GUIDANCE IS NOT A HYPERPARAMETER: LEARNING DYNAMIC CONTROL IN DIFFUSION LANGUAGE MODELS

Fan Zhou, Tim Van de Cruys
KU Leuven
{fan.zhou,tim.vandecruys}@kuleuven.be

ABSTRACT

Classifier-Free Guidance (CFG) is a widely used mechanism for controlling diffusion-based generative models, yet its guidance scale is typically treated as a fixed hyperparameter throughout generation. This static design yields a suboptimal controllability–quality tradeoff, as the optimal degree of guidance varies across tasks and across different stages of the diffusion process, especially in NLP domain. We recast CFG scale selection as a sequential decision-making problem and propose to learn dynamic guidance trajectories via reinforcement learning. Specifically, we model the guidance scale as a discrete control action selected at each generation step based on the evolving diffusion state, and optimize a policy using Proximal Policy Optimization (PPO) under task-level rewards. Experiments on three controlled NLP generation tasks using discrete diffusion language models demonstrate that adaptive guidance consistently achieves a better balance between controllability and generation quality than fixed-scale strategies. Further analysis of the learned policies reveals distinct and interpretable guidance trajectories across tasks, underscoring the importance of treating guidance as a dynamic control process rather than a static design choice.

1 INTRODUCTION

Diffusion-based generative models have emerged as a dominant paradigm across multiple domains. In computer vision, they have become the standard approach for high-fidelity image generation (Rombach et al., 2022), while recent advances in NLP domain, especially the discrete diffusion language models (dLLMs) (Nie et al., 2025; Ye et al., 2025) have demonstrated their effectiveness for text generation, providing a viable alternative to conventional autoregressive language models. A key factor underlying this success is the ability to explicitly control the generation process through guidance mechanisms. Among these, classifier-free guidance (CFG) (Ho & Salimans, 2022) has become the de facto standard, exposing a single control variable—the guidance scale—that directly modulates the influence of conditioning information during generation. By adjusting this scale, CFG enables controllable generation but also induces an inherent tradeoff between adherence to the desired conditions and generation quality, suggesting that effective control hinges on how this tradeoff is managed throughout the generation process.

While classifier-free guidance provides a simple and effective control mechanism, its guidance scale is almost always treated as a fixed hyperparameter throughout the entire generation process (Ho & Salimans, 2022; Dhariwal & Nichol, 2021). In practice, this scale is typically selected via offline tuning and then applied uniformly across all generation steps and inputs (Nichol & Dhariwal, 2021). However, diffusion-based generation is inherently sequential and context-dependent: the role and effect of guidance vary across generation stages, and the balance between controllability and generation quality can differ substantially across tasks and inputs(Wang et al., 2024; Zhang & Wan, 2025). As a result, applying a single, globally fixed guidance scale introduces a structural mismatch between the stage-wise requirements of the generation process and the control mechanism, leading to suboptimal controllability–quality tradeoffs in practice.

To address the limitations of static guidance, prior work—particularly in computer vision—has explored heuristic strategies for adjusting guidance strength during diffusion sampling, typically by prescribing a fixed functional schedule shared across generation scenarios, such as exponential curves (Gao et al.) or distributions parameterized by simple forms (e.g., Beta schedules) (Malarz et al., 2025). While these heuristics can be effective in specific settings, they fundamentally rely on pre-defined schedules that are determined prior to sampling and remain independent of the evolving generation state or task-level objectives. As a result, they implicitly assume that a single schedule—or a small family of hand-crafted schedules—can generalize across diverse inputs, tasks, and controllability requirements. In practice, however, optimal guidance behavior often varies substantially across scenarios, making it difficult for static heuristic schedules to capture task-specific needs without extensive manual tuning, particularly in NLP tasks with heterogeneous objectives. More importantly, such heuristic approaches lack a systematic mechanism for incorporating task-level feedback during generation, highlighting the need for a more flexible and adaptive approach to guidance control.

These observations suggest that guidance control in diffusion models is inherently a sequential decision-making problem. At each generation step, the choice of guidance strength influences not only the current output but also the future trajectory of the generation process, with its effect only fully reflected in task-level outcomes at the end of sampling. Importantly, there is no oracle supervision for selecting guidance strength at intermediate steps, and downstream task objectives are typically non-differentiable and only observable upon completion of the generation process. As a result, manually specifying guidance schedules or relying on myopic, step-wise rules is insufficient for optimizing long-term task performance. Reinforcement learning provides a principled framework for this setting by enabling sequential decisions to be optimized under sparse and delayed rewards while accounting for long-term effects across generation steps. In this work, we adopt a policy-based reinforcement learning approach to adaptively select guidance scales during generation, enabling dynamic control that responds to the evolving generation context.

In this work, we focus on learning task-specific dynamic guidance trajectories for diffusion-based text generation, rather than adapting guidance on a per-instance basis, which would require reliable instance-level feedback during generation and risk overfitting to noisy trajectory outcomes. Our goal is to capture characteristic guidance behaviors that are optimal for a given generation task and can be consistently applied across different inputs. We instantiate the proposed reinforcement learning formulation in discrete diffusion language models, using LLaDA (Nie et al., 2025) as the underlying generative model. We study three representative NLP scenarios with distinct controllability objectives: keyword-conditioned sentence generation, sentiment-controlled style transfer, and length-controlled sentence rewriting. For each task, we optimize a guidance policy using task-specific rewards and analyze the resulting guidance trajectories to understand how controllability–quality tradeoffs evolve across diffusion steps.

Beyond overall generation quality, our primary goal is to understand how guidance behaviors evolve over diffusion steps to effectively balance controllability and generation quality. By learning task-specific guidance policies, we aim to characterize the structure of guidance trajectories that emerge under different controllability objectives. In particular, we seek to answer the following questions: how optimal guidance behaviors differ across tasks, how guidance strength evolves over the course of generation, and to what extent learned guidance trajectories provide insights beyond fixed or heuristic schedules.

Taken together, these considerations motivate our formulation of adaptive guidance as a learning problem. In the following sections, we detail our reinforcement learning approach and present empirical results that illustrate the resulting task-specific guidance behaviors across different controllability objectives.

2 Related work

2.1 Controllability–quality tradeoffs in generation

A fundamental challenge in controllable text generation is balancing controllability with fluency. Prior work has repeatedly observed that enforcing stronger control over attributes such as sentiment, keywords, or style often degrades fluency and naturalness, leading to repetition or semantic drift

(Hu et al., 2017; Dathathri et al., 2019; Krause et al., 2021). This tradeoff has been studied across different generation paradigms. In autoregressive language models, recent work primarily relies on generation-time control mechanisms such as weighted decoding, critic-guided decoding, prefix-based control, and instruction-based prompting (Zhou et al., 2023; Kim et al., 2023; Pei et al., 2023; Zhang et al., 2023; Shin et al., 2025). In diffusion-based text generation, controllability is commonly introduced through guidance mechanisms that modify sampling dynamics, including plug-and-play approaches based on external classifiers (Horvitz et al., 2024), classifier-free guidance (CFG) via scaling conditional signals during sampling (Ho & Salimans, 2022), as well as training-time methods that incorporate control signals during model learning (Zhou et al., 2025). In contrast to these approaches, our work focuses on generation-time controllability in diffusion-based language models and explicitly studies how guidance strength should be allocated over the diffusion process to balance controllability and generation quality, rather than modifying model parameters or decoding objectives.

2.2 CFG GUIDANCE SCHEDULING FOR CONTROL

Recent work revisits classifier-free guidance (CFG) by treating the guidance scale as a dynamic control variable rather than a fixed hyperparameter. A common approach is to design time-dependent heuristic schedulers that vary guidance strength across diffusion steps, often supported by empirical studies on "how much to guide" and by analyses that explain why different noise regimes favor different guidance magnitudes (Wang et al., 2024; Zhang & Wan, 2025; Malarz et al., 2025). Beyond purely step-based schedules, recent methods incorporate online feedback signals (e.g., attribute confidence or constraint satisfaction) to adapt guidance strength on-the-fly during sampling, providing sample-level adjustment without learning a controller (Papalampidi et al., 2025). In parallel, theory- or control-inspired formulations revisit CFG in discrete diffusion settings and derive more principled adaptive scaling rules from modeling assumptions or hand-designed objectives, including theory-informed improvements for discrete diffusion and stochastic optimal control perspectives (Rojas et al., 2025; Azangulov et al., 2025). While these methods introduce adaptive or scheduled guidance during diffusion sampling, they primarily rely on hand-designed rules, instantaneous feedback signals, or analytical scaling laws, and do not explicitly learn a guidance policy optimized for task-level objectives. In contrast, our approach formulates guidance selection as a sequential decision-making problem and learns task-specific guidance trajectories via reinforcement learning under sparse, terminal rewards.

