generation tasks.

4 Experiments

4.1 Evaluation Metric

We test our method on three types of controllable text generation tasks: (1) Sentiment Control, (2) Topic Control, and (3) Detoxification.

Automatic Evaluation. We automatically evaluate the completed sentences from three aspects. (1) **Accuracy** assesses how well the generated examples align with the target attributes. We use RoBERTa$_{\text{large}}$-based (Liu, 2019) attribute classifiers trained on widely used benchmark datasets: IMDB movie reviews (Maas et al., 2011) for sentiment control, and AGNews (Zhang et al., 2015) for topic control to compute the accuracy of generated sentences that contain corresponding attribute.The two classifiers achieve accuracies of 95.52% and 95.18%. For the detoxification task, we utilize the Perspective API[1] to measure the average toxicity for the generated texts.(2) **Fluency** is measured using the perplexity (PPL) scores of the generated sentences, evaluated by GPT-2$_{\text{Base}}$, GPT-2$_{\text{Medium}}$ and GPT-2$_{\text{Large}}$ versions of GPT-2 (Radford et al., 2019), and we report the average perplexity as the final result. (3) **Diversity** is measured by the distinctness (Li et al., 2015) of the generated sentences. Concretely, we compute the fraction of unique 1-grams, 2-grams, and 3-grams of total texts. These metrics are denoted as Dist-1, Dist-2, and Dist-3.

Human Evaluation. Following (Zhong et al., 2023), we conduct human evaluation for sentiment and topic control on three aspects:(1) the text **relevance** of generation text with the desired control attribute; (2) The **fluency** from human perspective; (3) **Topicality** evaluate the consistency between the generated text and the input prompt. Each sentence is rated on a scale from 1 to 5, with higher scores indicating better performance. For each task, we

[1] https://www.perspectiveapi.com/

randomly select 50 sentences for each length and ask three annotators to rate them based on the two metrics. The final human evaluation score is obtained by averaging the collected ratings across all annotators.

4.2 Baselines

We compare the proposed method with a wide range of baselines as follows, all of which are implemented using their official codebases. We retrain these methods on IMDB dataset for sentiment control, on AGnews dataset for topic control, and on Jigsaw Toxic Comment Classification Challenge Dataset for detoxification. For all experiments, We adopt GPT-2_{Medium} as the frozen backbone LM. Other hyperparameters details are described in Appendix A. Learning-free: (1) **FreeCtrl** (Feng et al., 2024), a learning-free approach manipulate the feedforward layers' weight to guide the generation towards the desired directionat at generation process. In our implementation, we remove its *filtering process*, which was originally designed to discard outputs that did not sufficiently align with the target attribute. (2) **DATG** (Liang et al., 2024) employs attribute classifiers to assess PLM-generated texts, constructing dynamic attribute graphs that identify key words aligned with or opposed to target attribute dimensions, we employ their DATG-P method in our experiment. Prefix/Prompt-based: (3) **Tailor** (Yang et al., 2023) represents each attribute as a pre-trained soft prompt and concatenated it with the input, which guides the generation of a frozen PLM to satisfy a pre-specified attribute. Decoding-time: (4)**Air-Decoding** (Zhong et al., 2023) uses a attribute distribution reconstruction method to reconstruct the original attribute distribution to keep fluency of generated texts. (5)**GeDi** (Krause et al., 2021) finetunes external class-conditional LMs to hint a base model's generation. (6) **DExpert** (Liu et al., 2021) finetuns GPT-2 as an expert/anti-expert to manipulate a base model's logits at inference time.

4.3 Experimental Setup

The Reinforced-Decoding approach involves dividing the continuation into blocks of b tokens. Each time b generation steps are completed at time step t, the policy network takes the language model's hidden state as input and decide whether to insert prefiexs.

Sentiment Control. Following previous work (Krause et al., 2021), we first train two CC-LMs with a prefix length of 20 on IMDb movie reviews (Maas et al., 2011),which contains 12.5K positive and 12.5K negative samples. During the evaluation phase, the 15 attribute-unrelated prompts used for the model's generation are identical to those in with PPLM (Dathathri et al., 2019). For each prompt, we generate 50 sentences with varying maximum generation lengths of 64, 128, 192, 256, 384, and 512 tokens.

