

Reliability-Calibrated Guidance for Few-Step Categorical Flow-Map Text Generation

Zihao Liu¹ Hongchen Wu¹ Xiaochang Fang¹
Hongxuan Li¹ Huaxiang Zhang¹ Jiande Sun^{1,2}

¹School of Computer Science and Artificial Intelligence,

²Interdisciplinary Research Center of General Artificial Intelligence,

Shandong Normal University, Jinan 250358, Shandong, China

lzhcs32@163.com wuhongchen@sdu.edu.cn 2021317109@stu.sdu.edu.cn

lihongxuan0117@gmail.com huaxzhang@hotmail.com jiandesun@hotmail.com

Abstract

Few-step flow-map guidance provides an efficient way to steer generation without retraining, but its guidance updates are naturally defined in a continuous or relaxed state space. When such guidance is applied to categorical text generation, the final output is instead determined by simplex-constrained token distributions and hard decoding. Under low NFE, where each guidance update induces a relatively large finite displacement, this continuous-to-discrete mismatch can make terminal reward guidance unreliable: updates may overshoot the base transition, conflict with generation quality, become unnecessarily aggressive after reward saturation, or fail to transfer from relaxed reward improvement to decoded tokens.

We present Reliability-Calibrated Flow Guidance (RCFG), a sampling-time reliability-calibration framework for terminal reward pull-back in few-step categorical flow maps. Rather than replacing the underlying guidance primitive, RCFG calibrates its magnitude and direction according to transition scale, reward saturation, local reward-quality conflict, and categorical endpoint structure. Experiments on LM1B and Text8 show more favorable aggregate control-likelihood operating points after calibration: on LM1B, RCFG maintains or improves verifier Reward and Target while reducing NLL and PPL relative to terminal reward-pullback guidance, while Text8 shows a modest aggregate improvement in lexical-control balance. Code and reproduction instructions are available at: <https://github.com/ZionLau/RCFG>

1 Introduction

Diffusion and flow-based generative models have recently moved rapidly toward few-step and even one-step generation (Song et al., 2023; Geng et al., 2025; Boffi et al., 2025b; Sabour et al., 2025; Lee et al., 2026). This trend is particularly important for language generation, where reducing the

number of sampling steps can make parallel non-autoregressive generation substantially more practical. Recent flow-map and consistency-style models show that high-quality samples can be produced with only a small number of function evaluations, suggesting a promising direction for efficient text generation (Roos et al., 2026; Potapchik et al., 2026).

However, fast generation alone is not enough. In many applications, a generator must also be steered toward a user-specified attribute or reward, such as topic, grammaticality, sentiment, safety, or lexical constraints. In text generation, these controls are ultimately judged on decoded sequences: a useful guidance method should satisfy the requested attribute or lexical constraint while avoiding repetitive, low-likelihood, or verifier-exploiting outputs. Classical diffusion guidance modifies the sampling trajectory using classifier gradients or classifier-free combinations of conditional and unconditional scores (Dhariwal and Nichol, 2021; Ho and Salimans, 2021). More recent training-free guidance methods further extend this idea to arbitrary differentiable rewards, external energy functions, and source- or trajectory-level optimization (Bansal et al., 2024; Yu et al., 2023; Feng et al., 2025). Most existing guidance methods were developed for many-step samplers, where small reward-based corrections can be applied gradually. When directly reused under low NFE budgets, each update must influence a much longer remaining horizon, so the same reward signal can become overly aggressive, weakly coupled to the terminal sample, or unstable after decoding.

This issue is especially pronounced for categorical text generation. In categorical flow-map models, guidance is applied in a relaxed continuous state space, while terminal token probabilities are obtained through simplex normalization and final text is produced by hard decoding (Roos et al., 2026; Potapchik et al., 2026). Thus, the reward used

during guidance is often evaluated on a relaxed terminal representation, whereas final evaluation is performed on decoded text. A small relaxed perturbation may leave the decoded tokens unchanged, while a perturbation near a decoding boundary may abruptly flip tokens. Moreover, reward improvement and generation quality are not always aligned: a reward-increasing direction may also reduce fluency, increase repetition, or exploit a reward-model shortcut.

Figure 1 illustrates this low-NFE mismatch on a categorical probability simplex. Directly reusing many-step guidance with only a few updates can make the late-stage update boundary-sensitive and poorly aligned with hard decoding, motivating our view that few-step categorical guidance should calibrate when and how strongly the pulled-back reward direction can be trusted.

Recent few-step flow-map guidance methods, such as FMRG, mitigate the horizon problem by using a learned flow map to pull terminal reward gradients back to intermediate states (Huang et al., 2026). However, this pullback remains an ambient-space reward-gradient update whose reliability is not guaranteed in categorical flow maps. It is computed in relaxed continuous coordinates, whereas terminal text is determined only after simplex normalization and hard decoding. As a result, a relaxed reward-increasing perturbation may be absorbed within the same decoding region, while a small perturbation near an argmax boundary may flip tokens and degrade fluency or increase repetition. Thus, terminal reward pullback is informative, but its strength and direction must be calibrated in categorical few-step generation.

In this work, we present **Reliability-Calibrated Flow Guidance (RCFG)** as a sampling-time reliability-calibration framework for terminal reward pullback in few-step categorical flow-map generation. RCFG treats the reliability of terminal reward pullback as the central design object. It calibrates the pulled-back update along three coupled aspects: the effective magnitude of the update relative to the base flow-map transition and remaining target headroom, the local directional conflict between reward ascent and quality preservation, and the interaction between relaxed terminal rewards and categorical endpoint structure. The additive update is applied in the relaxed ambient state space, while categorical structure enters through terminal simplex normalization and a simplex-valid mirror target. This design is intended to improve the reli-

bility of the terminal reward-pullback update under few-step categorical decoding.

We evaluate RCFG on two few-step text guidance settings. On LM1B attribute-guided generation, we consider four verifier-based control tasks: AGNews-Sports, CoLA-Acceptable, IMDb-Positive, and TweetEval-NonOffensive, following recent flow-map language-model guidance protocols (Roos et al., 2026; Huang et al., 2026). We report external verifier reward, target success rate, negative log-likelihood, and GPT-2-large perplexity to evaluate both attribute control and generation quality. On Text8 target-word guidance, we evaluate exact lexical control using Exact-1, which requires the target word to appear exactly once and therefore penalizes both missing the target and over-insertion. Together, these settings test classifier-based attribute control and lexical exact-count control in the low-NFE regime.

Our experiments indicate a more favorable aggregate control–likelihood trade-off after reliability calibration. On LM1B, RCFG maintains or improves average Reward and Target compared with terminal reward-pullback guidance while reducing NLL and PPL, with clearer gains at larger few-step budgets. On Text8, RCFG provides a modest aggregate improvement in lexical-control balance over FMRG. Scale-control diagnostics and matched component analyses further support the role of reliability calibration in these gains.

Our contributions are summarized as follows:

- We revisit few-step categorical flow-map guidance from the perspective of *pullback reliability*. Instead of treating terminal reward pullback as either sufficient or ineffective, we show that it provides a useful terminal direction whose effectiveness depends on the sampling step, endpoint sensitivity, reward state, categorical relaxation, and generation-quality interaction.
- We develop **Reliability-Calibrated Flow Guidance (RCFG)**, a sampling-time framework that calibrates the reliability of terminal reward pullback in few-step categorical generation. RCFG keeps pullback guidance as the central primitive and calibrates its use according to the reliability of the current transition, rather than replacing it with an unrelated objective or relying on a fixed global guidance scale.

