FreeCtrl: Constructing Control Centers with Feedforward Layers for Learning-Free Controllable Text Generation

Zijian Feng[1,3], **Hanzhang Zhou**[1,3], **Zixiao Zhu**[1,3], and **Kezhi Mao**[2,3,*]

[1]Institute of Catastrophe Risk Management, Interdisciplinary Graduate Programme, Nanyang Technological University, Singapore
[2]School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
[3]Future Resilient Systems Programme, Singapore-ETH Centre, CREATE Campus, Singapore
{feng0119, hanzhang001, zixiao001}@e.ntu.edu.sg, ekzmao@ntu.edu.sg

Abstract

Controllable text generation (CTG) seeks to craft texts adhering to specific attributes, traditionally employing learning-based techniques such as training, fine-tuning, or prefix-tuning with attribute-specific datasets. These approaches, while effective, demand extensive computational and data resources. In contrast, some proposed learning-free alternatives circumvent learning but often yield inferior results, exemplifying the fundamental machine learning trade-off between computational expense and model efficacy. To overcome these limitations, we propose FreeCtrl, a learning-free approach that dynamically adjusts the weights of selected feedforward neural network (FFN) vectors to steer the outputs of large language models (LLMs). FreeCtrl hinges on the principle that the weights of different FFN vectors influence the likelihood of different tokens appearing in the output. By identifying and adaptively adjusting the weights of attribute-related FFN vectors, FreeCtrl can control the output likelihood of attribute keywords in the generated content. Extensive experiments on single- and multi-attribute control reveal that the learning-free FreeCtrl outperforms other learning-free and learning-based methods, successfully resolving the dilemma between learning costs and model performance[1].

1 Introduction

Controllable text generation (CTG) focuses on directing language models to produce diverse and fluent sentences that adhere to predefined single or multiple attributes such as topics and sentiment (Yang et al., 2023; Gu et al., 2023; Zhang et al., 2023; Zhong et al., 2023). Recent works on CTG can be roughly categorized into two groups based on their dependency on a learning process:

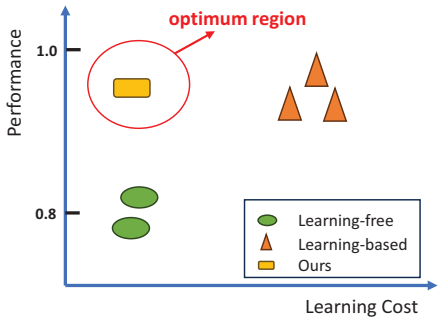

Figure 1: Trade-off between learning cost and performance for CTG. Learning-based methods excel in delivering superb results but demand significant training resources. Conversely, learning-free methods are more resource-efficient but tend to yield inferior performance. Numerical performance details are available in §5.

learning-based methods and learning-free methods (Mireshghallah et al., 2022).

Learning-based methods usually involve training (Yang and Klein, 2021; Krause et al., 2021; Lin and Riedl, 2021), fine-tuning (Ficler and Goldberg, 2017; Keskar et al., 2019; Wang et al., 2021), or prefix-tuning (Qian et al., 2022; Zhang and Song, 2022; Gu et al., 2022, 2023; Yang et al., 2023; Zhong et al., 2023) language models or discriminators based on attribute-specific data. Despite their effectiveness, these approaches come with high computational costs for training and a dependency on vast, attribute-specific datasets, posing challenges for deployment in environments with limited data or computational capacity.

Only a few existing methods are learning-free, avoiding the need for training. For instance, K2T (Pascual et al., 2021) employs attribute-focused keywords to influence token output probability during generation. Another method, Mix&Match (Mireshghallah et al., 2022), integrates diverse black-box experts as a probabilistic energy model to steer large language model (LLM) outputs. These learning-free methods, despite bypassing the

* Corresponding author
[1]Code is available at https://github.com/zijian678/FreeCtrl

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 7627–7640

training process, tend to fall short in performance compared to advanced learning-based approaches.

The analysis spotlights the classic dilemma in machine learning between the cost of learning and model performance, as shown in Figure 1. To overcome this obstacle, particularly in avoiding learning expenses for CTG while ensuring high performance, we propose **FreeCtrl**: Constructing Control Centers with Feedforward Layers for Learning-Free Controllable Text Generation. FreeCtrl's central idea is to manipulate FFN vectors[2] to regulate the outputs of LLMs, inspired by a recent finding that the tokens generated by LLMs can be attributed to the weights of vectors in FFN layers (Geva et al., 2022).

Specifically, the key principle is that increasing a single FFN vector's weight alters the output distribution, raising specific tokens' output probability. This strategy enables the targeted enhancement of certain FFN vector weights to raise attribute keywords' output probability, directing LLM generation towards preferred attributes. Our study first examines the possibility of this strategy by pinpointing three key characteristics of FFN vectors: 1) **Convergence**: Increasing the weight of an FFN vector can result in a stable and convergent output distribution in LLMs, thereby elevating the probabilities of specific tokens. 2) **Diversity**: Diverse FFN vectors can increase the output probabilities for most tokens in the LLM vocabulary, covering keywords relevant to general attributes in CTG; 3) **Prompt-Invariance**: the observed effects of convergence and diversity remain consistent across different input prompts. These characteristics suggest FFN vectors can enable stable, diverse controls for LLM outputs, directing sentence generation toward desired attributes.

However, we also identify a major limitation of FFN vectors: the **high-maintenance** challenge, where adjusting their weights for precise control proves difficult. Low weights lack the power to steer LLMs, while high weights compromise output diversity and fluency. To mitigate this, FreeCtrl initially sets up control centers using FFN vectors for various attributes, then navigates LLM output via a cycle of initialization, monitoring, adaptation, and filtering during the generation process. Continuous monitoring ensures that token generation is assessed at each step, allowing for adaptive weight adjustments of the control centers. Ultimately, a score-based filtering mechanism is employed to refine the outputs. Notably, this framework requires no training or attribute-specific data yet surpasses the efficacy of advanced learning-based models. Therefore, FreeCtrl addresses the cost-performance dilemma, situating it at the optimal upper-left corner in Figure 1, denoting learning-free but high performance. Our main contributions are summarized as follows:

1. We conduct a systematic analysis of using FFN vectors for CTG in §3, identifying three key characteristics: convergence, diversity, and prompt-invariance, alongside a notable challenge of high maintenance.

2. We propose FreeCtrl in §4, a learning-free approach that identifies and utilizes FFN vectors governing various attributes to establish control centers, thus enabling precise management of LLM outputs through initialization, monitoring, adaptation, and filtering.