3 PRELIMINARIES

3.1 DISCRETE DIFFUSION LANGUAGE MODELS

We consider a discrete diffusion language model (ddLM) defined over token sequences (Lou et al., 2023; Nie et al., 2025). Let $x_0 = (x_0^1, \ldots, x_0^L)$ denote a clean sequence of length L, and let M be a special [MASK] token. ddLM defines a continuous-time forward noising process that independently masks each token.

Given a noise level $t \in [0, 1]$, the forward process is defined as

$$q_t(x_t^i \mid x_0^i) = \begin{cases} t, & x_t^i = M, \\ 1 - t, & x_t^i = x_0^i, \end{cases} \qquad q_t(x_t \mid x_0) = \prod_{i=1}^{L} q_t(x_t^i \mid x_0^i), \tag{1}$$

where x_t denotes the corrupted sequence at noise level t.

The reverse process is parameterized by a neural *mask predictor* $p_\theta(\cdot \mid x_t)$, which models the conditional distribution of the original token at each masked position given the partially masked sequence.

For conditional generation, ddLM models the distribution $p_\theta(r_0 \mid p_0)$ by applying the forward masking process only to the response segment while keeping the prompt fixed. At inference time, generation is performed by simulating the reverse diffusion process from a fully masked response. Let $1 = t_K > t_{K-1} > \cdots > t_0 = 0$ denote a discretization of the time interval. At each reverse step, the mask predictor fills in the masked tokens, followed by a remasking operation that ensures consistency with the forward process, yielding a valid reverse diffusion process.

3.2 CLASSIFIER-FREE GUIDANCE

Classifier-free guidance (CFG) (Ho & Salimans, 2022) controls conditional diffusion sampling by amplifying the effect of conditioning information during the reverse process. Let c denote the condition (e.g., a prompt). For a corrupted sequence x_t, the mask predictor produces token-level log-probabilities

$$\ell^i_{\text{cond}}(x) = \log p_\theta(x \mid x_t, c), \qquad \ell^i_{\text{uncond}}(x) = \log p_\theta(x \mid x_t), \tag{2}$$

for each masked position i.

In dLLMs, CFG is implemented by defining guided logits as

$$\ell^i_{\text{CFG}}(x) = \ell^i_{\text{uncond}}(x) + (1+\gamma)\big(\ell^i_{\text{cond}}(x) - \ell^i_{\text{uncond}}(x)\big), \tag{3}$$

where $\gamma \geq 0$ is the guidance scale.

The guidance scale γ is an inference-time control variable that can be adjusted during sampling to modify the reverse diffusion dynamics without changing the learned model parameters.

3.3 POLICY LEARNING FOR SEQUENTIAL DECISION

We formulate guidance selection as a policy-based sequential decision-making problem, where a parameterized policy $\pi_\phi(a \mid s)$ selects actions based on the current state and induces a trajectory of decisions. The objective is to maximize the expected cumulative reward over a trajectory.

In this work, we optimize the policy using Proximal Policy Optimization (PPO) (Schulman et al., 2017), a standard on-policy algorithm for discrete action spaces.

Policy-based methods such as PPO are well suited for sequential decision problems with stochastic dynamics and delayed rewards, which motivates their use in our setting.

4 METHOD

4.1 MOTIVATION: WHY REINFORCEMENT LEARNING

We start from the observation that different downstream tasks evaluate generated outputs using distinct criteria. As a result, we assume that the optimal guidance schedule over diffusion steps is task-dependent, and no single guidance curve is universally optimal across tasks.

Importantly, our goal is not to fit guidance decisions to individual samples. Instead, we aim to learn guidance strategies that generalize across samples drawn from the same task distribution. This task-level perspective is necessary to avoid overfitting to noisy instance-level outcomes and to enable robust generalization. Formally, let $\mathcal{T}$ denote a distribution over tasks, and let τ represent a complete diffusion sampling trajectory induced by a guidance strategy. Our objective is to maximize the expected task-level reward:

$$\max_{\pi} \ \mathbb{E}_{\mathcal{T}} \left[\mathbb{E}_{\tau \sim \pi(\mathcal{T})} \left[R(\tau) \right] \right], \tag{4}$$

where $R(\tau)$ evaluates the quality of the final generated output.

Many existing guidance strategies are motivated by heuristic intuitions, such as using smaller guidance scales at highly noisy stages, larger scales at intermediate steps, and smaller scales again near convergence. Such intuitions often lead to parameterized guidance curves with a small number of degrees of freedom. Let a guidance curve be parameterized as $g_\theta(t)$ with parameters $\theta \in \mathbb{R}^d$. Searching for a task-specific guidance strategy then amounts to solving the following optimization problem:

$$\theta^\star = \arg\max_{\theta} \ \mathbb{E}_{\tau \sim g_\theta} \left[R(\tau) \right]. \tag{5}$$

However, extending these approaches to task-specific guidance quickly becomes computationally infeasible. Searching over flexible curve families significantly enlarges the action space, and evaluating each candidate curve requires multiple diffusion sampling runs. Given the high computational cost and low sampling efficiency of discrete diffusion language models, curve-level search in Eq. equation 5 is prohibitively expensive in practice.

Purely heuristic guidance strategies further rely on fixed, deterministic schedules of the form $a_k = g(k)$ and do not incorporate task-level feedback. Moreover, there is no oracle supervision for optimal guidance decisions at intermediate diffusion steps, and task evaluation is only available upon completion of the sampling process. As a result, selecting guidance actions based solely on intuition or static rules is insufficient for task-adaptive control.

Taken together, these considerations indicate that adaptive guidance selection should be treated as a sequential decision-making problem with delayed and noisy feedback. This naturally motivates a reinforcement learning formulation that directly optimizes expected trajectory-level returns.

4.2 Policy Learning with PPO

Following the formulation in Section 4.1, we model adaptive guidance selection during diffusion sampling as a policy learning problem. A guidance policy interacts with the diffusion process over multiple steps, producing a sequence of guidance decisions that jointly determine the final generated output.

Policy Formulation. Let $\tau = (s_K, a_K, \ldots, s_1, a_1, s_0)$ denote a complete diffusion sampling trajectory, where s_k represents the sampling state at diffusion step k and a_k denotes the guidance action selected at that step. A policy $\pi_\phi(a_k \mid s_k)$ maps the current sampling state to a guidance action. One complete diffusion run corresponds to a single episode.

The objective of policy learning is to maximize the expected cumulative reward over trajectories:

$$J(\pi_\phi) = \mathbb{E}_{\tau \sim \pi_\phi} \left[\sum_{k=0}^{K} r_k \right], \tag{6}$$

where r_k denotes the reward received at diffusion step k.

Sparse Reward Design. We adopt a sparse reward formulation aligned with downstream task evaluation. Specifically, rewards are assigned only at the final diffusion step:

$$r_k = \begin{cases} R(\tau), & k = 0, \\ 0, & k > 0, \end{cases} \tag{7}$$

where $R(\tau)$ evaluates the quality of the completed generated sequence.

Downstream tasks define their objectives solely on the final output. In contrast, intermediate diffusion states s_k contain masked tokens and stochastic noise and do not correspond to semantically interpretable text. As a result, assigning step-wise rewards based on such states requires evaluating the final task objective using partial and noisy representations.

Formally, for any step-wise reward $\hat{r}_k = f(s_k)$ derived from an intermediate state, the best possible predictor of the final task reward $R(\tau)$ in the mean-squared sense is the conditional expectation $\mathbb{E}[R(\tau) \mid s_k]$. The informativeness of any step-wise reward is therefore governed by the variance of this quantity:

$$\mathrm{Var}(R) = \mathbb{E}[\mathrm{Var}(R \mid s_k)] + \mathrm{Var}(\mathbb{E}[R \mid s_k]), \tag{8}$$

where $\mathrm{Var}(\mathbb{E}[R \mid s_k])$ measures the portion of reward variability that can be explained by the intermediate state s_k.