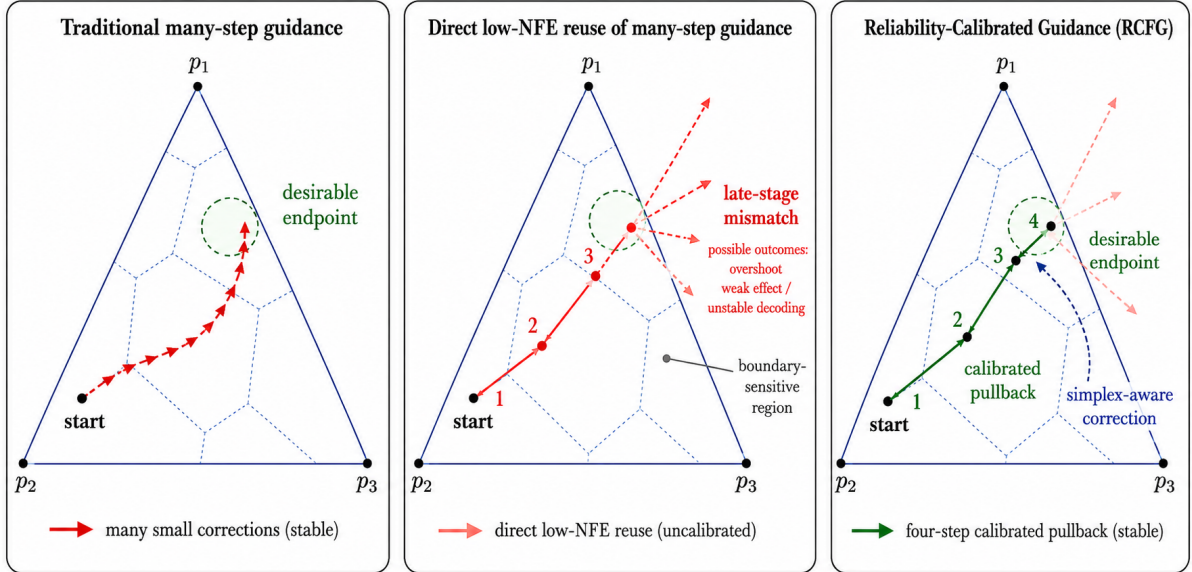


Figure 1: Illustration of guidance on a categorical probability simplex. Left: many-step guidance accumulates small corrections gradually. Middle: directly reusing many-step guidance in the low-NFE regime can lead to late-stage mismatch, making the final update boundary-sensitive or unstable after hard decoding. Right: RCFG preserves the guidance signal while calibrating the trajectory toward a more stable desirable endpoint region.

- We support this view through diagnostic analyses and experiments on LM1B attribute-guided generation and Text8 target-word guidance. The analyses explain why pullback reliability can vary in categorical few-step sampling, and the experiments show more favorable aggregate control–likelihood operating points after calibration, with gains that are not explained by fixed-scale tuning or target over-insertion alone.

2 Related Work

Guidance for diffusion and flow models.

Inference-time guidance steers a pretrained generative model toward a desired condition or reward without updating model parameters. Classical diffusion guidance uses classifier gradients or classifier-free score combinations (Dhariwal and Nichol, 2021; Ho and Salimans, 2021), while recent training-free methods extend guidance to differentiable objectives, external energies, source/noise optimization, source-distribution modification, optimal-control views, search-based steering, and adaptive schedules (Bansal et al., 2024; Yu et al., 2023; Ben-Hamu et al., 2024; Wang et al., 2025; Guo et al., 2025; Koulischer et al., 2025). Guidance has also been formulated directly for flow matching, including energy guidance, source-level guidance, and exact guidance for discrete flow

matching (Feng et al., 2025; Wan et al., 2026). These methods provide flexible control mechanisms, but most are designed for many-step samplers, continuous image/latent spaces, or discrete transition-state models. In few-step categorical text generation, directly reusing such updates can be unreliable because each step must affect a long future horizon, while the guidance signal is computed on relaxed states and realized only after hard token decoding.

Flow maps and few-step generation.

Flow matching and stochastic interpolants learn continuous-time generative dynamics from a simple source distribution to data (Lipman et al., 2023; Albergo et al., 2025). To reduce inference cost, consistency-style, mean-flow, and flow-map models approximate large transitions or solution operators between time points, enabling one-step or few-step generation (Song et al., 2023; Boffi et al., 2025a; Geng et al., 2025; Boffi et al., 2025b; Sabour et al., 2025). For language generation, recent flow-map language models distill continuous denoising over one-hot token encodings into efficient few-step or one-step samplers (Lee et al., 2026), while categorical and discrete flow-map formulations connect trajectory compression with simplex-valued or categorical endpoint structure (Roos et al., 2026; Potapchik et al., 2026). For guidance under low NFE budgets, FMRG uses

flow-map look-ahead to pull terminal reward gradients back to intermediate states (Huang et al., 2026). However, in categorical text flow maps, this pullback is computed on relaxed states, evaluated through terminal simplex-normalized token distributions, and realized after hard decoding, making its reliability state-dependent. Our work follows the few-step flow-map perspective, but focuses on calibrating the reliability of terminal reward pullback for categorical text.

Guidance for discrete text generation. Another line of work studies diffusion language models with discrete noising processes, including masked diffusion language models and uniform-state discrete diffusion models (Sahoo et al., 2024, 2025). Their few-step generation efficiency can be improved through distillation, such as self-distillation through time and discrete consistency distillation (Deschenaux and Gulcehre, 2025; Sahoo et al., 2025). Guidance in these models typically modifies discrete transition probabilities, denoising distributions, or objectives on noised/intermediate states, and recent exact-guidance work derives guidance at the level of discrete transition rates (Wan et al., 2026). These methods are well suited to CTMC, masked diffusion, or discrete transition-state samplers, whereas categorical flow maps sample through relaxed continuous states and obtain terminal categorical probabilities by simplex normalization. We therefore treat discrete diffusion and discrete-flow guidance as cross-family references, while comparing mainly against same-backbone categorical flow-map guidance.

Reward-guided text generation and evaluation. Recent few-step LM work evaluates reward-guided generation using external verifiers for topic, grammaticality, sentiment, and safety control (Lee et al., 2026). Since verifier reward alone can be inflated by repetition, classifier shortcuts, or low-likelihood text, we report reward together with target rate and NLL/PPL, and include diagnostics for scale tuning and target over-insertion.

Position of this work. RCFG is designed for few-step categorical flow-map guidance. Unlike ordinary continuous-space guidance, it accounts for the fact that a pulled-back reward direction must pass through relaxed categorical endpoints and hard decoding before affecting the final text. Unlike prior flow-map guidance, it does not treat terminal reward pullback as either sufficient or in-

effective; instead, it treats pullback reliability as state-dependent and calibrates how strongly the direction should be trusted according to the current transition, endpoint sensitivity, reward state, categorical relaxation, and reward-quality interaction. Unlike discrete diffusion or discrete-flow guidance, RCFG does not require a CTMC transition kernel, a noised-state classifier, or an exact transition-rate correction, and can be applied directly at sampling time to categorical flow maps. Accordingly, we view RCFG as a structured reliability-calibration refinement of terminal reward-pullback guidance rather than a separate guidance paradigm.

3 Method

3.1 Problem Setup

We study guidance for few-step categorical flow-map generation. Let $x_s \in \mathcal{X} \subseteq \mathbb{R}^{L \times V}$ denote the relaxed continuous state at time s , where L is the sequence length and V is the vocabulary size. The additive guidance update is applied in this ambient relaxed state space, whereas categorical structure is introduced when terminal predictions are converted into simplex-normalized vocabulary distributions. The categorical endpoint structure is summarized in Appendix F.

A learned categorical flow map $\Phi_{s \rightarrow t} : \mathcal{X} \rightarrow \mathcal{X}$ maps the current relaxed state from time s to time t . In few-step sampling, a small number of such maps are composed to produce the terminal prediction. Given a current state x_s , we denote the terminal categorical prediction by

$$p_1 = \sigma_{\Delta}(\Phi_{s \rightarrow 1}(x_s)) \in \Delta^{L \times V}, \quad (1)$$

where σ_{Δ} denotes the differentiable simplex normalization, implemented by softmax normalization in our experiments. In our implementation, the hard decoder $\mathcal{D}$ first selects the highest-probability token at each position and then converts the resulting token IDs into text using the tokenizer. This hard-decoding step induces piecewise-constant argmax regions, as discussed in Appendix D.