3. Comprehensive experiments in §5 on both single and multi-attribute control demonstrate that FreeCtrl, without incurring any learning costs, outperforms existing learning-free baselines and cutting-edge learning-based models.

2 Related Work

Learning-based Methods Initial research efforts concentrate on adapting language models into attribute-conditional language models, utilizing methods like fine-tuning (Ficler and Goldberg, 2017; Keskar et al., 2019; Wang et al., 2021) and reinforcement learning (Ziegler et al., 2019; Khalifa et al., 2020; Kim et al., 2023). Weighted decoding stands out as another key strategy in CTG. These methods are primarily learning-oriented, involving updates to the model's hidden states based on decoded logits (Dathathri et al., 2019) or training attribute discriminators to modify model output probabilities (Yang and Klein, 2021; Krause et al., 2021; Lin and Riedl, 2021). Amidst the growth of large language models like GPTX (Radford et al., 2019; OpenAI, 2023) and LLaMA2 (Touvron et al., 2023), recent techniques often preserve LLM parameters and utilize lightweight fine-tuning methods such as prefix-tuning (Li and Liang, 2021; Lester et al., 2021) followed by decoding-time control (Qian et al., 2022; Zhang and Song, 2022; Gu et al., 2022, 2023; Yang et al., 2023; Zhong et al.,

[2]FFN vectors refer to the value vectors in the second weight matrix of the FFN layer. More details and definitions can be found in §3.1.

2023). These methods generally necessitate a large volume of attribute-specific data and considerable computing resources for training either prefixes or discriminators.

Learning-free Methods The realm of learning-free approaches, which eschew any training process, is sparsely populated with research. One such example is K2T (Pascual et al., 2021), which shifts the output logit distribution by calculating the semantic similarity between vocabulary words and target attribute keywords. Notably, Mix&Match (Mireshghallah et al., 2022) stands as the pioneer in introducing a "learning-free" control framework. This innovation leverages external black-box scoring experts to evaluate the attributes of generated content, thereby regulating the model outputs. Compared to cutting-edge learning-based methods like PriorControl (Gu et al., 2023), these approaches often fall short in performance.

3 FFN Vectors for CTG

This section first details the theoretical basis for using FFN vectors to control LLM outputs, then examines their characteristics and challenges.

3.1 Theoretical Foundations

In line with previous findings (Sukhbaatar et al., 2015, 2019; Geva et al., 2021, 2022; Feng et al., 2024), the outputs from FFNs can be viewed as linear vector combinations:

$$\begin{aligned}\mathrm{FFN}^{\ell}(\mathbf{x}^{\ell}) &= f(W_K^{\ell}\mathbf{x}^{\ell})W_V^{\ell} \\ &= \sum_{i=1}^{d_m} f(\mathbf{x}^{\ell}\cdot\mathbf{k}_i)\mathbf{v}_i = \sum_{i=1}^{d_m} m_i^{\ell}\mathbf{v}_i\end{aligned}$$

where f is the activation function, W_K^{ℓ} and W_V^{ℓ} are the weight matrices, and $\mathbf{x}^{\ell}$ is the input at layer ℓ. FFN then can be conceptualized as a neural key-value memory system, where the columns in W_K represent the keys and rows in W_V are the values. Given an input vector $\mathbf{x}^{\ell}$, the keys generate the coefficients $\mathbf{m}^{\ell} = f\left(W_K^{\ell}\mathbf{x}^{\ell}\right) \in \mathbb{R}^{d_m}$, which assign weights to the values in W_V.

In other words, within each layer, the **value vectors** denoted as $\mathbf{v}_i$ are extracted from the rows of the secondary weight matrix, W_V. Taking GPT2-medium (Radford et al., 2019) as an example, W_V is dimensioned at 4096×1024. This dimensionality implies the existence of 4096 value vectors, each extending to a 1024-dimensional space within each individual FFN layer. With 24 such layers incorporated within the GPT2-medium, the model encompasses $24 \times 4096 = 98,304$ value vectors in total.

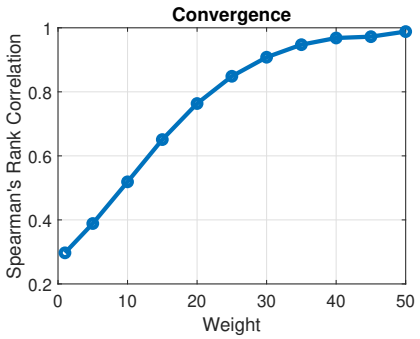

Figure 2: Convergence. As the value vector weight increases, its corresponding output distribution converges.

Prior research (Geva et al., 2022) has verified that the outputs generated by LLMs can be explained by examining the weights associated with the value vectors. The weights of various value vectors directly influence the probabilities of different token outputs. Building on this foundation, our study explores the identification of value vectors controlling different attributes in CTG and the feasibility of manipulating their weights to achieve attribute-specific control.

3.2 Characteristics of Value Vectors

Effective control via value vectors depends on three core requirements: stable impact on output distributions, the ability to manage a wide range of CTG attributes, and consistent behavior under different prompts. We highlight three key properties that affirm the value vectors' utility in achieving trustworthy CTG. Utilizing GPT2 as an example, we iteratively select a single value vector, denoted as $\mathbf{v}_i$, and then incrementally increase its weight, denoted as u. The resultant model output distribution, represented as $\mathbf{p}_i^u \in \mathbb{R}^{|\mathcal{V}|}$, is observed, where $\mathcal{V}$ signifies the GPT2's vocabulary and $|\mathcal{V}|$ is its size.

Convergence While progressively increasing the weight u from 1 to 50, the distribution influenced by each value vector progressively attains a state of stability and constancy, as shown in Figure 2. Specifically, we treat the output distribution controlled by a weight of 50[3] as the ground-truth $\mathbf{p}_i^g$ and calculate Spearman's rank correlation between distributions controlled by different weights and

[3]A weight of 50 is considered exceptionally large according to Geva et al. (2022).

Attribute	Keywords & Positions
POLITICS	politics (20, 1651), government (22, 3127), election (17, 1620), republic (0, 2991), state (19, 84)
SPORTS	sports (14, 1078), champion (21, 4020), football (17, 573), game (23, 1928), coach (17, 1773)

Table 1: Politics and sports-related keywords along with their respective value vectors in GPT2. The position is denoted by (a, b), where a represents the layer number and b is the position of the value vector within that layer.

$\mathbf{p}_i^g$. The mean correlation across all 98,304 resultant distributions is reported in Figure 2. Spearman's rank correlation is used because it directly compares token ranks and mitigates the impact of topk/p sampling and temperature variations. In summary, increasing the weights u establishes stable token ranking and distribution patterns. Such convergence enables the discovery of patterns related to target attributes in CTG and facilitates stable control.

