P-Masking: Power Law Masking Improves Multi-attribute Controlled Generation

Anonymous ACL submission

Abstract

We introduce LingGen, a novel approach for controlled text generation that offers precise control over a wide array of linguistic attributes, even as the number of attributes varies. LingGen employs a dynamic P-MASKING strategy, which samples masking rates from a power law distribution during training. This innovative approach enables the model to develop robust representations and adapt its attribute control capabilities across a variable number of attributes, from a single attribute to multiple complex configurations. The P-MASKING technique enhances LingGen's ability to manage different levels of attribute visibility, resulting in superior performance in multi-attribute generation tasks. Our experiments demonstrate that LingGen surpasses current state-of-the-art models in both attribute control accuracy and text fluency, particularly excelling in scenarios with varying attribute demands. Additionally, our ablation studies highlight the effectiveness of P-MASKING and the influence of different base language models on performance. These findings demonstrate LingGen's potential for applications requiring precise and adaptable control over multiple linguistic attributes in text generation.

1 Introduction

The demand for controlled text generation (CTG) has surged across various domains, including content creation, personalized communication, and automated writing. This task involves generating text that adheres to specific constraints, which is crucial for meeting diverse user requirements (Prabhumoye et al., 2020). However, achieving fine-grained control over linguistic features remains a significant challenge (Liu et al., 2023a).

Existing CTG methods have shown promise in controlling high-level attributes like sentiment or topic, but they often struggle with finer-grained linguistic features. Traditional models tend to suffer from inefficiencies and quality degradation when handling multiple controls, especially with complex linguistic attributes (Li et al., 2018; Liu et al., 2023a).

Recent advancements in language model pretraining have highlighted the complementary role of denoising objectives alongside traditional causal language modeling (CLM) (Raffel et al., 2020; Tay et al.; Zeng et al.). Denoising objectives, often referred to as infilling tasks, enable models to learn to "fill in the blanks" within a sequence, thereby enhancing their ability to handle tasks requiring bidirectional context, such as infilling and long-range dependency modeling (Wettig et al., 2023; Clark et al., 2020). This mixture of denoising and CLM has been shown to improve model robustness and sample efficiency, particularly in scenarios where both prefix and suffix contexts are available (Brown et al., 2020; Hoffmann et al., 2022).

In this paper, we introduce LingGen, a novel approach for CTG that leverages a dynamic masking strategy inspired by denoising objectives. Our method, **P-MASKING**, samples masking rates from a power law distribution, allowing the model to learn robust representations and generalize its attribute control capabilities to a variable number of attributes (from 1 to k). This approach addresses the limitations of existing techniques by incorporating the strengths of denoising objectives, enabling improved performance in multi-attribute generation tasks.

Our contributions are as follows: (1) We propose a novel P-MASKING strategy that enhances the flexibility and effectiveness of CTG models by enabling control over a variable number of attributes. (2) We demonstrate the superior performance of LingGen in multi-attribute generation tasks compared to state-of-the-art baselines, particularly excelling in scenarios with varying attribute demands. (3) We provide insights into the impact of different base language models on performance. The rest

of the paper is organized as follows: Section 2 discusses the background and related work, Section 3 details our methodology, and Section 4 presents our experimental results.

2 Background

Controlled text generation has increasingly focused on methods to regulate multiple attributes simultaneously, such as sentiment, tense, formality, or specific keywords (Shen et al., 2017). However, traditional models often lack the flexibility to adapt to new configurations, leading to inefficiencies and quality degradation when handling multiple controls, especially with finer-grained linguistic attributes (Li et al., 2018; Liu et al., 2023a).

2.1 Compositional Text Control

Recent advancements have explored compositional text control in latent space by leveraging compact, differentiable representations. Techniques based on ordinary differential equations (ODEs) and latent space samplers have shown promise in efficiently composing multiple control operations, significantly reducing computational overhead and maintaining high text quality (Liu et al., 2023a; Ding et al., 2023). These methods align with the growing interest in developing models that adapt to dynamic and flexible control inputs across various domains without the need for extensive retraining or costly optimizations (Yang et al., 2023).

2.2 Denoising Objectives

In parallel, research into Masked Language Models (MLMs) has also highlighted the importance of masking strategies to improve model efficiency and performance (Devlin et al., 2019). Conventional wisdom in MLM training has prescribed masking 15% of tokens (Devlin et al., 2019), but recent work challenges this approach, showing that higher masking rates—up to 40% or even 80%—can enhance performance in certain scenarios without sacrificing representational quality (Wettig et al., 2023).

Building on these findings, we propose P-MASKING, a novel masking strategy that samples the masking rate from a power law distribution (Clauset et al., 2009). This approach leverages the flexibility of variable masking rates, allowing the model to better handle a diverse and dynamic set of attribute controls, ranging from 1 to k attributes. By incorporating a power law distribution, P-MASKING addresses limitations found in fixed-rate masking strategies, enabling improved control over multi-attribute text generation (Clark et al., 2020). The power law distribution favors lower masking rates, which introduces less noise and helps the model learn more effectively in typical cases (Newman, 2005). However, the model also learns to handle edge cases with higher masking rates, ensuring robust performance across varying levels of attribute visibility (Wettig et al., 2023).

Our method extends principles from prior work, such as PMI-Masking (Levine et al., 2021), which aimed to move beyond uniform masking strategies, and infilling objectives like those explored in UL2 and GLM-130B (Tay et al.; Zeng et al.; Levine et al., 2021). With P-MASKING, we introduce a principled approach that allows for a smoother and more effective composition of multiple attributes, ensuring better alignment between the generated text and the desired attribute configurations.

