Palette of Language Models: A Solver for Controlled Text Generation

Zhe Yang,Yi Huang*, Yaqin Chen, Xiaoting Wu, Junlan Feng and Chao Deng
JIUTIAN Team, China Mobile Research Institute
{yangzhe,huangyi,chenyaqin,wuxiaoting,fengjunlan,dengchao}@chinamobile.com

Abstract

Recent advancements in large language models have revolutionized text generation with their remarkable capabilities. These models can produce controlled texts that closely adhere to specific requirements when prompted appropriately. However, designing an optimal prompt to control multiple attributes simultaneously can be challenging. A common approach is to linearly combine single-attribute models, but this strategy often overlooks attribute overlaps and can lead to conflicts. Therefore, we propose a novel combination strategy inspired by *the Law of Total Probability* and *Conditional Mutual Information Minimization* on generative language models. This method has been adapted for single-attribute control scenario and is termed the **Palette of Language Models** due to its theoretical linkage between attribute strength and generation style, akin to blending colors on an artist's palette. Moreover, *positive correlation* and *attribute enhancement* are advanced as theoretical properties to guide a rational combination strategy design. We conduct experiments on both single control and multiple control settings, and achieve surpassing results.

1 Introduction

The purpose of controlled text generation is to modify the output of the language models with a pregiven attribute, so that the final output conforms to the attribute (Hu et al., 2017; Madaan et al., 2021; Zhang et al., 2023). It is common to utilize Bayes Rules (Li et al., 2022b) to modify the language model $p(X)$ to form with the conditional variable $p(X|a)$, and by virtue of the corresponding discriminative model (generally a classification model), implement the constraints on the generated results. This approach will face two problems: On the one hand, the discriminative model usually needs to be fine-tuned according to the generation task scenario to better assist controlled text generation, because the accuracy of the discriminative model plays a key role in the generation effect. Nevertheless, it is time-consuming to collect the corresponding classification data. On the other hand, the classification effect of the discriminative model often depends on certain words or phrases related to attributes, therefore, in the process of predicting the next token, the discriminative model will guide the language model to bias these words, which makes the generated results lack diversity.

In recent years, with the rapid development of large language models (Chowdhery et al., 2022; Du et al., 2022; Hoffmann et al., 2022), models represented by ChatGPT (Achiam and et.al, 2023) have powerful text generation capabilities. Many generation tasks can be converted to prompt engineering to use large language models to get better solutions. Similarly, attributes can be designed as prompts, so that with the powerful generation ability of the large language model, the final generated text will also conform to such attributes. When the number of attributes that need to be met is large, it is difficult to design a suitable prompt to cover these attributes. Also, due to the inherent ambiguity and sensitivity of prompts, we can't guarantee that the generated results will be perfectly reproduced according to the attributes mentioned in prompts.

Model Arithmetic (Dekoninck et al., 2024) proposes a framework to ensemble multiple attributes, in which through simple arithmetic operations, such as linear composition, multiple attribute-related language models or discriminative models are combined to obtain a multi-attribute controlled text generation strategy. This framework treats the language models associated with different attributes as independent. However, in reality, each language model may have multiple attributes, and the attributes between language models may overlap which can not be effectively modeled with simple arithmetic operations. For example, the

*Corresponding authors

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 9868–9881

main attribute of language model A is "formal response", and that of language model B is "child's tone" (which suggests an attribute of "informal response" as well). With linear combination, the attribute of "formal response" is faded, making the final generation strategy incomplete.

Deriving from *the Law of Total Probability* and *Conditional Mutual Information Minimization*, we propose **Palette of Language Models**, which alleviates the latent attribute overlapping problem when attributes assembled. The improved combination strategy has following contributions:

• Leveraging *the Law of Total Probability*, we decompose the final generation distribution as attribute satisfied event and its complementary event of which the latter never appears in previous works.

• We model the attributes overlapping problem as *Conditional Mutual Information Minimization*, with which a dynamic coefficient on each attribute (as a part of attribute strength) is derived for attribute enhancement. We also take theoretical analysis on attribute strength and conclude the positive correlation between it and the final generation style.

• We propose two pieces of theoretical properties (*positive correlation* and *attribute enhancement*) which guide a rational attribute combination.

2 Related work

Some works for controlling language models are training or aligning language models conditioned on static or iterative updated control codes that make desired features of generated text more explicit (Keskar et al., 2019; Zhang et al., 2020; Lu et al., 2022). These methods often depend on a large amount of training data, which will result in a lot of annotation costs and resource consumption. In order to solve the above problems, some researchers have proposed methods of fine-tuning (Bender et al., 2021; Yuan et al., 2023), prompt tuning (Li et al., 2022a; Pagnoni et al., 2021) or Prefix Tuning (Qian et al., 2022; Clive et al., 2022; Ma et al., 2023) large models to generate controlled text. However, due to the use of GPT series models, it will cause infeasible and cannot solve the toxicity and bias problems of the model (Tonmoy et al., 2024). Although Timo Schick et al. (Schick et al., 2021) found that pre-trained models can largely recognize their bad biases and the toxicity of the content they produce, by giving textual descriptions of bad behavior when decoding, the probability of generating problematic text can be reduced. But this method is greedy in nature when supporting or opposing a decision must always use a specific word, taking into account only the context in which it has been generated.

Another category of method uses gradients of single or combination of attribute discriminators based on energy to update generating language model's hidden states to guide decoding process (Dathathri et al., 2020; Mireshghallah et al., 2022; Qin et al., 2022; Kumar et al., 2022, 2021), which do not need labeled training data. Despite the use of Langevin dynamics, samples the sequence of token embeddings instead of logits and Gibbs sampling methods. The main issue of this category is that multiple sampling iterations are required to converge, which slow down the decoding progress. BOLT (Liu et al., 2023), instead, preserves token dependencies and simultaneously optimizes via autoregressive decoding to constrain by adding biases. But BOLT requires stricter constraints, for example, keyword control requires more than three keywords. BOLT also requires careful tuning of the different hyperparameters constituting the energy function—a problem prevalent in energy-based controlled power generation approaches.

Some methods are inspired by Bayes Rules that use attribute probability from discriminators to steer language generation towards desired attributes. The attribute discriminators can be discriminative or generative. FUDGE (Yang and Klein, 2021) learns a binary discriminator for whether attribute will become true in future, and the output probabilities of this discriminator are multiplied with generator's original probabilities to get the desired probabilities. Gedi (Krause et al., 2021) uses class-conditional language models as generative discriminators, which results in faster computation speed due to the parallel computation of all candidate tokens. Compared with these methods, PREADD (Pei et al., 2023) considers the language generator with a prefix-prepended prompt as the role of attribute discriminator, which does not require an external model and corresponding additional training data.