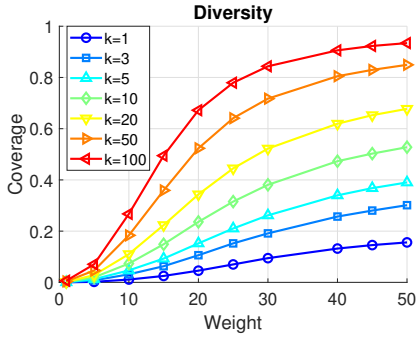

Figure 3: Diversity. The percentage of top-k tokens in the whole vocabulary grows with increasing weights.

Diversity The second question explores whether value vectors can sufficiently control a wide range of tokens representing various attributes in CTG. Our analysis is twofold: first, we assess the percentage of top-k controllable tokens in $\mathbf{p}_i^u$ over the whole covabulary; second, we identify specific value vectors that govern general attributes. Figure 3 shows that with increasing weights, the top-1 tokens controlled by the 98,304 vectors account for up to 20% of the GPT2 vocabulary, and this figure rises to over 80% for top-50 tokens. This demonstrates the vectors' capacity to influence most tokens in the entire vocabulary. To provide further clarity, Table 1 lists several value vectors alongside their corresponding attributes. More attributes and their associated vectors are detailed in Appendix A. These findings confirm the value vectors' ability to control a broad spectrum of attributes in CTG.

Prompt-Invariance To ensure effective control in LLMs, it is critical that value vectors maintain their properties across various input prompts. Our analysis involves feeding GPT2 with 35 different prompts, as provided by Dathathri et al. (2019). The results for various prompts mirror the earlier results, showcasing the characteristics of prompt-invariance.

3.3 Limitation of High Maintenance

While value vectors show promise for CTG, a significant challenge is their **high maintenance** due to the difficulty in setting optimal weights. Low weights fail to effectively direct LLMs towards desired attribute-specific tokens, whereas high weights can reduce output diversity and fluency. Furthermore, the ideal weights for various value vectors can be different. As illustrated in Appendix B, a weight of 1 for politics-related vectors does not steer the model towards political content, but increasing the weight to 5 results in constrained and lower-quality outputs. Conversely, a weight of 5 is effective for sports-related attributes, producing relevant and high-quality generations.

4 Methodology

To maximize the benefits and mitigate the limitations of FFN value vectors, we introduce FreeCtrl: Constructing Control Centers with Feedforward Layers for Learning-Free Controllable Text Generation. FreeCtrl begins by gathering attribute keywords, subsequently constructing a control center for each attribute. It then guides the LLM to produce outputs relevant to the target attribute through a systematic pipeline of initialization, monitoring, adaptation, and filtering. The overall framework is illustrated in Figure 4.

4.1 Attribute Keyword Collection

For a given attribute a_i in the attribute set $\mathbf{A} = \{a_1, \cdots, a_n\}$, we begin by gathering its associated keywords to promote diverse generations. Various external knowledge bases such as WordNet (Miller, 1994), ConceptNet (Speer et al., 2017), RelatedWords[4], and ChatGPT (OpenAI, 2023) can be utilized for this purpose. To reduce the noises

[4] https://relatedwords.org/

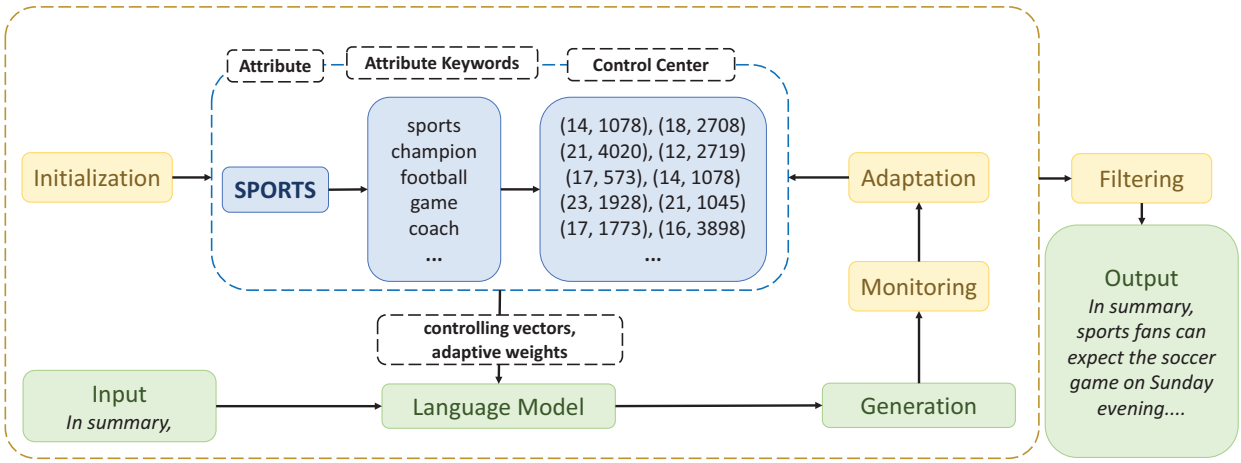

Figure 4: Overview of FreeCtrl. For the target attribute "SPORTS", FreeCtrl initially identifies related keywords and value vectors to establish a control center. Throughout the generation phase, it dynamically adjusts the control center's weights based on real-time output monitoring, ensuring adaptive feedback for subsequent token generation. Finally, a filter is applied to verify compliance with the required attribute. Notably, the position (a, b) specifies the layer number a and the value vector's position b within that layer.

from these external sources, we apply a refinement function:

$$G(z) = r\left(z, a_i\right) \frac{|\mathbf{A}| - 1}{\sum_{a_j \in \mathbf{A}, a_j \neq a_i} r\left(z, a_j\right)} \quad (1)$$

where z represents the collected keyword for attribute a_i, and $r(\cdot)$ indicates the cosine similarity.

This function incorporates ideas from both TF-IDF (Spärck Jones, 2004) and KPT (Hu et al., 2022). It operates on the premise that a suitable attribute keyword should have a relevance score for its corresponding attribute that is higher than the average relevance score for other attributes. Consequently, keywords where $G(z) < 1.0$ are deemed less relevant and are filtered out for refinement. As this is not the primary focus and contribution of our work, we offer only a brief introduction here. For more detailed information, please refer to KPT (Hu et al., 2022).

4.2 Control Center Construction

Based on attribute-specific keywords, we can identify corresponding value vectors to direct the LLM toward outputs focused on these keywords and attributes. Building on §3.2, we iteratively assign a weight of 50 to each vector and examine the output distribution to locate dominant value vectors that control the attribute keywords.