Topic Control. We experiment with four CC-LMs with a prefix length of 20 on AGnews dataset (Zhang et al., 2015),which consists of four topics: World, Sports, Business and Science, each containing 30K samples. During the evaluation phase, we use 20 prompts identical to those in PPLM (Dathathri et al., 2019). For each prompt, 50 sentences are generated, with the same maximum generation lengths as in sentiment control.

Detoxification. We use the Jigsaw Toxic Comment Classification Challenge Dataset to train CC-LMs. The length of prefix is set to 20. Following previous work,we use 203 prompts collected by (Zhong et al., 2023) from RealToxicityPrompts (Gehman et al., 2020). For each prompt, 20 sentences are generated. Dynamic Prefix Insertion is not used cause dynamic its need a feedback of Perpective API. While accessing the API is feasible, the training duration may extend or interrupted due to delays associated with the API.

4.4 Results and Analysis

Sentiment Control. As shown in Table 1, Reinforced-Decoding significantly outperforms all other baselines in at least one metric, with the exception of generation length 384 and 512. Our accuracy on length of 384 and 512 just fall a little behind FreeCtrl. In contrast, Reinforced-Decoding outperforms FreeCtrl thoroughly across four length(64, 128, 192, 256). while achieving a high accuracy, FreeCtrl tends to generate repetitive content, leading to lower diversity, especially in long (384, 512) texts. Notably, our method maintains a high Dist score even at the maximum length of 512 tokens and keep a low perplexity(33.19), demonstrating its robustness in generating diverse and fluent content over long texts. Tailor and Air-Decoding demonstrates a balanced performance across various metrics; however, Tailor's attribute control remains relatively weak. On the other hand, GeDi attains SOTA diversity but at the cost of severe fluency degradation, making it challenging to produce fluent and coherent text. This issue becomes even