2.3 Controlled Text Generation

Controlled Text Generation has become a vital tool in NLP, enabling the creation of text tailored to specific requirements. Works like Shi et al. (2024) introduced fine-grained control codes (LiFi) for sentiment manipulation, while Liu et al. (2023b) proposed BOLT, enabling tunable biases for factual consistency. Pei et al. (2023) further explored prefix-adaptive decoding for controlling text style. However, these methods primarily focus on high-level properties like sentiment, factual accuracy, or style in general.

The integration of denoising objectives in pre-training, as seen in models like UL2 and GLM, has demonstrated the potential for enhancing model capabilities in handling diverse linguistic tasks (Tay et al.; Zeng et al.). These objectives complement traditional CLM by providing models with the ability to process and generate text with both prefix and suffix contexts, a feature particularly beneficial for applications such as code generation and document completion (Chowdhery et al., 2023; Roberts et al., 2023).

Our work focuses on fine-grained control over multiple linguistic attributes, building on the insights from denoising objectives to enhance the flexibility and effectiveness of CTG models.

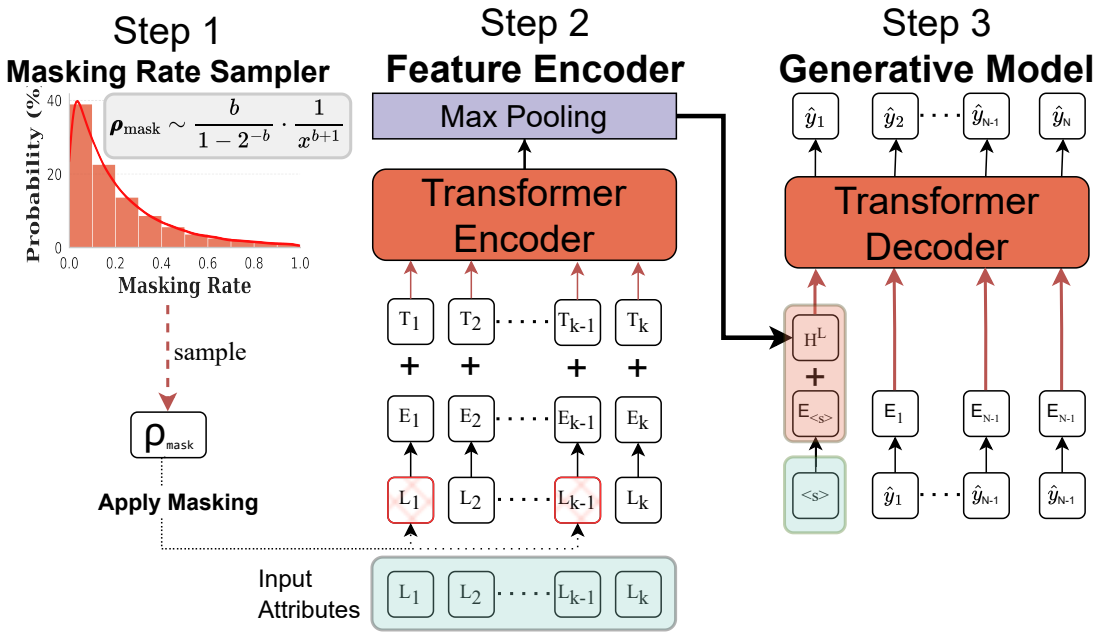

Figure 1: Overview of the LingGen architecture for controlled text generation. 1) **Masking Rate Sampler**: During training, masking rates (ρ_{mask}) are sampled from a truncated Pareto distribution, balancing attribute visibility. This dynamic masking ensures robust learning by varying the number of attributes visible to the model. 2) **Feature Encoder**: The linguistic attributes ($L_1, \ldots, L_K$) are encoded into embeddings ($E_1, \ldots, E_K$) using a linear layer that maps $\mathbb{R}^1$ to $\mathbb{R}^d$, where d is the transformer hidden size. These embeddings are combined with token type embeddings ($T_1, \ldots, T_K$) to generate a global feature representation that feeds into the generative model. 3) **Language Model**: The Transformer Decoder generates text tokens ($\hat{y}_1, \ldots, \hat{y}_n$) conditioned on the encoded attributes. The special start token ($< s >$) combines with the global feature representation to guide generation, enabling fine-grained control over multiple linguistic attributes.

3 Linguistic Generation with LingGen

Given a set of desired linguistic attributes, $\boldsymbol{a} = \{L_1, ..., L_k\}$, where each L_i represents a specific linguistic feature (e.g., sentence length, presence of keywords, number of unique sophisticated words), the task is to generate text that exhibits those attributes. We use 40 attributes, with the specific indices used described in Section A. Let Y be the space of possible generated texts. Our goal is to find a model G that takes the desired attributes $\boldsymbol{a}$ as input and generates a text $\boldsymbol{y} = G(\boldsymbol{a})$ that minimizes a loss function $L(V(\boldsymbol{y}), \boldsymbol{a})$, where $V : Y \to \mathbb{R}^k$ is a function that extracts a fixed-size vector representation of the attributes present in a given text (Hu et al., 2017). This can be expressed as finding $\boldsymbol{y} = \arg\min_{\boldsymbol{y} \in Y} L(V(y), \boldsymbol{a})$. Note that there can be multiple solutions $\boldsymbol{y}$ that minimize this loss. For example, if $\boldsymbol{a}$ specifies a sentence of length 10, there are many possible sentences of length 10 that could be generated. However, as the number of attributes in $\boldsymbol{a}$ increases and the granularity of these attributes becomes finer (e.g., specifying not just sentence length but also specific keywords, syntactic structure, and sentiment), the set of possible solutions shrinks. In the extreme case, with a sufficiently large and specific set of attributes, there may be only one or a very small number of sentences $\boldsymbol{y}$ that satisfy all the constraints (Holtzman et al., 2020).