This explainable variance is fundamentally limited by the mutual information between s_k and the final output. Since $R(\tau)$ is a deterministic function of the final generated sequence x_0, the data processing inequality yields

$$I(R; s_k) \leq I(x_0; s_k), \tag{9}$$

where $I(\cdot;\cdot)$ denotes mutual information. At early and intermediate diffusion steps, s_k remains highly noisy and masked, implying limited mutual information with the final output x_0. Consequently, $\mathrm{Var}(\mathbb{E}[R \mid s_k])$ is small, and any step-wise reward derived from s_k is inherently noisy and weakly correlated with the true task objective.

For these reasons, dense step-wise rewards are not considered. Using sparse terminal rewards avoids introducing proxy objectives based on low-information intermediate states and directly aligns policy optimization with the true task-level evaluation defined in Eq. equation 4.

Action Repetition and Temporal Abstraction. As the number of diffusion steps K increases, sparse terminal rewards in Eq. equation 7 induce long-horizon credit assignment with high variance. To reduce the effective decision horizon, we adopt action repetition, where a single guidance action is held constant for n consecutive diffusion steps. Let $m = \lceil K/n \rceil$ denote the number of decision blocks, and let $\tilde{a}_j$ be the action selected for block $j \in \{1, \ldots, m\}$. The per-step action is defined as

$$a_k = \tilde{a}_{\lceil k/n \rceil}. \tag{10}$$

This temporal abstraction reduces the number of policy decisions from K to m, mitigating the variance of policy updates under sparse terminal feedback. Moreover, holding actions constant over short intervals encourages smooth guidance trajectories by construction, preventing abrupt changes between consecutive diffusion steps.

Discrete Action Space for Generalized Trajectories. Our goal is to learn task-generalized guidance behaviors characterized by coarse trajectory properties, such as early-, mid-, and late-stage guidance strength, overall trends, and peak values, rather than fine-grained per-step control. Accordingly, we restrict the action space to a small discrete set of guidance scales:

$$\mathcal{A} = \{0, 0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0, 2.25, 2.5, 2.75, 3.0\}, \qquad |\mathcal{A}| = 13. \tag{11}$$

This discrete design provides sufficient expressive power to capture distinct guidance regimes across diffusion stages while substantially reducing exploration complexity compared to continuous control. In practice, a categorical policy over a finite action set yields more stable on-policy updates under sparse and noisy trajectory-level rewards.

Policy Optimization with PPO. To optimize the guidance policy under sparse and delayed terminal rewards, we adopt Proximal Policy Optimization (PPO). We use the standard clipped surrogate objective described in Section 3.3, which provides stable on-policy updates under the stochastic dynamics of diffusion sampling. PPO naturally supports discrete action spaces and long-horizon trajectories, making it well suited for adaptive guidance selection without requiring differentiability of the task-level reward.

4.3 Sampling Task-Generalized Guidance Trajectories

After training, the learned guidance policy π_ϕ does not produce a single deterministic guidance schedule. Instead, it induces a distribution over guidance trajectories through stochastic policy sampling and the inherent randomness of diffusion sampling. Each rollout corresponds to one realization of guidance decisions over diffusion steps.

Policy-Induced Trajectory Distribution. Recall that policy optimization in Eq. equation 4 aims to maximize the expected task-level reward under the policy-induced trajectory distribution. Let $\tilde{\tau}^{(i)} = \{\tilde{a}_1^{(i)}, \ldots, \tilde{a}_m^{(i)}\}$ denote the sequence of guidance actions selected for the m decision blocks defined in Eq. equation 10 during the i-th rollout. Sampling the learned policy yields a set of trajectories $\{\tilde{\tau}^{(i)}\}_{i=1}^N$ drawn i.i.d. from the distribution induced by π_ϕ.

Under this formulation, no single trajectory is expected to be optimal. Instead, the policy captures a distribution over guidance behaviors that collectively maximize expected task performance.

Monte Carlo Estimation via Mean Trajectory. To obtain a task-generalized guidance strategy, we estimate the expected guidance behavior under the learned policy by aggregating sampled trajectories. Specifically, we compute the empirical mean trajectory:

$$\bar{a}_j = \frac{1}{N} \sum_{i=1}^{N} \tilde{a}_j^{(i)}, \qquad j = 1, \ldots, m, \tag{12}$$

where $\tilde{a}_j^{(i)}$ denotes the action selected for decision block j in the i-th rollout.

Equation equation 12 constitutes a Monte Carlo estimator of the expected guidance action at each decision block. Under standard assumptions, the estimator is unbiased and converges to the policy-induced expectation as the number of sampled trajectories increases. Averaging across trajectories reduces variance arising from stochastic sampling while preserving the policy's task-level generalization.

Frequency-Weighted Monte Carlo Aggregation. While uniform averaging treats all sampled trajectories equally, low-frequency trajectories are more likely to arise from stochastic exploration under sparse terminal rewards. To emphasize guidance patterns consistently selected by the policy, we further consider a frequency-weighted aggregation.

Let f_i denote the empirical frequency of the i-th guidance interval or pattern across sampled trajectories, and let $\bar{v}_i$ be the average guidance value associated with that interval. We define the frequency-weighted mean as

$$\mathrm{CFG}^{\mathrm{freq}} = \frac{\sum_{i=1}^{K} f_i^{\,p}\, \bar{v}_i}{\sum_{i=1}^{K} f_i^{\,p}}, \tag{13}$$

where $p \geq 1$ controls the strength of frequency amplification.

This formulation can be interpreted as a power-transformed Monte Carlo estimator of the expected guidance behavior under the empirical trajectory distribution. Using $p > 1$ biases aggregation toward dominant guidance patterns while suppressing low-frequency trajectories that are more likely to reflect exploratory noise. In practice, this provides a smooth compromise between uniform expectation and mode selection, yielding robust task-generalized guidance trajectories.

Inference-Time Application. The resulting mean or frequency-weighted mean trajectory defines a deterministic, task-generalized guidance schedule. This schedule can be applied directly at inference time without additional policy sampling or optimization, providing a stable approximation of the policy's expected behavior that remains fully aligned with the training objective.

5 Experimental Setup

5.1 Tasks and Evaluation

We evaluate our method on three controlled text generation tasks that exhibit distinct controllability–quality trade-offs and require different guidance behaviors across diffusion stages. These tasks serve as representative testbeds for assessing task-adaptive guidance strategies, as no single static guidance schedule is expected to perform optimally across all settings.

Keyword-conditioned sentence generation. The model is required to generate a fluent sentence that contains all keywords from a given set of 10 words. Controllability is measured by the success rate of including all required keywords. Generation quality is assessed by fluency, measured using perplexity (PPL) computed with a pretrained GPT-2 language model.

Length-controlled generation. The model rewrites an input sentence such that the output length falls within 40%–80% of the original sentence length measured in number of words. Controllability is measured by whether the generated output satisfies the length constraint. Generation quality is evaluated using content preservation and GPT-2 perplexity.

Sentiment style transfer. Given an input sentence, the model rewrites it to the opposite sentiment while preserving the original content. We consider both positive-to-negative and negative-to-positive transfer. Controllability is measured by sentiment transfer accuracy using a pretrained binary sentiment classifier. Generation quality is assessed by content preservation and fluency, measured by GPT-2 perplexity.

All metrics are computed on the final generated outputs and averaged over the evaluation set. Training and evaluation are conducted on disjoint prompt sets sampled from the same task distribution to assess task-level generalization.

5.2 Reward Definition

For all tasks, we adopt a sparse terminal reward aligned with downstream task evaluation. The reward is defined as a weighted combination of controllability and generation quality:

$$R(\tau) = \lambda_1\, R_{\mathrm{ctrl}}(\tau) + \lambda_2\, R_{\mathrm{PPL}}(\tau)(+\lambda_3\, R_{\mathrm{semantic}}(\tau)), \tag{14}$$

where R_{ctrl} denotes the task-specific controllability score, R_{PPL} denotes the generation fluency and R_{semantic} denotes the semantic preservation. Both components are computed only on the final generated output.