Formally, let $\mathbf{P}_{u_{max}} \in \mathbb{R}^{N \times |\mathcal{V}|}$ represent the softmaxed output distribution across the vocabulary $\mathcal{V}$, modulated by all the N value vectors with a set weight of $u_{max} = 50$. The control effect for a specific keyword z is captured by $\mathbf{P}_{u_{max}}[:, d_{\mathcal{V}}(z)] \in \mathbb{R}^N$, where $d_{\mathcal{V}}(z)$ denotes the index of token z in the vocabulary, and N signifies the total number of value vectors. To pinpoint vectors controlling the keyword z, we choose value vectors with top-k probabilities:

$$\mathbf{c}_z = d_{vec}\{\max_k(\mathbf{P}_{u_{max}}[:, d_{\mathcal{V}}(z)])\} \quad (2)$$

where $d_{vec}(\cdot)$ retrieves the index of value vectors in N. For instance, the positions (21,4020) and (12,2719) in Figure 4 of value vectors correspond to the top-2 output probabilities for the attribute keyword "champion".

Finally, the control center for an attribute a_i is established by aggregating the value vectors of the keywords related to the attribute a_i:

$$\mathbf{C}_{a_i} = \bigcup \mathbf{c}_z, z \in \mathcal{Z}(a_i) \quad (3)$$

where $\mathcal{Z}(a_i)$ denotes the set of keywords for the attribute a_i.

4.3 Single-Attribute Control

To precisely control LLMs through control centers, we adopt a structured process that includes initialization, monitoring, adaptation, and filtering. We constantly monitor the LLM's generation and then adaptively adjust control parameters to steer output toward the desired attribute. A final filtering step verifies the output's compliance with the specified attribute.

Initialization For a specified attribute a_i, we first locate value vectors to construct the control center $\mathbf{C}_{a_i}$.

Monitoring At this stage, each token produced by the LLM in response to a prompt is evaluated for its relevance to the attribute a_i. We construct current output $\mathbf{s}_t^{a_i}$ by integrating the input prompt with tokens generated by the LLM at timestamp t for attribute a_i. The embedding for token s_i in $\mathbf{s}_t^{a_i}$ is derived as $E[s_i] \in \mathbb{R}^{d_e}$, where $E[\cdot]$ is the LLM's embedding matrix with dimension d_e. The attribute embedding $E[\mathcal{Z}(a_i)]$ can be obtained in a similar way. The correlation between current output $\mathbf{s}_t^{a_i}$ and target attribute a_i can be calculated as:

$$\rho_t^{a_i} = \frac{1}{l_t} \sum_{j=1}^{l_t} \max\{r(E[s_j], E[\mathcal{Z}(a_i)]\} \quad (4)$$

where l_t denotes the length of the current sentence and the correlation score $\rho_t^{a_i}$ ranges from 0 to 1. This equation initially computes the maximum cosine similarity between each token in the current output and all target attribute keywords, subsequently calculating the mean value of these correlation scores.

To enhance the correlation with the target attribute while minimizing it with other attributes, we utilize Eq.5, which resembles Eq.1 and is designed to calculate the score of the current output. By continuously tracking the sentence score $\mu_t^{a_i}$, we are able to quickly modify the weights of $\mathbf{C}_{a_i}$, ensuring a coherent and seamless output that aligns with the targeted attribute.

$$\mu_t^{a_i} = \rho_t^{a_i} \frac{|\mathbf{A}| - 1}{\sum_{a_j \in \mathbf{A}, a_j \neq a_i} \rho_t^{a_j}} \quad (5)$$

Adaptation Utilizing $\mu_t^{a_i}$, we can dynamically adjust the weights for timestamp $t+1$, facilitating smooth control and generation. To clarify, the model is required to generate a token at each timestamp. The primary reason for this adaptation is the high maintenance of value vector weights, as discussed in §3.3. The weight for timestamp $t+1$ is determined as Eq.6. Here, μ_ω denotes a preset hyperparameter of the sentence score, λ is a scaling parameter, and $\mu_{s_{l_t}}^{a_i}$ is the last token score obtained by Eqs.4 and 5. We regard $\widehat{\mu}_t^{a_i} = \max(\mu_t^{a_i}, \mu_{s_{l_t}}^{a_i})$ as final score to ensure the fluency.

$$\omega_{t+1}^{a_i} = \begin{cases} \frac{\lambda}{1+\exp[-(\mu_\omega - \widehat{\mu}_t^{a_i}) \cdot l_t]} & \mu_\omega - \widehat{\mu}_t^{a_i} > 0 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

To clarify, a value of $\mu_\omega - \widehat{\mu}_t^{a_i} > 0$ indicates that the score of the current sentence or the last-generated token is below the predefined threshold, and as a result, the weight should be determined by the difference between μ_ω and $\widehat{\mu}_t^{a_i}$. Additionally, the sentence length l_t implies that weights at the outset of generation will be higher than those assigned later. This is grounded in the rationale that initially higher weights are necessary to guide the generation towards the desired direction. Once this direction is established, the LLM tends to continue generating tokens along this path, allowing for reduced weights in later stages to maintain fluency. Conversely, when $\mu_\omega - \widehat{\mu}_t^{a_i} < 0$, it signifies that the sentence or the last-generated token at timestamp t has an adequate score and aligns with the target attribute, eliminating the need for a higher weight. Setting the weight to 0 is a deliberate strategy to prevent the LLM from generating outputs associated with contrary attributes.

Filtering Through continuous monitoring and adaptation, the LLM can be steered to generate outputs focused on target attributes. However, some generations might not meet the μ_ω threshold throughout the generation process, despite maintaining high weights. To filter out such non-compliant generations, a final screening is conducted using Eq.7. Only those sentences that achieve scores in accordance with Eq.7 are considered valid outputs.

$$\mu_T^{a_i} > \mu_\omega \quad (7)$$

where T represents the final timestamp in the generation process, with a token produced at each timestamp.