During guidance, we optimize a differentiable surrogate reward evaluated on the relaxed terminal endpoint:

$$r(p_1) = \log P_{\psi}(c \mid p_1), \quad (2)$$

where P_{ψ} is the reward model and c is the target attribute. The relaxed reward interface, including the distinction between relaxed categorical inputs and

hard decoded text, is discussed in Appendix F. Our goal is to improve the decoded text with respect to the target attribute while preserving generation quality.

We use the relaxed reward as a differentiable guidance signal, rather than explicitly learning a soft-to-hard reward alignment model. Nevertheless, in few-step categorical generation, raw soft-reward guidance can become unreliable under terminal categorical structure, hard-decoding discontinuities, target saturation, and reward-quality conflict. We use these reliability issues as motivation for the calibrated design below, with the soft-hard discrepancy analyzed diagnostically in Appendix B.

3.2 Terminal Reward Pullback

Following terminal reward pullback in flow-map reward guidance such as FMRG (Huang et al., 2026), we use terminal reward differentiation as the basic guidance primitive. The terminal reward objective viewed from time s is

$$\mathcal{J}_s(x_s) = r(\sigma_\Delta(\Phi_{s \rightarrow 1}(x_s))). \quad (3)$$

The corresponding terminal reward-pullback direction is

$$g_R = \nabla_{x_s} \mathcal{J}_s(x_s). \quad (4)$$

This direction applies the update at the current relaxed state while evaluating it by its terminal effect through the remaining flow map. A short derivation and few-step sensitivity discussion are provided in Appendix A. We additionally provide a Gaussian toy example in Appendix G, which shows that even for a linear-Gaussian flow map with a quadratic reward, the effective terminal movement depends on the reward scale, the step magnitude, and the endpoint sensitivity of the flow map.

Raw terminal pullback is useful, but it is agnostic to several reliability factors that become prominent in few-step categorical generation: the limited reward headroom of saturated samples, local conflict between reward ascent and generation quality, and the instability caused by large relaxed perturbations before hard decoding. These effects do not invalidate soft reward guidance, but they make raw reward pullback less reliable as a standalone update rule. Therefore, we introduce a reliability-calibrated guidance rule for categorical flow maps.

3.3 Reliability-Calibrated Flow Guidance

RCFG is a sampling-time calibration framework around terminal reward pullback: it retains the

same primary guidance primitive while calibrating its magnitude and direction. At each step from s to t , the guided relaxed state is

$$x_s^+ = x_s + \eta(t-s)a_{\text{sat}}(u_s)\|v_s\|_2 d_{\text{RCFG}}, \quad (5)$$

where the addition is performed in the relaxed ambient state space $\mathcal{X}$. The categorical simplex enters through the terminal normalization in Equation (1) and through the simplex-valid mirror target introduced below. The base flow-map velocity is

$$v_s = \frac{\Phi_{s \rightarrow t}(x_s) - x_s}{t-s}, \quad (6)$$

η is the guidance scale, and d_{RCFG} is a normalized reliability-calibrated direction.

Scaling the guidance step by $\|v_s\|_2$ makes the guidance magnitude comparable to the natural scale of the base flow-map transition, instead of using an absolute gradient magnitude independent of the base transition scale. Because d_{RCFG} is normalized and $a_{\text{sat}}(u_s) \in [a_{\min}, 1]$, the relaxed-state displacement satisfies

$$\begin{aligned} \|x_s^+ - x_s\|_2 &\leq \eta(t-s)a_{\text{sat}}(u_s)\|v_s\|_2 \\ &\leq \eta(t-s)\|v_s\|_2. \end{aligned} \quad (7)$$

This bound should be read as a magnitude calibration property, not as a global guarantee on decoded reward. The need for such calibration is illustrated by the Gaussian toy example in Appendix G, where the effective terminal step size depends on flow-map endpoint sensitivity, and by the hard-decoding boundary analysis in Appendix D, where the number of potentially changed decoded tokens depends on the size of the relaxed perturbation. The code-aligned form of the full update is given in Appendix E.

Saturation-aware step scale. Let $u_s = P_\psi(c | p_1)$ be the target confidence at the predicted endpoint. The soft reward in Equation (2) uses the target log-probability, while u_s uses the corresponding target probability. When u_s is already high, the remaining target-reward gain is limited, while aggressive guidance may still degrade fluency or diversity. We therefore use the saturation scale

$$a_{\text{sat}}(u_s) = a_{\min} + (1 - a_{\min})(1 - u_s)^\rho, \quad (8)$$

where $a_{\min}$ is a floor value and ρ controls how quickly the update is reduced as the target confidence approaches one. This scale preserves strong

guidance for unsatisfied samples and becomes conservative for saturated ones. Its state-dependent behavior and displacement effect are detailed in Appendix E.

Conflict-aware quality correction. To reduce reward-quality conflict, we introduce a differentiable quality surrogate

$$Q(p_1) = \alpha_H H(p_1) - \alpha_{KL} D_{KL}(p_1 \| p_1^{\text{base}}), \quad (9)$$

where $H(p_1)$ is the average token entropy and p_1^{base} is the detached endpoint predicted by the unguided flow map. The quality pullback direction is

$$g_Q = \nabla_{x_s} Q(\sigma_\Delta(\Phi_{s \rightarrow 1}(x_s))). \quad (10)$$

If the reward and quality directions conflict, i.e., $\langle g_R, g_Q \rangle < 0$, moving along g_R decreases the quality surrogate to first order. We therefore remove from g_R the component that is opposed to g_Q :

$$g_R^{\text{oq}} = g_R - \frac{\min\{0, \langle g_R, g_Q \rangle\}}{\|g_Q\|_2^2 + \epsilon} g_Q. \quad (11)$$

The primary quality-corrected direction is then

$$g_P = u(g_R^{\text{oq}}) + \lambda_Q u(g_Q), \quad u(z) = \frac{z}{\max(\|z\|_2, \epsilon)}. \quad (12)$$

All vector norms and normalizations are applied per sample. This branch is designed to retain alignment with the terminal reward direction while reducing local first-order conflict with the quality surrogate. It is a local correction rather than a global multi-objective optimality guarantee; see Appendix E.

Simplex-aware residual correction. The previous branch corrects reward-quality conflict in the relaxed state space, but it does not explicitly use the simplex structure of the terminal categorical prediction. Since the reward is evaluated through $p_1 = \sigma_\Delta(\Phi_{s \rightarrow 1}(x_s))$, we construct an auxiliary terminal target directly on the vocabulary simplex and then pull it back to the current relaxed state. The relevant terminal simplex structure is discussed in Appendix C and Appendix F.

We first compute the terminal reward gradient on the simplex $r_p = \nabla_{p_1} r(p_1)$, and construct a mirror target on the vocabulary simplex:

$$q = \sigma_\Delta(\log(p_1 + \epsilon) + \eta_m [r_p - \text{mean}(r_p)]). \quad (13)$$

Here the mean is taken over vocabulary coordinates at each position. The target q is a valid categorical distribution and can be interpreted as a

KL-regularized reward-ascent target on the simplex; its mirror-ascent interpretation is given in Appendix F.3.

We then pull this terminal target back to the current relaxed state by minimizing the terminal KL divergence:

$$g_M = -\nabla_{x_s} D_{KL}(q \| \sigma_\Delta(\Phi_{s \rightarrow 1}(x_s))). \quad (14)$$

To avoid using a mirror correction that directly opposes the reward direction, we remove the opposing component when necessary:

$$\tilde{g}_M = g_M - \frac{\min\{0, \langle g_M, g_R \rangle\}}{\|g_R\|_2^2 + \epsilon} g_R. \quad (15)$$

The mirror correction is then combined with the reward direction through a confidence-dependent gate:

$$g_C = u(g_R) + \lambda_C \text{sigmoid}\left(\frac{u_s - \tau_g}{\kappa}\right) u(\tilde{g}_M), \quad (16)$$

where τ_g and κ control the activation threshold and smoothness. The construction of this mirror correction and its confidence-dependent gate is provided in Appendix E.

