LingGen: Scalable Multi-Attribute Linguistic Control via Power-Law Masking

Mohamed Elgaar and Hadi Amiri
University of Massachusetts Lowell
{melgaar,hadi}@cs.uml.edu

Abstract

We present LingGen, a controlled text generation model that allows fine-grained control over a large number of real-valued linguistic attributes. It encodes target attribute values with a dedicated linguistic attribute encoder and conditions the language model by injecting the resulting representation into the language model using the beginning-of-sequence (BOS) embeddings. To improve robustness when controlling different attribute subsets, we introduce P-MASKING, which samples per-example attribute masking rates from a truncated Pareto distribution during training. Across 1–40 control attributes, LingGen achieves the lowest average control error among evaluated methods, while remaining efficient at inference and receiving the highest fluency scores in human evaluation. Ablations show that Pareto-sampled masking and BOS-based injection are effective choices compared to alternative masking and integration variants.[1]

1 Introduction

Controlled text generation (CTG) aims to produce text that satisfies user-specified constraints, with applications in education, accessibility, and personalized communication (Dathathri et al., 2020a; Zhang et al., 2023; Prabhumoye et al., 2020). While existing approaches can control coarse properties (e.g., sentiment), fine-grained linguistic control remains difficult (Liu et al., 2023a), especially when (i) target linguistic attributes are real-valued, (ii) the number of controlled attributes varies at inference time, and (iii) the attribute set is large enough such that interactions and trade-offs become unavoidable. In this work, we study scalable linguistic control with $k = 40$ expert-crafted linguistic complexity attributes spanning surface counts, syntactic structure, and psycholinguistic complexity (Appendix C).

Existing CTG methods can steer high-level attributes such as sentiment or topic, but they are less effective for finer-grained linguistic attributes and suffer from inefficiencies when many attributes must be handled jointly (Li et al., 2018; Liu et al., 2023a). In parallel, recent work shows that combining denoising objectives with causal language modeling can improve robustness (Raffel et al., 2020; Tay et al., 2023; Zeng et al., 2023; Wettig et al., 2023). Building on these insights, we propose LingGen–a controlled generation model that supports fine-grained control over a large number of real-valued linguistic attributes.

LingGen conditions a base language model on target attribute values using a dedicated attribute encoder and a lightweight integration mechanism: it injects the encoded attribute representation into the beginning-of-sequence (BOS) embedding, allowing self-attention to propagate conditioning information without modifying every token embedding. To generalize control to arbitrary subsets of attributes (from 1 to k), we introduce P-MASKING, which stochastically hides attributes during training by sampling a masking rate from a truncated Pareto distribution (a bounded power law distribution). This sampling scheme emphasizes near-complete attribute sets while preserving a tail of sparse configurations to improve robustness when the number of controlled attributes varies at test time.

Our contributions are as follows: (**1**) we propose LingGen, a CTG model for real-valued, fine-grained linguistic control that scales to 40 attributes and supports controlling any subset size (1–40) at inference time; (**2**) we introduce P-MASKING, a Pareto-sampled, per-example attribute masking scheme that improves robustness across attribute subset sizes; (**3**) we provide empirical analysis of attribute interactions and benchmark LingGen against fine-tuning, inference-time control, and prompting baselines, and highlight the trade-offs between *control* accuracy, *fluency*, and *efficiency*.

[1] Code & data: https://github.com/CLU-UML/LingGen

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 1925–1942

Capability	LingGen (Ours)	Fine-tuning (MCTune, PTG)	Inference-time (PPLM, Guided Decoding)	LLM Prompting (Llama 3.1)
Fine-grained control	✓	✓	✓	✗
Maintains high fluency	✓	✓	✗	✓
Efficient at inference	✓	✓	✗	✗
Robust & scalable attribute control†	✓	✗	✗	✗

Table 1: Comparison of LingGen with other CTG approaches. †Maintains effective control over any subset of attributes, including high-count subsets (>20).

2 Background

CTG 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).

Recent advancements have explored compositional text control in latent space by leveraging compact, differentiable representations. Techniques based on ordinary differential equations and latent space samplers can efficiently compose multiple control operations while reducing computational overhead and maintaining text quality (Liu et al., 2023a; Ding et al., 2023). These methods align with the growing interest in developing models that adapt to diverse control inputs across various domains (Yang et al., 2023).

In parallel, research into Masked Language Models (MLMs) has highlighted the effectiveness of masking strategies (Devlin et al., 2019). Our method draws on prior work such as PMI-Masking (Levine et al., 2021) and infilling objectives in UL2 and GLM-130B (Tay et al., 2023; Zeng et al., 2023; Levine et al., 2021), but targets a different problem: robust conditioning on a variable-size set of linguistic attribute controls. While traditional MLMs use a fixed 15% masking rate (Devlin et al., 2019), recent work has shown benefits from higher rates—up to 40% or 80%—in some settings (Wettig et al., 2023). Building on these insights, we design P-MASKING, which masks linguistic control attributes during training according to a power law distribution (Clauset et al., 2009). This distribution exposes the model to both dense and sparse attribute sets.

CTG enables text creation tailored to specific requirements, with recent works exploring various approaches such as sentiment manipulation via fine-grained control codes (Shi et al., 2024), tunable biases for factual consistency (Liu et al., 2023b), and prefix-adaptive decoding for style control (Pei et al., 2023). However, these methods primarily focus on high-level properties rather than low-level linguistic attributes. Models incorporating denoising objectives in pretraining, such as UL2 and GLM, have demonstrated enhanced capabilities in diverse linguistic tasks (Tay et al., 2023; Zeng et al., 2023; Chowdhery et al., 2023; Roberts et al., 2023). Unlike existing approaches that use instruction-tuning (Nguyen et al., 2024), prompt-tuning (Bandel et al., 2022; Alhafni et al., 2024; Yang et al., 2023), concatenation (Huang et al., 2023), or simple fusion (Liu et al., 2023a), LingGen introduces a dedicated attribute embedding network and selective masking to control a variable number of attributes while preserving base LLM capabilities. In addition, although multi-attribute controlled *paraphrase* generation has been studied (Elgaar and Amiri, 2025), LingGen targets *free text* generation with robust control over *variable-size* attribute subsets. Table 1 provides a summary of how LingGen compares to existing approaches.