Recent work proposes controlling text generation via language model arithmetic (Dekoninck et al., 2024), which enables to combine multiple language models of different attributes into a formula-based composite. The key distinguishing feature of our method is that we explicitly model the attributes overlapping between language models, so that they are not faded during combination.

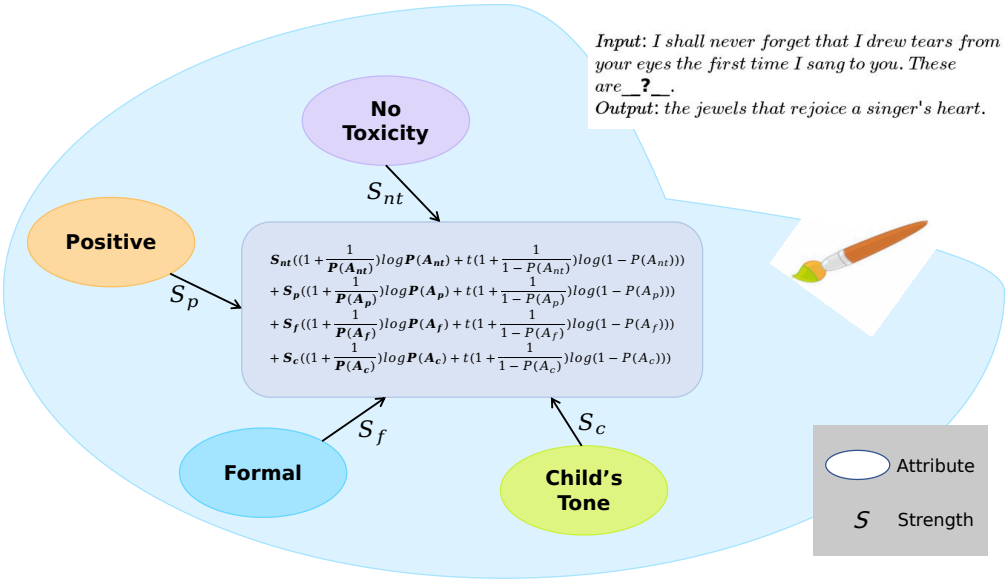

Figure 1: Overview of **Palette of Language Models**. Each ellipse in the figure represents a generative language model with a specific attribute, and S represents the strength of the corresponding model. Employing Equation 8, the final generation under multiple constraints is derived.

3 Proposed method

In this paper, we focus on **n** (where $n >= 2$) attributes combination strategy to derive a specific output distribution on controlled text generation which indicates a mixture style. Analogously, single attribute control can be treated as the combination with the basic model. We attach each attribute to a generative language model for ease of the formula derivation.

3.1 Problem definition

Assuming **n** generative language models, with each owning a certain attribute, such as "able to generate text with positive sentiment", "speak in a child's tone", or "tell a topic about the solar system", etc. For the **i**th model A_i, the prediction probability of the next token is $p_{A_i}(x_t = x|x_{1:t-1})$, abbreviated as $p(A_i = x)$ (where $x \in V$, the vocabulary of current language model). Thus, we desire a function $\mathcal{F}$ on $p(A_i)$ and express the final distribution as:

$$\begin{aligned} p(Z) &= \mathcal{F}(p(A_1), ..., p(A_i), ..., p(A_n)) \\ \textbf{s.t. } & p(A_i, A_j) = p(A_i)p(A_j), \\ & min(\mathcal{M}(A_i, A_j)) \ (for\ 1 \leq i, j \leq n) \end{aligned} \quad (1)$$

where $\mathcal{M}(.)$ is a metric to gauge the overlapping between an attribute couple under the final distribution. It is noteworthy that the joint distribution item, i.e., $p(A_i, A_j)$, means the probability that "the next predicated tokens for both attributes A_i and A_j are x". Obviously, they are separate neural networks and none inter-influence exists, which ensures the independence between them.

For the attribute couple (A_i, A_j), the *conditional mutual information* under the final distribution Z is always no less than 0. And **if it is greater than 0, the overlapping between them happens**. Therefore, the overlapping metric $\mathcal{M}(.)$ mentioned in Equation 1 is defined as:

$$\begin{aligned} \mathcal{M}(A_i, A_j) &= \mathbf{I}(A_i, A_j|Z) = \\ \sum_z p(z) \sum_{a_i, a_j} p(a_i, a_j|z) log \frac{p(a_i, a_j|z)}{p(a_i|z)p(a_j|z)} &\geq 0 \end{aligned} \quad (2)$$

where $z \in Z$, $a_i \in A_i$ and $a_j \in A_j$ are satisfied.

3.2 Proposed solution

For the final generation distribution, i.e., $p(Z = x)$, we use *the Law of Total Probability*, which, in the case of a single attribute A_i, can be written as:

$$\begin{aligned} p(Z = x) &= \lambda_i * p(A_i = x) + \lambda_{i'} * p(A_i \neq x) \\ \textbf{s.t. } & \lambda_i = p(Z = x|A_i = x), \\ & \lambda_{i'} = p(Z = x|A_i \neq x) \end{aligned} \quad (3)$$

Similarly, for the attribute couple (A_i, A_j), taking into account the independence between them (see Equation 1), the final generation strategy can be written as:

$$\begin{aligned} p(Z=x) &= \lambda_{ij} * p(A_i=x)p(A_j=x) \\ &+\lambda_{i'j'} * p(A_i\neq x)p(A_j\neq x) \\ &+\lambda_{ij'} * p(A_i=x)p(A_j\neq x) \\ &+\lambda_{i'j} * p(A_i\neq x)p(A_j=x) \\ \textbf{s.t. } \lambda_{ij} &= p(Z=x|A_i=x, A_j=x), \\ \lambda_{i'j'} &= p(Z=x|A_i\neq x, A_j\neq x), \\ \lambda_{ij'} &= p(Z=x|A_i=x, A_j\neq x), \\ \lambda_{i'j} &= p(Z=x|A_i\neq x, A_j=x) \end{aligned} \tag{4}$$

Employing the convexity of the logarithmic function, i.e., $log(ax+by)-log(a+b) \geq \frac{a}{a+b}log(x)+\frac{b}{a+b}log(y)$, we add both single and couple factorization equations aforementioned, and obtain an approximate expression (details are shown in Section A):

$$\begin{aligned} log3p(Z) &\approx \sum_{s\in\{i,j\}} \alpha_s logp(A_s=x) \\ &+ \sum_{s\in\{i,j\}} \beta_s logp(A_s\neq x), \\ \textbf{s.t. } \alpha_i &= \lambda_i+\lambda_{ij}+\lambda_{ij'},\ \beta_i=\lambda_{i'}+\lambda_{i'j}+\lambda_{i'j'} \end{aligned} \tag{5}$$