Instead of directly adding g_C to the primary quality-corrected direction, we keep only the component orthogonal to g_P :

$$g_{\text{res}} = g_C - \frac{\langle g_C, g_P \rangle}{\|g_P\|_2^2 + \epsilon} g_P. \quad (17)$$

The final RCFG direction is

$$d_{\text{RCFG}} = u(u(g_P) + \lambda_M u(g_{\text{res}})). \quad (18)$$

The orthogonal residual lets the simplex-aware branch provide complementary terminal categorical information without canceling the primary reward-quality direction. The corresponding orthogonal composition is detailed in Appendix E. In our implementation, λ_C and λ_M are controlled by the same mixing coefficient unless otherwise specified.

3.4 Sampling Procedure

At each sampling step from s to t , RCFG first predicts the terminal categorical distribution p_1 using the learned flow map, computes the calibrated direction d_{RCFG} , applies the relaxed-state update in Equation (5), and then advances the state with the base flow map:

$$x_t = \Phi_{s \rightarrow t}(x_s^+). \quad (19)$$

Table 1: Terminal reward pullback and the three reliability calibrations in RCFG.

Component	Reliability question	Main operation
Raw FMRG	How is the terminal reward direction obtained?	Terminal reward pullback
Magnitude calibration	Is the update transition-compatible?	Transition-relative and saturation scaling
Quality correction	Does reward ascent conflict with local quality?	Removes harmful first-order conflict
Simplex residual	Is the change realizable after categorical decoding?	Mirror target and simplex-feasible residual

Thus, RCFG calibrates the terminal reward-pullback update at sampling time before each categorical flow-map transition; it does not retrain the base flow map. The controlled variants used in the Text8 diagnostic comparisons are defined in Appendix H.3. For completeness, Appendix E.8 provides the step-by-step sampling procedure.

4 Experiments

All experiments build on the flow-map backbone from Categorical Flow Maps (Roos et al., 2026). We use a checkpoint trained for 500K steps on LM1B and a checkpoint trained for 100K steps on Text8. All guidance methods operate only at sampling time; RCFG requires no retraining or modification of the pretrained flow-map backbone.

4.1 LM1B Attribute-Guided Generation

We evaluate LM1B attribute-guided generation on AGNews-Sports, CoLA-Acceptable, IMDB-Positive, and TweetEval-NonOffensive. For each task–NFE–method group, we select hyperparameters by validation verifier reward and average over tasks. We report Reward, Target, NLL, and GPT-2-large PPL; higher Reward/Target and lower NLL/PPL are better. Guidance uses task-specific differentiable soft-BERT reward models, whereas final Target success is evaluated by separate frozen Hugging Face classifiers that are never used for guidance; NLL/PPL use an independent causal language model. Selection rules are summarized in Table 23. ATG is a stop-gradient endpoint baseline that uses the same reward and update as FMRG but removes the flow-map pullback.

Table 2 shows that RCFG yields a more favorable aggregate control–likelihood operating point than FMRG at each tested NFE. For example, at NFE= 8, RCFG raises Reward/Target from 0.592/0.617 to 0.621/0.648 while reducing NLL/PPL from 3.92/50.4 to 3.71/40.9; we next provide the task-wise breakdown. RCFG improves the aggregate operating point without uniformly dominating every task–NFE cell.

Table 2: Main LM1B comparison using the 500K-step flow-map backbone. Results are averaged over four attribute-control tasks with 512 samples per condition. Higher Reward and Target are better; lower NLL and PPL are better. Values after $\pm$ denote standard deviations.

NFE	Method	Reward $\uparrow$	Target $\uparrow$	NLL $\downarrow$	PPL $\downarrow$
2	Base	0.465 $\pm$ 0.009	0.500 $\pm$ 0.018	4.83 $\pm$ 0.033	125.6 $\pm$ 4.5
	ATG	0.422 $\pm$ 0.008	0.410 $\pm$ 0.018	4.61 $\pm$ 0.039	109.8 $\pm$ 5.0
	FMRG	0.555 $\pm$ 0.011	0.580 $\pm$ 0.021	4.32 $\pm$ 0.039	75.2 $\pm$ 5.1
	RCFG	0.571$\pm$0.012	0.600$\pm$0.022	4.17$\pm$0.036	64.8$\pm$4.6
4	Base	0.486 $\pm$ 0.010	0.521 $\pm$ 0.020	4.87 $\pm$ 0.036	130.4 $\pm$ 5.0
	ATG	0.441 $\pm$ 0.009	0.430 $\pm$ 0.020	4.63 $\pm$ 0.040	182.7 $\pm$ 9.2
	FMRG	0.574 $\pm$ 0.012	0.599 $\pm$ 0.022	4.06 $\pm$ 0.037	58.0 $\pm$ 4.3
	RCFG	0.597$\pm$0.013	0.625$\pm$0.023	3.87$\pm$0.034	47.9$\pm$3.9
8	Base	0.505 $\pm$ 0.011	0.535 $\pm$ 0.021	4.91 $\pm$ 0.039	135.4 $\pm$ 5.6
	ATG	0.462 $\pm$ 0.010	0.461 $\pm$ 0.022	4.57 $\pm$ 0.040	99.9 $\pm$ 5.0
	FMRG	0.592 $\pm$ 0.013	0.617 $\pm$ 0.023	3.92 $\pm$ 0.033	50.4 $\pm$ 3.6
	RCFG	0.621$\pm$0.014	0.648$\pm$0.024	3.71$\pm$0.031	40.9$\pm$3.3

Table 3: LM1B results on IMDB-Positive using the 500K-step backbone.

NFE	Method	Reward $\uparrow$	Target $\uparrow$	NLL $\downarrow$	PPL $\downarrow$
2	Base	0.625 $\pm$ 0.018	0.650 $\pm$ 0.034	4.81 $\pm$ 0.035	123.1 $\pm$ 5.4
	ATG	0.390 $\pm$ 0.015	0.320 $\pm$ 0.030	5.21 $\pm$ 0.051	182.7 $\pm$ 9.2
	FMRG	0.815 $\pm$ 0.018	0.820 $\pm$ 0.033	4.47 $\pm$ 0.044	86.8 $\pm$ 6.4
	RCFG	0.820$\pm$0.019	0.840$\pm$0.034	4.32$\pm$0.041	74.8$\pm$5.7
4	Base	0.645 $\pm$ 0.019	0.670 $\pm$ 0.036	4.85 $\pm$ 0.037	127.8 $\pm$ 5.8
	ATG	0.420 $\pm$ 0.016	0.360 $\pm$ 0.032	5.08 $\pm$ 0.049	160.2 $\pm$ 8.5
	FMRG	0.790$\pm$0.019	0.806 $\pm$ 0.034	4.21 $\pm$ 0.042	67.0 $\pm$ 5.6
	RCFG	0.786 $\pm$ 0.020	0.815$\pm$0.034	4.02$\pm$0.039	55.3$\pm$5.0
8	Base	0.670 $\pm$ 0.020	0.700 $\pm$ 0.038	4.89 $\pm$ 0.040	132.7 $\pm$ 6.5
	ATG	0.455 $\pm$ 0.017	0.430 $\pm$ 0.034	4.96 $\pm$ 0.046	142.7 $\pm$ 7.8
	FMRG	0.760 $\pm$ 0.020	0.788$\pm$0.033	4.07 $\pm$ 0.038	58.2 $\pm$ 4.7
	RCFG	0.773$\pm$0.021	0.787 $\pm$ 0.033	3.86$\pm$0.036	47.2$\pm$4.3

Table 4: LM1B results on TweetEval-NonOffensive using the 500K-step backbone.