3 Linguistic Generation with LingGen

Given a set of desired linguistic attributes, $\boldsymbol{a} = \{L_1, \ldots, L_k\}$, where each L_i represents a specific linguistic attribute (e.g., sentence length, number of unique sophisticated words), the task is to generate text that exhibits those attributes. In our setup, target attribute values are obtained by running the deterministic attribute extractor on held-out reference texts and z-normalizing each attribute using training-set mean and standard deviation; at test time we sample a reference text, randomly select the requested number of attributes (e.g., 1/5/10/20/40), and use their normalized values as the control targets. We use 40 attributes (detailed in Appendix C) representing diverse dimensions of linguistic style with low mean Pearson correlation (0.29), indicating minimal redundancy. Fine-grained stylistic control often requires simultaneous manipulation of complex features; for instance,

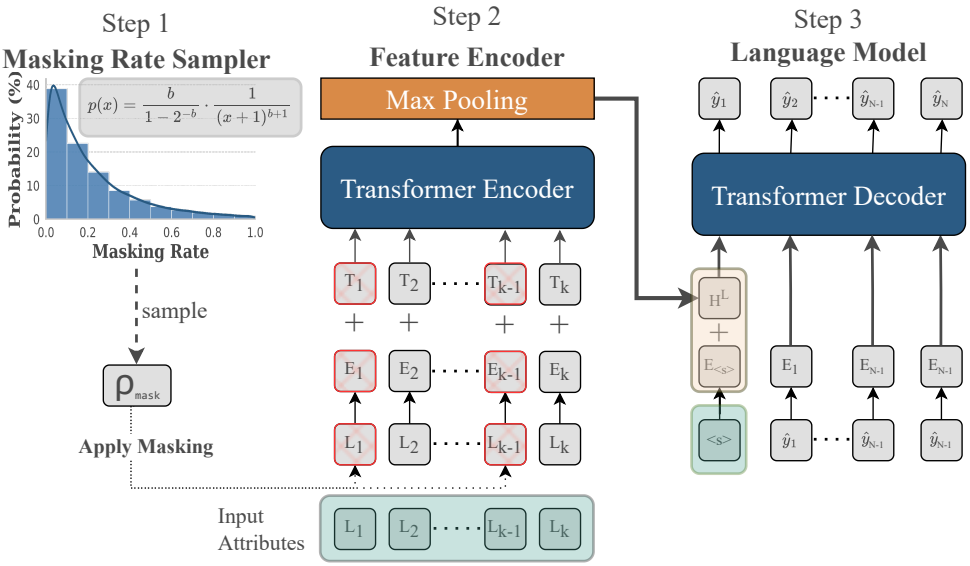

Figure 1: Overview of the LingGen architecture for controlled text generation. 1) **Masking Rate Sampler (Training):** Implements P-MASKING by sampling attribute masking rates (ρ_{mask}) from a Pareto distribution per sample. 2) **Attribute Encoder:** Encodes linguistic attributes ($L_1..L_K$) into a combined representation using embeddings and attribute-specific token types ($T_1..T_K$). 3) **Language Model:** A Transformer Decoder generates text ($\hat{y}_1..\hat{y}_n$) conditioned on the attribute representation, which is injected into the BOS token (<s>) embedding to steer generation.

generating accessible materials for individuals with aphasia requires control over word frequency, syntactic complexity, and lexical density beyond simple readability scores. Our P-MASKING strategy is designed to improve control across a variable and large number of attributes (from 1 to k). 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 $V : Y \rightarrow \mathbb{R}^k$ 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 $\boldsymbol{y} = \arg\min_{\boldsymbol{y} \in Y} L(V(\boldsymbol{y}), \boldsymbol{a})$. 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).

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 computed using explicit linguistic analysis algorithms from prior work (Lu, 2020, 2012; Lee and Lee, 2023). These algorithms provide the function $V(\boldsymbol{y})$ that maps generated text $\boldsymbol{y}$ to a vector of attribute values in $\mathbb{R}^k$. These same algorithms are used during evaluation to provide reliable, deterministic measurement of attribute control error, ensuring that our reported MSE values reflect true deviations from the target attributes rather than relying on approximations.

LingGen consists of a Masking Rate Sampler, an Attribute Encoder, and an LLM (Figure 1), working together to allow for flexible control over a variable number of attributes.

Method	MSE ↓	PPL ↓	Time/token (ms) ↓	
No Control				
Reference	0.00	35.4	-	-
Vanilla OPT-350M (Zhang et al., 2022)	2.00	12.7	25	(1×)
Inference-time Control				
PPLM (Dathathri et al., 2020b)	5.99	<u>10.7</u>	1515	(61×)
Guided Decoding (Glandorf et al., 2025)	3.74	20.5	112	(5×)
COLD (Qin et al., 2022)	3.99	11.3	3846	(155×)
Mix&Match (Mireshghallah et al., 2022)	1.58	165.0	5882	(237×)
BOLT (Liu et al., 2023b)	2.08	12.0	114	(5×)
PiLM (Yang et al., 2024)	2.12	17.7	2102	(84×)
Llama 3.1 (Dubey et al., 2024)	3.11	14.5	162	(7×)
Fine-tuned LLM				
MCTune Llama 2-7B (Nguyen et al., 2024)	2.44	**5.8**	68	(3×)
MCTune OPT-350M (Nguyen et al., 2024)	11.46	16.4	46	(2×)
PTG (Alhafni et al., 2024)	<u>1.71</u>	21.5	76	(3×)
LingGen (ours)	**1.08**	15.4	**25**	**(1×)**

Table 2: Comparison of model performance across different methods. The table presents the average MSE across varying numbers of attributes (1, 5, 10, 20, 40), perplexity (PPL), and time per token.

3.1 Attribute Integration

The k linguistic attributes $\boldsymbol{a} = \{L_1, \ldots, L_k\}$ are processed through an Attribute Encoder architecture. First, the encoder employs a linear embedding layer that maps each scalar attribute value L_i to a d_{attr}-dimensional embedding space through the transformation $E_i = W_{emb}L_i + b_{emb}$ where $W_{emb} \in \mathbb{R}^{d_{attr} \times 1}$ and $b_{emb} \in \mathbb{R}^{d_{attr}}$. These attribute embeddings are then combined with learned token-type embeddings T_i that indicate the attribute identity (e.g., sentence length vs. lexical sophistication), which is particularly important when only a subset of attributes is provided as input.