Additionally, by minimizing the conditional mutual information in Equation 2, we derive that (details are shown in Section B):

$$\begin{aligned} \lambda_{ij} &= \frac{\lambda_i\lambda_j}{p(Z=x)}, \lambda_{i'j'} = \frac{\lambda_{i'}\lambda_{j'}}{p(Z=x)}, \\ \lambda_{ij'} &= \frac{\lambda_i\lambda_{j'}}{p(Z=x)}, \lambda_{i'j} = \frac{\lambda_{i'}\lambda_j}{p(Z=x)} \end{aligned} \tag{6}$$

We traverse the pairwise combinations of **n** attributes (altogether C_n^2 items) and add them in the form of Equation 5:

$$\begin{aligned} logp(Z=x) &\propto \sum_{i=1}^{n} \varphi_i logp(A_i=x) \\ &+ \sum_{i=1}^{n} \omega_i logp(A_i\neq x) \\ \textbf{s.t. } \varphi_i &= \frac{(n-1)\lambda_i+\frac{p(A_i=x|Z=x)}{p(A_i=x)}\sum_{j\neq i}(\lambda_j+\lambda_{j'})}{C_n^2}, \\ \omega_i &= \frac{(n-1)\lambda_{i'}+\frac{p(A_i\neq x|Z=x)}{p(A_i\neq x)}\sum_{j\neq i}(\lambda_j+\lambda_{j'})}{C_n^2} \end{aligned} \tag{7}$$

Inspired by previous works, we introduce the generative language model with standard output (which means no bias to any specific attributes) as the basic part in aid of generation stability, and simplify the expression in Equation 7 (details are shown in Section C). Therefore, both $single$ and $multiple$ attributes can achieve the final generation solution in a consolidated manner (where the $single$ setting can be treated as the combination with the basic part):

$$\begin{aligned} logp(Z=x) &\propto \frac{\sum_{i=1}^{n} s_i c_i logp(A_i=x) + logP_b}{M_1} \\ &+t\ \frac{\sum_{i=1}^{n} s_{i'} c_{i'} logp(A_i\neq x)}{M_2} \\ \textbf{s.t. } c_i &= 1+\frac{1}{p(A_i=x)},\ c_{i'} = 1+\frac{1}{p(A_i\neq x)} \end{aligned} \tag{8}$$

where M_1 and M_2 are normalization values with which the combination logits will be as the same order of magnitude as the basic language model. P_b is the generation probability of the basic part. s_i and $s_{i'}$ are variables to express the strength (the proportion to the final generation) of the current attribute. t is a small coefficient which servers the complementary event, i.e., $p(A_i \neq x)$, as an auxiliary for the final generation strategy.

3.3 Strategy properties

Property 1. *It demonstrates a positive correlation between attribute strength, i.e., s_i, and the final generation style about our improved attribute combination strategy.* (The proof is shown in Section D)

Specifically, we introduce the attribute token $x_{attr} \in V$ (vocabulary of the language model) which can better express the property of current model, hence:

$$p(A_i=x_{attr}) > p(A_i=x_t)\ (\forall t \neq attr) \tag{9}$$

Property 2. *Our strategy gets attribute enhancement compared to linear combination strategy.* (The proof is shown in Section D)

As for two combination strategies $\mathcal{F}_1$ and $\mathcal{F}_2$, the attribute A_i get enhancement from $\mathcal{F}_1$ to $\mathcal{F}_2$, if:

$$p(Z_2=x_{attr}) > p(Z_1=x_{attr}) \tag{10}$$

where $p(Z_k)$ means the final generation probability over strategy $\mathcal{F}_k$, x_{attr} is the attribute token of A_i.

Attribute	Prompt
reducing toxicity	The following conversation is one that does not perpetuate negative stereotypes, is not threatening and does not contain any sexually explicit or profane language.
enabling positive sentiment	The following conversation is one that is happy and positive.
enabling fluency	Continue writing after the conversation to make it more fluency and readable.

Table 1: Attributes and prompts for toxicity reduction.

	Llama2-7b		Pythia-12b		MPT-7b	
	Tox.	Perpl.	Tox.	Perpl.	Tox.	Perpl.
M	0.270	12.87	0.250	21.29	0.266	**18.61**
Self-Debiasing ($\lambda = 10$)	0.257	14.25	0.241	25.37	0.268	21.19
FUDGE ($M + C$)	0.233	13.56	0.208	21.80	0.225	19.53
PREADD (M–$0.6M_{toxic}$)	0.215	12.66	**0.176**	32.88	**0.200**	23.01
Linear (M–$0.96union(M_{toxic}, M)$)	0.199	10.61	0.179	22.74	0.204	19.32
Ours	**0.159**	**9.42**	0.186	**20.34**	0.213	24.07

Table 2: Toxicity and perplexity of various methods on the /pol/ dataset. M and M_{toxic} denotes the methods without/with conditioning to toxicity respectively. C is a toxicity classifier. Perplexity is measured with respect to **M**. Lower is better.

4 Experiments

We evaluate the improved attribute combination strategy in both $single$ and $multiple$ control scenarios to testify the attribute enhancement. Specifically, Sections 4.1 and 4.2 are for the single setting, and the Section 4.4 is for multi-attribute control which also demonstrates the overlapping alleviation. We conduct experiment for positive correlation and complementary event verification in Sections 4.3. All the experiments are implemented on a single GPU of *Tesla V100S-PCIE-32GB*.

Base models. We conduct experiments on several popular autoregressive large language models for strategy evaluation: Llama2-7b[1](Touvron et al., 2023), Pythia-12b[2](Biderman et al., 2023) and MPT-7b[3](Team, 2023).

Baselines for comparison. The baselines we make comparison with are as follows: **M**(the basic language model without any prompt), **Self-Debiasing (Schick et al., 2021)**, **FUDGE (Yang and Klein, 2021)**, **PREADD (Pei et al., 2023)** and **Linear combination (Dekoninck et al., 2024)** which is derived from the KL-Optimality and satisfies:

$$logp(Z = x) \propto \sum_{i=1}^{n} \lambda_i logp(A_i = x) \qquad (11)$$

Linear combination is a concise strategy to merge multiple attributes, and makes operation on λ_i to bias the overall output toward ($\lambda_i > 0$) or away from ($\lambda_i < 0$) the attribute A_i. However, it does neglect the overlapping between attributes which might cause attribute conflict when combination.

As for the linear combination method, we employ its best ARITHMETIC strategy, $M - 0.96Union(.)$, for comparison.