Instead of using reinforcement learning, which has drawbacks like lower effectiveness compared to supervised learning and reliance on a potentially difficult-to-train attribute discriminator V, we train the model using cross-entropy loss on the predicted token sequence, conditioned on the input attributes. Cross-entropy loss is particularly useful because it aligns with the model's training objective of predicting the next word in a sequence, thus reducing the discrepancy between training and test conditions. This helps mitigate the accumulation of errors during sequence generation, as the model learns to generate text that is both fluent and coherent while conforming to the desired attributes (Ranzato et al., 2016; Bengio et al., 2000). Training on

a large and diverse dataset with a wide variety of attribute combinations allows the model to learn the underlying relationship between attributes and text, enabling it to generate text that is both fluent and coherent while conforming to the desired attributes (Radford et al., 2019). The attribute values themselves are derived using linguistic analysis tools (Lu, 2020, 2012; Lee and Lee, 2023). These tools provide the function $V(\boldsymbol{y})$ that maps generated text $\boldsymbol{y}$ to a vector of attribute values in $\mathbb{R}^k$.

LingGen consists of three main components: a Masking Rate Sampler, a Feature Encoder, and a Language Model (illustrated in Figure 1). These components work together to apply the P-MASKING mechanism and integrate attributes into the language model, allowing for flexible control over a variable number of attributes.

3.1 Attribute Integration

The k linguistic attributes $\boldsymbol{a} = \{L_1, ..., L_k\}$ are encoded into a hidden representation, which is then added element-wise to the embedding of a special Start-Of-Sequence (SOS) token. This injection of attribute information at the beginning of the generation process provides a strong signal to the model about the desired text characteristics. We chose to use the OPT-350M (Zhang et al., 2022) model as the base model for our experiments due to its balance of strong performance and computational efficiency. We also experimented with GPT-2 (Radford et al., 2019) and Pythia-410M (Biderman et al., 2023) to assess the impact of model size on performance.

The Feature Encoder integrates linguistic attributes into the model's latent space. Each linguistic attribute (L_i) is transformed into an embedding (E_i) using a linear layer that maps $\mathbb{R}^1$ to $\mathbb{R}^d$, where d is the transformer hidden size. These embeddings are combined with token type embeddings ($T_1, \ldots, T_K$) to generate a global feature vector, encapsulating the overall attribute information, which is passed to the Language Model. The feature encoder utilizes a novel strategy, **P-MASKING**, where masking rates (ρ_{mask}) are sampled from a truncated Pareto distribution. This balances attribute visibility during training, allowing the model to handle varying levels of visibility and generalize across attributes.

The Language Model , a Transformer Decoder, generates text tokens ($\hat{y}_1, \ldots, \hat{y}_n$) conditioned on the input embeddings and the global attribute feature vector. The global feature vector, along with a special start token ($< s >$), is fed into the decoder, enabling precise control over linguistic attributes. The decoder attends to both attribute representations and input context, ensuring alignment with desired attribute configurations.

3.2 P-MASKING: A Dynamic Masking Strategy

During training, we employ P-MASKING, a dynamic masking strategy. Masking attributes prevents memorization of training data and mitigates spurious correlations. By randomly masking some attributes, the model learns to infer missing information, leading to more robust control and disentangled attribute representations.

P-MASKING samples masking rates from a truncated Pareto distribution (Burroughs and Tebbens, 2001), enabling the model to learn robust representations and generalize its attribute control capabilities to a wider range of attribute visibility levels. For each sample, a masking rate m is sampled, controlling the proportion of attributes masked. This allows the model to learn to control any number of attributes. The probability of masking m percent of the attributes is given by:

$$P(\rho_{\text{mask}} = m) = \frac{b}{1 - 2^{-b}} \frac{1}{m^{b+1}} - 1 \quad (1)$$

for $0 \leq m \leq 1$, where b is a shape parameter. In our experiments, b is tuned such that the distribution yields a masking rate of 30% or lower in over 60% of samples. The sampled masking rate m determines how many attributes are masked. Masked attributes are replaced with a zero vector and excluded from the self-attention (Vaswani et al., 2017).

Advantages over Fixed-Rate Masking: This dynamic strategy offers several advantages. It introduces more randomness during training, forcing robust and generalizable representations. The power law distribution allows a nuanced approach: frequent lower masking rates preserve the learning of accurate attribute representations, while less frequent higher rates force the model to handle missing attributes, crucial for multi-attribute control.

4 Experiments

We evaluate LingGen by comparing it against several state-of-the-art CTG baselines. All baselines

Method	MSE (↓)	Fluency (↑)	Time per token (ms) (↓)	
No Control				
Reference	0.00	72.7	-	-
Vanilla LLM (Zhang et al., 2022)	2.03	57.6	25	(1×)
Decoding-time Control				
PPLM (Dathathri et al., 2020)	5.99	67.7	1515	(61×)
Fudge (Yang and Klein, 2021)	3.24	65.1	112	(5×)
COLD (Qin et al., 2022)	3.99	49.1	3846	(155×)
BOLT (Liu et al., 2023b)	2.59	88.9	114	(5×)
LLama3.1 (Dubey et al., 2024)	2.27	**94.5**	162	(7×)
Mix&Match (Mireshghallah et al., 2022)	1.58	42.5	5882	(237×)
Fine-tuned LLM				
MCTune (LLama-7B) (Nguyen et al., 2024)	2.51	97.3	68	(3×)
MCTune (OPT-350M) (Nguyen et al., 2024)	9.90	76.3	46	(2×)
LingGen (P-MASKING)	**0.90**	83.6	25	(1×)

Table 1: Comparison of model performance across different methods. The table presents the MSE, fluency scores, and time per token for each model. Lower MSE and time per token values indicate better performance, while higher fluency scores are preferred.