The combined embeddings are processed by a transformer encoder consisting of 2 layers with 5 attention heads, which captures the interactions between different attributes. The encoder's output undergoes max-pooling across the sequence dimension to obtain a fixed-size representation, which is then mapped to the model's hidden dimension through a final projection layer $W_{proj} \in \mathbb{R}^{d_{attr} \times d_{LLM}}$.

A key design choice is to add the resulting attribute representation only to the beginning-of-sequence (BOS) token embedding. This allows the conditioning signal to influence generation through self-attention while avoiding the distribution shift that can arise when adding conditioning to every token embedding.

3.2 P-MASKING: A Sample-based Attribute Masking Strategy

During training, we employ P-MASKING, a strategy that selects the number of linguistic control attributes to mask for each sample. Intuitively, varying which attributes are visible encourages the model to rely on the provided controls (rather than a fixed attribute configuration) and improves robustness when the requested subset of attributes changes at inference time.

P-MASKING samples masking rates from a truncated Pareto distribution (Burroughs and Tebbens, 2001). For each training sample, a masking proportion ρ_{mask} is sampled, determining the proportion of attributes to mask. This allows the model to learn to control any number of attributes. Notably, this strategy introduces only a single hyperparameter (b) keeping tuning complexity minimal. The probability density function for masking proportion ρ_{mask} is given by:

$$p(x) = \frac{b}{1 - 2^{-b}} \frac{1}{(x+1)^{b+1}} \tag{1}$$

for $0 \leq x \leq 1$, where b is a shape parameter. We sweep $b \in \{1, 2, 5, 10, 20\}$ and measure the resulting masking-rate histogram (Appendix F). As shown in Figure 3, the selected value $b = 5$ concentrates probability mass on near-complete attribute sets (0–30% masking) while still assigning some

probability to sparse configurations (50–99% masking). The sampled masking proportion ρ_{mask} determines how many attributes are masked. Masked attributes are excluded from the self-attention and from the output representation pooling.

This sample-based strategy introduces per-example variation in the visible attribute set. The power law distribution yields frequent low masking rates (dense conditioning) while still sampling sparse cases that are necessary for subset robustness.

We aim for the model to perform well on any subset of attributes. However, using a fixed masking rate would train the model primarily on subsets of a certain size (e.g., $k \times (1 - \rho_{\text{mask}})$ visible attributes), potentially limiting its ability to handle the full set of k attributes together or single attributes in isolation. Therefore, P-MASKING samples the number of masked attributes from a power law distribution: it places high probability on low masking (nearly complete visible sets) and a low-probability tail on high masking (sparse visible sets). Dense cases support learning precise multi-attribute control, while sparse cases train the model to remain functional when few attributes are provided.

4 Experiments

4.1 Baselines

We compare against the following baselines:

Vanilla OPT-350M (Zhang et al., 2022) generates text by sampling from the LLM without attribute conditioning, serving as a control to verify attribute learning.

Reference uses the reference sentence as output, providing an upper performance bound by assuming perfect reproduction of the reference text.

Mix&Match (Mireshghallah et al., 2022) treats controllable generation as sampling from an energy-based model, using MLM scores and an attribute discriminator.

Plug and Play Language Models (PPLM) (Dathathri et al., 2020b) combines a pre-trained LLM with attribute classifiers for attribute control without LLM retraining.

Guided Decoding (Glandorf et al., 2025) is an adaptation of FUDGE (Yang and Klein, 2021) that steers generation by modifying output probabilities to satisfy constraints.

Method	Fluency ↑
BOLT	3.73
PiLM	3.80
PTG	4.20
LingGen (ours)	**4.60**

Table 3: Human evaluation of fluency for each method. The scores range from 1 to 5.

Llama 3.1-70B (Dubey et al., 2024) is an LLM prompting baseline using the instruction-tuned Llama 3.1 (70B) chat model.

Biases Over Logits (BOLT) (Liu et al., 2023b) directly modifies LLM output logits using learned biases, tuned to minimize attribute and perplexity losses.

COLD Decoding (Qin et al., 2022) frames constrained generation as an energy minimization problem, employing gradient-based sampling for constraint adherence.

Multi-Control Tuning (MCTune) (Nguyen et al., 2024) uses instruction tuning for linguistic attribute control, incorporating attribute definitions into a meta prompt and appending target values to input prompts.

Personalized Text Generation with Fine-Grained Linguistic Control (PTG) (Alhafni et al., 2024) controls linguistic attributes via a learned representation prepended to the input text.

Plug-in Language Model (PiLM) (Yang et al., 2024) adjusts LLM latent states for control using black-box attribute discriminators.

4.2 Experimental Setup

All baselines (except Llama 3.1) are re-implemented using OPT-350M as a base-model. For the LLM prompting baseline, we query the instruction-tuned Llama 3.1-70B chat model (Dubey et al., 2024) using a fixed two-message chat template: a system meta-instruction (similar to the meta-prompt used by MCTune (Nguyen et al., 2024)) that defines the available attribute tags (corresponding to Appendix C) and their meanings, and a user message that lists the desired subset of k targets as bracketed `[tag: value]` pairs; the model is instructed to output a JSON object with a single `text` field, and generations are capped at 100 tokens to

Method	# Sophisticated	# Total	# Lexical	Unique Word Ratio
Vanilla OPT-350M	3.56	14.08	6.89	11%
COLD	3.75	19.31	9.52	17%
Mix&Match	3.83	10.67	6.55	10%
PPLM	7.06	50.64	23.56	30%
PiLM	3.61	18.65	9.62	14%
Guided Decoding	5.91	40.63	17.93	14%
BOLT	3.58	25.21	10.71	14%
MCTune OPT-350M	5.84	17.04	10.04	12%
MCTune Llama 2-7B	5.16	16.24	9.19	11%
Llama 3.1-70B	4.30	10.43	6.04	8%
PTG	3.10	8.44	5.20	9%
LingGen	**1.28**	**1.76**	**1.50**	**6%**

Table 4: Mean Absolute Error (MAE) for selected linguistic attributes across different models. The values represent the error between the attributes of the generated text and the target attributes.

match other baselines. Moreover, inference-time algorithms use 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. It is trained on attributes generated by the explicit linguistic attribute extraction algorithms.