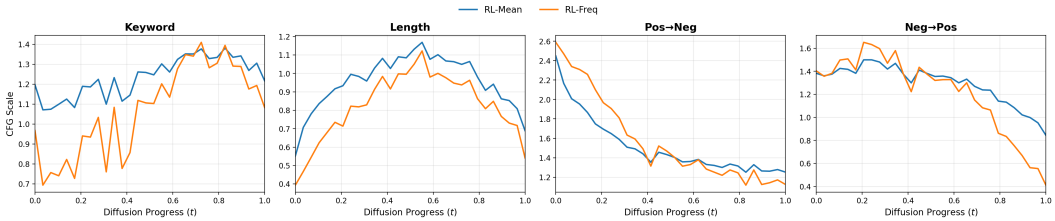

Figure 1: Mean guidance trajectories learned by the RL policy across diffusion progress for different tasks under 60-step sampling.

Method	Keyword Generation		Length Control			Sentiment Transfer(Pos2Neg)			Sentiment Transfer(Neg2Pos)		
	Ctrl(%) ↑	PPL ↓	Acc(%) ↑	Content(%) ↑	PPL ↓	Acc(%) ↑	Content(%) ↑	PPL ↓	Acc(%) ↑	Content(%) ↑	PPL ↓
Fixed CFG	71.4	61.3	76.0	90.4	301.9	99.0	66.8	184.7	32.8	68.7	137.2
Linear Increase	73.8	62.0	85.4	89.8	374.8	98.6	**74.6**	345.9	38.0	71.9	301.3
Linear Decrease	66.2	89.2	65.6	86.8	396.5	98.4	59.9	164.9	27.0	62.9	98.7
Cosine Increase	70.6	66.7	87.2	90.0	378.9	98.2	74.3	339.4	38.2	72.1	326.2
Cosine Decrease	68.0	88.3	63.2	86.1	398.9	98.6	58.7	150.8	26.0	62.1	**95.4**
Beta	73.8	65.0	78.0	90.0	294.7	97.6	71.7	199.3	**38.6**	72.7	163.1
Inverted Beta	68.8	76.3	75.0	86.5	542.9	97.8	60.6	241.0	29.8	63.5	141.2
RL-Mean(ours)	**74.6**	56.2	**92.8**	**91.8**	**205.6**	99.4	67.2	**145.2**	40.2	75.9	110.1
RL-Freq(ours)	74.2	**54.6**	92.2	91.7	211.0	**99.6**	63.2	150.7	**40.6**	**76.1**	106.6

Table 1: Main results on three controlled text generation tasks with 60 diffusion sampling steps. Fixed CFG on keywords, Length, Pos2Neg and Neg2Pos are 1.5, 1, 1.5, 1.5 respective to achieve the best results over all fixed CFG scales. Other heuristic CFG scales ranges from 0 to 3.

5.3 BASELINES

We compare our method against a set of non-learning guidance baselines that rely on fixed, predefined guidance schedules and do not adapt guidance decisions based on task-level feedback.

Constant guidance A fixed classifier-free guidance scale is applied uniformly across all diffusion steps.

Heuristic guidance schedules We evaluate commonly used hand-crafted guidance curves, including linear increase, linear decrease, cosine increase, cosine decrease, Beta-shaped schedules, and inverted-Beta schedules. All heuristic baselines are tuned to their best-performing configurations for each task.

These baselines represent standard heuristic approaches to guidance scheduling and serve as natural points of comparison for assessing the benefits of adaptive, policy-learned guidance.

5.4 MAIN RESULTS

Table 5.3 presents the full comparison across all four tasks with 60 diffusion steps. We highlight three key findings.

RL-learned schedules consistently outperform fixed and heuristic baselines. Across all four tasks, both RL-MEAN and RL-FREQ either achieve the best performance or remain competitive with the best heuristic on every metric. On **keyword generation**, RL-MEAN attains the highest keyword coverage of 74.6%, a 3.2 percentage point (pp) improvement over the fixed CFG baseline (71.4%), while simultaneously reducing perplexity from 61.3 to 56.2. RL-FREQ further improves fluency to a perplexity of 54.6 with comparable coverage (74.2%). On **length control**, the gains are most pronounced: RL-MEAN achieves 92.8% accuracy, a 16.8 pp absolute improvement over fixed CFG (76.0%), while also delivering the best content preservation (91.8%) and the lowest perplexity (205.6)—a Pareto improvement across all three metrics. Even the best-performing heuristic (cosine increase, 87.2% accuracy) falls 5.6 pp short and incurs nearly double the perplexity (378.9 vs. 205.6). For **pos→neg sentiment transfer**, where all methods already achieve near-saturated accuracy ($\geq$97.6%), RL-MEAN reduces perplexity to 145.2 (vs. 184.7 for fixed CFG) while maintaining 99.4% accuracy, and RL-FREQ reaches the highest accuracy of 99.6%. On the more chal-

lenging **neg→pos** direction, RL-FREQ achieves 40.6% accuracy and 76.1% content similarity—improvements of 7.8 pp and 7.4 pp over fixed CFG (32.8% and 68.7%), respectively—while also lowering perplexity from 137.2 to 106.6.

Heuristic schedules exhibit a controllability–fluency trade-off that RL resolves. Among the heuristic baselines, a clear and consistent pattern emerges: increasing schedules (linear increase, cosine increase) improve task controllability but severely degrade fluency, while decreasing schedules exhibit the opposite behavior. For instance, on neg→pos sentiment, cosine increase raises accuracy to 38.2% but inflates perplexity to 326.2; conversely, cosine decrease achieves the lowest perplexity (95.4) but suffers the worst accuracy (26.0%), a 14.6 pp drop from RL-FREQ. On length control, cosine increase reaches 87.2% accuracy but at a perplexity of 378.9, while linear decrease drops accuracy to 65.6% with a perplexity of 396.5. The beta distribution schedule, which peaks in the middle, generally falls between these two extremes but does not consistently excel on any task. In contrast, the RL-learned schedules break this trade-off: they simultaneously improve both controllability and fluency by learning when to apply strong versus weak guidance throughout the generation process.

Learned trajectories reveal task-dependent guidance strategies. Figure 1 visualizes the CFG scale trajectories produced by RL-MEAN and RL-FREQ across the four tasks. On **keyword generation** and **length control**, the learned schedules exhibit a *hump-shaped* pattern: guidance increases during the early-to-middle diffusion steps ($t \approx 0.1$–0.5), where the overall token structure is being established, then gradually decreases toward the end of generation. This suggests that strong guidance is most beneficial during the coarse structure-forming phase, while lighter guidance during the final refinement steps helps preserve fluency. In contrast, for **pos→neg sentiment transfer**, the policy learns a *monotonically decreasing* schedule, starting with high CFG scales ($\sim$2.5) that rapidly decline. This indicates that establishing the target sentiment polarity early is critical, after which the model can focus on generating coherent and fluent content with reduced guidance. The **neg→pos** direction follows a similar but more moderate decreasing trend, consistent with this task being inherently harder (reflected by the lower absolute accuracy across all methods). Notably, RL-FREQ consistently uses lower guidance scales than RL-MEAN, particularly in later steps, which explains its tendency toward better fluency (lower perplexity) on several tasks.

6 CONCLUSION

We have presented a reinforcement learning framework for learning dynamic classifier-free guidance schedules in discrete diffusion language models. By formulating guidance selection as a sequential decision-making problem with sparse terminal rewards, our approach discovers task-specific guidance trajectories that adapt to the evolving generation state—without modifying the pretrained model.Experiments on keyword-conditioned generation, length-controlled rewriting, and sentiment style transfer show that the learned schedules consistently outperform fixed and heuristic baselines, achieving simultaneous improvements in controllability and fluency that static schedules cannot. Analysis of the learned trajectories reveals distinct guidance patterns across tasks: hump-shaped profiles for structural constraints (keywords, length) and monotonically decreasing profiles for stylistic control (sentiment), suggesting that optimal guidance allocation is fundamentally task-dependent.

Limitations. While adaptive guidance scheduling significantly improves the controllability–quality trade-off within the discrete diffusion paradigm, a gap remains compared to autoregressive large language models, which benefit from stronger language modeling priors and more mature decoding strategies. The guidance mechanism can mitigate but not fully compensate for the inherent limitations of current diffusion language models, such as token repetition and fluency degradation under strong conditioning.