(except Llama 3.1) are re-implemented using OPT-350M as a base-model. We consider the following baselines:

5 Experimental Setup

We train LingGen using LoRA with $r = 64$, $alpha = 128$, a batch size of 140, using AdamW optimizer, for 3 epochs. We select the model from the best validation step. We use a max length of 100 tokens, and train on a single A100 GPU.

Next, we re-train MCTune on opt-350m for the same number of tokens and epochs as LingGen. Because MCTune requires In-Context Fine-Tuning (ICFT) with a long prompt, and a context up to 1024 tokens, it takes 216 GPU hours to train, while LingGen takes 18 GPU hours only. Subsequently, MCTune was trained on an HPC using 12 GPUs. We tune the hyper-parameters of all baselines using grid search.

5.1 Baselines

Vanilla LLM Generation (Zhang et al., 2022): This baseline generates text by randomly sampling from the probability distribution of the language model, without any conditioning on the desired attributes. This serves as a basic sanity check to ensure that our model is indeed learning to control for the attributes.

Reference: This baseline uses the reference sentence as the generated text. This baseline provides an upper bound on the performance, as it assumes that the model can perfectly reproduce the reference text.

Mix&Match (Mireshghallah et al., 2022): This baseline interprets controllable generation as sampling from an energy-based model whose energy values are derived from the scores of a masked language model (MLM) filling a mask token and an attribute discriminator.

PPLM (Dathathri et al., 2020): This baseline guides text generation by combining a pre-trained language model with attribute classifiers, allowing control over attributes without retraining the language model.

Fudge (Yang and Klein, 2021): This baseline adjusts the language model's probabilities by adding the likelihood of an attribute discriminator to the language model likelihood.

LLama3.1 (70B) Chat Model (Dubey et al., 2024): This baseline employs the LLama3.1 (70B) chat model, which has been instruction-tuned to follow instructions and complete tasks. We use this as a representative of large language models that have been trained on a massive dataset of text and code.

BOLT (Liu et al., 2023b): This baseline utilizes tunable biases to directly modify the output logits of the language model.

COLD Decoding (Qin et al., 2022): This baseline frames constrained generation as an energy

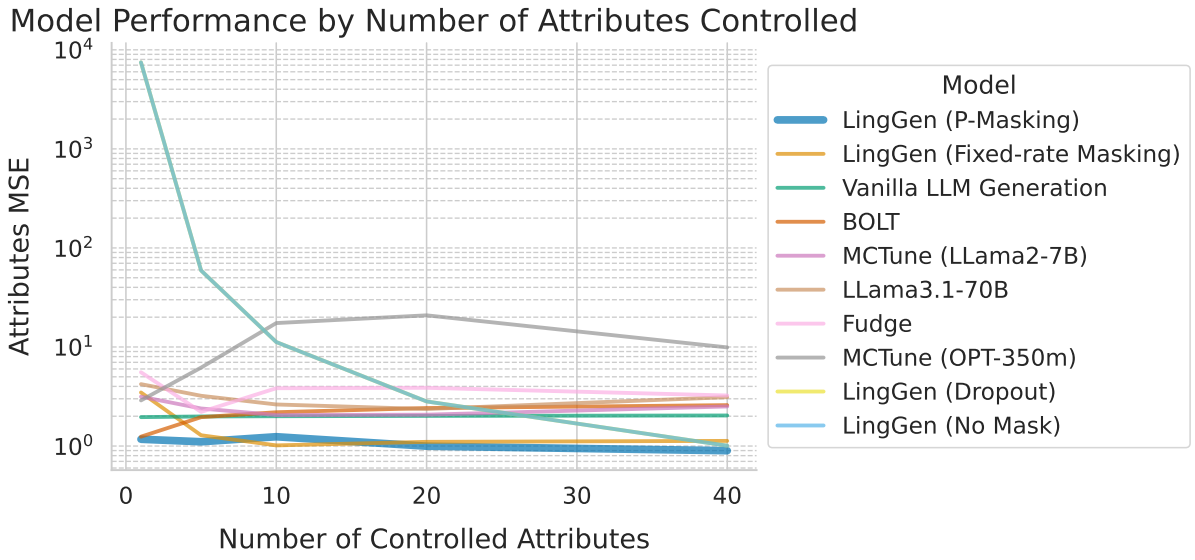

Figure 2: Model Performance by number of attributes controlled. The graph shows the MSE for different models as the number of controlled attributes increases. LingGen (P-Masking) consistently achieves the lowest MSE, indicating effective multi-attribute control.

minimization problem and uses gradient-based sampling to generate text that adheres to the given constraints.

MCTune (LLama2-7B) (Nguyen et al., 2024): This baseline leverages the MCTune method, which incorporates multiple linguistic complexity values as controls during instruction tuning to control the complexity of the generated text.

MCTune (OPT-350M) (Nguyen et al., 2024): This baseline adapts the MCTune method to the OPT-350M model, enabling control over multiple linguistic complexities in the generated text.

Decoding-time algorithms leverage a linguistic discriminator (LD) to estimate the linguistic attributes of generated text. This component is independently pre-trained and frozen, allowing for differentiable computation of linguistic attributes and backpropagation of the error. The LD is implemented using a DeBERTa encoder (He et al., 2021) with the token embedding layer replaced with that of OPT-350M, followed by a projection layer, trained to minimize the mean squared error between predicted and gold attributes. For further details, refer to Appendix B.

5.2 Datasets

We use 6.8M text samples, totaling 360M training tokens, drawn from diverse publicly available datasets. Each sample is truncated to a maximum of 100 tokens to prevent overfitting and ensure generalizability.