The **Linguistic Discriminator** (LD) is a crucial component for inference-time algorithms. It is pre-trained on data generated by the explicit linguistic attribute extraction algorithms (described in Section C) to provide an efficient estimation of linguistic attributes for these methods. It uses a DeBERTa encoder (He et al., 2023) with the token embedding layer replaced with that of OPT-350M, followed by a projection layer. The LD is trained to minimize the mean squared error between predicted attributes and gold attributes:

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

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.

We tune the hyperparameters of all baselines using grid search. The final hyperparameters used for each baseline are detailed in Appendix D.2. Further details on the training setup and computational infrastructure are provided in Appendix D.1.

4.3 Datasets

We use 6.8M samples (360M tokens) from publicly available datasets spanning multiple domains and writing styles, truncated to 100 tokens maximum. A detailed list is provided in Appendix B.

4.4 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. With sentence length as an example, MSE measures the squared difference between the length of the generation and the target length. Attribute measurements for MSE calculation are performed using the same explicit algorithms used to define the target attributes. This provides an exact measure, making human evaluation for attribute accuracy unnecessary. 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 **Perplexity (PPL)** according to GPT2-XL (Radford et al., 2019), following recent work such as Shen and Huang (2025) and Fathi et al. (2025). To complement this automated metric, we also conduct human evaluations of fluency, detailed in Appendix G.

5 Results

5.1 Main Results

Table 2 summarizes the performance of various models in controlled text generation tasks. The reported MSE is an average calculated across experiments controlling 1, 5, 10, 20, and 40 attributes simultaneously. Each attribute count setting was evaluated using 2,000 test samples per seed across 3 random seeds (6,000 generations per attribute count). For models evaluated across all five attribute counts, this totals 30,000 generations per model. Due to computational cost, the inference-time baselines (PPLM, Mix&Match, COLD, PiLM) were only run for the 40-attribute setting (6,000 generations), and their MSE in Table 2 corresponds to that setting. This design reflects overall performance across varying control complexities and mitigates potential bias from specific attribute selections or random seeds.

Vanilla OPT-350M, generating text without attribute control, provides a baseline with an average MSE of 2.00 and a perplexity of 12.7. Models achieving an average MSE lower than 2.0 include LingGen, Mix&Match, and PTG. Among those, Mix&Match exhibits very high perplexity, indicating a sacrifice of fluency for control. LingGen achieves the best overall average MSE (1.08) while maintaining competitive perplexity (15.4). LingGen also demonstrates consistently low MSE across attribute counts and seeds (Table 6), highlighting its scalability and robustness. In contrast, models like MCTune (OPT-350M), Guided Decoding, and Llama 3.1-70B show less stable performance as the number of attributes varies.

Furthermore, we conducted a human evaluation of text fluency for the top-performing fine-tuned models. As shown in Table 3, LingGen achieves the highest average fluency score. Further details on the human evaluation setup and annotator guidelines are provided in Appendix G. Qualitative samples in Appendix A demonstrate the gap in existing models: PTG and Llama 3.1 either break attribute constraints or collapse to incoherent text, whereas LingGen maintains usable generations while achieving the target attributes.

To better understand model performance on specific types of attributes, Table 4 shows the Mean Absolute Error (MAE) for four representative attributes: sophisticated word count, total words, lexical words, and unique word ratio. LingGen achieves the lowest error across all these attributes.

Attribute	LingGen	Llama 70B	Gap
Complex Nominals	0.84	2.36	2.8×
T-units	0.17	0.57	3.4×
Age of Acquisition	12.00	75.36	6.3×

Table 5: Comparison of Mean Absolute Error (MAE) for LingGen and Llama 3.1-70B on deep structural attributes. Gap denotes the ratio between the errors.

The Llama 3.1-70B model, despite its strong general capabilities, shows higher average MSE (3.11) and middling performance on specific attributes (Table 4), suggesting limitations in fine-grained control over linguistic attributes compared to specialized methods. Table 5 further highlights this gap on deep structural attributes (syntactic: Complex Nominals, T-units; psycholinguistic: Age of Acquisition).

Appendix E includes ablation studies on P-MASKING's effects compared to other strategies, different base models, and attribute integration methods. Results indicate that LingGen with P-MASKING outperforms other masking strategies when evaluated on variable attribute numbers. Our ablation study on attribute integration methods (Table 12) shows that injecting the attribute representation at the BOS token yields the best performance, lowering MSE by 36–65% compared with alternative injection points (adding to all tokens, outputs, or logits) while maintaining perplexity comparable to the baseline.

5.2 Robustness Across Attribute Counts

To evaluate control effectiveness over varying attribute numbers, we compare the MSE of models when controlling 1, 5, 10, 20, or 40 attributes simultaneously (Table 6). For each attribute count, 2,000 test samples were evaluated per random seed, and the experiment was repeated with three random seeds. This resulted in 6,000 evaluations per attribute count setting and 30,000 total evaluations per model for models evaluated across all five attribute counts. For a given attribute count, attributes were randomly selected for control from the full set of 40 for each seed. PPLM, Mix&Match, COLD, and PiLM were only run for the 40-attribute setting due to computational cost. The values reported in Table 6 are the average MSE across the three seeds for that specific count. The "Average" column shows the mean MSE across all five attribute counts (with standard deviation across the five means).

Method	1	5	10	20	40	Average
Reference	-	-	-	-	0.00	0.00
PPLM	-	-	-	-	5.99	5.99
COLD	-	-	-	-	3.99	3.99
Mix&Match	-	-	-	-	1.58	1.58
PiLM	-	-	-	-	2.12	2.12
Vanilla OPT-350M	1.96	1.99	2.00	2.02	2.03	2.00 ± 0.03
BOLT	1.24	1.96	2.19	2.42	2.59	2.08 ± 0.47
Llama 3.1-70B	4.22	3.22	2.63	2.36	3.11	3.11 ± 0.64
Guided Decoding	5.56	2.21	3.83	3.87	3.24	3.74 ± 1.09
MCTune OPT-350M	2.89	6.19	17.43	20.88	9.90	11.46 ± 6.75
MCTune Llama 2-7B	3.15	2.39	2.06	2.07	2.51	2.44 ± 0.40
PTG	2.11	1.96	2.29	1.61	**0.60**	1.71 ± 0.60
LingGen (ours)	**1.17**	**1.10**	**1.24**	**1.00**	0.90	**1.08** ± 0.12

Table 6: MSE for different models when controlling 1, 5, 10, 20, or 40 attributes simultaneously.