REFERENCES

Iskander Azangulov, Peter Potaptchik, Qinyu Li, Eddie Aamari, George Deligiannidis, and Judith Rousseau. Adaptive diffusion guidance via stochastic optimal control. *arXiv preprint arXiv:2505.19367*, 2025.

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

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Jiayang Gao, Tianyi Zheng, Jiayang Zou, Fengxiang Yang, Shice Liu, Luyao Fan, Zheyu Zhang, Hao Zhang, Jinwei Chen, Peng-Tao Jiang, et al. Beyond fixed: Aligning guidance with diffusion dynamics via exponential scaling.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Matthew Honnibal. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *(No Title)*, 2017.

Zachary Horvitz, Ajay Patel, Chris Callison-Burch, Zhou Yu, and Kathleen McKeown. Paraguide: Guided diffusion paraphrasers for plug-and-play textual style transfer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 18216–18224, 2024.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. Toward controlled generation of text. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2017.

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

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

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2023.

Dawid Malarz, Artur Kasymov, Maciej Zieba, Jacek Tabor, and Przemysław Spurek. Classifier-free guidance with adaptive scaling. 2025.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PMLR, 2021.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025.

Pinelopi Papalampidi, Olivia Wiles, Ira Ktena, Aleksandar Shtedritski, Emanuele Bugliarello, Ivana Kajic, Isabela Albuquerque, and Aida Nematzadeh. Dynamic classifier-free diffusion guidance via online feedback. *arXiv preprint arXiv:2509.16131*, 2025.

Jonathan Pei, Kevin Yang, and Dan Klein. Preadd: Prefix-adaptive decoding for controlled text generation. *arXiv preprint arXiv:2307.03214*, 2023.

Kevin Rojas, Ye He, Chieh-Hsin Lai, Yuta Takida, Yuki Mitsufuji, and Molei Tao. Theory-informed improvements to classifier-free guidance for discrete diffusion models. *arXiv preprint arXiv:2507.08965*, 2025.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. Style transfer from non-parallel text by cross-alignment. *Advances in neural information processing systems*, 30, 2017.

Seungmin Shin, Dooyoung Kim, and Youngjoong Ko. Eco decoding: Entropy-based control for controllability and fluency in controllable dialogue generation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 28297–28309, 2025.

Xi Wang, Nicolas Dufour, Nefeli Andreou, Marie-Paule Cani, Victoria Fernández Abrevaya, David Picard, and Vicky Kalogeiton. Analysis of classifier-free guidance weight schedulers. *arXiv preprint arXiv:2404.13040*, 2024.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025.

Huixuan Zhang and Xiaojun Wan. How much to guide: Revisiting adaptive guidance in classifier-free guidance text-to-vision diffusion models. In *Proceedings of the 7th ACM International Conference on Multimedia in Asia*, pp. 1–8, 2025.

Zhiling Zhang, Mengyue Wu, and Kenny Zhu. Semantic space grounded weighted decoding for multi-attribute controllable dialogue generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13230–13243, 2023.

Fan Zhou, Chang Tian, and Tim Van de Cruys. Controllable stylistic text generation with train-time attribute-regularized diffusion. *arXiv preprint arXiv:2510.06386*, 2025.

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

A Experimental Setup

A.1 Base Model

We use LLaDA-8B-Instruct (GSAI-ML/LLaDA-8B-Instruct), an 8-billion-parameter masked diffusion language model. The model is loaded in bfloat16 precision and kept frozen (eval mode) throughout training — only the lightweight RL policy network is trained. LLaDA generates text via an iterative denoising process: the generation region is initialized entirely with [MASK] tokens, and at each diffusion step, the model predicts logits over the vocabulary for all masked positions. Tokens with the highest prediction confidence are unmasked, and this process repeats until all tokens are revealed. We adopt the semi-autoregressive block-based generation scheme, where the generation length is divided into blocks that are filled sequentially. Classifier-free guidance (CFG) is applied at each step using the formulation logits $=$ logits$_{uc} + (\gamma + 1)$(logits$c -$ logitsuc), where γ is the CFG scale selected by the RL policy. Throughout all experiments, we use deterministic decoding (Gumbel temperature = 0) and the low-confidence remasking strategy.

A.2 RL Framework

We formulate the problem of learning a dynamic CFG schedule as a Markov Decision Process (MDP) and solve it with Proximal Policy Optimization (PPO). At each decision point during the diffusion process, the policy observes the current generation state and selects a CFG scale to apply. To reduce the length of the decision horizon while preserving diffusion quality, we employ action repeat: the same CFG scale is applied to multiple consecutive diffusion sub-steps per policy decision. All four tasks use 30 effective decision points.

The policy and value networks are implemented as separate MLPs with layer normalization. The actor network consists of two hidden layers (128 units each) with LayerNorm and ReLU activations, followed by an output head. The critic network shares the same architecture but outputs a scalar state

value. Weights are initialized with orthogonal initialization (gain = 0.01 for the actor, gain = 1.0 for the critic) to encourage initial exploration.

We use the TorchRL framework with TensorDict-based data flow. Rollouts are collected on-policy, advantages are computed via Generalized Advantage Estimation (GAE), and the policy is updated with clipped surrogate objectives. All tasks use sparse reward, meaning the reward signal is provided only at the end of each episode (i.e., after the full diffusion trajectory completes).

A.3 Tasks and Datasets

For the keyword-constrained generation and length control tasks, we construct datasets from WikiText-103 (Merity et al., 2016) by extracting sentences and using spaCy (Honnibal, 2017) for keyword extraction. For sentiment style transfer, we use the Yelp sentiment dataset (Shen et al., 2017), a standard benchmark in text style transfer research containing non-parallel positive and negative reviews.

We evaluate on three controlled generation tasks spanning four experimental configurations:

Keyword-Constrained Generation. Given a set of 10 keywords, the model must generate a fluent sentence containing all keywords. The prompt is formatted as: "Generate a fluent and coherent sentence that contains all of the following keywords: [keywords]." The dataset contains 3,000 training samples and 500 evaluation samples, each consisting of a reference sentence (20–50 words, mean = 34.5) paired with 10 extracted keywords. All 3,000 training samples are used during training.

Length-Controlled Sentence Compression. Given a sentence, the model must compress it to 40%–80% of its original word count while preserving meaning. The prompt is formatted as: "Compress to 40%-80% length. Output only the compressed sentence, no explanation. Input: [sentence] Output:" The dataset contains 5,000 training samples and 500 evaluation samples, each with a sentence and its word count. We use 3,000 training samples.

Sentiment Style Transfer (neg→pos and pos→neg). Given a sentence in the source sentiment, the model must rewrite it in the target sentiment while preserving meaning. We train two separate policies for the two transfer directions. The prompt follows the template: "Rewrite the following [source] sentence into a [target] sentence while preserving the original meaning." The sentiment dataset contains 50,000 training sentences per polarity (from which we sample 3,000) and 500 development sentences per polarity for evaluation.

A.4 State Space

1. Step ratio (t/T): the fraction of completed diffusion steps, providing a temporal signal.

2. Mask ratio: the fraction of generation tokens still masked, indicating generation progress.

3. Task-specific progress: keyword coverage ratio (fraction of keywords found in current text), compression length ratio (current word count / original word count), or sentiment score (binary: whether current text is classified as the target sentiment).

4. Previous CFG scale (normalized): the CFG scale used at the previous decision point, enabling the policy to condition on its own recent actions.

5. Model confidence: the mean softmax probability of predicted tokens at masked positions, reflecting how certain the model is about its current predictions.

All features are normalized to approximately [0, 1].

A.5 Action Space

For the three tasks, the action space is discrete with 13 choices: 0.0, 0.25, 0.50, ..., 3.0, parameterized by a Categorical distribution over logits. The discrete granularity of 0.25 provides sufficient resolution for CFG control while keeping the action space tractable for policy learning.