The datasets span a range of domains and writing styles, including web text (C4), paraphrase pairs (MRPC), question pairs (QQP), and natural language inference datasets (ANLI, RTE, STS-B, SNLI, MNLI, FeverNLI). All datasets are utilized as single text samples, focusing on characteristics such as user-generated content, formally written text, automatically generated text, etc. This diversity is crucial for training a robust model capable of handling various linguistic phenomena. The following datasets were used in our experiments: **Common Crawl (C4)** (Raffel et al., 2020), **Microsoft Research Paraphrase Corpus (MRPC)** (Dolan and Brockett, 2005), **Quora Question Pairs (QQP)** (Iyer et al., 2017), **Adversarial NLI (ANLI)** (Nie et al., 2020), **Recognizing Textual Entailment (RTE)** (Dagan et al., 2005), **Semantic Textual Similarity Benchmark (STS-B)** (Cer et al., 2017), **Stanford Natural Language Inference (SNLI)** (Bowman et al., 2015), **Multi-Genre Natural Language Inference (MNLI)** (Williams et al., 2018), and **FEVER NLI (FeverNLI)** (Thorne et al., 2018).

5.3 Metrics

Our model is evaluated on two key metrics. Most Controlled Text Generation (CTG) papers use two primary metrics for evaluation: **attribute accuracy** and **fluency**. Attribute accuracy measures how well the generated text adheres to the specified attributes, while fluency assesses the grammatical and logical

coherence of the text.

Mean Squared Error (MSE) of attributes calculates the error between attributes of the generation and the desired target attributes. For example, if the target attribute is sentence length, the MSE would measure the squared difference between the length of the generated sentence and the target length. Generations may achieve a good score on the target attributes while being non-fluent (i.e., logically or grammatically incorrect). To quantify the trade-off between control and fluency, we evaluate **Fluency** by prompting an LLM to answer whether the given sentence is fluent or not, and we report the fluency as the number of fluent paraphrases over total paraphrases. The LLM used is `Gemma 2 (9B)` (Team et al., 2024), and we include the prompt text in Appendix C.

6 Results

6.1 Main Results

Table 1 provides a comprehensive overview of the performance of various models in controlled text generation tasks. The Reference model, which uses the original text as the generated output, serves as an upper bound for fluency, achieving a score of 72.7. This sets a benchmark for what can be considered reasonable fluency. On the other hand, the Vanilla LLM, which generates text without any attribute control, provides a baseline for the average MSE of a random sample, with an MSE of 2.03 and a fluency score of 57.6. This indicates the typical performance of a model without any control mechanisms.

LingGen (P-MASKING) stands out by achieving the best MSE of 0.90, demonstrating its capability in controlled generation. This low MSE indicates that LingGen can effectively manage and adhere to specified linguistic attributes, outperforming other models in this regard. Importantly, LingGen maintains a fluency score of 83.6, which is not only significantly higher than the Vanilla LLM but also close to the Reference model, indicating that its fluency is not deteriorated by the control mechanisms.

The LLama3.1 model achieves the highest fluency score of 94.5. Although the instruction fine-tuned LLama3.1 70B is generally thought to be steerable for most instructions and controls, it still fails with unusual and numerous attributes, as evidenced by its MSE of 2.27. This suggests that while LLama3.1 excels in generating fluent text, it lacks the ability to control specific attributes effectively.

Other models, such as PPLM and Fudge, show varying degrees of success in controlled generation. PPLM, with an MSE of 5.99, struggles with attribute control, while Fudge performs better with an MSE of 3.24. These results align with their reliance on attribute classifiers, which may not be accurate enough to control the attributes effectively, and that become noisier when the number of attributes increases. The BOLT model achieves a relatively low MSE of 2.59 and a high fluency score of 88.9, indicating a good balance between control and fluency.

Overall, the results demonstrate that LingGen (P-MASKING) effectively balances attribute control and fluency, making it a robust choice for applications requiring precise and adaptable text generation.

For a detailed analysis of the ablation studies, please refer to Appendix B.1.

6.2 MSE Comparison Across Multiple Attributes

To assess the effectiveness of multi-attribute control, we conducted an experiment comparing the MSE of all models when controlling for different numbers of attributes. Specifically, we evaluate the MSE when 1, 5, 10, 20, or 40 attributes are controlled simultaneously. For each number of attributes, 2000 test samples are evaluated, each with a random selection of attributes to control. For each number of attributes, we repeat the experiment for three random seeds to account for biases due to the varying difficulty of the randomly chosen attribute(s). This experimental design allows us to evaluate the consistency and scalability of each model's attribute control capability across the linguistic indices described in Section A.

Figure 2 illustrates the MSE performance of various models as the number of controlled attributes increases. LingGen with P-Masking consistently achieves the lowest MSE across all attribute counts, demonstrating its superior ability to manage multiple attributes effectively. As the number of attributes increases, models like No Masking and Dropout show a significant rise in MSE, indicating challenges in handling complex attribute configurations. In contrast, LingGen with Fixed-rate Masking and BOLT maintain relatively stable MSEs, though not as low as LingGen with P-Masking. This highlights the robustness of the P-Masking strategy in multi-attribute control tasks.