5.3 Analysis of Attribute Interaction Effects

This experiment investigates the interaction effects between pairs of linguistic attributes in multi-attribute controlled text generation. Understanding these interactions is crucial, as controlling one attribute j may facilitate (synergy) or hinder (conflict) the model's ability to control another attribute i. We quantify this interaction effect, $\Delta\text{MSE}_{i \leftarrow j}$, as the difference in the expected squared error for attribute i. Let SE_i denote the squared error for attribute i. The interaction effect is then the difference in this value when attribute j is included in the set of controlled attributes (A_j) versus when it is excluded ($\neg A_j$):

$$\Delta\text{MSE}_{i \leftarrow j} = \mathbb{E}[\text{SE}_i \mid A_j] - \mathbb{E}[\text{SE}_i \mid \neg A_j]$$

A negative $\Delta\text{MSE}_{i \leftarrow j}$ indicates synergy (controlling j improves control over i), while a positive value signifies conflict (controlling j degrades control over i).

We conducted the analysis using the LingGen model. We evaluated 2,000 test samples per run across 8 random seeds (16,000 total samples). In each run, 20 attributes were randomly selected from the total 40 for simultaneous control. This setup ensures a balanced comparison, as any attribute j had approximately a 50% chance of being included in the controlled set. We report only statistically significant interactions (paired t-tests, $p < 0.05$). To facilitate comparison across different target attributes i, the effect sizes presented in Figure 2 are row-normalized, scaling the strongest significant positive or negative effect for each attribute i to $+1.0$ or -1.0, respectively.

Our analysis identifies attributes that have large effects when included as conditioning factors. *Sentence Count* shows synergies with metrics related to average sentence length, such as *Words per Sentence*, *Characters per Sentence*, and *Syllables per Sentence*, likely because fixing the sentence count provides a stable denominator for these averages. However, it conflicts with controlling *Dependent Clauses*, suggesting a tension between controlling sentence quantity and sentence complexity. Similarly, controlling *Words per Sentence* can help constrain related metrics such as total words and number of syllables.

Conversely, controlling for *Unique Word Ratio*, a measure of lexical diversity, consistently introduces conflicts. It degrades the ability to control *Total Words*, *Total lexical words*, *Age of Acquisition Score (AoA)*, and *Readability Level*. This suggests an inherent trade-off: enforcing high lexical diversity makes it harder for the model to adhere to specific length constraints (which might otherwise favor repetition) or to use simpler and more common vocabulary (lower AoA and higher readability).

Controlling for higher-level composite attributes could improve control over their constituent components. *Reading Time For Average Readers* shows strong synergistic effects, improving control over *Lexical Sophistication*, *Complex Nominals*, *Character Count*, and *Characters per Word*. This implies that targeting an overall reading time

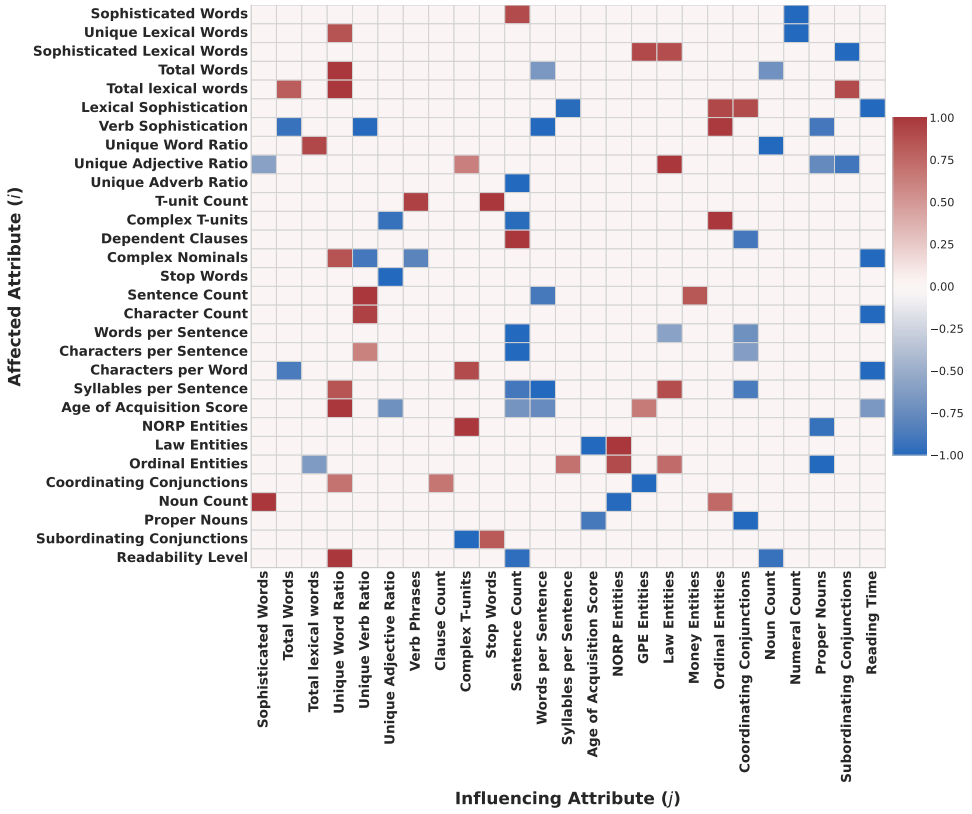

Figure 2: Interaction Effect ($\Delta\text{MSE}_{i \leftarrow j}$) of Controlling Attribute j on Controlling Attribute i (Statistically Significant Interactions, Row-Normalized). The cell values are the interaction effect ($\Delta\text{MSE}_{i \leftarrow j}$); Negative (blue) indicates synergy, while positive (red) indicates conflict.

forces the model to implicitly manage underlying factors like word complexity, length, and structure, thereby aiding explicit control over these factors.