Experiment Setting. We notice that implementation of the standard normalization M in Equation 8 remains elusive due to the fact that the variable c_i tends to infinite if $p(A_i = x)$ is trivial. With superseding it, we introduce $sigmoid$ function ($\sigma(p(A_i) = x)$) for its keeping the major properties in Section 3.3 (details are shown in Section F). In addition, instead of training language models with a specific attribute from scratch, we induce the attribute in the basic language model with the **prompt** engineering. Hence, variables for experi-

[1] https://huggingface.co/meta-llama/Llama-2-7b
[2] https://huggingface.co/EleutherAI/pythia-12b
[3] https://huggingface.co/mosaicml/mpt-7b

Negative ⟶ Positive	Llama2-7b		Pythia-12b		MPT-7b	
	Sentiment.	Perpl.	Sentiment.	Perpl.	Sentiment.	Perpl.
M	0.218	14.15	0.212	22.86	0.210	19.41
M_{pos}	0.239	13.69	0.244	21.69	0.244	18.25
Self-Debiasing ($\lambda = 10$)	0.270	14.99	0.244	25.17	0.251	19.79
FUDGE($M + C$)	0.339	13.73	0.337	22.63	0.326	18.81
PREADD (M–$0.6M_{pos}$)	0.373	14.20	**0.343**	30.86	0.343	19.25
Linear (M_{pos}–$0.96union(M_{pos}, M_{neg})$)	0.411	**12.82**	0.343	22.53	0.370	**17.88**
Ours	**0.426**	19.42	0.336	**15.33**	**0.405**	18.85
Positive ⟶ Negative	Llama2-7b		Pythia-12b		MPT-7b	
	Sentiment.	Perpl.	Sentiment.	Perpl.	Sentiment.	Perpl.
M	0.204	13.51	0.218	**22.25**	0.196	18.26
M_{neg}	0.299	14.17	0.340	22.73	0.284	17.83
Self-Debiasing ($\lambda = 10$)	0.339	15.02	0.364	25.59	0.285	20.02
FUDGE ($M + C$)	0.410	14.97	0.433	23.39	0.377	18.43
PREADD (M–$0.6M_{neg}$)	0.470	14.66	**0.514**	32.03	0.438	19.44
Linear (M_{neg}–$0.96union(M_{neg}, M_{pos})$)	0.502	**13.12**	0.499	24.93	0.452	18.23
Ours	**0.547**	16.31	0.469	30.41	**0.478**	**15.81**

Table 3: Sentiment score and Perplexity value of various methods on the IMDB dataset for **"negative to positive"** and **"positive to negative"** transition tasks. M_{pos} and M_{neg} denote the models with conditioning to positive/negative sentiments respectively. C is a sentiment classifier. Perplexity is measured with respect to **M**. Lower is better.

ments of the improved strategy are:

$$c_i = 1+\frac{1}{\sigma(p(A_i = x)}, \; c_{i'} = 1+\frac{1}{\sigma(p(A_i \neq x)},$$
$$M_1{=}1{+}(2+\frac{1}{e})\sum_{i=1}^{n} s_i, \; M_2{=}(2+\frac{1}{e})\sum_{i=1}^{n} s_{i'},$$
$$p(A_i = x) = P_b(x|prompt_{A_i}) \quad (12)$$

where $prompt_{A_i}$ are displayed in Tables 1, 4, 5 and 8.

4.1 Toxicity reduction

We first test our algorithm in terms of toxicity reduction on **/pol/** dataset (Papasavva et al.), which comprises contents from website 4chan[4] and attaches each item a toxicity score. We randomly select 2000 samples with their scores greater than 0.5. For toxicity reduction process, we construct a dialogue stuff pattern in which original toxicity texts are from the inquirer (i.e., **Person 1:**), and with that, the attribute mixture with combined strategy (as **Person 2:**) is compelled to generate toxicity-free utterance. We assemble three attributes of which the positive sentiment and fluency are as supplementaries for toxicity reduction. Each attribute and its corresponding prompt are in Table 1. The metrics picked in this setting are **Toxicity Score** (which measures the virulent degree by Perspective API[5]) and **PPL** (which estimates generation consistency according to perplexity calculation).

As is shown in Table 2, with Llama2-7b as the basic language model, both toxicity score and PPL value are at the first-rate where the toxicity probability degrades to a high-quality level with 4% compared to traditional SOTA methods, which embodies advancement of attribute enhancement in our algorithm. The proposed strategy perform a slight inferior (about 1% at a disadvantage in toxicity) than its comparators under Pythia-12b and MPT-7b settings.

4.2 Sentiment control

Attribute	**Prompt**
positive reply	The following is a positive movie review, with a very positive sentiment and a very positive tone.
negative reply	The following is a negative movie review, with a very negative sentiment and a very negative tone.

Table 4: Attributes and prompts for sentiment transition.

We make a subset of **IMDB movie review** dataset (Maas et al., 2011) with 1000 samples separately for both positive and negative sentiment control. Following the setting of Dekoninck et al., we also keep the first 32 tokens of original sen-

[4] https://boards.4chan.org/pol/
[5] https://perspectiveapi.com/

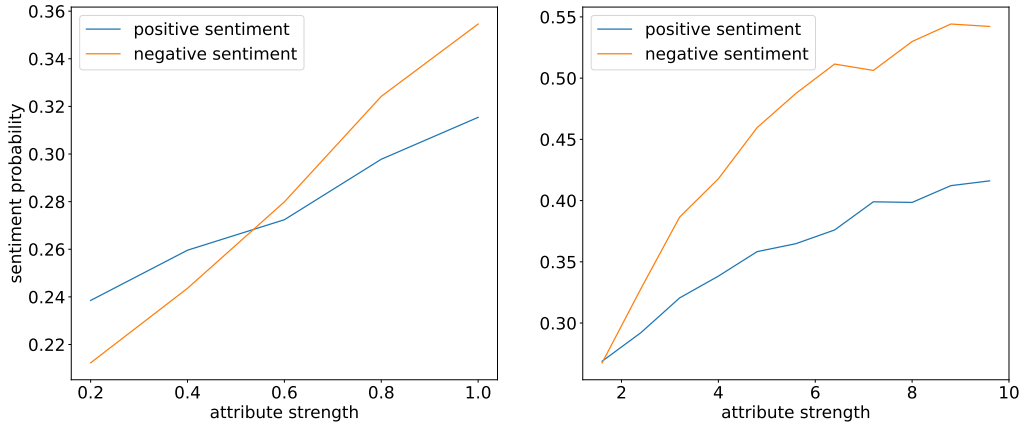

Figure 2: Positive Correlation between attribute strength & sentiment score (Left: s<1, Right: s>1).

tences retained and force language models to write after them with an opposite sentiment. This setting demonstrates hostile performance for language models owing to their generally decoding in complying with previous tokens (in both structural and semantic consistency). Inspired by (Liu et al., 2021; Dekoninck et al., 2024), we enhance the accent on required sentiment via deducting logits of its antagonistic stuff. That is:

$$logp(Z=x) \propto \frac{\sum f(i) * s_i c_i logp(A_i = x) + logP_b}{M_1} + t \frac{\sum f(i) * s_{i'} c_{i'} logp(A_i \neq x)}{M_2}$$

$$\textbf{s.t. } i \in \{main, anti\}, f(i) = \begin{cases} 1 & main \\ -1 & anti \end{cases} \quad (13)$$

For instance, if the required sentiment is **positive**, then **main** is for positive attribute and **anti** is for negative. The prompts for both positive and negative sentiments are enumerated in Table 4. Following Dekoninck et al., we take the **twitter-sentiment** discriminative model[6] for sentiment score derivation.