Length	Method	Sentiment			Topic		
		Acc ↑	PPL ↓	Dist-1/2/3 ↑	Acc ↑	PPL ↓	Dist-1/2/3 ↑
	FreeCtrl[2024]	95.80	40.77	0.10 / 0.45 / 0.79	74.43	29.42	0.08 / 0.42 / 0.78
	DATG[2024]	70.76	42.64	0.12 / 0.49 / 0.82	-	-	- / - / -
	Tailor[2023]	80.57	24.40	0.12 / 0.51 / 0.84	76.60	35.82	0.14 / 0.58 / 0.87
64	Air-Decoding[2023]	96.07	25.66	0.13 / 0.56 / 0.80	96.51	30.67	0.08 / 0.47 / 0.78
	GeDi[2021]	95.93	151.40	**0.32 / 0.82 / 0.94**	92.18	150.39	**0.27 / 0.81 / 0.96**
	DExperts[2021]	81.60	**20.06**	0.20 / 0.65 / 0.87	-	-	- / - / -
	Ours	**96.47**	28.09	0.23 / 0.68 / 0.88	**96.83**	**26.71**	0.18 / 0.65 / 0.88
	FreeCtrl	95.21	29.86	0.09 / 0.40 / 0.75	68.10	28.24	0.07 / 0.38 / 0.74
	DATG	68.53	49.27	0.11 / 0.50 / 0.82	-	-	- / - / -
	Tailor	76.37	30.97	0.10 / 0.50 / 0.85	68.60	38.42	0.13 / 0.59 / 0.89
128	Air-Decoding	94.76	27.65	0.13 / 0.52 / 0.81	91.93	31.71	0.11 / 0.50 / 0.79
	GeDi	94.73	195.07	**0.30 / 0.83 / 0.96**	83.40	189.235	**0.24 / 0.82 / 0.97**
	DExperts	83.53	**16.45**	0.16 / 0.58 / 0.82	-	-	- / - / -
	Ours	**95.47**	28.30	0.19 / 0.67 / 0.92	**92.35**	**27.37**	0.16 / 0.64 / 0.90
	FreeCtrl	95.16	25.72	0.07 / 0.36 / 0.72	66.53	28.36	0.06 / 0.34 / 0.71
	DATG	64.37	55.12	0.11 / 0.52 / 0.81	-	-	- / - / -
	Tailor	82.17	38.61	0.10 / 0.49 / 0.86	65.70	41.90	0.13 / 0.58 / 0.90
192	Air-Decoding	93.62	28.83	0.12 / 0.53 / 0.80	88.19	34.63	0.10 / 0.49 / 0.82
	GeDi	93.07	263.52	**0.29 / 0.85 / 0.97**	77.85	266.97	**0.24 / 0.83 / 0.98**
	DExperts	83.27	**16.40**	0.13 / 0.52 / 0.75	-	-	- / - / -
	Ours	**95.40**	29.40	0.18 / 0.66 / 0.92	**90.58**	**28.21**	0.15 / 0.62 / 0.89
	FreeCtrl	94.68	27.36	0.07 / 0.33 / 0.69	63.98	**25.96**	0.06 / 0.32 / 0.68
	DATG	62.29	63.65	0.12 / 0.51 / 0.83	-	-	- / - / -
	Tailor	86.83	45.04	0.10 / 0.50 / 0.86	63.60	38.42	0.12 / 0.57 / 0.89
256	Air-Decoding	92.28	31.16	0.11 / 0.52 / 0.78	86.46	37.27	0.09 / 0.48 / 0.82
	GeDi	92.13	360.91	**0.29 / 0.86 / 0.98**	76.25	370.71	**0.24 / 0.85 / 0.98**
	DExperts	82.13	**18.83**	0.12 / 0.48 / 0.71	-	-	- / - / -
	Ours	**94.73**	29.91	0.17 / 0.64 / 0.91	**88.65**	31.72	0.14 / 0.61 / 0.90
	FreeCtrl	**95.23**	25.43	0.06 / 0.29 / 0.63	62.45	**24.49**	0.05 / 0.28 / 0.63
	DATG	61.78	68.93	0.10 / 0.54 / 0.85	-	-	- / - / -
	Tailor	88.23	55.13	0.11 / 0.52 / 0.87	63.70	53.83	0.12 / 0.56 / 0.89
384	Air-Decoding	90.11	35.03	0.10 / 0.49 / 0.78	83.31	41.76	0.08 / 0.49 / 0.79
	GeDi	89.13	602.45	**0.39 / 0.89 / 0.98**	74.18	599.12	**0.24 / 0.87 / 0.98**
	DExperts	81.89	**15.26**	0.09 / 0.40 / 0.62	-	-	- / - / -
	Ours	93.53	31.36	0.15 / 0.61 / 0.91	**85.20**	33.29	0.11 / 0.56 / 0.89
	FreeCtrl	**95.93**	24.29	0.05 / 0.26 / 0.60	62.95	**23.55**	0.04 / 0.25 / 0.59
	DATG	60.14	74.67	0.09 / 0.55 / 0.84	-	-	- / - / -
	Tailor	86.50	62.14	0.12 / 0.54 / 0.87	77.50	62.94	0.11 / 0.54 / 0.88
512	Air-Decoding	87.89	38.91	0.10 / 0.49 / 0.80	80.32	44.21	0.08 / 0.43 / 0.81
	GeDi	86.73	862.92	**0.28 / 0.9 / 0.99**	71.25	844.49	**0.24 / 0.88 / 0.98**
	DExperts	80.53	**14.61**	0.08 / 0.34 / 0.54	-	-	- / - / -
	Ours	92.13	33.89	0.14 / 0.59 / 0.90	**82.17**	36.44	0.11 / 0.55 / 0.89

Table 1: The main experimental results for sentiment and topic controllable text generation. ↑ indicates that a higher score is better, whereas ↓ signifies the opposite. We bold the **best results**, underline the runner-up.

more pronounced as text length increases: GeDi's PPL rises drastically with longer texts (e.g., from 151.40 to 862.92). Human evaluations from Table 3 confirm our method's superiority in relevance (4.02/5) and Ttopicality (3.76/5), outperforming baselines across both subjective and objective metrics.