Moreover, strong synergies are observed between definitionally linked attributes such as *Subordinating Conjunctions* and *Complex T-units* likely because complex T-units often require subordinating conjunctions. We also observe an unexpected synergy where controlling *Numeral Count* improves control over *Sophisticated Words* and *Unique Lexical Words*. One hypothesis is that numeral constraints induce a shift toward a enumerations, measurements, and comparisons, which increase reliance on self-contained constructs like nouns and domain-specific words, which perhaps improve control over sophisticated/unique words.

Finally, attribute compatibility matters: some controls systematically synergize while others conflict, suggesting that some attribute combinations may be difficult to satisfy simultaneously. They also motivate controlling a broader set of attributes when targeting a complex style, since changing one attribute can induce measurable shifts in others.

6 Conclusion

We presented LingGen, a controlled text generation method for fine-grained linguistic control with a variable number of real-valued attributes. LingGen combines a dedicated attribute encoder with BOS-based integration and achieves the strongest overall control-fluency-efficiency trade-off among the evaluated baselines, including settings that control up to 40 attributes simultaneously. We also introduced P-MASKING, a Pareto-sampled attribute masking strategy that improves robustness across attribute subset sizes. Beyond aggregate control error, our interaction analysis indicates that fine-grained linguistic control is affected by synergies and conflicts among attributes. First, for composite targets, controlling a broader set of related attributes can stabilize control by anchoring shared components. Second, some attribute pairs are inherently difficult to satisfy simultaneously, and effective control may require relaxing or reweighting constraints. Future work includes extending beyond a fixed attribute set (e.g., natural-language attribute descriptions) and interaction-aware constraint selection.

Limitations

Our study focuses on a fixed set of 40 attributes computed by deterministic extractors, and LingGen does not directly support controlling unseen attributes without defining (and validating) new extractors. We also evaluate generations truncated to 100 tokens; maintaining fine-grained control over longer documents may introduce additional challenges (e.g., drift in attribute satisfaction over time).

We note that this work does not address "few-to-many" generalization in the sense of training on a small fixed subset of attributes and generalizing to control unseen attributes at test time. Instead, our focus is on robustness across the combinatorial space of attribute subsets: given a fixed set of k attributes, P-MASKING enables the model to robustly handle any subset of size 1 to k at inference, without retraining. Table 6 demonstrates this capability, showing stable performance across varying numbers of simultaneously controlled attributes.

Ethical Statement

Fine-grained control over linguistic attributes can be beneficial (e.g., readability adaptation) but may also enable misuse, such as producing persuasive or targeted text optimized for manipulation, impersonation, or misinformation. Because our controls include stylistic and complexity attributes, a malicious user could attempt to tailor text to specific audiences or contexts. Mitigations include restricting deployment to trusted settings, applying safety filters and content policies at generation time, and auditing attribute extractors and training data for biases that could be amplified through controlled generation.

References

Bashar Alhafni, Vivek Kulkarni, Dhruv Kumar, and Vipul Raheja. 2024. Personalized text generation with fine-grained linguistic control. In *Proceedings of the 1st Workshop on Personalization of Generative AI Systems (PERSONALIZE 2024)*, pages 88–101, St. Julians, Malta. Association for Computational Linguistics.

Elron Bandel, Ranit Aharonov, Michal Shmueli-Scheuer, Ilya Shnayderman, Noam Slonim, and Liat Ein-Dor. 2022. Quality controlled paraphrase generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 596–609, Dublin, Ireland. Association for Computational Linguistics.

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, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, 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.

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, and 1 others. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

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. 2020a. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020b. 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, and 1 others. 2024. The llama 3 herd of models. *ArXiv preprint*, abs/2407.21783.

Mohamed Elgaar and Hadi Amiri. 2025. Linguistically-controlled paraphrase generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 20842–20864, Suzhou, China. Association for Computational Linguistics.

Nima Fathi, Torsten Scholak, and Pierre-Andre Noel. 2025. Unifying autoregressive and diffusion-based sequence generation. In *ICLR 2025 Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy*.

Dominik Glandorf, Peng Cui, Detmar Meurers, and Mrinmaya Sachan. 2025. Grammar control in dialogue response generation for language learning chatbots. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 9820–9839, Albuquerque, New Mexico. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

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.

Xuancheng Huang, Zijun Liu, Peng Li, Tao Li, Maosong Sun, and Yang Liu. 2023. An extensible plug-and-play method for multi-aspect controllable text generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15233–15256, Toronto, Canada. Association for Computational Linguistics.

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.

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. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

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, and 1 others. 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.

Ying Shen and Lifu Huang. 2025. LLM braces: Straightening out LLM predictions with relevant sub-updates. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7977–7992, Vienna, Austria. Association for Computational Linguistics.

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, Huaixiu Steven Zheng, Denny Zhou, Neil Houlsby, and Donald Metzler. 2023. UL2: unifying language learning paradigms. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

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.

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.

Nai-Chi Yang, Wei-Yun Ma, and Pu-Jen Cheng. 2024. Plug-in language model: Controlling text generation with a simple regression model. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2165–2181, Mexico City, Mexico. Association for Computational Linguistics.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Zhiyuan Liu, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. GLM-130B: an open bilingual pre-trained model. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

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.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, and 1 others. 2022. Opt: Open pre-trained transformer language models. *ArXiv preprint*, abs/2205.01068.

A Representative Sample Comparison

Table 7 shows generated texts for four sets of target attributes.

B Datasets Used

The datasets used in our experiments 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. 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 FeverNLI (Thorne et al., 2018).

C List of Linguistic Attributes

We use expert-crafted linguistic complexity attributes as the control attributes for CTG. 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. Table 8 lists all the attributes that we use.

D Additional Experimental Details

D.1 Training Details

LingGen is trained using LoRA with parameters $r = 64$ and $\alpha = 128$, and a batch size of 140, using the AdamW optimizer over 3 epochs. The model is selected based on the best validation step. We set a maximum sequence length of 100 tokens, and train on a single A100 40GB GPU.

We evaluate two variants of MCTune: one using **Llama 2-7B** as the base model, and another using **OPT-350M** and trained on the same dataset 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 only takes 18 GPU hours. Subsequently, MCTune was trained on an HPC using 12 GPUs.

D.2 Baseline Hyperparameters

We tune the hyperparameters of all baselines using grid search over common ranges suggested in their respective papers and report the best performing configuration found on our validation set. The key hyperparameters used in our experiments are shown in Table 9.