4.4 Multi-Attribute Control

A significant strength of our method is its seamless adaptability to controlling multiple attributes. Specifically, in the case of multi-attributes $\{a_1, \cdots, a_M\}$, we initially gather the respective control centers for these attributes based on Eqs.2 and 3. We then calculate sentence scores and weights for all M attributes by Eqs.4- 6. At each timestamp, we select the control center with the highest weight for control. Assuming the control

Methods	Sentiment↑ (%)			Topic↑ (%)					Detox.	PPL	Dist.-1/2/3
	Avg.	Neg.	Pos.	Avg.	P.	S.	B.	T.	↑(%)	↓	↑
Learning-based Methods											
PPLM	80.0	97.2	62.7	70.6	74.9	46.5	62.4	98.6	93.2	63.2	31.1/70.9/85.9
GeDi	88.4	96.6	80.2	90.8	84.3	92.6	87.1	99.2	95.4	134.1	47.5/88.9/93.0
Contra. Prefix	89.5	88.4	90.6	86.7	74.5	85.3	93.5	93.6	93.8	37.7	17.3/47.0/71.1
Discrete	92.5	99.1	85.9	90.4	84.5	95.0	84.6	97.5	90.1	46.2	36.9/76.3/87.0
PriorControl	97.1	**99.9**	94.3	95.9	**95.5**	**99.3**	90.2	98.7	90.7	54.3	29.1/70.1/86.9
Learning-free Methods											
Mix&Match	82.8	99.2	63.3	75.6	79.5	57.4	69.6	99.3	96.9	65.2	31.5/74.8/88.8
FreeCtrl (Ours)	**97.7**	**99.9**	**95.4**	**96.5**	93.7	96.1	**96.5**	**99.6**	**97.3**	27.2	20.2/61.3/84.1

Table 2: Automatic results on single-attribute control. Results are reported for the attributes of **Pos**itive, **Neg**ative, **P**olitics, **S**ports, **B**usiness, **T**echnology, and **Detox**ification, in addition to the computed **Aver**ag**e** score. Fluency is measured using perplexity (**PPL**), and diversity (**Dist-1/2/3**) is evaluated by distinct uni-, bi-, and tri-grams.

center $\mathbf{C}_{a_m}$ for the m-th attribute a_m has the maximum weight $\omega_{t+1}^{a_m}$, then the weight for $\mathbf{C}_{a_m}$ is set to $\omega_{t+1}^{a_m}$, while weights for control centers of all other attributes are set to 0. Through this process, different control centers are activated at different timestamps, ultimately yielding an output that integrates multiple attributes after undergoing a filtering process as Eq.7.

5 Experiments

5.1 Experimental Setups

To align with established methods for CTG and ensure fair comparisons, our experimental setups rigorously follow Discrete (Gu et al., 2022) and PriorControl(Gu et al., 2023).

Tasks Our analysis includes three CTG tasks: topic, sentiment, and detoxification, under both single- and multi-attribute control scenarios. Following PPLM (Dathathri et al., 2019), we utilize 35 neutral prompts. The GPT2-medium (Radford et al., 2019) is employed to generate sentence completions. For single-attribute control, GPT2 produces 5 completions for each attribute across all prompts, culminating in a total of 35 prompts × (2+4+1) attribute scenarios × 5 completions=1225 sentences. In the multi-attribute control, the model generates a total of 1,400 sentences, calculated as 35 prompts × (2×4×1) attribute combinations × 5 completions.

Implementation Details Implementation details and hyperparameter settings for our methods and comparative baselines are detailed in Appendix C.

Baselines We compare our FreeCtrl with both learning-based and learning-free methods. The learning-based approaches include (1) **PPLM** (Dathathri et al., 2019), which leverages classifiers' gradients as bias indicators to guide the model's outputs; (2) **GeDi** (Krause et al., 2021), which steers the decoding stage using compact conditional generative models; (3) **Contrastive Prefix** (Qian et al., 2022), incorporating contrastive learning into the prefix strategies to control the LLM generations; (4) **Discrete** (Gu et al., 2022), employing discrete sampling to map the distribution of attributes within latent space to guide the LLM output; and (5) **PriorControl** (Gu et al., 2023), which transfers complex distributions as simple Gausian distributions by using normalizing flow. For learning-free baselines, we compare with the advanced **Mix&Match** (Mireshghallah et al., 2022), which uses external scoring experts to assess generated content attributes.

Evaluation We conduct both automatic and human evaluation. For **automatic evaluation**, we leverage classifiers from prior research (Gu et al., 2022, 2023) to assess topic relevance and sentiment accuracy. Additionally, we utilize the Google Perspective API [5] to evaluate the effectiveness of detoxification. We also report the generation fluency using the mean perplexity and diversity calculated by the mean number of unique n-grams (Li et al., 2016). In **human evaluation**, three annotators evaluate each output based on text quality and the relevance of the specified attribute. These elements are scored on a 1 to 5 scale, where higher scores signify superior performance.

5.2 Single-Attribute Control

Table 2 presents the automatic evaluation results on single-attribute control. When compared to the advanced learning-free approach Mix&Match, our

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

Methods	Average ↑ (%)	Sentiment ↑ (%)	Topic ↑ (%)	Detoxification ↑ (%)	PPL. ↓	Dist. ↑
Learning-based Methods						
PPLM	71.0 ± 21.4	64.7 ± 24.8	63.5 ± 22.7	84.9 ± 6.5	62.6	62
GeDi	81.4 ± 14.7	76.1 ± 17.2	73.8 ± 11.3	94.2 ± 1.9	116.6	75.1
Contra. Prefix	81.3 ± 16.5	74.4 ± 19.6	76.9 ± 16.7	92.7 ± 3.5	31.9	43.3
Discrete	87.4 ± 10.9	86.7 ± 10.5	84.8 ± 14.2	90.7 ± 7.4	28.4	49.5
PriorControl	89.9 ± 8.7	88.0 ± 10.6	87.4 ± 8.5	94.3 ± 3.2	34.7	55.5
Learning-free Methods						
Mix&Match	79.7 ± 21.8	73.5 ± 25.9	69.9 ± 21.1	**95.8 ± 1.9**	63.0	61.8
FreeCtrl (Ours)	**93.4 ± 6.9**	**95.7 ± 8.4**	**89.7 ± 5.8**	94.7 ± 2.2	25.7	53.4

Table 3: Automatic evaluation results on multi-attribute control. The overall and individual average scores for sentiments, topics, and detoxification are reported. ± denotes the standard deviation, which reflects the stability of models among different attribute combinations

Method	Quality↑	Attribute↑	Avg.↑
Single-Attribute Control			
Mix&Match	3.2	3.4	3.3
PriorControl	**4.2**	4.3	**4.3**
FreeCtrl (Ours)	4.1	**4.5**	**4.3**
Multi-Attribute Control			
Mix&Match	3.0	3.1	3.1
PriorControl	**3.9**	4.1	4.0
FreeCtrl (Ours)	3.8	**4.3**	**4.1**

Table 4: Human evaluation results. Quality and Attribute are assessed on a 1 to 5 scale, focusing on text quality and relevance to the specified attribute. The inter-annotator agreement is 0.33 based on Fleiss' Kappa.

proposed method, FreeCtrl, demonstrates superior performance across all attributes, with average improvements of 14.9% in sentiment control, 20.9% in topic control, and 0.4% in detoxification. These results distinctly showcase FreeCtrl's significant advancement over current state-of-the-art (SOTA) learning-free techniques in CTG. Compared to learning-based approaches, FreeCtrl demonstrates competitive or superior performance against the SOTA PriorControl. Specifically, FreeCtrl achieves an average improvement of 0.6% over PriorControl in both sentiment and topic control domains and significantly outpaces PriorControl by a notable margin of 6.6% in detoxification. The results from human evaluation, as shown in Table 4, further reveal a similar trend as automatic evaluations. It is noteworthy that FreeCtrl operates without the need for a learning/training phase or training data, yet it still secures the best results. This underlines FreeCtrl's potential in addressing the challenge of optimizing the balance between cost and performance, as depicted in Figure 1.