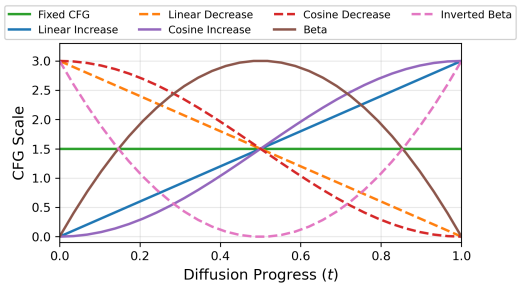

Figure 2: Seven heuristic guidance schedules used as baselines. All schedules operate within $[0, \gamma_{\max}]$ where $\gamma_{\max} = 3.0$. Fixed CFG uses a constant scale of 1.5. Increasing and decreasing variants are defined for linear and cosine families. The beta schedule follows a $\mathrm{Beta}(2, 2)$ density, peaking at the midpoint, while inverted beta is its complement.

A.6 Reward Design

All tasks combine task-specific quality metrics with a fluency measure (GPT-2 perplexity). Perplexity rewards are computed as $r_{ppl} = 1 - \text{clip}\left(\frac{\text{PPL}-1}{\text{PPL}_{max}-1}, 0, 1\right)$, capped at a task-specific maximum.

Keyword generation: $R = 0.5 \cdot r_{coverage} + 0.5 \cdot r_{ppl}$, where $r_{coverage}$ is a strict binary indicator (1.0 if all 10 keywords appear as exact word matches, 0.0 otherwise), and $\text{PPL}_{max} = 120$.

Length control: $R = 0.45 \cdot r_{length} + 0.10 \cdot r_{content} + 0.45 \cdot r_{ppl}$, where r_{length} is 1.0 if the compressed word count falls within the target range and decays linearly outside it, $r_{content}$ is the cosine similarity between Sentence-BERT embeddings of the original and compressed sentences, and $\text{PPL}_{max} = 500$.

Sentiment transfer (neg→pos): $R = 0.6 \cdot r_{cls} + 0.3 \cdot r_{ppl} + 0.1 \cdot r_{semantic}$, where r_{cls} is the target-class probability from a fine-tuned RoBERTa sentiment classifier (accuracy = 0.97), and $\text{PPL}_{max} = 300$.

Sentiment transfer (pos→neg): $R = 0.3 \cdot r_{cls} + 0.6 \cdot r_{ppl} + 0.1 \cdot r_{semantic}$, with the same classifier and $\text{PPL}_{max} = 300$. The higher PPL weight for this direction reflects the empirical observation that negative-style generation requires stronger fluency regularization.

A.7 Heuristic Schedules

To contextualize the learned guidance schedules, we compare against seven heuristic baselines (Figure 2). **Fixed CFG** applies different fixed CFG scale over different tasks throughout generation. **Linear** and **cosine** variants monotonically increase or decrease the guidance scale between 0 and $\gamma_{\max} = 3.0$, representing the intuition that guidance should either strengthen as generation progresses (to reinforce constraints on partially formed tokens) or weaken (to avoid disrupting already-denoised content). The **beta** schedule, parameterized as a $\mathrm{Beta}(2, 2)$ density scaled to $[0, \gamma_{\max}]$, concentrates guidance in the middle diffusion steps, while **inverted beta** applies strong guidance at both endpoints and minimal guidance at the midpoint. Together, these baselines span a diverse set of monotonic, symmetric, and constant guidance strategies.

B Sensitivity Analysis

B.1 Effect of Stochastic Temperature

We vary the sampling temperature of the guidance policy to control the level of stochasticity during trajectory sampling. Figure 3 shows the effect of temperature on controllability, fluency (PPL), and content preservation.

As temperature increases, controllability generally improves due to increased exploration of stronger guidance actions, while fluency and content preservation gradually degrade. Moderate temperatures achieve the best tradeoff, indicating that limited stochastic exploration is beneficial for discovering

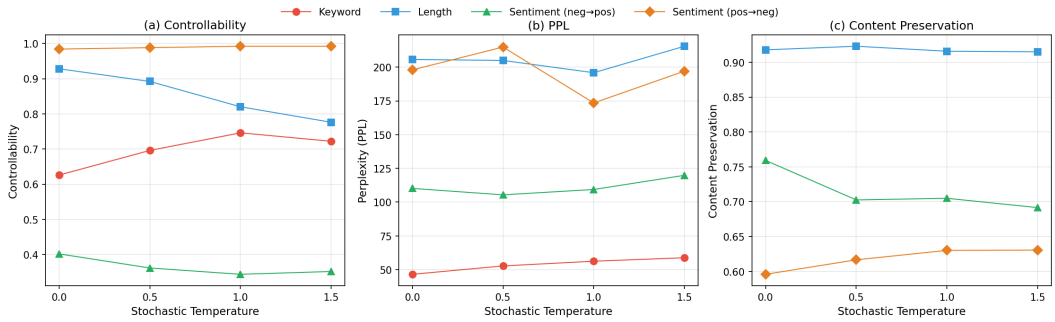

Figure 3: Ablation study on policy sampling temperature. We report controllability, fluency (GPT-2 perplexity), and content preservation across tasks.

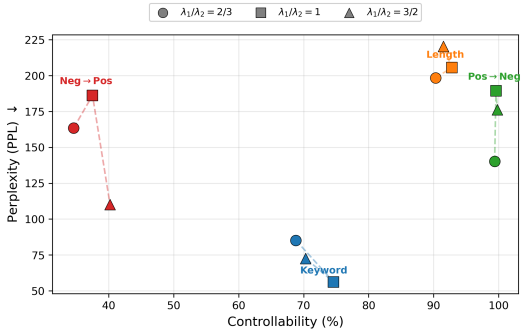

Figure 4: Controllability–fluency Pareto front under different reward weight ratios (λ_1/λ_2) across four tasks. Each task traces a trade-off curve: higher λ_1/λ_2 improves controllability at the cost of fluency.

effective guidance behaviors, whereas excessive randomness leads to unstable or overly aggressive guidance.

B.2 Effect of Reward Weights

Figure 4 visualizes the controllability–fluency trade-off under different reward weight ratios $\lambda_1/\lambda_2 \in \{2/3, 1, 3/2\}$. The optimal ratio differs across tasks and reflects the inherent difficulty of each controllability objective.For **pos→neg** sentiment transfer, where all configurations already achieve near-perfect accuracy ($\geq$99.4%), controllability is effectively saturated. Accordingly, allocating more weight to fluency ($\lambda_1/\lambda_2 = 2/3$) yields the best overall result, as the policy can focus its optimization budget on reducing perplexity without sacrificing accuracy. Conversely, for **neg→pos**—the hardest task with accuracy ranging from 34.6% to 40.2%—a higher controllability weight ($\lambda_1/\lambda_2 = 3/2$) is necessary to push the policy toward stronger guidance that improves sentiment transfer accuracy. For **keyword generation** and **length control**, a balanced ratio ($\lambda_1/\lambda_2 = 1$) achieves the best trade-off, as neither controllability nor fluency is trivially solved.

B.3 Effect of Number of Diffusion Sampling Timesteps

We compare adaptive guidance learned under 30-step and 60-step diffusion sampling (Tables C and 5.3). The effect of increasing the number of diffusion steps varies across tasks. On **length control** and **neg→pos sentiment**, more steps consistently improve all metrics—for instance, RL-Mean accuracy on length control increases from 83.8% to 92.8% while perplexity drops from 222.7 to 205.6. On **keyword generation**, additional steps substantially improve fluency (RL-Mean PPL: 87.6→56.2) but slightly reduce keyword coverage (78.8%→74.6%), suggesting a trade-off between controllability and fluency as the sampling horizon lengthens. On **pos→neg sentiment**, accuracy and content preservation improve, while perplexity slightly increases for the RL methods (RL-Mean:

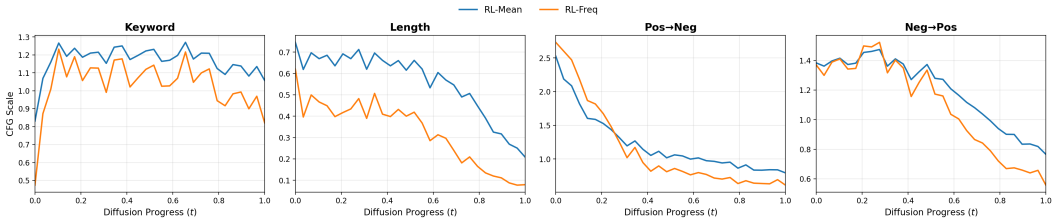

Figure 5: Mean guidance trajectories learned by the RL policy across diffusion progress for different tasks under 30-step sampling.