Topic Control. In the topic task, Reinforced-Decoding continues to maintain a high diversity across different sequence lengths. Furthermore, it achieves the SOTA accuracy in topic control, outperforming all baselines on all generation length. These results indicate that our approach effectively balances attribute alignment, fluency, and diversity, making it a robust solution for topic-controllable text generation. FreeCtrl, continues to show the weakest diversity in the topic task. Additionally, its accuracy drops significantly, reaching only around

60–70%, far below its performance in sentiment control. Tailor, Air-Decoding, and GeDi exhibit a performance pattern similar to that observed in sentiment task. However, given that topic control is inherently more complex than sentiment control, their accuracy scores decline across the board. We exclude DExperts and DATG from topic evaluations due to their inherent architectural limitations in handling multi-category attributes. The human evaluation results in Table 3 also demonstrate the superiority of our method, mainly in fluency and topicality.

Length	Method	Detoxification		
		Tox. ↓	PPL ↓	Dist-1/2/3 ↑
64	DATG[2024]	38.91	45.17	0.09 / 0.41 / 0.72
	Tailor[2023]	40.50	47.77	0.08 / 0.36 / 0.65
	Air-Decoding[2023]	22.13	48.54	0.12 / 0.48 / 0.73
	GeDi[2021]	20.91	173.60	**0.18 / 0.59 / 0.74**
	DExperts[2021]	25.13	**18.47**	0.11 / 0.44 / 0.67
	Ours	**19.35**	47.12	0.12/ 0.47 / 0.71
128	DATG	37.64	52.49	0.08 / 0.42 / 0.74
	Tailor	40.50	47.77	0.08 / 0.38 / 0.73
	Air-Decoding	24.60	49.24	0.11 / 0.48 / 0.79
	GeDi	21.94	199.50	**0.18 / 0.66 / 0.85**
	DExperts	25.54	**18.59**	0.09 / 0.42 / 0.68
	Ours	**21.66**	48.46	0.11 / 0.49 / 0.78
192	DATG	37.13	57.98	0.09 / 0.44 / 0.77
	Tailor	40.83	53.89	0.09 / 0.39 / 0.75
	Air-Decoding	24.56	51.86	0.10 / 0.48 / 0.80
	GeDi	22.87	249.80	**0.18 / 0.70 / 0.89**
	DExperts	25.69	**19.61**	0.07 / 0.38 / 0.64
	Ours	**21.83**	38.58	0.11 / 0.48 / 0.80
256	DATG	38.31	61.22	0.10 / 0.43 / 0.79
	Tailor	40.89	60.10	0.09 / 0.41 / 0.77
	Air-Decoding	24.60	53.37	0.09 / 0.46 / 0.81
	GeDi	22.41	305.70	**0.18 / 0.72 / 0.90**
	DExperts	25.73	**23.15**	0.07 / 0.35 / 0.61
	Ours	**21.55**	49.8	0.17 / 0.64 / 0.92
384	DATG	39.02	66.81	0.09 / 0.45 / 0.80
	Tailor[2023]	40.85	70.37	0.11 / 0.43 / 0.80
	Air-Decoding	24.29	51.44	0.08 / 0.44 / 0.80
	GeDi	22.88	445.10	**0.19 / 0.77 / 0.93**
	DExperts	25.93	**24.54**	0.05 / 0.29 / 0.53
	Ours	**21.88**	36.80	0.10 / 0.47 / 0.80
512	DATG	38.46	74.26	0.11 / 0.48 / 0.81
	Tailor	41.24	81.18	0.12 / 0.46 / 0.81
	Air-Decoding	24.10	50.92	0.08 / 0.43 / 0.80
	GeDi	22.48	606.90	**0.19 / 0.80 / 0.94**
	DExperts	26.01	**29.66**	0.04 / 0.25 / 0.46
	Ours	**21.74**	53.90	0.09 / 0.44 / 0.80

Table 2: The main experimental results for detoxification. "Tox." measures toxicity (lower is better), PPL represents perplexity (lower is better), and Dist-1/Dist-2/Dist-3 quantify diversity. We bold the **best results**.