NFE	Method	Reward $\uparrow$	Target $\uparrow$	NLL $\downarrow$	PPL $\downarrow$
2	Base	0.870 $\pm$ 0.004	1.000$\pm$0.000	4.84 $\pm$ 0.032	126.9 $\pm$ 4.1
	ATG	0.878 $\pm$ 0.004	1.000$\pm$0.000	4.51 $\pm$ 0.035	90.7 $\pm$ 3.8
	FMRG	0.880 $\pm$ 0.007	1.000$\pm$0.000	4.27 $\pm$ 0.039	71.1 $\pm$ 4.9
	RCFG	0.884$\pm$0.008	1.000$\pm$0.000	4.12$\pm$0.036	61.3$\pm$4.6
4	Base	0.875 $\pm$ 0.004	1.000$\pm$0.000	4.88 $\pm$ 0.034	131.7 $\pm$ 4.4
	ATG	0.883 $\pm$ 0.004	1.000$\pm$0.000	4.40 $\pm$ 0.034	81.2 $\pm$ 3.5
	FMRG	0.891 $\pm$ 0.008	1.000$\pm$0.000	4.01 $\pm$ 0.037	54.8 $\pm$ 3.9
	RCFG	0.892$\pm$0.009	1.000$\pm$0.000	3.82$\pm$0.034	45.3$\pm$3.8
8	Base	0.880 $\pm$ 0.005	1.000$\pm$0.000	4.92 $\pm$ 0.036	136.7 $\pm$ 4.8
	ATG	0.889 $\pm$ 0.005	1.000$\pm$0.000	4.28 $\pm$ 0.033	72.2 $\pm$ 3.2
	FMRG	0.903$\pm$0.009	1.000$\pm$0.000	3.87 $\pm$ 0.033	47.6 $\pm$ 3.4
	RCFG	0.896 $\pm$ 0.010	1.000$\pm$0.000	3.66$\pm$0.031	38.7$\pm$3.3

Configuration-matched statistical analyses further support these results: paired bootstrap intervals for NFE= 2 and 4 and three-seed paired tests for NFE= 8 are reported in Tables 11 and 12.

We further compare RCFG with D-Flow (Ben-

Table 5: LM1B results on AGNews-Sports using the 500K-step backbone.

NFE	Method	Reward $\uparrow$	Target $\uparrow$	NLL $\downarrow$	PPL $\downarrow$
2	Base	0.310 $\pm$ 0.008	0.340 $\pm$ 0.035	4.82 $\pm$ 0.034	124.3 $\pm$ 4.4
2	ATG	0.285 $\pm$ 0.007	0.300 $\pm$ 0.033	4.71 $\pm$ 0.036	110.8 $\pm$ 4.0
2	FMRG	0.335 $\pm$ 0.010	0.405 $\pm$ 0.033	4.17 $\pm$ 0.036	64.3 $\pm$ 4.4
2	RCFG	0.365$\pm$0.011	0.430$\pm$0.034	4.02$\pm$0.033	55.4$\pm$3.9
4	Base	0.350 $\pm$ 0.010	0.400 $\pm$ 0.038	4.86 $\pm$ 0.038	129.1 $\pm$ 5.1
4	ATG	0.300 $\pm$ 0.008	0.325 $\pm$ 0.035	4.68 $\pm$ 0.040	107.4 $\pm$ 4.6
4	FMRG	0.370 $\pm$ 0.011	0.445 $\pm$ 0.036	3.91 $\pm$ 0.034	49.6 $\pm$ 3.7
4	RCFG	0.420$\pm$0.012	0.480$\pm$0.037	3.72$\pm$0.031	41.0$\pm$3.3
8	Base	0.365 $\pm$ 0.011	0.410 $\pm$ 0.039	4.90 $\pm$ 0.041	134.0 $\pm$ 5.8
8	ATG	0.315 $\pm$ 0.009	0.345 $\pm$ 0.036	4.62 $\pm$ 0.042	101.5 $\pm$ 5.0
8	FMRG	0.410 $\pm$ 0.012	0.475 $\pm$ 0.038	3.77 $\pm$ 0.030	43.1 $\pm$ 3.0
8	RCFG	0.465$\pm$0.013	0.520$\pm$0.039	3.56$\pm$0.028	35.0$\pm$2.7

Table 6: LM1B results on CoLA-Acceptable using the 500K-step backbone.

NFE	Method	Reward $\uparrow$	Target $\uparrow$	NLL $\downarrow$	PPL $\downarrow$
2	Base	0.055 $\pm$ 0.004	0.010 $\pm$ 0.004	4.85 $\pm$ 0.031	128.1 $\pm$ 4.1
2	ATG	0.135 $\pm$ 0.007	0.020 $\pm$ 0.008	4.01$\pm$0.032	55.0$\pm$3.1
2	FMRG	0.190 $\pm$ 0.009	0.095 $\pm$ 0.018	4.37 $\pm$ 0.037	78.6 $\pm$ 4.7
2	RCFG	0.215$\pm$0.010	0.130$\pm$0.020	4.22 $\pm$ 0.034	67.7 $\pm$ 4.2
4	Base	0.075 $\pm$ 0.005	0.015 $\pm$ 0.005	4.89 $\pm$ 0.035	133.0 $\pm$ 4.8
4	ATG	0.160 $\pm$ 0.008	0.035 $\pm$ 0.012	4.36 $\pm$ 0.036	78.0 $\pm$ 3.7
4	FMRG	0.245 $\pm$ 0.010	0.145 $\pm$ 0.018	4.11 $\pm$ 0.035	60.6 $\pm$ 4.0
4	RCFG	0.290$\pm$0.011	0.205$\pm$0.021	3.92$\pm$0.032	50.0$\pm$3.5
8	Base	0.105 $\pm$ 0.006	0.030 $\pm$ 0.008	4.93 $\pm$ 0.037	138.2 $\pm$ 5.3
8	ATG	0.190 $\pm$ 0.010	0.070 $\pm$ 0.016	4.42 $\pm$ 0.038	83.2 $\pm$ 4.1
8	FMRG	0.295 $\pm$ 0.011	0.205 $\pm$ 0.021	3.97 $\pm$ 0.031	52.7 $\pm$ 3.3
8	RCFG	0.350$\pm$0.012	0.285$\pm$0.024	3.76$\pm$0.029	42.7$\pm$2.9

Hamu et al., 2024), SGFM (Wang et al., 2026), Tree-G (Guo et al., 2025), OC-Flow (Wang et al., 2025), and reward-based Best-of-8 reranking at NFE $\in \{2, 4, 8\}$ on the same LM1B backbone (Table 7). On the four-task macro-average, RCFG achieves the strongest aggregate control results and the lowest NLL/PPL across all three NFE budgets.

Figure 2 places RCFG in the high-Target, low-PPL region among the compared NFE=8 guidance baselines, showing that its reward gains are not accompanied by degradation in the reported automatic likelihood-based quality metrics.

Guidance-scale sweeps further support a more favorable control-likelihood trade-off for RCFG over FMRG (Figure 4).

4.2 Target-word Guidance and Extended Control on Text8

We further evaluate Text8 target-word guidance to isolate lexical control from external classifier effects. We use five target sets and focus on NFE=8. The primary metric is Exact-1, which requires the target word to appear exactly once and therefore penalizes both missing and over-insertion.

For Text8, we retain the validation-selected controlled variants as an operating-point comparison rather than a configuration-matched component attribution. Component necessity is assessed sepa-

Table 7: Comparison with broader guidance baselines on LM1B using the 500K-step backbone. Results are averaged over four attribute-control tasks. Higher Reward and Target are better; lower NLL and PPL are better.