Method	Keyword Generation		Length Control			Sentiment Transfer(Pos2Neg)			Sentiment Transfer(Neg2Pos)		
	Ctrl(%) ↑	PPL ↓	Acc(%) ↑	Content(%) ↑	PPL ↓	Acc(%) ↑	Content(%) ↑	PPL ↓	Acc(%) ↑	Content(%) ↑	PPL ↓
Fixed CFG	74.2	107.6	70.8	89.4	395.4	98.4	63.9	196.2	31.2	66.1	162.7
Linear Increase	76.8	150.1	83.0	88.9	522.5	98.4	73.0	412.7	**41.8**	72.5	378.9
Linear Decrease	72.6	111.9	54.6	85.6	393.2	**99.6**	56.6	122.3	19.6	60.0	**96.4**
Cosine Increase	74.8	169.3	84.2	88.3	551.4	97.8	**73.7**	422.0	**41.8**	71.6	391.6
Cosine Decrease	70.6	110.6	53.4	84.7	406.0	**99.6**	56.6	127.7	22.8	59.9	96.7
Beta	**78.8**	157.2	68.0	89.0	406.4	97.8	70.3	249.7	39.2	70.4	212.6
Inverted Beta	75.2	86.9	67.0	86.0	550.5	99.0	57.7	179.6	26.2	60.4	130.7
RL-Mean(ours)	**78.8**	87.6	83.8	**91.5**	**222.7**	98.6	59.5	**116.6**	40.1	**72.6**	126.7
RL-Freq(ours)	77.0	**81.0**	**90.0**	90.1	228.8	**99.6**	62.0	122.6	39.4	61.2	121.1

Table 2: Main results on three controlled text generation tasks with 30 diffusion sampling steps.

116.6→145.2), indicating that the policy leverages additional steps primarily to strengthen style transfer at a modest fluency cost.

Notably, while the qualitative task-dependent patterns of the learned guidance trajectories are preserved across sampling horizons (Figures 5 and 1), the specific scale values and trajectory shapes differ, indicating that the policy adapts its guidance strategy to the available number of diffusion steps rather than simply rescaling a fixed schedule.

C Case Studies

To provide qualitative insight into how different guidance strategies affect generation quality, we present representative outputs from all methods on each task (Table 3 - Table 10).

Keyword Generation Prompt. Use the following 10 keywords to generate one fluent sentence: `entered, tons, long, flooding, meaning, hole, wounded, ship, water, men`.

Method	Generated Text
Fixed CFG	As the ship entered entered into the long ocean, carrying tons tons long tons of valuable cargo, a sudden storm caused flooding flooding flooding meaning meaning that the ship was at risk of water water water water due to a hidden hole hole that threatened to wounded men men men among the ship ship's water water wounded men.
Linear Increase	As the old ship entered the treacherous waters, it encountered tons upon tons of debris, causing a long and relentless flooding that meant a significant hole where it had once sturdy, ultimately leaving leaving the wounded and injured ship clinging to a precarious existence, while its brave men desperately fought for survival amidst the encroaching water.
Linear Decrease	As the ship entered entered into the long long long ocean carrying tons tons long tons tons, it encountered a long flooding long flooding event, meaning meaning that the ship ship had entered entered a hole hole long hole hole where wounded men wounded men had been surrounded by water water long water water water wounded men wounded men.
Cosine Increase	As the long ship entered the treacherous waters, it massive hull began to flood with tons of water, meaning that it had encountered a catastrophic hole that threatened to engulf the entire vessel, leaving its wounded men to scramble desperately for safety as they fought desperately to keep the ship afloat amidst the relentless, encroaching water.
Cosine Decrease	As the ship entered entered the long tons tons tons long tons tons of water, it began flooding flooding revealing flooding flooding revealing flooding flooding revealing revealing revealing revealing flooding revealing revealing revealing a hidden hole meaning hole filled hole filled men wounded men men men men men men men wounded men men who were desperately in need of help.
Beta	As the old ship entered entered the treacherous coastline, it encountered tons tons long long long term flooding flooding, meaning meaning that once a small hole gaping hole in the hole was now spilling ship tons water water and threatening ship men wounded men who had been bravely fighting but were injured during the harrowing ordeal.
Inverted Beta	As the ship entered entered into the long long coastline, it was flooded with tons tons of water, causing a large hole in the hull, which meant that the ship was at risk of being further wounded by water rising water rising water, while the crew worked tirelessly to repair the hole to save the wounded men.
RL-Mean (ours)	As the old ship entered the treacherous waters, it encountered tons upon tons of water that threatened to create a long and dangerous hole in the hole, which meant that the ship was flooding at an alarming rate, posing a significant danger to the wounded men who were desperately trying to stay afloat amidst the rapidly rising water.
RL-Freq (ours)	As the old ship entered the treacherous waters after a long voyage, tons of water seeped in, causing severe flooding and a gaping hole, which meant that the wounded men on board had no choice but to fight to save themselves and the ship, as the rising water threatened threatened to engulf the entire vessel.

Table 3: Qualitative results for the keyword-conditioned generation task with **30 diffusion steps**. All methods are evaluated under the same keyword prompt. Adaptive guidance (RL-Mean and RL-Freq) produces more coherent sentences with fewer repetitions compared to fixed and heuristic guidance schedules.

Keyword Generation Prompt. Use the following 10 keywords to generate one fluent sentence: `entered, tons, long, flooding, meaning, hole, wounded, ship, water, men`.

Method	Generated Text
Fixed CFG	As the old ship entered the stormy waters, carrying tons of water and causing severe flooding due to a long hole in its hull, it meant that the wounded men on board desperately needed to find a way to escape the rising water before the ship could completely sink, meaning their very lives hung in the balance.
Linear Increase	As the long ship entered the treacherous waters, it encountered tons of water that threatened to cause catastrophic flooding, meaning that a significant hole had developed in its hull, severely endangering the wounded men who were desperately navigating the ship's survival amidst the relentless influx of water threatening to engulf their beloved vessel.
Linear Decrease	As the ship entered entered tons of water during its long and harrowing journey, a sudden flooding occurred, meaning that a gaping hole had developed in its fragile hull, forcing the wounded men to scramble for safety amidst the rising water engulfing the ship, as they desperately tried to navigate through the treacherous waters.
Cosine Increase	As the old ship entered the treacherous waters, a massive hole in its hull began to frantically swallowing tons of water, causing a long and alarming flooding that meant imminent danger for the men on board, including several wounded men as they desperately fought to stabilize the stricken vessel amidst the relentless, encroaching water.
Cosine Decrease	As the ship entered entered the harbor, carrying tons of cargo, its long hull was suddenly compromised by a flooding breach, meaning that a gaping hole had developed, allowing massive amounts of water to enter, threatening the safety of the men on board, including wounded individuals who desperately needed to escape the treacherous situation.
Beta	As the old ship entered the treacherous stormy sea, tons of water began to pour in due to a disarmingly long hole, which meant that the-mean-meaning crew members, including several wounded men, were desperately trying to navigate the treacherous vessel while also fighting for their lives amidst the perilous flooding.
Inverted Beta	As the ship entered entered the harbor, it was carrying tons of cargo and long passengers, but unfortunately, a large hole had developed in its hull, causing severe flooding, which meant that the wounded men on the ship were now in grave danger as they desperately fought to escape and stay afloat amidst the rising water.
RL-Mean (ours)	As the wounded ship entered the treacherous waters, a long hole began to widen, causing tons of water to flood in and threatening the lives of the men on board, meaning that the vessel was now in dire need of immediate attention to prevent further damage and ensure the safety of its crew amidst the perilous situation.
RL-Freq (ours)	As the long ship entered the treacherous waters, it encountered tons of water pouring in, causing severe flooding that ultimately meant the discovery of a hole in the ship's hull, further endangering the wounded men who now desperately needed to find a way to escape the rising water that threatened to engulf the entire vessel.

Table 4: Qualitative results for the keyword-conditioned generation task with **60 diffusion steps**. Compared to 30-step sampling, longer diffusion trajectories amplify the differences between guidance strategies. Learned adaptive guidance yields more fluent and semantically consistent outputs.