5.3 Multi-Attribute Control

Table 3 details the results of multi-attribute control evaluations, where FreeCtrl markedly outperforms both the learning-based SOTA PriorControl and the learning-free SOTA Mix&Match by significant margins. Specifically, FreeCtrl exceeds Mix&Match's performance by 22.2% and PriorControl's by 7.7% in sentiment control, and by 19.8% and 2.3% in topic control, respectively. Furthermore, FreeCtrl enhances the overall average score by 13.7% over Mix&Match and by 3.5% over PriorControl. The human evaluation results presented in Table 4 further highlight the superior performance of our method. These findings underscore FreeCtrl's efficiency in CTG, demonstrating its capability to excel without relying on a training set or undergoing a learning process.

5.4 Ablation Study

To elucidate the impact of each FreeCtrl component, we conduct an ablation study focusing on topic control as follows:

- Random Monitoring (**Ran. Mon**): We replace the monitoring score in Eq. 5 with a random score generator, which produces scores uniformly distributed between 0 and 2.
- Without Adaptation (**w/o Ada**): In this variant, we remove the adaptation component entirely and apply static weights (1.5 and 0.5) to examine how the system performs without the ability to adjust control weights dynamically based on monitoring feedback.
- Without Filtering (**w/o Fil**): This setup tests FreeCtrl's performance without the filtering process.

Method	P. ↑	S. ↑	B. ↑	T. ↑	Avg. ↑	PPL ↓	Dist ↑
FreeCtrl	93.7	96.1	96.5	99.6	96.5	28.9	20.2/61.3/84.1
Ran. Mon	86.1	83.2	84.2	95.5	88.0 (-8.5)	31.1 (+2.2)	16.2/51.7/75.5
w/o Ada (0.5)	74.9	79.2	77.8	92.1	81.0 (-15.5)	15.7 (-13.2)	25.6/59.2/83.7
w/o Ada (1.5)	88.5	90.3	96.4	98.4	93.4 (-3.1)	39.9 (+11.0)	15.2/46.2/70.4
w/o Fil	89.7	88.6	91.7	98.3	92.1 (-4.4)	26.4 (-2.5)	19.9/59.4/81.2

Table 5: The ablation study on topic control using different components of FreeCtrl.

Table 5 summarizes the results, demonstrating that replacing or removing monitoring, adaptation, or filtering components leads to performance drops. Specifically, using a constant weight of 0.5 reduces performance by 15.5% but increases diversity due to reduced control. Conversely, a constant weight of 1.5 significantly lowers both fluency and diversity, indicating that excessive control in LLMs is detrimental. These outcomes highlight the critical role of adaptive control in FreeCtrl for balancing performance, fluency, and diversity in text generation.

5.5 Diversity Analysis

The experimental results presented in Tables 2 and 3 reveal a slight decrease in the diversity of generated content. To mitigate this, we propose increasing the temperature settings of the LLMs during generation. Tables 6 and 7 display the outcomes of applying our FreeCtrl method with an increased temperature setting, compared against other prominent baselines for both single-attribute and multi-attribute control tasks. The results demonstrate that increasing the temperature enables FreeCtrl to achieve the highest diversity scores while also maintaining superior control accuracy and fluency, as evidenced by Perplexity (PPL) scores.

Method	Sentiment	Topic	Detox.	PPL	Dist.
Discrete	92.5	90.4	90.1	46.2	66.7
PriorControl	97.1	95.9	90.7	54.3	62.0
Max&Match	82.8	75.6	96.9	65.2	65.0
FreeCtrl	97.4	96.1	97.1	39.3	71.2

Table 6: Results of increased temperature on single-attribute control.

Method	Sentiment	Topic	Detox.	PPL	Dist.
Discrete	86.7	84.8	90.7	28.4	49.5
PriorControl	88.0	87.4	94.3	34.7	55.5
Max&Match	73.5	69.9	95.8	63.0	61.8
FreeCtrl	95.8	88.9	95.0	34.6	68.1

Table 7: Results of increased temperature on multi-attribute control.

5.6 Further Analysis

Further analysis is provided as follows:

- Hyperparameter analysis: We examine three hyperparameters in FreeCtrl for adjusting the control strength in Appendix D.
- Case study: For a visual illustration of control effects, output examples along with their corresponding control weights are presented in Appendix E.
- Inference speed: Given that monitoring, adaptation, and filtering could add additional time costs, we assess FreeCtrl's inference speed and compare it with other methods in Appendix F.
- Scalability: To verify its scalability, we extend FreeCtrl to a larger language model, LLaMA2-7B (Touvron et al., 2023), with detailed results presented in Appendix G.

6 Conclusions

In this paper, we introduce FreeCtrl, a learning-free approach for controllable text generation (CTG). FreeCtrl employs FFN value vectors to establish control centers tailored to each attribute, enabling dynamic control via a structured process of initialization, monitoring, adaptation, and filtering. Comprehensive experiments demonstrate that FreeCtrl markedly outperforms both learning-based and learning-free methods.

7 Limitations

Our FreeCtrl approach effectively navigates the trade-off between learning expenses and model efficacy. We believe its control mechanism could be further streamlined while maintaining satisfactory outcomes. Additionally, delving deeper into the dynamics of value vectors, including their interactions, can enrich our comprehension and enhance CTG design strategies. These areas offer promising directions for future research.

Acknowledgements

We express our sincere gratitude to the reviewers for their insightful and constructive feedback. We would like to acknowledge that this project is an outcome of the Future Resilient Systems initiative at the Singapore-ETH Centre (SEC). Additionally, we extend our thanks to the National Research Foundation, Prime Minister's Office, Singapore, for their invaluable support through the Campus for Research Excellence and Technological Enterprise (CREATE) programme.

References

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. In *International Conference on Learning Representations*.

Zijian Feng, Hanzhang Zhou, Zixiao Zhu, Junlang Qian, and Kezhi Mao. 2024. Unveiling and manipulating prompt influence in large language models. In *The Twelfth International Conference on Learning Representations*.

Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. In *Proceedings of the Workshop on Stylistic Variation*, pages 94–104, Copenhagen, Denmark. Association for Computational Linguistics.

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*, pages 3356–3369, Online. Association for Computational Linguistics.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yuxuan Gu, Xiaocheng Feng, Sicheng Ma, Lingyuan Zhang, Heng Gong, and Bing Qin. 2022. A distributional lens for multi-aspect controllable text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1023–1043, Abu Dhabi, United Arab Emirates. 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)*, pages 12590–12616, Toronto, Canada. Association for Computational Linguistics.

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, and Maosong Sun. 2022. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2225–2240, Dublin, Ireland. Association for Computational Linguistics.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Muhammad Khalifa, Hady Elsahar, and Marc Dymetman. 2020. A distributional approach to controlled text generation. In *International Conference on Learning Representations*.

Minbeom Kim, Hwanhee Lee, Kang Min Yoo, Joonsuk Park, Hwaran Lee, and Kyomin Jung. 2023. Critic-guided decoding for controlled text generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4598–4612, Toronto, Canada. Association for Computational Linguistics.

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. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4929–4952, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. 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 4582–4597, Online. Association for Computational Linguistics.

Zhiyu Lin and Mark Riedl. 2021. Plug-and-blend: A framework for controllable story generation with blended control codes. In *Proceedings of the Third Workshop on Narrative Understanding*, pages 62–71, Virtual. Association for Computational Linguistics.

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.

George A. Miller. 1994. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Fatemehsadat Mireshghallah, Kartik Goyal, and Taylor Berg-Kirkpatrick. 2022. Mix and match: Learning-free controllable text generationusing energy language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 401–415, Dublin, Ireland. Association for Computational Linguistics.

Finn Årup Nielsen. 2011. A new anew: Evaluation of a word list for sentiment analysis in microblogs. In *Workshop on'Making Sense of Microposts: Big things come in small packages*, pages 93–98.

OpenAI. 2023. Chatgpt. `https://openai.com`. Version used: GPT-3.5.

Damian Pascual, Beni Egressy, Clara Meister, Ryan Cotterell, and Roger Wattenhofer. 2021. A plug-and-play method for controlled text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3973–3997, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jing Qian, Li Dong, Yelong Shen, Furu Wei, and Weizhu Chen. 2022. Controllable natural language generation with contrastive prefixes. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2912–2924, Dublin, Ireland. Association for Computational Linguistics.

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

Karen Spärck Jones. 2004. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 60(5):493–502.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1).

Sainbayar Sukhbaatar, Edouard Grave, Guillaume Lample, Herve Jegou, and Armand Joulin. 2019. Augmenting self-attention with persistent memory. *arXiv preprint arXiv:1907.01470*.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. *Advances in neural information processing systems*, 28.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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, Online. Association for Computational Linguistics.

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*, pages 3511–3535, Online. 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, Toronto, Canada. Association for Computational Linguistics.

Hanqing Zhang and Dawei Song. 2022. DisCup: Discriminator cooperative unlikelihood prompt-tuning for controllable text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3392–3406, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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.

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. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8233–8248, Singapore. Association for Computational Linguistics.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

A Attributes and Corresponding Value Vectors in GPT2

Table 8 details the general attribute keywords in CTG along with their associated value vectors.

B Examples of High Maintenance

We begin by identifying value vectors linked to particular attributes and then vary their weights to assess the impact on LLM outputs, as summarized in Table 9. A weight of 1 for politics-related vectors is insufficient to direct the model's focus towards political themes. Elevating the weight to 5 leads to diminished output diversity and quality. In contrast, applying a weight of 5 to sports-related vectors successfully generates relevant and high-quality content. These results verify the high maintenance of value vectors.

C Implementation Details

Learning-based methods typically require extensive attribute-specific datasets. In line with prior studies, we provide them with the AGNews (Zhang et al., 2015), IMDB (Maas et al., 2011), and Jigsaw Toxic datasets [6] for topics, sentiments, and detoxification, respectively. Our approach is learning-free and obviates the need for training datasets. Following KPT (Hu et al., 2022), we gather and refine topic-attribute keywords using RelatedWords[7] and source sentiment-related keywords for positive and negative attributes from the AFINN (Nielsen, 2011) sentiment lexicon. Our method constructs a control center using positive versus toxic keywords from Gehman et al. (2020) for single-attribute detoxification and filters out toxic words from negative keywords to enable the generation of non-toxic, negative content for multi-attribute control. In this way, each attribute contains 200 to 300 keywords. Our proposed FreeCtrl features three hyperparameters: the number of value vectors k for each attribute keyword in Eq.2, the sentence threshold μ_ω, and the scaling factor λ in Eq.6. Hyperparameter configurations for single- and multi-attribute control experiments are detailed in Table 10.

[6] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/

[7] https://relatedwords.org/

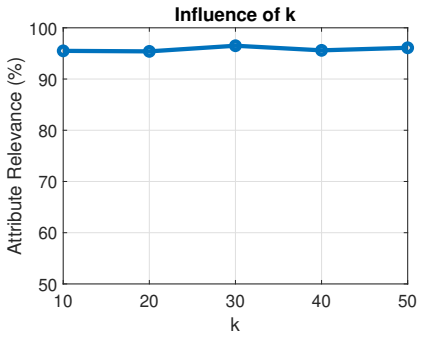

Figure 5: Influence of k on topic control.

D Hyperparameter Analysis

Our proposed FreeCtrl has three hyperparameters: the number of value vectors k for each attribute keyword in Eq.2, the sentence threshold μ_ω, and the scaling factor λ in Eq.6.

The hyperparameter k represents the number of value vectors used to regulate a single attribute keyword. Empirical evidence suggests that $k = 30$ for single-attribute control and $k = 200$ for multi-attribute control yield satisfactory results. The necessity for a greater number of value vectors in multi-attribute control arises from the increased complexity and heightened competition among attributes. Figure 5 outlines the average impact of varying k on topic control. Observations reveal that when k ranges from 10 to 50, the control effects fluctuate slightly between 95.4% and 96.5%, illustrating the robustness of FreeCtrl.

The second hyperparameter, μ_ω, defines the sentence score threshold for control, affecting the final output collection. Figure 6 shows the effect of altering μ_ω between 1.0 and 1.2 on topic control effectiveness. As μ_ω increases from 1.0 to 1.1, there is a gradual improvement in performance. Adjusting μ_ω further, from 1.1 to 1.2, results in stable and satisfactory performance, ranging between 95% and 97%.