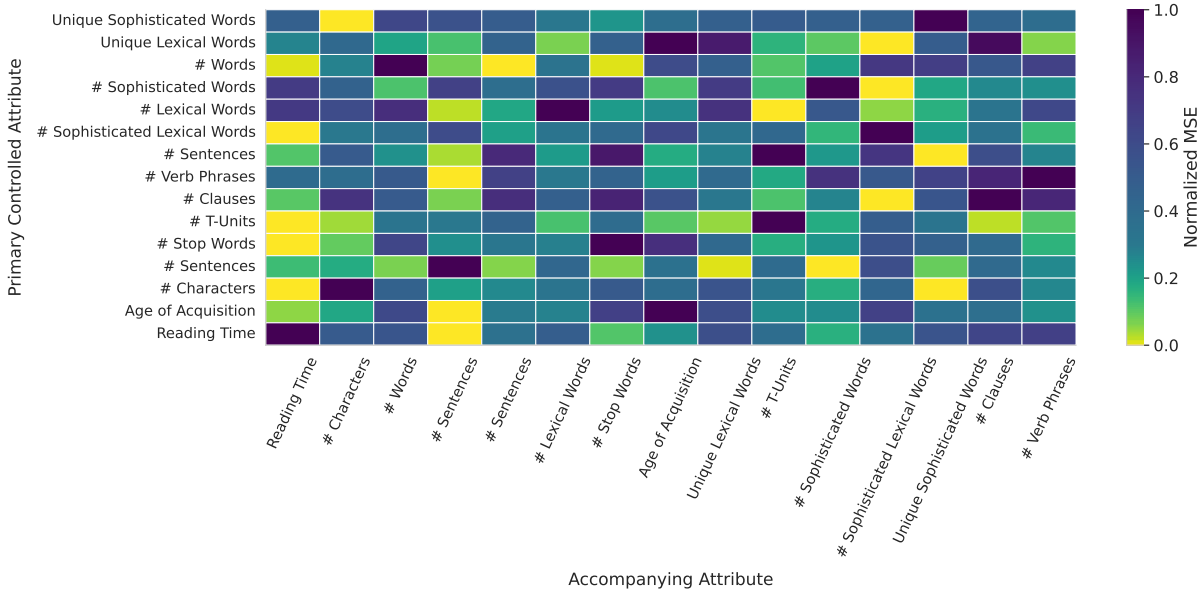

Figure 3: Pairwise attribute relations. Each cell represents the normalized MSE for a primary controlled attribute (rows) when paired with an accompanying attribute (columns). Lighter colors indicate higher errors.

The poor performance of models such as No Masking and Dropout is due to their inability to dynamically adapt to multiple attribute configurations. These models lack the mechanisms to effectively manage and integrate diverse attributes, leading to higher MSEs. Additionally, models like Fudge and LLama3.1-70B struggle with scalability, as their attribute control methods are not optimized for handling a large number of attributes, resulting in performance degradation.

6.3 Further Analysis of LingGen with P-MASKING

To gain deeper insights into the behavior of LingGen with our proposed P-MASKING strategy, we conducted a detailed analysis of pairwise attribute interactions.

We computed the pairwise MSE between all pairs of attributes. This analysis highlights which attributes become easier to control when accompanied by others and identifies those that effectively facilitate control over additional attributes.

Marginal Effects of Accompanying Attributes: "Reading Time" emerges as a positive accompanying attribute, often reducing the error when paired with others. This suggests that controlling for reading time can facilitate better management of other attributes.

Impact of Sophisticated Lexical Words: "# Sophisticated Lexical Words" serves as an effective accompanying attribute for controlling both the number of characters and the number of sentences. When "# Sophisticated Lexical Words" is included, the model demonstrates a reduced error in managing these two attributes. This can be attributed to the fact that sophisticated lexical words often have a higher character count, which directly influences the total number of characters. Additionally, the presence of sophisticated lexical words tends to structure sentences more clearly, thereby aiding in the control of sentence count.

Highlighting Key Interactions: The interaction between "Unique Sophisticated Words" and "Sophisticated Lexical Words" results in a notable increase in error, suggesting difficulty in controlling these nuanced aspects simultaneously.

This analysis provides valuable insights into the complexities of controlling multiple attributes, guiding future strategies for optimizing model performance in multi-attribute settings.

7 Conclusion

We have presented LingGen, a novel approach to controlled text generation that leverages a dynamic P-MASKING strategy to achieve precise control over a wide array of linguistic attributes. Our method demonstrates significant improvements in multi-attribute control, outperforming existing state-of-the-art models in both attribute accuracy and text fluency. The experimental results highlight LingGen's robustness and adaptability, particularly in scenarios with varying attribute demands. Future

research directions include expanding the range of controllable attributes and applying LingGen to larger, more diverse datasets to further enhance its applicability and performance in real-world text generation tasks.

8 Limitations

First, the model's performance is heavily dependent on the quality and diversity of the training data. Although we utilized a wide range of datasets, the model may still struggle with attributes or contexts not well-represented in the training set. This limitation suggests that the model's generalizability could be improved by incorporating more diverse, comprehensive, longer-text datasets.

Another limitation is the computational cost associated with training and deploying LingGen, which can be substantial, especially for larger models or when scaling to extensive datasets. The P-MASKING strategy, while effective in enhancing attribute control, introduces additional complexity in tuning the model for specific applications. This requires careful calibration of the masking distribution to ensure optimal performance, which can be resource-intensive and time-consuming. Moreover, the current implementation focuses on a predefined set of linguistic attributes, which may not encompass all the nuances required for certain specialized applications. This restricts the model's applicability in domains where unique or highly specific attributes are critical.

Future work should address these limitations by exploring more efficient training methods and expanding the attribute set. Additionally, extending LingGen into instruction fine-tuning could be a promising direction. This approach would allow the model to gain the benefits of a general-purpose language model (LLM) capable of performing a wide range of tasks and adapting to new circumstances. Instruction fine-tuning could enhance LingGen's flexibility and utility, enabling it to handle diverse tasks beyond controlled text generation, thereby broadening its applicability and impact in various real-world applications.