Length Control Prompt. Rewrite the input sentence so that the output length is within **40%–80%** of the original length. *Input:* "Pennsylvania Representative Joseph Ingersoll , a Whig , spoke against the bill , noting that Patterson opposed the new denominations , and that the idea had been repeatedly turned down , whenever considered."

Method	Generated Text
Fixed CFG	Pennsylvania Representative Joseph Ingersoll spoke against the bill, noting Patterson Patterson Patterson Patterson Patterson opposed the new denominations, and the idea had been repeatedly turned down whenever considered.
Linear Increase	'unsylvania Whig Ingersoll criticized the bill, citing Patterson's opposition to new denominations and repeated idea idea whenever considered.'
Linear Decrease	Representative Ingersgersoll opposed Whig denominations bill, citing Patterson Patterson's opposition and noting the idea had been repeatedly rejected.
Cosine Increase	Wh Pennsylvania Whig Ingersoll spoke against bill, citing Patterson's opposition to new denominations and noting the idea idea'd repeatedly turned down whenever considered.
Cosine Decrease	Representative Ingersgersoll opposed Whig denominations bill, citing Patterson Patterson's opposition, noting the idea repeatedly repeatedly rejected.
Beta	Representative Ingersoll Pennsylvania Whig bill bill. Patterson opposed Patterson new denominations. Idea turned down repeatedly whenever repeatedly whenever considered.
Inverted Beta	Representative Ingersgersoll Joseph Whig Pennsylvania spoke bill, noting Patterson Patterson opposed new denominations. Idea had idea repeatedly turned down repeatedly whenever considered.
RL-Mean (ours)	Representative Joseph Ingersoll, a Whig, spoke against the bill bill, citing Patterson's opposition to new denominations and repeated rejections.
RL-Freq (ours)	Whig Representative Joseph Ingersoll opposed the bill, citing Patterson's stance and repeated rejections.

Table 5: Qualitative results for the length-controlled rewriting task with **30 diffusion steps**. All methods are prompted to rewrite the input sentence within the target length range (40%–80% of the original). Adaptive guidance better preserves semantic content while satisfying length constraints.

Length Control Prompt. Rewrite the input sentence so that the output length is within **40%–80%** of the original length. *Input:* "Pennsylvania Representative Joseph Ingersoll , a Whig , spoke against the bill , noting that Patterson opposed the new denominations , and that the idea had been repeatedly turned down , whenever considered."

Method	Generated Text
Fixed CFG	Whig Rep Joseph Ingersoll opposed Pennsylvania bill, citing Patterson Patterson's opposition to new denominations and noting the idea had been repeatedly rejected.
Linear Increase	'unsylvania Whig Ingersoll spoke against bill, referencing Patterson's opposition to new denominations and noting idea turned down repeatedly whenever considered.'
Linear Decrease	Representative Ingersoll opposed Patterson bill, stating Patterson disapproved of new denominations, idea repeatedly rejected.
Cosine Increase	'unsylvania Whig Ingersoll spoke against bill, referencing Patterson's opposition to new denominations and noting idea turned down repeatedly whenever considered.'
Cosine Decrease	Representative Ingersoll opposed Patterson bill, stating Patterson disapproved of new denominations, idea repeatedly rejected.
Beta	Representative Ingersoll (Whig) opposed Pennsylvania bill, citing Patterson's opposition to new denominations and noting the idea was repeatedly rejected.
Inverted Beta	Representative Ingersoll opposed Pennsylvania bill, citing Patterson's stance to new denominations and noting the idea was repeatedly rejected.
RL-Mean (ours)	Representative Ingersoll spoke against the bill, citing Patterson's opposition to new denominations and noting that the idea had been repeatedly rejected.
RL-Freq (ours)	Representative Ingersoll spoke against the bill, citing Patterson's opposition to new denominations and noting that the idea had been repeatedly rejected.

Table 6: Qualitative results for the length-controlled rewriting task with **60 diffusion steps**. With longer diffusion trajectories, adaptive guidance produces more concise and faithful rewrites, while heuristic schedules often introduce redundancy or fragmentation.

Sentiment Transfer (Pos→Neg). Rewrite the sentence to **negative** sentiment while preserving content. *Input:* "the pepper steak is another good one ."

Method	Generated Text
Fixed CFG	pepper steak is another bad one .
Linear Increase	The pepper steak is not another good one .
Linear Decrease	pepper steak is another bad one .
Cosine Increase	the pepper steak lacks another good one .
Cosine Decrease	pepper steak is another bad one .
Beta	The pepper steak is not another good one.
Inverted Beta	pepper steak: another mediocre one .
RL-Mean (ours)	pepper steak is another bad one .
RL-Freq (ours)	pepper steak is another bad one .

Table 7: Qualitative results for sentiment style transfer from positive to negative with **30 diffusion steps**. All methods are evaluated on the same input sentence. Adaptive guidance achieves correct sentiment transfer with minimal distortion of the original content.

Sentiment Transfer (Pos→Neg). Rewrite the sentence to **negative** sentiment while preserving content. *Input:* "the pepper steak is another good one ."

Method	Generated Text
Fixed CFG	pepper steak is another bad one .
Linear Increase	the pepper steak is not another good one .
Linear Decrease	pepper steak is another bad one .
Cosine Increase	the pepper steak isn't another good one .
Cosine Decrease	pepper steak is another bad one .
Beta	The pepper steak isn't another good one.
Inverted Beta	pepper steak: another mediocre one .
RL-Mean (ours)	pepper steak is another bad one .
RL-Freq (ours)	pepper steak is another bad one .

Table 8: Qualitative results for sentiment style transfer from positive to negative with **60 diffusion steps**. Compared to shorter trajectories, longer sampling further stabilizes sentiment control under adaptive guidance.

Sentiment Transfer (Neg→Pos). Rewrite the sentence to **positive** sentiment while preserving content. *Input:* "never will i be back to this place "	
Fixed CFG	I will revisit revisit this place at some point.AlternativelyAlternatively:Positive: I will definitely be back to this place in the future, although it may not be under the original circumstances.Both rewritten sentences change the negative sentence into a positive sentence while preserving the original meaning.
Linear Increase	I shall never revisit this place.
Linear Decrease	Positive: revisit revisit revisit revisit this place .(I chose this rewritten sentence to preserve the original meaning of the negative sentence by using r̈eisitd̈o convey that the speaker speaker will not return or revisit the place. This replaces the negative word n̈everẅith a positive action of revisiting.)
Cosine Increase	I shall never revisit this place.
Cosine Decrease	Positive: never revisit revisit revisit this place .Or can be rewritten as: I will not return to this place.Or can be rewritten as: I will not return to this place.(Note, I have rewritten the negative sentence into a positive sentence while preserving the original meaning)
Beta	I will definitely return revisit this place.
Inverted Beta	Positive: never revisit revisit revisit this place .Alternatively, you could also use:Positive: I will not return to this location.OrPositive: I will not be here again.These sentences transform the negative sentence into a positive sentence while preserving its original meaning.
RL-Mean (ours)	I will never revisit this place.
RL-Freq (ours)	I will never revisit this place.

Table 9: Qualitative results for sentiment style transfer from negative to positive with **30 diffusion steps**. Heuristic guidance often leads to inconsistent polarity or repetitive artifacts, whereas adaptive guidance yields clearer sentiment reversal.

Sentiment Transfer (Neg→Pos). Rewrite the sentence to **positive** sentiment while preserving content. *Input:* "never will i be back to this place "	
Fixed CFG	I will never revisit this place.
Linear Increase	I no longer revisit this place.
Linear Decrease	I revisit revisit this place.
Cosine Increase	I shall never revisit this place.
Cosine Decrease	Positive: I revisit revisit this place.
Beta	I will not revisit this place. I reaffirm my intention to never return to this place or Positive: I choose not to return .
Inverted Beta	I revisit revisit this place.
RL-Mean (ours)	I will never revisit this place.
RL-Freq (ours)	Positive: I will not revisit this place.In this rewritten sentence, n̈everïs replaced with n̈ot revisitd̈o preserve the original meaning that the person will not return to the place. The positive sentence Ï will not revisit this placec̈onveys the same idea as the original negative sentence.

Table 10: Qualitative results for sentiment style transfer from negative to positive with **60 diffusion steps**. Adaptive guidance produces more stable and semantically faithful sentiment inversion under longer diffusion trajectories.