For Guided Decoding, we adapt the logit adjustment formula of FUDGE (Yang and Klein, 2021) to our multi-attribute control task.

$$\log p + \lambda \times \varphi, \tag{2}$$

where φ is the favor signal. Due to the varying difficulty and MSE scales across different attribute sets, we set

$$\varphi = 1/\text{MSE} \tag{3}$$

(as lower MSE is better) and then normalize it per sample as

$$\varphi_{\text{norm}} = \frac{\varphi - \varphi_{\min}}{\varphi_{\max} - \varphi_{\min}}. \tag{4}$$

This normalization makes the weighting parameter λ less sensitive to the scale of individual attribute errors.

E Ablation Studies

To understand the contributions of our proposed P-MASKING strategy, the impact of different base models, and the effect of attribute integration methods, we conducted 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.

Sample Task	PTG (Baseline)	Llama 3.1 (Prompted)	LingGen (Ours)
Task 1: Deep Control. Target: *Sentences (4), Soph. Words (11), Complex Nominals (7)*			
Text	*...and is a web-based application that allows you to manage your business... For example, Firefox, Chrome, Opera, Safari, Internet Explorer, Mozilla, etc.*	*The intricate complexities of modern society necessitate an elevated comprehension of nuanced dialectics... an examination of societal hierarchies reveals a penchant for grandiose rhetoric.*	*The new version of the "Sonic" is now available for download. This entry was posted on Friday, June 1st, 2018... Pinging is currently not allowed.*
Analysis	✓ `Sentences`: 4 → **4** ✗ `Soph. Words`: 11 → **9** ✗ `Complex Nom.`: 7 → **5** *Fluency Failure: Incoherent.*	✓ `Sentences`: 4 → **4** ✗ `Soph. Words`: 11 → **30**	✓ `Sentences`: 4 → **4** ✗ `Soph. Words`: 11 → **9** ✗ `Complex Nom.`: 7 → **4**
Task 2: Simple Lexical Constraint. Target: *Soph. Word Count (0), Unique Word Ratio (1)*			
Text	*The following is a list of all the articles on the website. This entry was posted in Uncategorized and tagged American History...*	*cat the dog*	*A man is sitting down on his bike.*
Analysis	✗ `Soph. Words`: 0 → **10** ✗ `Unique Ratio`: 1.0 → **0.73** *Fluency Failure: Catastrophic collapse.*	✓ `Soph. Words`: 0 → **0** ✓ `Unique Ratio`: 1.0 → **1.0**	✓ `Soph. Words`: 0 → **0** ✓ `Unique Ratio`: 1.0 → **1.0** *Fluent and coherent.*
Task 3: Syntactic Structure. Target: *Sentences (2), Dependent Clauses (3), Words/Sentence (12.5)*			
Text	*This is a great way to get your kids involved in the arts and crafts! ... For ages 3-5.*	*The sun was shining brightly in the clear blue sky. The birds were singing their sweet melodies from the trees.*	*The first episode of the new series, The Vampire Diaries. This is the first episode of The Vampire Diaries.*
Analysis	✗ `Sentences`: 2 → **4** ✗ `Dep. Clauses`: 3 → **0**	✓ `Sentences`: 2 → **2** ✗ `Dep. Clauses`: 3 → **0**	✓ `Sentences`: 2 → **2** ✗ `Dep. Clauses`: 3 → **0**
Task 4: Surface Constraints. Target: *Sentences (1), Words (5), Characters (21)*			
Text	*http://www.c-c.org/c/b/p/n/d/...*	*They are very happy now.*	*There are people on bikes.*
Analysis	✗ `Sentences`: 1 → **0** ✗ `Words`: 5 → **1** ✗ `Characters`: 21 → **131** *Fluency Failure: Generated a URL.*	✓ `Sentences`: 1 → **1** ✗ `Words`: 5 → **4** ✗ `Characters`: 21 → **22**	✓ `Sentences`: 1 → **1** ✓ `Words`: 5 → **5** ✗ `Characters`: 21 → **25**

Table 7: Quantitative analysis of model performance on four diverse generation tasks. Values are computed on the displayed excerpts. For discrete targets, ✓denotes an exact match; ✗denotes a mismatch or a clear fluency failure. The examples illustrate both successes and failures across methods.

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 Acquisition 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 8: Linguistic attributes used in this paper.

- **LingGen (Fixed Rate)**: A fixed masking rate of 0.3 is applied, providing a consistent level of attribute masking.

- **LingGen (P-MASKING)**: Our proposed P-MASKING strategy, which adapts the masking rate based on a power law distribution.

Table 10 shows that P-MASKING achieves lower MSE (0.90) than No Masking (1.01) and Fixed Rate masking (1.13), while also maintaining the lowest perplexity among these variants (16.3).

Method	Hyperparameter	Value
Guided Decoding	λ	5
PPLM	window_length	5
	grad_length	20
	stepsize	0.01
	gamma	1
	num_iterations	10
COLD	stepsize	0.1
	constraint-weight	0.5
	topk	10
	num-iters	2000
BOLT	learning_rate	0.05
Mix&Match	max_iter	8
	n_samples	2
	alpha	10
	beta	1
	temperature	1
PiLM	stepsize	0.1
	M	2
	future_n_tokens	5
	ppl_weight	-0.3
MCTune	batch_size	48
	lr	2e-5
	warmup_steps	1000
PTG	batch_size	320
	lr	5e-5

Table 9: Hyperparameters used for baseline methods.

Method	MSE ↓	PPL ↓
No Masking	1.01	17.4
Fixed Rate	1.13	16.4
P-MASKING	**0.90**	**16.3**

Table 10: Comparison of masking strategies using OPT-350M base model. P-MASKING achieves the lowest MSE and perplexity among the masking variants.

Impact of Base Model We further evaluated LingGen with our proposed P-MASKING strategy using different base LLMs, specifically GPT-2 (Radford et al., 2019) and Pythia-410M (Biderman et al., 2023).

As shown in Table 11, P-MASKING yields lower MSE than both No Masking and Fixed Rate masking for GPT-2 and Pythia-410M.

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 BOS): Our proposed method, where the encoded attribute representation is added to the BOS token embedding.