References

Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. 2000. A neural probabilistic language model. In *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA*, pages 932–938. MIT Press.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Stephen M Burroughs and Sarah F Tebbens. 2001. Upper-truncated power laws in natural systems. *Pure and Applied Geophysics*, 158:741–757.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. 2009. Power-law distributions in empirical data. *SIAM review*, 51(4):661–703.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, pages 177–190. Springer.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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

Hanxing Ding, Liang Pang, Zihao Wei, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. 2023. MacLaSa: Multi-aspect controllable text generation via efficient sampling from compact latent space. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4424–4436, Singapore. Association for Computational Linguistics.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *ArXiv preprint*, abs/2407.21783.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *Preprint*, arXiv:2111.09543.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pages 30016–30030.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1587–1596. PMLR.

Shankar Iyer, Nikhil Dandekar, and Kornél Csernai. 2017. Quora question pairs dataset. `https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs`. Accessed: 2024-10-15.

Bruce W. Lee and Jason Lee. 2023. LFTK: Handcrafted features in computational linguistics. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 1–19, Toronto, Canada. Association for Computational Linguistics.

Yoav Levine, Barak Lenz, Opher Lieber, Omri Abend, Kevin Leyton-Brown, Moshe Tennenholtz, and Yoav Shoham. 2021. Pmi-masking: Principled masking of correlated spans. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1865–1874, New Orleans, Louisiana. Association for Computational Linguistics.

Guangyi Liu, Zeyu Feng, Yuan Gao, Zichao Yang, Xiaodan Liang, Junwei Bao, Xiaodong He, Shuguang Cui, Zhen Li, and Zhiting Hu. 2023a. Composable text controls in latent space with ODEs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16543–16570, Singapore. Association for Computational Linguistics.

Xin Liu, Muhammad Khalifa, and Lu Wang. 2023b. BOLT: Fast energy-based controlled text generation with tunable biases. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 186–200, Toronto, Canada. Association for Computational Linguistics.

Xiaofei Lu. 2012. The relationship of lexical richness to the quality of esl learners' oral narratives. *The Modern Language Journal*, 96(2):190–208.

Xiaofei Lu. 2020. Automatic analysis of syntactic complexity in second language writing. *ArXiv preprint*, abs/2005.02013.

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.

Mark EJ Newman. 2005. Power laws, pareto distributions and zipf's law. *Contemporary physics*, 46(5):323–351.

Dang Nguyen, Jiuhai Chen, and Tianyi Zhou. 2024. Multi-objective linguistic control of large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 4336–4347, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Jonathan Pei, Kevin Yang, and Dan Klein. 2023. PREADD: Prefix-adaptive decoding for controlled text generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10018–10037, Toronto, Canada. Association for Computational Linguistics.

Shrimai Prabhumoye, Alan W Black, and Ruslan Salakhutdinov. 2020. Exploring controllable text generation techniques. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1–14, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Lianhui Qin, Sean Welleck, Daniel Khashabi, and Yejin Choi. 2022. Cold decoding: Energy-based constrained text generation with langevin dynamics. *Advances in Neural Information Processing Systems*, 35:9538–9551.

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

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

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Adam Roberts, Hyung Won Chung, Gaurav Mishra, Anselm Levskaya, James Bradbury, Daniel Andor, Sharan Narang, Brian Lester, Colin Gaffney, Afroz Mohiuddin, et al. 2023. Scaling up models and data with t5x and seqio. *Journal of Machine Learning Research*, 24(377):1–8.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi S. Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6830–6841.

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

Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal Schuster, Steven Zheng, et al. Ul2: Unifying language learning paradigms. In *The Eleventh International Conference on Learning Representations*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *ArXiv preprint*, abs/2408.00118.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Alexander Wettig, Tianyu Gao, Zexuan Zhong, and Danqi Chen. 2023. Should you mask 15% in masked language modeling? In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2985–3000, Dubrovnik, Croatia. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. 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.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. Glm-130b: An open bilingual pre-trained model. In *The Eleventh International Conference on Learning Representations*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *ArXiv preprint*, abs/2205.01068.

A List of Linguistic Attributes

We use expert-crafted linguistic indices as the control attributes for CTG. Table 2 lists all the indices that we use. For the full descriptions please refer to Lu (2020), Lu (2012), and Lee and Lee (2023). Briefly, **Automated Readability Index** measures text complexity, **Lexical words** are content words (nouns, verbs, adjectives, adverbs), and **Sophisticated words** are less frequent words in the American National Corpus. **Age of acquisition** refers to the age at which a word is typically learned.

B Linguistic Attribute Predictor

The **Linguistic Discriminator** (LD) is a crucial component for decoding-time algorithms, providing an efficient estimation of linguistic attributes. It is pre-trained using a DeBERTa encoder (He et al., 2021) with the token embedding layer replaced with that of OPT-350M, followed by a projection layer, trained to minimize the mean squared error between predicted attributes ($\text{LD}(x) = l^p$) and gold attributes (l^x) as shown in Equation 2:

$$\ell_{disc}(x) = \|\text{LD}(x) - l^x\|_2^2. \quad (2)$$

The final MSE loss of the pre-trained LD is 0.16 on our test set. The correlation between the predicted MSE by the LD and the real MSE by the original linguistic attribute extractor tool is 0.8, which is sufficiently high for reliable utilization.

B.1 Ablation Studies

To understand the contributions of our proposed P-MASKING strategy and the impact of different base models, we conducted two ablation studies.

Ablation Study: Impact of P-MASKING We evaluated various versions of our model using different methods of masking attributes during training. The methods included:

- **LingGen (No Masking)**: Attributes are not masked during training, serving as a baseline to assess the impact of masking.
- **LingGen (Dropout)**: A fixed dropout rate of 0.3 is applied to the attributes, introducing randomness to the training process.
- **LingGen (Fixed Rate)**: A fixed masking rate of 0.3 is applied, providing a consistent level of attribute masking.
- **LingGen (P-MASKING)**: Our proposed dynamic P-MASKING strategy, which adapts the masking rate based on a power law distribution.