Detoxification. The results presented in the Table 3 demonstrate that our approach achieves the lowest toxicity score among all baselines, while maintaining competitive fluency compared to DExperts and Air-Decoding. This suggests effective toxicity reduction without severe fluency degradation. Interestingly, Reinforced-Decoding does not achieve the same level of diversity as it does in sentiment and topic tasks. A possible reason for this discrepancy could be that detoxification involves a more constrained generation process. Despite this, Reinforced-Decoding outperforms baselines in toxicity reduction while maintaining competitive fluency and diversity. This balance highlights the effectiveness of our approach in controlling toxicity without severely compromising fluency and diversity.

Method	Sentiment			Topic		
	Rel.	Flu.	Top.	Rel.	Flu.	Top.
DATG[2024]	2.31	3.37	3.04	-	-	-
FreeCtrl[2024]	3.87	**3.92**	3.69	2.26	3.28	2.85
Tailor[2023]	2.32	3.63	3.02	2.55	3.16	3.20
Air-Decoding[2023]	3.91	3.84	3.72	3.93	**3.73**	3.71
GeDi[2021]	3.41	2.28	3.27	3.79	2.11	2.97
DExperts[2021]	3.46	3.16	3.38	-	-	-
Ours	**4.02**	3.81	**3.76**	**3.97**	3.72	**3.79**

Table 3: The human evaluation for sentiment and topic controllable text generation on 128 length. We bold the **best results**.

5 Conclusions

In this paper, we first identify the phenomenon of performance degradation as the length setting increases, and propose a novel lightweight framework that leverages reinforcement learning to determine whether to insert prefixes during the text generation phase, enabling timely adjustments to the generation trajectory. Specifically, we train prefixes to obtain CC-LMs and utilize a reinforcement learning approach to explore an optimal policy that determine whether to insert prefixes to enhance the influence of prefixes towards CC-LMs' distribution. Then we reconstruct the base LM's distributional to guide the generation towards desired attributes. We conduct experiments on three typical CTG tasks, and the results demonstrate that our approach performs well in long-text generation. Overall, Reinforced-Decoding holds promise for enhancing a variety of prompt-based or prefix-based methods, offering a flexible and adaptive solution for controlling text generation.

Limitations

Reinforced-Decoding combines the efficiency of prefix-tuning with the precision of decoding-time control, utilizing a policy network to guide generation and mitigate the degradation of attribute control in long-text generation. However, as generation continues, the growing length of past keys and values may impact the model's ability to maintain effective control. Future work should investigate methods to manage the length increase of prefixes in attention layers. Moreover, extending this insertion method for fine-grained multi-attribute control remains an areas for future research. These aspects provide avenues for future research.

Acknowledgments

We express our sincere gratitude to the reviewers for their insightful and constructive feedback.

References

Rohan Deepak Ajwani, Zining Zhu, Jonathan Rose, and Frank Rudzicz. 2024. Plug and play with prompts: A prompt tuning approach for controlling text generation. *arXiv preprint arXiv:2404.05143*.

Dhananjay Ashok and Barnabas Poczos. 2024. Controllable text generation in the instruction-tuning era. *arXiv preprint arXiv:2405.01490*.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: A simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164*.

Jasper Dekoninck, Marc Fischer, Luca Beurer-Kellner, and Martin T. Vechev. 2024. Controlled text generation via language model arithmetic. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Zijian Feng, Hanzhang Zhou, Zixiao Zhu, and Kezhi Mao. 2024. Freectrl: Constructing control centers with feedforward layers for learning-free controllable text generation. *arXiv preprint arXiv:2406.09688*.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 3356–3369. Association for Computational Linguistics.

Yuxuan Gu, Xiaocheng Feng, Sicheng Ma, Lingyuan Zhang, Heng Gong, Weihong Zhong, and Bing Qin. 2023. Controllable text generation via probability density estimation in the latent space. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 12590–12616. Association for Computational Linguistics.