NFE	Method	Reward $\uparrow$	Target $\uparrow$	NLL $\downarrow$	PPL $\downarrow$
2	D-Flow	0.386 $\pm$ 0.013	0.401 $\pm$ 0.029	5.82 $\pm$ 0.052	337.6 $\pm$ 19.8
	SGFM	0.375 $\pm$ 0.014	0.395 $\pm$ 0.030	6.16 $\pm$ 0.055	476.2 $\pm$ 29.1
	Tree-G	0.283 $\pm$ 0.016	0.309 $\pm$ 0.032	7.42 $\pm$ 0.073	1668.0 $\pm$ 112.4
	OC-Flow	0.487 $\pm$ 0.013	0.505 $\pm$ 0.027	4.86 $\pm$ 0.043	129.2 $\pm$ 8.7
	Best-of-8	0.443 $\pm$ 0.012	0.460 $\pm$ 0.025	4.29 $\pm$ 0.039	72.8 $\pm$ 4.8
	RCFG	0.571$\pm$0.012	0.600$\pm$0.022	4.17$\pm$0.036	64.8$\pm$4.6
4	D-Flow	0.411 $\pm$ 0.012	0.428 $\pm$ 0.027	5.56 $\pm$ 0.049	260.1 $\pm$ 16.0
	SGFM	0.401 $\pm$ 0.013	0.420 $\pm$ 0.028	5.96 $\pm$ 0.052	388.2 $\pm$ 24.8
	Tree-G	0.305 $\pm$ 0.015	0.332 $\pm$ 0.031	7.18 $\pm$ 0.069	1310.6 $\pm$ 86.3
	OC-Flow	0.521 $\pm$ 0.012	0.540 $\pm$ 0.025	4.57 $\pm$ 0.041	96.7 $\pm$ 6.5
	Best-of-8	0.470 $\pm$ 0.011	0.488 $\pm$ 0.023	3.96 $\pm$ 0.035	52.5 $\pm$ 3.8
	RCFG	0.597$\pm$0.013	0.625$\pm$0.023	3.87$\pm$0.034	47.9$\pm$3.9
8	D-Flow	0.438 $\pm$ 0.011	0.455 $\pm$ 0.025	5.30 $\pm$ 0.046	200.3 $\pm$ 13.2
	SGFM	0.426 $\pm$ 0.012	0.447 $\pm$ 0.026	5.71 $\pm$ 0.049	301.8 $\pm$ 19.6
	Tree-G	0.339 $\pm$ 0.014	0.365 $\pm$ 0.029	6.94 $\pm$ 0.064	1031.4 $\pm$ 66.7
	OC-Flow	0.551 $\pm$ 0.011	0.572 $\pm$ 0.024	4.31 $\pm$ 0.039	74.4 $\pm$ 5.1
	Best-of-8	0.501 $\pm$ 0.010	0.520 $\pm$ 0.022	3.78 $\pm$ 0.032	43.9 $\pm$ 3.3
	RCFG	0.621$\pm$0.014	0.648$\pm$0.024	3.71$\pm$0.031	40.9$\pm$3.3

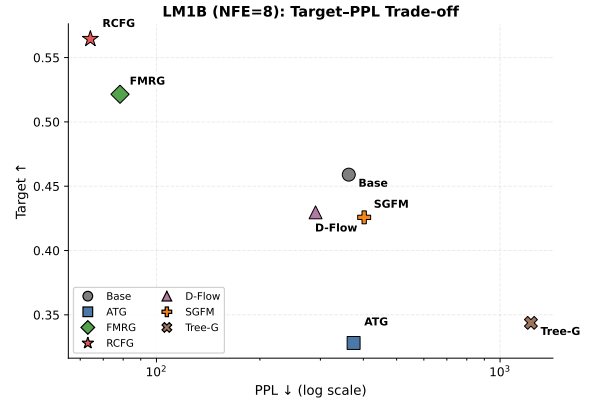


Figure 2: Target-PPL trade-off on LM1B under NFE=8. Each point reports the averaged result over four target-control tasks. RCFG lies in the high-Target and low-PPL region, achieving the best averaged Target and PPL among the compared NFE=8 guidance baselines.

rately by the matched LM1B ablation in Table 10.

Figure 3 shows that RCFG is best or tied-best on all five targets. More importantly, the gain reflects a better lexical-control trade-off. It does not obtain higher success by repeatedly inserting the target word.

We also evaluate four structured Text8 tasks at NFE = 8: multi-target, position-specified, forbidden-word, and rare-word exact-count control. All methods use the same checkpoint, 512 samples, random seed, and hyperparameter selection protocol. We report task success rate averaged over target groups; details are in Appendix H.5.

Table 8 provides a broader control check beyond single-target Exact-1. RCFG improves the average from 85.73% to 86.63%, although it is not best on

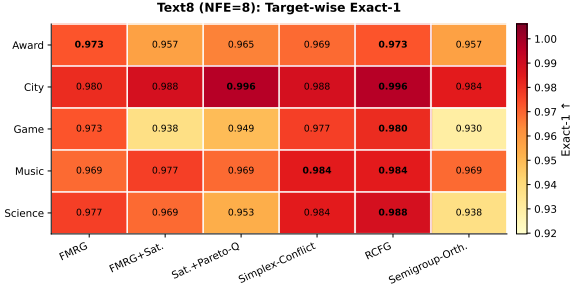


Figure 3: Target-wise Exact-1 results for validation-selected operating points on Text8 under NFE= 8.

Table 8: Extended Text8 control at NFE= 8. Multi-target requires all target words; Position requires a target at a specified position; Forbidden requires avoiding a target word; Rare exact-count requires a rare target word to appear exactly once.

Method	Multi-target $\uparrow$	Position $\uparrow$	Forbidden $\uparrow$	Rare exact-count $\uparrow$	Avg. $\uparrow$
FMRG	93.36	99.28	91.02	59.24	85.73
FMRG+Sat.	93.23	99.48	91.50	48.76	83.24
Sat.+Pareto-Q	88.28	96.29	90.62	45.18	80.09
Simplex-Conflict	94.40	99.61	<u>91.11</u>	60.22	86.34
RCFG	94.73	<u>99.54</u>	<u>91.11</u>	<u>61.13</u>	86.63
Semigroup-Orth.	93.82	99.48	91.02	61.78	<u>86.52</u>

every individual task. Overall, RCFG provides a modest aggregate improvement in lexical-control balance over FMRG.

5 Discussion

The experiments suggest that reliability calibration yields a more favorable aggregate control-likelihood operating point for terminal reward pullback. On LM1B, RCFG obtains higher average Reward and Target and lower NLL/PPL than FMRG at each tested NFE. These automatic likelihood-based metrics support the reported trade-off but do not by themselves establish human-perceived quality. Additional configuration-transfer, sensitivity, reliability-diagnostic, and matched qualitative analyses are reported in Appendix Tables 13, 14, 16, and 17.

Because RCFG uses additional gradient products, equal NFE does not imply equal wall-clock cost. Detailed per-NFE runtime measurements are provided in Appendix Table 15. Table 9 compares FMRG and RCFG at approximately matched sampling budgets. At the smaller matched budget, FMRG is slightly better in Reward and likelihood metrics while Target is essentially tied; at the larger budget, validation-selected RCFG is slightly better on all four metrics. Accordingly, we do not claim an unconditional RCFG advantage over higher-NFE FMRG at matched wall-clock cost.

Table 9: Compute-matched LM1B comparison. Times are measured wall-clock seconds per sample. Higher Reward and Target are better; lower NLL and PPL are better.

Budget	Method	Time	Reward $\uparrow$	Target $\uparrow$	NLL $\downarrow$	PPL $\downarrow$
≈ 0.032	FMRG, NFE= 4	0.0308	0.574	0.599	4.06	58.0
	RCFG, NFE= 2	0.0328	0.571	0.600	4.17	64.8
≈ 0.062	FMRG, NFE= 8	0.0615	0.592	0.617	3.92	50.4
	RCFG, NFE= 4	0.0631	0.597	0.625	3.87	47.9

The Text8 experiment provides a complementary lexical-control view without relying on an external attribute classifier. RCFG improves the trade-off between hitting the target word and avoiding over-insertion: it increases Exact-1 while keeping the target count close to one. Together, the LM1B and Text8 results provide complementary evidence for the calibration framework under classifier-based attribute control and lexical exact-count control.