Table 3 presents the results, demonstrating that P-MASKING significantly outperforms baseline methods in both MSE and fluency. The results show the effectiveness of P-MASKING in achieving a balance between attribute control and text quality.

Impact of Base Model We further evaluated LingGen with our proposed P-MASKING strategy using different base language models, specifically GPT-2 (Radford et al., 2019) and Pythia-410M (Biderman et al., 2023). Table 4 shows the results, highlighting that P-MASKING consistently delivers superior performance across different base models. This demonstrates that P-MASKING consistently enhances attribute control compared to other methods across different language model architectures.

Impact of Different Integration Methods We also explored the effects of different methods for integrating linguistic attributes into the model. The integration methods compared were:

- **LingGen (Add to SOS):** Our proposed method, where the encoded attribute representation is added to the Start-Of-Sequence (SOS) token embedding.

- **LingGen (Add to All):** The encoded attribute representation is added to all decoder inputs at each time step.

- **LingGen (Add to Output):** The encoded attribute representation is added to the decoder output at each time step.

- **LingGen (Add to Logits):** The encoded attribute representation is added to the logits at each time step.

All methods utilized the same P-MASKING strategy. As shown in Table 5, adding the encoded attribute representation to the SOS token embedding yields the best performance, outperforming other integration methods in terms of MSE. This improvement can be attributed to several factors: adding the attribute information to all input tokens can excessively distort the language model, while adding to the logits is computationally expensive and introduces noise due to the large vocabulary size. Furthermore, integrating attributes into the outputs (hidden representations) proves to be less effective. In contrast, incorporating the attribute information at the SOS token allows for efficient and effective propagation of this information throughout the entire sequence during generation, leveraging self-attention mechanisms. Notably, this approach demonstrates effectiveness both with and without masking, showing its reliability. This insight contributes to our understanding of how best to integrate attribute information into language models.

C Fluency Evaluation Prompt

We use the following prompt to evaluate fluency of the outputs. The prompt is adapted from (Liu et al., 2023b). Additionally, we post-process the output logits and only select the top token out of "yes" and "no".

The annotation task will provide texts created by different models.
Annotator is required to classify to answer whether the text is fluent or not fluent.
Fluency is defined as the ease and naturalness with which a text can be understood.
A fluent text should be straightforward to read or hear, without any structural or lexical awkwardness or ambiguity.
When evaluating fluency, annotators should consider two factors.
Grammaticality: Does the text follow standard grammatical rules?
Coherence: Does the text make sense in the context in which it is presented?
Here are some positive and negative samples corresponding to each factor.
First, Grammaticality.
Positive example: "The cat is sleeping peacefully on the soft, fluffy pillow." This text follows standard grammatical rules, with proper subject-verb agreement and adjective placement.
Negative example: "The cat are sleep peaceful on the soft pillow." This text contains grammatical errors, with a subject-verb disagreement and a missing adjective ending.
Second, Coherence.
Positive example: "After finishing her work, she decided to take a walk in the park." This text makes sense and flows logically, with a clear cause-and-effect relationship.
Negative example: "The concert was great, but I forgot my keys at home." This text lacks coherence, as there is no clear connection between the two clauses.
Annotators should not take into account the factual correctness or completeness of the text.
If the annotator finds it challenging to select a clear winner, they should select the text that is most similar in fluency to the other two texts.
Annotators should rely on their judgment and knowledge while assessing fluency, but consistency in their annotations should also be a priority.
Answer only using "yes" or "no", with no additional commentary or explanation.
Sentence:

Unique sophisticated words
Unique lexical words
Unique sophisticated lexical words
Total words
Total sophisticated words
Lexical sophistication (unique)
Verb sophistication
Ratio of unique words
Ratio of unique verbs
Ratio of unique adjectives
Ratio of unique adverbs
Dependent clauses
Clauses
T-units
Complex T-units
Complex nominals
Stop Words
Sentences
Characters
Average Words Per Sentence
Average Characters Per Sentence
Average Characters Per Word
Average Syllables Per Sentence
Total Age Of Acquistion Of Words
Named Entities Norp
Named Entities Gpe
Named Entities Law
Named Entities Money
Named Entities Ordinal
Coordinating Conjunctions
Nouns
Numerals
Proper Nouns
Subordinating Conjunctions
Automated Readability Index
Reading Time For Average Readers

Table 2: Linguistic indices used in this paper.

Method	MSE	Fluency
No Masking	1.01	86.15%
Dropout	1.00	86.22%
Fixed Rate	1.13	86.35%
P-MASKING	**0.90**	**86.50%**

Table 3: Comparison of masking strategies, demonstrating that P-MASKING outperforms baseline methods in both MSE and fluency.

Model	MSE	Fluency
Pythia-410M		
No Masking	2.58	78.15
Fixed Rate	2.39	46.70
P-MASKING	**2.04**	**78.75**
GPT-2		
No Masking	2.69	**64.25**
Fixed Rate	3.75	49.85
P-MASKING	**2.47**	61.55

Table 4: Performance of P-MASKING across different base models, highlighting its superior efficacy compared to other masking strategies.

Method	MSE	Fluency
No Masking		
SOS	**1.01**	**86.2**
All	2.60	0.0
Output	1.11	80.2
Logits	1.52	80.5
P-Masking		
SOS	**0.90**	**83.5**
All	3.52	0.0
Output	1.76	80.7
Logits	1.31	81.9

Table 5: Evaluation of integration methods, justifying that the SOS method provides the best performance in terms of mean squared error and fluency.