Sara Kangaslahti and David Alvarez-Melis. 2024. Continuous language model interpolation for dynamic and controllable text generation. *arXiv preprint arXiv:2404.07117*.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation.

Minbeom Kim, Hwanhee Lee, Kang Min Yoo, Joonsuk Park, Hwaran Lee, and Kyomin Jung. 2022. Critic-guided decoding for controlled text generation. *arXiv preprint arXiv:2212.10938*.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2021. GeDi: Generative discriminator guided sequence generation. pages 4929–4952, Punta Cana, Dominican Republic.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

Wendi Li, Wei Wei, Kaihe Xu, Wenfeng Xie, Dangyang Chen, and Yu Cheng. 2024. Reinforcement learning with token-level feedback for controllable text generation. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1704–1719, Mexico City, Mexico. Association for Computational Linguistics.

Xiang Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *ACL*.

Xun Liang, Hanyu Wang, Shichao Song, Mengting Hu, Xunzhi Wang, Zhiyu Li, Feiyu Xiong, and Bo Tang. 2024. Controlled text generation for large language model with dynamic attribute graphs. *arXiv preprint arXiv:2402.11218*.

Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A Smith. 2024. Tuning language models by proxy. *arXiv preprint arXiv:2401.08565*.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A Smith, and Yejin Choi. 2021. Dexperts: Decoding-time controlled text generation with experts and anti-experts. *arXiv preprint arXiv:2105.03023*.

Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364.

Ximing Lu, Sean Welleck, Jack Hessel, Liwei Jiang, Lianhui Qin, Peter West, Prithviraj Ammanabrolu, and Yejin Choi. 2022. Quark: Controllable text generation with reinforced unlearning. *Advances* *in neural information processing systems*, 35:27591–27609.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

Fatemehsadat Mireshghallah, Kartik Goyal, and Taylor Berg-Kirkpatrick. 2022. Mix and match: Learning-free controllable text generation using energy language models. *arXiv preprint arXiv:2203.13299*.

Jing Qian, Li Dong, Yelong Shen, Furu Wei, and Weizhu Chen. 2022. Controllable natural language generation with contrastive prefixes. *arXiv preprint arXiv:2202.13257*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*.

Chufan Shi, Deng Cai, and Yujiu Yang. 2024. Lifi: lightweight controlled text generation with fine-grained control codes. *arXiv preprint arXiv:2402.06930*.

Hye Ryung Son and Jay-Yoon Lee. 2024. Locate&edit: Energy-based text editing for efficient, flexible, and faithful controlled text generation. *arXiv preprint arXiv:2407.00740*.

Jiao Sun, Yufei Tian, Wangchunshu Zhou, Nan Xu, Qian Hu, Rahul Gupta, John Frederick Wieting, Nanyun Peng, and Xuezhe Ma. 2023. Evaluating large language models on controlled generation tasks. *arXiv preprint arXiv:2310.14542*.

Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*. MIT Press.

Yufei Wang, Ian Wood, Stephen Wan, Mark Dras, and Mark Johnson. 2021. Mention flags (mf): Constraining transformer-based text generators. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 103–113.

Kevin Yang and Dan Klein. 2021. FUDGE: controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 3511–3535. Association for Computational Linguistics.

Kexin Yang, Dayiheng Liu, Wenqiang Lei, Baosong Yang, Mingfeng Xue, Boxing Chen, and Jun Xie. 2023. Tailor: A soft-prompt-based approach to attribute-based controlled text generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 410–427.

Dian Yu, Zhou Yu, and Kenji Sagae. 2021. Attribute alignment: Controlling text generation from pre-trained language models. *arXiv preprint arXiv:2103.11070*.

Sangwon Yu, Changmin Lee, Hojin Lee, and Sungroh Yoon. 2024. Controlled text generation for black-box language models via score-based progressive editor. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 14215–14237. Association for Computational Linguistics.