Referring to Table 3, with both Llama2-7b and MPT-7b as the basic models, our strategy exceeds other methods on the metric of sentiment score. Specifically, in Llama2-7b setting, we obtain an average score of 3% (1.5% for neg2pos and 4.5% for pos2neg transition) greater than the SOTA method. The values are 3.5% (neg2pos) and 2.6% (pos2neg) in MPT-7b setting. As for PPL value, our strategy demonstrates a modicum of weakness at a manageable level in which the average relative values on the SOTA method are $+4.9$, $+0.9$ and -0.5 for the three basic models. Therefore, without sacrificing the generation quality, our algorithm gains attribute enhancement in the sentiment control setting.

[6] https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment-latest

4.3 Evaluations on attribute strength s_i and the complementary event $A_i \neq x$

We verify the positive correlation property between attribute strength s and the generation style (see **Property 1** in Section 3.3) in this part. Following the setting of Section 4.2, we dynamically vary strength of the main attribute s_{main} in Equation 13 and chronicle sentiment scores. The basic model we choose is Llama2-7b. From Figure 2, we judge that the final generation performance inclines to ascend on account of gaining strength in both cases that $s < 1$ and $s > 1$.

As is processed in Equation 8, the complementary event $A_i \neq x$ is treated as the auxiliary for the main part $A_i = x$. We make evaluations of the impact it makes on the final generation style. Similar to Section 4.3, we also conduct experiments on sentiment control with Llama2-7b and vary the coefficient t of $logp(A_i \neq x)$ in Equation 13.

Shown in Figures 3 and 4 (see in Section G), the dash lines represent sentiment probabilities with $t = 0$ (which equal to 0.4161 and 0.5442 in positive & negative settings, separately). From both figures, we discover that the final generation performance fluctuates nearby the dash line with the value of t.

4.4 Multi-attribute control

In this part, we utilize a dual-attribute control setting (toxicity & sentiment) to evaluate the overlap-

Settings	Attribute	Prompt
Setting one	**toxicity**	The following movie review is not threatening and does not contain any sexually explicit or profane language, and is with a slight **negative** sentiment.
	positive reply	Continue writing after the movie review to change it with a very **positive** sentiment and a very **positive** tone.
Setting two	**toxicity**	The following movie review is not threatening and does not contain any sexually explicit or profane language, and is with a slight **positive** sentiment.
	negative reply	Continue writing after the movie review to change it with a very **negative** sentiment and a very **negative** tone.

Table 5: Attributes and prompts for the multi-control with overlapping setting.

Positive		**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**	**Average**
Perplexity(↓)	**Linear**	14.95	15.48	15.36	15.84	15.70	15.17	15.71	15.57	15.51	15.86	15.51
	Union	13.72	14.25	13.50	13.97	13.82	14.16	13.71	13.98	13.95	13.90	13.90
	Ours	**6.40**	**6.54**	**6.63**	**6.52**	**6.60**	**6.55**	**6.66**	**6.42**	**6.60**	**6.68**	**6.56**
Sentiment(↑)	**Linear**	0.543	0.530	0.539	0.533	0.526	0.535	0.534	0.525	0.531	0.534	0.533
	Union	0.443	0.439	0.441	0.445	0.437	0.444	0.439	0.443	0.446	0.440	0.442
	Ours	**0.565**	**0.562**	**0.556**	**0.543**	**0.559**	**0.564**	**0.558**	**0.551**	**0.557**	**0.544**	**0.556**
Negative		**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**	**Average**
Perplexity(↓)	**Linear**	15.70	15.78	15.55	15.81	15.67	15.57	15.93	16.37	15.79	16.86	15.90
	Union	14.22	14.32	14.17	14.23	14.73	14.57	14.63	14.45	14.78	14.90	14.50
	Ours	**6.46**	**6.42**	**6.49**	**6.46**	**6.60**	**6.71**	**6.68**	**6.62**	**6.78**	**6.63**	**6.59**
Sentiment(↑)	**Linear**	0.570	0.557	0.551	0.558	0.555	0.553	0.556	0.557	0.548	0.554	0.556
	Union	0.452	0.455	0.474	0.475	0.465	0.462	0.469	0.457	0.466	0.474	0.465
	Ours	**0.574**	**0.571**	**0.575**	**0.581**	**0.569**	**0.571**	**0.569**	**0.562**	**0.553**	**0.560**	**0.569**

Table 6: Multi-attributes on positive sentiment and negative sentiment control evaluation.

ping alleviation in the multi-attribute combination scenario. Concretely, we conduct experiment on dataset exact to that in Section 4.2, nevertheless, we write after the first 32 tokens for consistency without any sentiment transition. We employ toxicity and sentiment as the control aspects, and manually add overlapping to bring conflict between them. Prompts for both the attributes are shown in Table 5. Taking **Setting one** for example, the **toxicity** attribute covers an extra antagonistic sentiment, i.e., *slight negative*, to the **positive reply** attribute, which begets overlapping when combined in a linear manner. We select Llama2-7b as the basic language model. According to Table 6, where the coefficients from 0.1 to 1.0 mean the relative strength ratio, i.e., s_1/s_2, our strategy shows superiority in both Positive and Negative sentiment control settings that the average relative scores on sentiment are $+2.3\%$ and $+1.3\%$. As for the **Union** baseline, we set the sentiment attribute as the base, and vary the coefficient of $max(toxi, senti)$ from 0.1 to 1.0. In comparison with it, our method still dominates in both control settings. Hence it proves that the improved attribute combination can diminish the attribute conflict degree to a certain extent.

Moreover, we design another overlapping type with toxicity attribute covering a same sentiment trend in comparison with the sentiment attribute (details are shown in Section H). Intuitively, we will derive a higher sentiment score in contrast to the previous design. In addition, the score growth of our method should fall short of the linear combination, as the consequence of that the enhanced overlapping should get deflated. Referring to Table 9, our method obtains increased sentiment scores of 0.189% and 0.192% by average on positive and negative sentiment settings separately, which are both less than the values, i.e., 0.577% and 0.694%, of the linear strategy.

Referring to $n > 2$ attributes combination, we incorporate a new attribute that influences the sentiment reply as well on the basis of setting one in Table 5:

> **Child's tone:** Writing after the movie review with mimicking the child's tone, and with some negative sentiment.

	0.2	0.4	0.6	0.8	1.0
Linear	0.531	0.530	0.528	0.521	0.527
Ours	**0.562**	**0.559**	**0.550**	**0.543**	**0.544**

Table 7: Attributes and prompts for sentiment transition.

where the coefficients above means $s_1/s_3 = s_2/s_3$ with also the *sentiment control* as the reference.