6 Conclusion

We presented Reliability-Calibrated Flow Guidance (RCFG), a sampling-time reliability-calibration framework built around terminal reward pullback for few-step categorical flow-map generation. RCFG keeps terminal reward pullback as the main guidance primitive while calibrating the update magnitude and direction with transition-relative scaling, saturation-aware step control, local quality correction, and a simplex-aware residual branch based on a KL-regularized terminal mirror target. Our analyses highlight why few-step categorical guidance is sensitive to endpoint scale, soft-hard transfer discrepancy, hard-decoding boundaries, reward saturation, and reward-quality conflict. Experiments on LM1B attribute-guided generation and Text8 lexical control support more favorable aggregate control-likelihood operating points after calibration, with Text8 showing a modest aggregate improvement in lexical-control balance. Together with the scale-control and exact-count analyses, these results suggest that the gains come from state-dependent reliability calibration of terminal reward pullback, rather than fixed guidance-scale tuning or target over-insertion alone. Overall, RCFG is best viewed as a structured reliability-calibration refinement of terminal reward-pullback guidance for few-step categorical flow maps.

Limitations

This work has several limitations. First, our experiments focus on LM1B attribute-guided generation and Text8 target-word guidance. Although these settings cover verifier-based control and lexical exact-count control, they do not test long-form generation, dialogue, complex instructions, or large-scale pretrained language models.

Second, RCFG relies on differentiable terminal rewards or differentiable surrogates. It calibrates how the reward signal is used during few-step sampling, but cannot correct a reward model that is itself misaligned with human preferences or text quality.

Third, RCFG mitigates but does not fully close the gap between relaxed terminal representations and hard decoded text. As a result, improvements in the relaxed reward space may not always translate perfectly to decoded-token improvements. RCFG also incurs approximately $2\times$ wall-clock sampling overhead relative to FMRG in our current implementation due to the additional gradient computations, so equal-NFE improvements should not be interpreted as unconditional compute-matched advantages.

Finally, RCFG uses several calibration hyperparameters and depends on the quality of the underlying categorical flow map. Future work could explore more adaptive calibration rules, joint training, larger-scale models, human evaluation, and broader control objectives.

Ethical Considerations

RCFG is a method for controllable text generation. Although our experiments focus on standard attribute and lexical-control benchmarks, similar test-time guidance could be used to steer generated text toward harmful, biased, or manipulative attributes if paired with inappropriate reward models. RCFG also inherits biases and failure modes from the underlying flow-map language model and from the external reward or verifier used during guidance. We therefore view RCFG as a controlled-generation technique that should be used with task-appropriate reward models, safety filtering, and human evaluation before deployment in open-ended generation settings. We use publicly available datasets, pretrained models, and baseline implementations following their original licenses and terms of use. Our released code is intended for research purposes. Because open-domain corpora may contain real-world

names or potentially offensive text, we use them only for research evaluation and do not collect new user data or attempt to identify individuals.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under Grants 62572288 and 61702312, in part by the Key Research and Development Plan of Shandong Province under Grant 2019GGX101075, in part by the Natural Science Foundation of Shandong Province of China under Grant ZR2017BF019, and in part by the Taishan Scholar Project of Shandong Province under Grant ts20190924.

References

- Michael Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. 2025. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80.
- Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Universal guidance for diffusion models. In *International Conference on Learning Representations*.
- Heli Ben-Hamu, Omri Puny, Itai Gat, Brian Karrer, Uriel Singer, and Yaron Lipman. 2024. D-flow: Differentiating through flows for controlled generation. In *Proceedings of the 41st International Conference on Machine Learning*.
- Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. 2025a. Flow map matching with stochastic interpolants: A mathematical framework for consistency models. *Transactions on Machine Learning Research*.
- Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. 2025b. How to build a consistency model: Learning flow maps via self-distillation. In *Advances in Neural Information Processing Systems*.
- Justin Deschenaux and Caglar Gulcehre. 2025. Beyond autoregression: Fast LLMs via self-distillation through time. In *International Conference on Learning Representations*.
- Prafulla Dhariwal and Alexander Quinn Nichol. 2021. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*.
- Ruiqi Feng, Chenglei Yu, Wenhao Deng, Peiyan Hu, and Tailin Wu. 2025. On the guidance of flow matching. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 16993–17029. PMLR.

- Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. 2025. Mean flows for one-step generative modeling. In *Advances in Neural Information Processing Systems*.
- Yingqing Guo, Yukang Yang, Hui Yuan, and Mengdi Wang. 2025. Training-free guidance beyond differentiability: Scalable path steering with tree search in diffusion and flow models. In *Advances in Neural Information Processing Systems*.
- Jonathan Ho and Tim Salimans. 2021. Classifier-free diffusion guidance. In *NeurIPS Workshop on Deep Generative Models and Downstream Applications*.
- Jerry Y. Huang, Justin Lin, Sheel Shah, Kartik Nair, and Nicholas M. Boffi. 2026. How to guide your flow: Few-step alignment via flow map reward guidance. In *International Conference on Machine Learning*.
- Felix Koulischer, Florian Handke, Johannes Deleu, Thomas Demeester, and Luca Ambrogioni. 2025. Feedback guidance of diffusion models. In *Advances in Neural Information Processing Systems*.
- Chanhyuk Lee, Jaehoon Yoo, Manan Agarwal, Sheel Shah, Jerry Huang, Aditi Raghunathan, Seunghoon Hong, Nicholas M. Boffi, and Jinwoo Kim. 2026. Flow map language models: One-step language modeling via continuous denoising. *arXiv preprint arXiv:2602.16813*.
- Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. 2023. Flow matching for generative modeling. In *International Conference on Learning Representations*.
- Peter Potaptchik, Jason Yim, Adhi Saravanan, Peter Holderrieth, Eric Vanden-Eijnden, and Michael S. Albergo. 2026. *Discrete flow maps*. Preprint, arXiv:2604.09784.
- Daan Roos, Oscar Davis, Floor Eijkelboom, Michael Bronstein, Max Welling, Ismail Ilkan Ceylan, Luca Ambrogioni, and Jan-Willem van de Meent. 2026. Categorical flow maps. In *International Conference on Machine Learning*.
- Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. 2025. Align your flow: Scaling continuous-time flow map distillation. In *Advances in Neural Information Processing Systems*.
- Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander M. Rush, and Volodymyr Kuleshov. 2024. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems*.
- Subham Sekhar Sahoo, Justin Deschenaux, Aaron Gokaslan, Guanghan Wang, Justin Chiu, and Volodymyr Kuleshov. 2025. The diffusion duality. *arXiv preprint arXiv:2506.10892*.
- Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. 2023. Consistency models. In *International Conference on Machine Learning*.
- Zhengyan Wan, Yidong Ouyang, Liyan Xie, Fang Fang, Hongyuan Zha, and Guang Cheng. 2026. Discrete guidance matching: Exact guidance for discrete flow matching. In *International Conference on Learning Representations*.
- Luran Wang, Chaoran Cheng, Yizhen Liao, Yanru Qu, and Ge Liu. 2025. Training free guided flow matching with optimal control. In *International Conference on Learning Representations*.
- Zifan Wang, Alice Harting, Matthieu Barreau, Michael M. Zavlanos, and Karl H. Johansson. 2026. Source-guided flow matching. In *International Conference on Learning Representations*.
- Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. 2023. FreeDoM: Training-free energy-guided conditional diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

A Terminal Pullback and Few-Step Sensitivity

A.1 Terminal objective viewed from an intermediate state

Let $x_s \in \mathcal{X}$ denote the relaxed state at time $s \in [0, 1]$, and let $\Phi_{s \rightarrow t} : \mathcal{X} \rightarrow \mathcal{X}$ be a differentiable learned flow map. For compactness, we write $\Phi_s(x) := \Phi_{s \rightarrow 1}(x)$, $y_s := \Phi_s(x_s)$. Here y_s is the endpoint predicted from the intermediate state x_s by the remaining flow map.