Hanqing Zhang and Dawei Song. 2022. Discup: Discriminator cooperative unlikelihood prompt-tuning for controllable text generation. *arXiv preprint arXiv:2210.09551*.

Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. 2023. A survey of controllable text generation using transformer-based pre-trained language models. *ACM Computing Surveys*, 56(3):1–37.

Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. 2024a. A survey of controllable text generation using transformer-based pre-trained language models. *ACM Comput. Surv.*, 56(3):64:1–64:37.

Hanqing Zhang, Si Sun, Haiming Wu, and Dawei Song. 2024b. Controllable text generation with residual memory transformer. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1048–1066, Bangkok, Thailand. Association for Computational Linguistics.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Tianqi Zhong, Quan Wang, Jingxuan Han, Yongdong Zhang, and Zhendong Mao. 2023. Air-decoding: Attribute distribution reconstruction for decoding-time controllable text generation. pages 8233–8248, Singapore.

Wangchunshu Zhou, Yuchen Eleanor Jiang, Ethan Wilcox, Ryan Cotterell, and Mrinmaya Sachan. 2023. Controlled text generation with natural language instructions. In *International Conference on Machine Learning*, pages 42602–42613. PMLR.

A Implement Details

We detail the hyperparameters and baselines as follows. For all baseline, we use the GPT-2$_{\text{Medium}}$ as base language model, and trained on IMDB datasets.

Sentiment Conrtol. In our method, we train a transformer encoder to coding two prefixes, each with a prefix length of 20. The training batch size is 4, the weight decay is 0.01,the learning rate is 5e-5, the number of training epochs is 10, and the insertion interval is set to 32. During the generation stage, we use ω=140.0 top-k=200, top-p=1.0.

For FreeCtrl, we set k-values=30, λ=0.20, top-k=25, temperature=100 as released in in their codes.

For Tailor, the length of soft prompt is set to 128, and we set top-k=10, top-p=1.0 as provided in the released codes.

For Air-Decoding, we set ω=140.0, top-k=200, top-p=1.0.

For GeDi, we train the generative discriminator based on GPT-2_{Medium}, we set ω=30.0 , top-p=0.8, τ=0.8 as reported in their implementation.

For Dexperts, we finetune two GPT-2_{Medium} as expert and anti-expert to guide a GPT-2_{Medium} ,set α=3.2, top-k=200, top-p=0.9.

Topic Conrtol. In our method, we train a transformer encoder to coding four prefixes, each with a prefix length of 20. The training batch size is 4, the weight decay is 0.01,the learning rate is 5e-5, and the number of training epochs is 10. During the generation stage, we use ω=60.0 top-k=200, top-p=1.0.

For FreeCtrl, we set k-values=30, λ=0.30, top-k=25, temperature=100 as released in in their codes.

For Tailor, the length of soft prompt is set to 128, and we set top-k=10, top-p=1.0 as provided in the released codes.

For Air-Decoding, we set ω=60.0, top-k=200, top-p=1.0.

For GeDi, we train the generative discriminator based on GPT-2_{Medium}, we set ω=30.0 , top-p=0.8, τ=0.8 .

Detoxification. In our method, we train a transformer encoder to coding four prefixes, each with a prefix length of 20. The training batch size is 4, the weight decay is 0.01,the learning rate is 5e-5, and the number of training epochs is 10. During the generation stage, we use ω=120.0 top-k=200, top-p=1.0.

For Tailor, the length of soft prompt is set to 128, and we set top-k=10, top-p=1.0 as provided in the released codes.

For Air-Decoding, we set ω=120.0, top-k=200, top-p=1.0.

For GeDi, we train the generative discriminator based on GPT-2_{Medium}, we set ω=30.0 , top-p=0.8, τ=0.8.

For Dexperts, we finetune two GPT-2_{Medium} as expert and anti-expert to guide a GPT-2_{Medium} ,set α=2.0, top-k=200, top-p=0.9.

B Ablation Study

To evaluate the impact of different components in our approach, we conduct an ablation study that investigates three variations on sentiment and topic tasks, and we define the settings without prefix insertion, with fixed interval prefix insertion and with policy prefix insertion as w/o, w/ and dynamic insertion. The results are presented in Table 4.