5 Conclusions

Considering underlying overlapping between attributes, we deduce **Palette of Language Models**, an improved linear combination strategy for multi-attribute control, with *the Law of Total Probability* and *Conditional Mutual Information Minimization*. Different from previous linear combination methods, the derived formula owns a dynamic coefficient to each model which can enhance the attribute expression. Additionally, two pivotal properties are proposed that serve as guiding principles for designing a rational attribute combination strategy. We conduct comprehensive experiments on both single and multiple attributes control settings which further underscore the effectiveness of our method.

Limitations

The **Palette of Language Models** we proposed has undergone relatively rigorous theoretical derivation and is suitable for autoregressive language models (with specific attributes) combination scenario. However, in actual applications, language models may be of different types, and the corresponding vocabulary may also vary. Therefore, we will solve the decoding problems caused by inconsistent vocabularies in the future work, trying to pave the way for more sophisticated and nuanced control mechanisms that can better cater to the extensive needs of language model applications.

Acknowledgements

This work is supported by Beijing Natural Science Foundation (L222006) and China Mobile Holistic Artificial Intelligence Major Project Funding (R22105ZS, R22105ZSC01).

References

OpenAI Josh Achiam and et.al. 2023. Gpt-4 technical report.

Emily M. Bender, Timnit Gebru, Angelina Mcmillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*.

Stella Biderman, Hailey Schoelkopf, Quentin G. 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. *ArXiv*, abs/2304.01373.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. *Preprint*, arXiv:2204.02311.

Jordan Clive, Kris Cao, and Marek Rei. 2022. Control prefixes for parameter-efficient text generation. *Preprint*, arXiv:2110.08329.

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.

Jasper Dekoninck, Marc Fischer, Luca Beurer-Kellner, and Martin Vechev. 2024. Controlled text generation via language model arithmetic. In *The Twelfth International Conference on Learning Representations*.

Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten Bosma, Zongwei Zhou, Tao Wang, Yu Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc V Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. 2022. Glam: Efficient scaling of language models with mixture-of-experts. *Preprint*, arXiv:2112.06905.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford,

Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training compute-optimal large language models. *Preprint*, arXiv:2203.15556.

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*, volume 70 of *Proceedings of Machine Learning Research*, pages 1587–1596. PMLR.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A conditional transformer language model for controllable generation. *CoRR*, abs/1909.05858.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2021. GeDi: Generative discriminator guided sequence generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4929–4952, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sachin Kumar, Eric Malmi, Aliaksei Severyn, and Yulia Tsvetkov. 2021. Controlled text generation as continuous optimization with multiple constraints. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 14542–14554.

Sachin Kumar, Biswajit Paria, and Yulia Tsvetkov. 2022. Gradient-based constrained sampling from language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 2251–2277. Association for Computational Linguistics.

Junyi Li, Tianyi Tang, Jian-Yun Nie, Ji-Rong Wen, and Wayne Xin Zhao. 2022a. Learning to transfer prompts for text generation. *Preprint*, arXiv:2205.01543.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. 2022b. Diffusion-lm improves controllable text generation. *Preprint*, arXiv:2205.14217.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021. DExperts: Decoding-time controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6691–6706, Online. Association for Computational Linguistics.

Xin Liu, Muhammad Khalifa, and Lu Wang. 2023. Bolt: Fast energy-based controlled text generation with tunable biases. *Preprint*, arXiv:2305.12018.

Ximing Lu, Sean Welleck, Jack Hessel, Liwei Jiang, Lianhui Qin, Peter West, Prithviraj Ammanabrolu, and Yejin Choi. 2022. Quark: Controllable text generation with reinforced unlearning. In *Advances in Neural Information Processing Systems*, volume 35, pages 27591–27609. Curran Associates, Inc.

Congda Ma, Tianyu Zhao, Makoto Shing, Kei Sawada, and Manabu Okumura. 2023. Focused prefix tuning for controllable text generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1116–1127, Toronto, Canada. Association for Computational Linguistics.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Nishtha Madaan, Inkit Padhi, Naveen Panwar, and Diptikalyan Saha. 2021. Generate your counterfactuals: Towards controlled counterfactual generation for text. *Preprint*, arXiv:2012.04698.

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), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 401–415. Association for Computational Linguistics.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with frank: A benchmark for factuality metrics. *Preprint*, arXiv:2104.13346.

A Papasavva, S. Zannettou, E. De Cristofaro, G. Stringhini, and J. Blackburn. Raiders of the lost kek: 3.5 years of augmented 4chan posts from the politically incorrect board. *Proceedings of the International AAAI Conference on Weblogs and Social Media*.

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, Toronto, Canada, July 9-14, 2023*, pages 10018–10037. Association for Computational Linguistics.

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

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*.

Timo Schick, Sahana Udupa, and Hinrich Schütze. 2021. Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in nlp. *Preprint*, arXiv:2103.00453.

MosaicML NLP Team. 2023. Introducing mpt-7b: A new standard for open-source, commercially usable llms. Accessed: 2023-05-05.

S. M Towhidul Islam Tonmoy, S M Mehedi Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. 2024. A comprehensive survey of hallucination mitigation techniques in large language models. *Preprint*, arXiv:2401.01313.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

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

Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, and Songfang Huang. 2023. How well do large language models perform in arithmetic tasks? *Preprint*, arXiv:2304.02015.

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. *Preprint*, arXiv:2201.05337.

Yizhe Zhang, Guoyin Wang, Chunyuan Li, Zhe Gan, Chris Brockett, and Bill Dolan. 2020. POINTER: constrained text generation via insertion-based generative pre-training. *CoRR*, abs/2005.00558.

A Details for Approximation Combination over Attribute Couple

We add single factorization on attributes $A_i \& A_j$ with the couple factorization on them, and get:

$$\begin{aligned}3 * p(Z) &= \lambda_i * p(A_i = x) + \lambda_{i'} * p(A_i \neq x) \\ &+\lambda_j * p(A_j = x) + \lambda_{j'} * p(A_j \neq x) \\ &+\lambda_{ij} * p(A_i = x)p(A_j = x) \\ &+\lambda_{i'j'} * p(A_i \neq x)p(A_j \neq x) \\ &+\lambda_{ij'} * p(A_i = x)p(A_j \neq x) \\ &+\lambda_{i'j} * p(A_i \neq x)p(A_j = x) \end{aligned} \tag{14}$$