The third hyperparameter, λ, acts as a scaling factor for the control weight. Figure 7 indicates that setting λ to 0.5 yields a 94.1% effectiveness in topic control. As λ increases from 1.0 to 2.5, per-

Attribute	Keywords & Positions
POLITICS	politics (20, 1651), government (22, 3127), election (17, 1620), republic (0, 2991), state (19, 84)
SPORTS	sports (14, 1078), champion (21, 4020), football (17, 573), game (23, 1928), coach (17, 1773)
BUSINESS	business (21, 1631), commerce (16, 2225), trade (17, 3938), market (22, 876), finance (22, 2709)
TECHNOLOGY	technology (0, 3260), engineering (0, 3780), science (13, 3160), internet (15, 547), robotics (0, 3260)
POSITIVE	admire (10, 459), great (23, 318), wonderful (12, 3475), good (20, 841), happy (20, 2959)
NEGATIVE	worse (17, 3792), bad (19, 3834), abuse (23, 2534), corrupt (0, 2890), fake (21, 1027)
FOOD	food (21, 3922), rice (14, 423), meat (15, 3011), milk (19, 2113), salt (13, 1992)
AMERICAN	America (19, 684), us (12, 3116), Trump (16, 558), bush (22, 819), American (23, 1417)
ASIAN	Asia (2, 1409), Japan (18, 1794), Korea (7, 2880), Singapore (18, 1794), China (19, 3818)
COMPUTER	laptop (19, 741), hardware (16, 1933), cpu (4, 283), processor (18, 3717), disk (18, 2619)
MILITARY	military (14, 2816), war (6, 989), army (23, 3142), navy (23, 1396), soldier (11, 469)
LEGAL	legal (18, 1137), court (19, 999), justice (18, 4022), legislation (15, 596), rule (21, 634)
RELIGION	religion (18, 3564), faith (21, 3294), god (8, 1710), bless (20, 691), church (14, 3094)

Table 8: Commonly-used attribute keywords and their corresponding positions in GPT2. The position is denoted by (a, b), where a represents the layer number and b signifies the position of the value vector within that layer.

Attribute	Weight	Output
politics	1.0	This essay discusses there is sufficient evidence to support the conclusion that there is ...
politics	3.0	This essay discusses political philosophy, including how philosophy can aid us as ...
politics	5.0	This essay discusses a state of state mind is a state of state...
sports	5.0	This essay discusses soccer in America. It's about the beautiful games that we watch...

Table 9: GPT2 outputs controlled by value vectors of different weights. The input prompt is "This essay discusses".

Hyperparameter	k	μ_ω	λ
Single -Attribute			
Topic	30	1.15	1.5
Sentiment	30	1.15	0.3
Detoxification	30	1.15	0.3
Multi-Attribute			
Topic	200	1.1	0.5
Sentiment	200	1.1	0.5
Detoxification	200	1.1	0.5

Table 10: Hyperparameter setting for single- and multi-attribute control tasks.

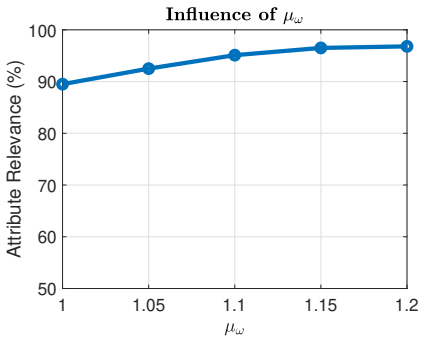

Figure 6: Influence of μ_ω on topic control.

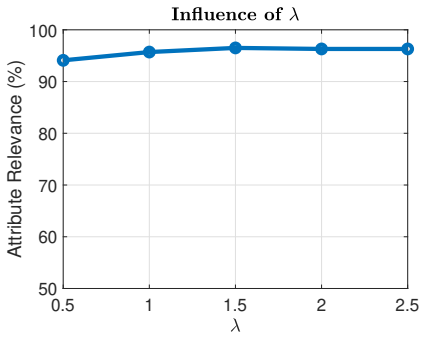

Figure 7: Influence of λ on topic control.

formance improves, ranging between 95.7% and 96.5%. This suggests that a higher λ enhances the control effect. Consistently achieving over 95.7% effectiveness with λ above 1.0 underscores FreeCtrl's efficacy and robustness.

E Case Study

To visually demonstrate the control effects, Figure 8 displays generation results alongside their respective controlling weights. The figure uses red to denote the weights of topic keywords and blue for the weights of sentimental keywords, with the intensity of each color reflecting the magnitude of the weight.

POLITICS	In summary, justice has state, political and social functions. In this respect, the concept...
SPORTS	In summary, soccer is a great sport but we need to be careful with the way we play it...
BUSINESS	In summary, trade is the means of exchange for goods or services and is a business activity...
TECHNOLOGY	In summary, power technology is now available to the military, and we are seeing the benefits...
POSITIVE	In summary, I am happy to report success in my quest for a new home, and I am grateful to...
NEGATIVE	In summary, the government has been guilty of gross negligence in the handling of sensitive...
POLITICS + POSITIVE	In summary, I love our country's great military, and government officials. The world is needed..
POLITICS + NEGATIVE	In summary, the government's failure to investigate and prosecute the perpetrators of the attacks...

Figure 8: Examples of FreeCtrl's control effects and results. The figure employs red to indicate the weights assigned to topic keywords and blue for sentimental keywords, with color saturation corresponding to the weight's intensity.

F Inference Speed

FreeCtrl comprises four main phases: initialization, monitoring, adaptation, and filtering, potentially adding to run-time. Initialization occurs pre-inference, incurring no extra time. Monitoring and adaptation involve evaluating model generation at each timestamp through simple calculations, with negligible added time. Filtering, however, eliminates outputs not meeting certain criteria, leading to wasted generation efforts and additional run-time. We benchmark FreeCtrl's inference time against the SOTA learning-free model, Mix&Match. We calculate FreeCtrl's average inference time using total run-time in §5 for all outputs (valid and invalid) divided by the number of valid outputs (1225+1400), resulting in an average of 9.8 seconds for FreeCtrl compared to 20 seconds for Mix&Match. Thus, FreeCtrl not only significantly enhances performance but also substantially reduces inference time.

G Scalability

We apply FreeCtrl to LLaMA2-7B, aiming to direct the model's outputs on specified topics including politics, sports, business, and technology. Comparative analysis between the original and FreeCtrl-influenced outputs is detailed in Table 11, which demonstrates that FreeCtrl effectively guides LLaMA2's outputs and significantly enhances attribute relevance scores.

Method	P	S	B	T
Original	24.7	12.1	23.9	83.9
FreeCtrl	81.5	83.6	79.2	98.7

Table 11: Results of using FreeCtrl on LLaMA2.