Without Prefix Insertion: This scenario serves as the baseline where no prefix is inserted into the generation process and achieves low accuracy but optimal fluency. **Fixed Interval Insertion**: In this setting, the prefixes is inserted at fixed intervals of 32 tokens during generation. While this approach provides periodic reinforcement of the desired attribute, it does so without considering the actual need for additional guidance. As a result, it slightly reduces fluency, leading to a moderate increase in perplexity. However, the impact is not substantial, as the accuracy and diversity metrics remain comparable to those of the dynamic insertion strategy. This suggests that the prefixes has a limited influence on the model's output. **Dynamic Prefix Insertion**: This approach achieves a similar level of accuracy and diversity as the fixed interval method but with slightly better fluency. The relatively small difference in perplexity between the two methods further supports the hypothesis that the prefix's influence is subtle. This adaptive mechanism ensures that the prefixes is utilized efficiently, providing attribute control with minimal interference in fluency.

Length	Method	Sentiment			Topic		
		Acc ↑	PPL ↓	Dist-1/2/3 ↑	Acc ↑	PPL ↓	Dist-1/2/3 ↑
64	w/o insertion	95.60	**24.70**	0.23 / 0.68 / 0.88	96.15	**25.50**	0.19 / 0.66 / 0.88
	w/ insertion	**96.73**	32.02	0.22 / 0.68 / 0.89	96.78	33.71	0.18 / 0.63 / 0.90
	dynamic insertion	96.47	**28.09**	0.23 / 0.68 / 0.88	**96.83**	**26.71**	0.18 / 0.65 / 0.88
128	w/o insertion	93.30	**25.85**	0.18 / 0.65 / 0.89	91.40	27.97	0.16 / 0.64 / 0.90
	w/ insertion	**95.92**	29.48	0.19 / 0.67 / 0.92	**92.75**	28.25	0.15 / 0.62 / 0.90
	dynamic insertion	95.47	28.30	0.19 / 0.67 / 0.92	92.53	**27.37**	0.16 / 0.64 / 0.90
192	w/o insertion	92.20	**25.42**	0.17 / 0.65 / 0.91	88.70	**27.21**	0.15 / 0.62 / 0.90
	w/ insertion	**95.81**	30.27	0.18 / 0.65 / 0.91	**91.23**	28.54	0.12 / 0.59 / 0.88
	dynamic insertion	95.40	29.40	0.18 / 0.66 / 0.92	90.58	28.21	0.15 / 0.62 / 0.89
256	w/o insertion	91.60	**24.96**	0.16 / 0.63 / 0.91	86.30	**27.72**	0.14 / 0.61 / 0.90
	w/ insertion	**95.00**	30.67	0.16 / 0.63 / 0.92	**89.53**	28.70	0.11 / 0.58 / 0.89
	dynamic insertion	94.73	29.91	0.17 / 0.64 / 0.91	88.65	31.27	0.14 / 0.61 / 0.90
384	w/o insertion	88.80	**24.51**	0.15 / 0.60 / 0.89	83.30	**29.39**	0.13 / 0.58 / 0.90
	w/ insertion	**94.01**	31.03	0.14 / 0.60 / 0.91	**86.03**	32.75	0.10 / 0.55 / 0.87
	dynamic insertion	93.53	31.36	0.15 / 0.61 / 0.91	85.20	33.29	0.11 / 0.56 / 0.89
512	w/o insertion	87.00	**24.64**	0.13 / 0.58 / 0.88	80.70	**29.86**	0.12 / 0.57 / 0.89
	w/ insertion	**92.73**	30.88	0.13 / 0.58 / 0.90	**83.65**	33.58	0.19 / 0.66 / 0.88
	dynamic insertion	92.13	33.89	0.14 / 0.59 / 0.90	82.17	36.44	0.11 / 0.55 / 0.89

Table 4: Ablation study results on sentiment and topic controllable text generation. ↑ indicates that a higher score is better, whereas ↓ signifies the opposite. We bold the **best results**.