Furthermore, employing the convexity of the logarithmic function, i.e., $log(ax+by) - log(a+b) \geq \frac{a}{a+b}log(x) + \frac{b}{a+b}log(y)$, we approximate the equation above as:

$$\begin{aligned}log3p(Z = x) &\approx \lambda_i logp(A_i = x) + \lambda_{i'} logp(A_i \neq x) \\ &+\lambda_j logp(A_j = x) + \lambda_{j'} logp(A_j \neq x) \\ &+\lambda_{ij} log(p(A_i = x)p(A_j = x)) \\ &+\lambda_{i'j'} log(p(A_i \neq x)p(A_j \neq x)) \\ &+\lambda_{ij'} log(p(A_i = x)p(A_j \neq x)) \\ &+\lambda_{i'j} log(p(A_i \neq x)p(A_j = x)) \\ &= (\lambda_i + \lambda_{ij} + \lambda_{ij'}) logp(A_i = x) \\ &+(\lambda_{i'} + \lambda_{i'j} + \lambda_{i'j'}) logp(A_i \neq x) \\ &+(\lambda_j + \lambda_{ij} + \lambda_{i'j}) logp(A_j = x) \\ &+(\lambda_{j'} + \lambda_{ij'} + \lambda_{i'j'}) logp(A_j \neq x) \end{aligned} \tag{15}$$

B Proof for Conditional Independent between Attribute Couple

As for conditional mutual information $I(A_i, A_j|Z)$, when over the token x (which means $Z = x$), we will obtain:

$$\begin{aligned}&\sum_{a_i, a_j} p(a_i)p(a_j)p(Z = x|a_i, a_j) * \\ &log\frac{p(Z = x|a_i, a_j)p(Z = x)}{p(Z = x|a_i)p(Z = x|a_j)}\end{aligned} \tag{16}$$

Therefore, the simple operation on minimization is to satisfy the equation $p(Z = x|a_i, a_j)p(Z = x) = p(Z = x|a_i)p(Z = x|a_j)$.

C Equation 7 Simplification

Based on **Cauchy inequality**, the inequalities establish that:

$$\begin{aligned}&2 \geq (\lambda_j + \lambda_{j'}) \geq \sqrt{\lambda_j^2 + \lambda_{j'}^2}, \\ &\sqrt{\lambda_j^2 + \lambda_{j'}^2} * \sqrt{p(A_j = x)^2 + p(A_j \neq x)^2} \\ &\geq (\lambda_j p(A_j = x) + \lambda_{j'} p(A_j \neq x)) = p(Z = x)\end{aligned} \tag{17}$$

hence, we derive that the value of $p(A_i|Z = x)\sum_{j \neq i}(\lambda_j + \lambda_{j'})$ is in the interval of $[(n-1)p(A_i = x|Z = x), 2(n-1)p(A_i = x|Z = x)]$, namely that $(n-1)\lambda_i$ and $p(A_i|Z = x)\sum_{j \neq i}(\lambda_j + \lambda_{j'})$ are in the same order of magnitude. Therefore, we approximate $p(A_i|Z = x)\sum_{j \neq i}(\lambda_j + \lambda_{j'})$ with $(n-1)\lambda_i$ for simplification.

Furthermore, we simplify φ_i and ω_i with highlighting the kernel part $1 + \frac{1}{p(A_i)}$ and condensing other variables as s_i to express the strength (the proportion to the final generation) of the current attribute. We consider $logp(A_i \neq x)$ part as an auxiliary to $logp(A_i = x)$, thus introduce a small coefficient t ahead.

D Proof for Strategy Properties

Proof for property 1 According to Equation 8, we focus on the main part (i.e. $p(A_i = x)$) and obtain that:

$$\begin{aligned}p(Z{=}x_k){=}&\frac{Q_k * p(A_i = x_k)^{s_i * (1 + \frac{1}{p(A_i = x_k)})}}{\sum_{v \in V} Q_v * p(A_i = x_v)^{s_i * (1 + \frac{1}{p(A_i = x_v)})}} \\ &= \frac{1}{\sum_{v \in V} \frac{Q_v}{Q_k} * p_{vk}^{s_i}} \\ \textbf{s.t. } Q_k &= P_b(x_k) \prod_{j \neq i} p(A_j = x_k)^{s_j * (1 + \frac{1}{p(A_j = x_k)})}, \\ p_{vk} &= \frac{p(A_i = x_v)^{1 + \frac{1}{p(A_i = x_v)}}}{p(A_i = x_k)^{1 + \frac{1}{p(A_i = x_k)}}}\end{aligned} \tag{18}$$

where x_k is the attribute token on A_i. Considering *attribute token* definition in Section 3.3, we conclude p_{vk} is less than 1 (details are shown in Section E), therefore, when attribute strength s_i increases (Q_k is fixed), the final generation probability will grow subsequently.

Proof for property 2 Similarly, we also consider the main part in Equation 8. Taking into account

the gaps between attribute token x_{attr} and non-attribute token x_v of both methods, we get:

$$\begin{aligned}(\textbf{ours})\ gap{=}s_i[(1{+}\frac{1}{p(A_i{=}x_{attr})})logp(A_i=x_{attr})\\ -(1{+}\frac{1}{p(A_i{=}x_v)})logp(A_i{=}x_v)]\\ (\textbf{linear})\ gap=s_i(logp(A_i=x_{attr})\\ -logp(A_i=x_v))\end{aligned} \tag{19}$$

Referring to a special function $f(x)=\frac{logx}{x}-logx$ which is always increasing within the interval of $(0,1)$, we derive that the gap of our strategy is greater than that of linear combination. Thus, if distributions on other attributes are fixed, the final generation tends to perform more like current attribute in our strategy.

E Proof for $p_{vk}<1$

$$\begin{aligned}p(A_i=x_k)^{1+\frac{1}{p(A_i=x_k)}}&>p(A_i=x_k)^{1+\frac{1}{p(A_i=x_v)}}\\ &>p(A_i=x_v)^{1+\frac{1}{p(A_i=x_v)}}\\ \textbf{s.t.}\ p(A_i=x_k)&>p(A_i=x_v)\end{aligned} \tag{20}$$

Hence the inequality $p_{vk}<1$ is satisfied.

F Proof for rationality of $\sigma(p(A_i=x))$

The surrogate stuff of $p(A_i=x)$ on the denominator (in Equation 8) is rational when it satisfies both the properties of **Theorem 1** and **Theorem 2**.

Proof for property 1 Similar to Equation 18, we substitute $\sigma(p(A_i)=x)$ for $p(A_i=x)$ on the denominator and obtain:

$$\begin{aligned}p(Z{=}x_k){=}&\frac{Q_k*p(A_i=x_k)^{s_i*(2+e^{-p(A_i{=}x_k)})}}{\sum_{v\in V}Q_v*p(A_i=x_v)^{s_i*(2+e^{-p(A_i{=}x_v)})}}\\ &=\frac{1}{\sum_{v\in V}\frac{Q_v}{Q_k}*p_{vk}^{s_i}}\\ \textbf{s.t.}\ Q_k&=P_b(x_k)\prod_{j\neq i}p(A_j=x_k)^{s_j*(2+e^{-p(A_j=x_k)})},\\ p_{vk}&=\frac{p(A_i=x_v)^{2+e^{-p(A_i=x_v)}}}{p(A_i=x_k)^{2+e^{-p(A_i=x_k)}}}\end{aligned} \tag{21}$$

Likewise, x_k is the *Attribute Token* which satisfies $p(A_i=x_k)>p(A_i=x_v)$. Furthermore, the inequality $e^{-p(A_i=x_k)}<e^{-p(A_i=x_v)}$ will hold. Hence, we will also obtain that $p_{vk}<1$ and there exists a positive correlation between s_i and $p(Z=x_k)$.