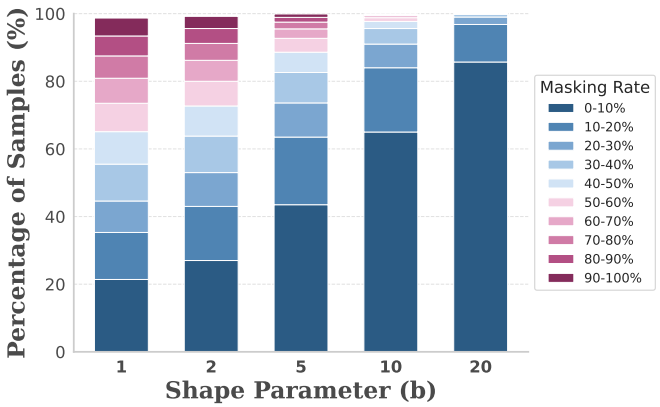

Figure 3: Effect of the Pareto shape parameter b on the distribution of sampled masking proportions. Each bar aggregates Monte Carlo samples into masking-rate buckets, with darker segments indicating higher masking.

Model	MSE ↓	PPL ↓
Pythia-410M		
No Masking	2.58	14.4
Fixed Rate	2.39	15.6
P-MASKING	**2.04**	**16.3**
GPT-2		
No Masking	2.69	**19.8**
Fixed Rate	3.75	32.9
P-MASKING	**2.47**	26.2

Table 11: Performance comparison of masking strategies across different base models (Pythia-410M, GPT-2). P-MASKING generally yields lower MSE compared to No Masking or Fixed Rate masking.

Integration Variant	MSE ↓	PPL ↓
No Masking		
BOS (ours)	**1.01**	**17.4**
All	2.60	768.9
Output	1.11	19.4
Logits	1.52	19.8
P-MASKING		
BOS (ours)	**0.90**	**16.3**
All	3.52	733.8
Output	1.76	16.3
Logits	1.31	16.4

Table 12: Evaluation of attribute integration methods (using OPT-350M base model), with and without P-MASKING. Adding attribute information to the BOS token consistently provides the best MSE and competitive perplexity.

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. These methods were evaluated both with and without P-MASKING, using the OPT-350M base model. As shown in Table 12, adding the encoded attribute representation to the BOS token embedding yields the lowest MSE in both settings. Notably, adding the attribute representation to all tokens produces extremely high perplexity (733–769), while BOS injection preserves competitive perplexity.

F Pareto Shape Parameter Study

Figure 3 compares candidate shape parameters by plotting the empirical masking-rate histograms (10k draws per setting). Smaller b values allocate substantial mass to high masking rates, which reduces the model's exposure to dense attribute configurations and harms control accuracy. Conversely, very large b values collapse the distribution toward minimal masking, curtailing the sparse cases needed for robustness. The selected $b = 5$ concentrates a clear majority of samples in the 0–30% bucket while leaving a visible tail across the higher buckets.

G Human Evaluation Details

To complement the automatic metrics (MSE and PPL), we conducted a human evaluation focusing on the fluency of the generated text.

We randomly selected 40 unique prompts (sets of target linguistic attributes) from our test set of 2,000 prompts. For each of these 40 prompts, we took the corresponding generated text output from the top-performing fine-tuned models identi-

fied in our experiments (BOLT, PiLM, PTG, and LingGen). Each of these generated text samples was independently rated by two human annotators, who are members of the institution conducting this research. Annotators were asked to evaluate the fluency of each text on a 5-point Likert scale, ranging from 1 (Not fluent at all) to 5 (Perfectly fluent). The average scores for each method are reported in Table 3. Cohen's Kappa was 0.32, which indicates fair agreement.

It is worth noting the potential discrepancy between automated perplexity (PPL) scores (Table 2) and human fluency judgments (Table 3). While PPL measures the model's confidence in predicting the sequence based on a general language model (GPT2-XL), human evaluation captures aspects like grammatical correctness, coherence, and naturalness. For instance, a model might generate repetitive or simplistic text that achieves low PPL but is rated lower by humans, or conversely, generate more complex but slightly awkward sentences that humans find less fluent despite potentially reasonable PPL. In our results, LingGen achieves the highest human fluency score, even though some models like PPLM or BOLT report slightly lower PPL values, highlighting the value of human assessment alongside automatic metrics.

The annotators were provided with the following guidelines to ensure consistency in their evaluations:

Human Evaluation Guidelines: Fluency

Task Overview: You will evaluate texts based on their **fluency** (how easy and natural they are to read).

Definition: Fluency measures whether a text reads smoothly and naturally, considering both grammatical correctness and logical flow. A fluent text should be effortless to process, with no structural awkwardness or confusing word choices.

Rating Scale:

Score 5 (Perfectly fluent): No grammatical errors, completely natural and effortless to read.

Example: "After finishing her work, she decided to take a walk in the park to enjoy the beautiful weather."

Score 4 (Very fluent): Minor imperfections that barely affect readability (e.g., slightly awkward phrasing, but no grammatical errors).

Example: "Having completed her work, she took a walk in the park for enjoying the weather."

Score 3 (Moderately fluent): Noticeable issues but main meaning is clear. May have minor grammatical errors or somewhat awkward phrasing that requires brief re-reading.

Example: "After she finish her work, she decide to walk in the park because nice weather."

Score 2 (Slightly fluent): Multiple grammatical errors or incoherent structure. Requires significant effort to understand, but meaning can be extracted.

Example: "She finishing work and walk park. Weather it was nice for her."

Score 1 (Not fluent at all): Severely broken grammar or completely incoherent. Nearly impossible to understand the intended meaning.

Example: "Work finish she park walking nice it weather the."

Decision Rules:

- **When in doubt:** Consider how much effort is required to understand the text. Choose a lower score if you had to re-read or mentally correct the text.
- **Grammar vs. coherence trade-off:** If grammar is perfect but logic is flawed (or vice versa), cap the score at 3. Both dimensions must be strong for scores of 4 or 5.
- **Context-specific:** Consider whether phrasing is natural for the specific linguistic context being evaluated (e.g., language learning level, domain-specific writing).

Important Notes:

- **Do not** consider factual correctness, content quality, or completeness. Only focus on linguistic fluency.
- **Do:** Use the anchor examples as reference points throughout your evaluation.