Proof for property 2 Following Equation 10, we tick off gaps of both the σ substitute and the linear combination, as:

$$\begin{aligned}(\sigma)\ gap&=s_i[(2+e^{-p(A_i=x_{attr})})logp(A_i=x_{attr})\\ &-(2+e^{-p(A_i=x_v)})logp(A_i=x_v)]\\ (\textbf{linear})\ gap&=s_i(logp(A_i=x_{attr})\\ &-logp(A_i=x_v))\end{aligned} \tag{22}$$

We introduce a special function and its derivative:

$$\begin{aligned}f(x)&=(2+e^{-x})logx-logx,\\ \nabla f(x)&=\frac{1+e^{-x}-xe^{-x}logx}{x}\end{aligned} \tag{23}$$

Obviously, $f(x)$ is increasing in the interval of $(0,1)$ due to $\nabla f(x)$ being always greater than 0. Therefore, we derive that $(2+e^{-p(A_i=x_{attr})})logp(A_i=x_{attr})-logp(A_i=x_{attr})$ is greater than $(2+e^{-p(A_i=x_v)})logp(A_i=x_v)-logp(A_i=x_v)$ which means the **gap** of the σ substitute outweighs that of the linear combination.

G Results for complementary event evaluation

We conduct experiments on the verification of t in Equation 8 and the results are demonstrated in Figures 3 & 4.

H The same trend overlapping setting for multiple attributes combination

As is shown in Table 8, both the settings are designed under the same trend overlapping condition, namely that attribute one will enhance the performance of attribute two.

We calculate the differential scores between overlapping settings of the same trend and the converse trend (Table 6), and report the results in Table 9.

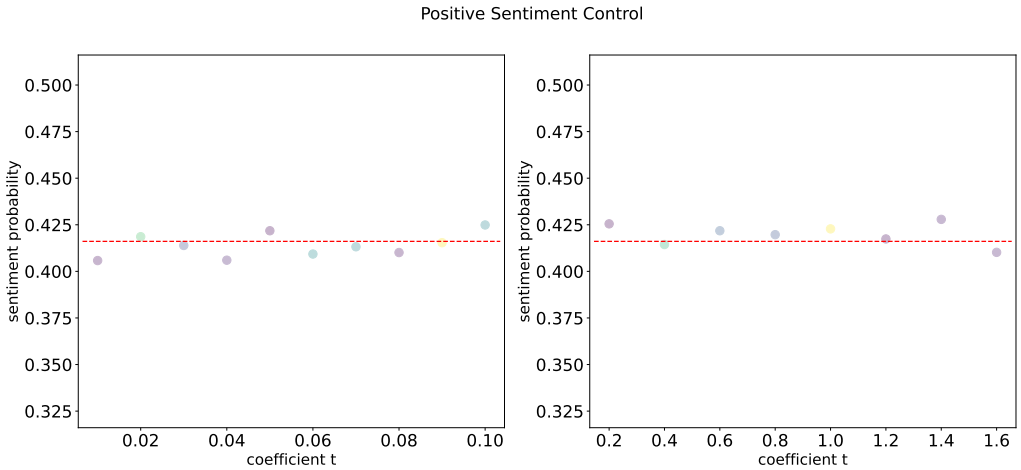

Figure 3: Coefficient t of $logp(A_i \neq x)$ evaluation on Positive Sentiment Control scenario.

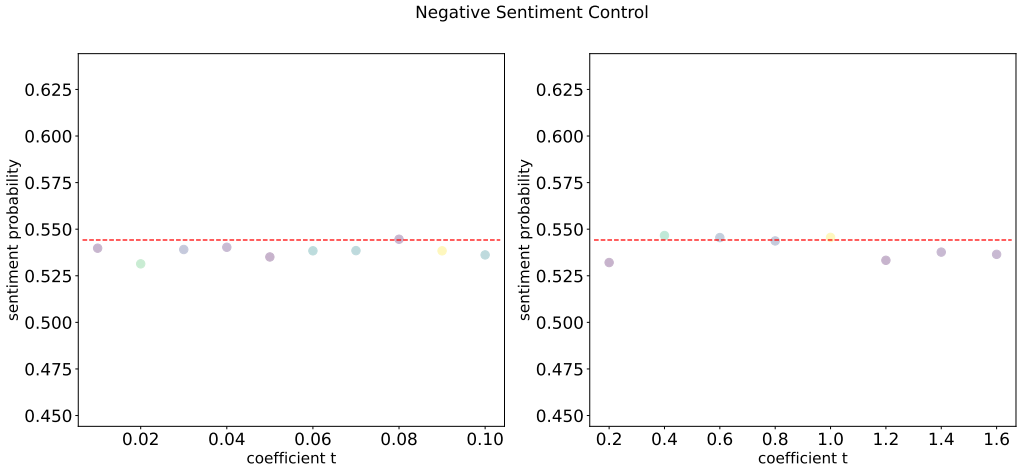

Figure 4: Coefficient t of $logp(A_i \neq x)$ evaluation on Negative Sentiment Control scenario.

Settings	Attribute	Prompt
Setting one	**toxicity**	The following movie review is not threatening and does not contain any sexually explicit or profane language, and is with a slight **positive** sentiment.
	positive reply	Continue writing after the movie review to change it with a very **positive** sentiment and a very **positive** tone.
Setting two	**toxicity**	The following movie review is not threatening and does not contain any sexually explicit or profane language, and is with a slight **negative** sentiment.
	negative reply	Continue writing after the movie review to change it with a very **negative** sentiment and a very **negative** tone.

Table 8: Attributes and prompts for the multi-control with the same sentiment trend overlapping setting.

Positive	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**	**Average**
Linear	0.020	0.990	0.530	1.010	1.030	-0.760	0.670	0.710	0.330	1.240	0.577
Ours	-0.430	0.590	0.240	1.180	-0.050	0.190	0.170	0.180	-0.670	0.490	**0.189**
Negative	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**	**Average**
Linear	-0.270	0.800	1.220	0.540	0.610	1.640	0.500	0.130	1.030	0.740	0.694
Ours	0.140	0.680	-0.280	-1.910	0.500	-0.370	0.090	0.400	1.420	1.250	**0.192**

Table 9: Sentiment Increase (%) on the basis of Table 6, less is better.