Let $R : \mathcal{X} \rightarrow \mathbb{R}$ be a differentiable terminal reward. In the main method, R corresponds to the soft reward evaluated at the terminal relaxed token distribution. We use σ_Δ to denote a differentiable simplex parameterization, implemented by softmax normalization in our experiments. Thus, $R(y_s) = r(\sigma_\Delta(y_s))$. This notation highlights that the reward is evaluated on relaxed categorical probabilities rather than on hard token samples.

The terminal objective viewed from time s is defined as

$$\mathcal{J}_s(x_s) := R(\Phi_s(x_s)). \quad (20)$$

Although $R(\Phi_s(x_s)) = R(y_s)$ numerically, the form $R(\Phi_s(x_s))$ makes explicit that the terminal reward depends on the intermediate state x_s through the remaining flow map. Terminal guidance then aims to choose a perturbation δx_s at the intermediate state such that $R(\Phi_s(x_s + \delta x_s))$ is increased.

A.2 A basic terminal-pullback identity

We first record the basic pullback identity used to define the terminal reward direction. This identity is a direct consequence of the chain rule, and is used only to fix notation for the reward-pullback primitive in our method.

Assume that the composed objective $\mathcal{J}_s = R \circ \Phi_s$ is differentiable in a neighborhood of x_s , and has a locally Lipschitz gradient. Let $J_s := D_x \Phi_s(x_s)$ be the Jacobian of the terminal flow map at x_s . Then

$$\nabla_{x_s} \mathcal{J}_s(x_s) = J_s^\top \nabla R(y_s), \quad y_s = \Phi_s(x_s). \quad (21)$$

Indeed, by the chain rule,

$$\begin{aligned} \nabla_{x_s} \mathcal{J}_s(x_s) &= \nabla_{x_s} R(\Phi_s(x_s)) \\ &= [D_x \Phi_s(x_s)]^\top \nabla R(\Phi_s(x_s)) \\ &= J_s^\top \nabla R(y_s). \end{aligned}$$

Moreover, since $\mathcal{J}_s$ has locally Lipschitz gradient, a first-order Taylor expansion around x_s gives, for sufficiently small δx_s ,

$$\begin{aligned} R(\Phi_s(x_s + \delta x_s)) - R(\Phi_s(x_s)) &= \mathcal{J}_s(x_s + \delta x_s) - \mathcal{J}_s(x_s) \\ &= \langle \nabla_{x_s} \mathcal{J}_s(x_s), \delta x_s \rangle + O(\|\delta x_s\|_2^2) \\ &= \langle J_s^\top \nabla R(y_s), \delta x_s \rangle + O(\|\delta x_s\|_2^2). \end{aligned} \quad (22)$$

Equation (21) defines the terminal-pullback direction

$$g_{\text{TP}}(x_s) := \nabla_{x_s} R(\Phi_s(x_s)) = J_s^\top \nabla R(y_s). \quad (23)$$

When $g_{\text{TP}}(x_s) \neq 0$, its normalized version gives the steepest first-order ascent direction under an ℓ_2 -bounded infinitesimal perturbation. This is the same terminal lookahead principle used in flow-map reward guidance methods such as FMRG (Huang et al., 2026). In our method, it serves as the reward-pullback primitive that is later calibrated by saturation scaling, quality correction, and simplex-aware residual guidance.

A.3 Local guidance versus terminal guidance

A local guidance rule may instead use the gradient of a reward evaluated around the current state, such as $\nabla R(x_s)$. This direction only describes the instantaneous reward change at the current state. It does not generally describe how a perturbation at time s changes the terminal reward after the remaining flow map.

By Equation (21), local guidance and terminal guidance coincide only when

$$\nabla R(x_s) = J_s^\top \nabla R(y_s), \quad y_s = \Phi_s(x_s). \quad (24)$$

This condition is restrictive in few-step flow-map generation. A nontrivial flow map usually satisfies $y_s \neq x_s$, $J_s \neq I$. Therefore, a local reward gradient can be misaligned with the terminal effect of the same perturbation. Terminal pullback addresses this horizon mismatch by differentiating the terminal reward through the remaining flow map.

A.4 Few-step sensitivity from larger per-step updates

We next explain why this mismatch becomes more consequential in few-step generation. Let $\mathcal{J}_{s_k}(x) = R(\Phi_{s_k \rightarrow 1}(x))$ be the terminal objective at the k -th guidance step, and assume that each $\mathcal{J}_{s_k}$ is β -smooth. For a normalized direction g_k with $\|g_k\|_2 \leq 1$, a guided update $x_{s_k}^+ = x_{s_k} + \eta_k g_k$ satisfies

$$\begin{aligned} \mathcal{J}_{s_k}(x_{s_k}^+) - \mathcal{J}_{s_k}(x_{s_k}) &\geq \eta_k \langle \nabla \mathcal{J}_{s_k}(x_{s_k}), g_k \rangle - \frac{\beta}{2} \eta_k^2. \end{aligned} \quad (25)$$

This is the standard smoothness lower bound, with the quadratic term measuring the local approximation error caused by a finite update.

If the sampler uses K guidance steps and the total guidance budget is fixed, $\sum_{k=1}^K \eta_k = E$, then, under the uniform allocation $\eta_k = \frac{E}{K}$, the accumulated smoothness-controlled remainder satisfies

$$\begin{aligned} \sum_{k=1}^K \frac{\beta}{2} \eta_k^2 &= \frac{\beta}{2} \sum_{k=1}^K \left(\frac{E}{K} \right)^2 \\ &= \frac{\beta K}{2} \frac{E^2}{K^2} = \frac{\beta E^2}{2K}. \end{aligned} \quad (26)$$

Therefore, with a fixed total guidance budget, fewer guidance steps imply larger per-step updates and a larger upper bound on the accumulated local Taylor remainder.

This observation does not claim that few-step guidance is always worse, nor does it provide a global trajectory error bound. It only shows that local approximation errors become more influential when guidance is concentrated into a small number of flow-map evaluations. This motivates the reliability-calibrated design used in the main method: terminal pullback provides the horizon-aware reward direction, while saturation scaling,

quality correction, and simplex-aware residual guidance reduce the risk of overly aggressive few-step updates.

B Diagnostic Soft-Hard Transfer in Categorical Flow-Map Guidance

This appendix analyzes the transfer between the relaxed reward used for guidance and the reward or metric evaluated on decoded text. The purpose is diagnostic: we do not treat the soft-hard discrepancy as a directly optimized objective, nor do we claim to eliminate it. Instead, the analysis clarifies why soft-reward ascent alone does not automatically imply decoded-text reward improvement in categorical flow-map guidance.

B.1 Relaxed guidance reward and decoded-text evaluation

In discrete flow-map guidance, the guidance signal is computed on a relaxed terminal distribution, while the final output is evaluated after hard decoding. Let $x_s \in \Delta^{L \times V}$ be the relaxed token state at time s , and let the predicted terminal relaxed distribution be $p_1 = \sigma_\Delta(\Phi_{s \rightarrow 1}(x_s))$, where σ_Δ denotes a differentiable simplex normalization, implemented as softmax normalization in our experiments.

Let $\mathcal{D} : \Delta^{L \times V} \rightarrow \mathcal{Y}$ be the hard decoder that maps a relaxed terminal distribution to a decoded text sequence. In our implementation, $\mathcal{D}$ selects the highest-probability token at each position and then converts the token IDs into text using the tokenizer.

We distinguish the differentiable guidance reward from the decoded-text reward: $R_{\text{soft}} : \Delta^{L \times V} \rightarrow \mathbb{R}$, $R_{\text{hard}} : \mathcal{Y} \rightarrow \mathbb{R}$. Here R_{soft} is evaluated on the relaxed terminal distribution, whereas R_{hard} is evaluated on decoded text. The two quantities may be implemented by related models or by separate evaluators; in either case, their difference should be interpreted as a diagnostic discrepancy rather than as an objective directly available during sampling.

For any relaxed terminal distribution p , define

$$e(p) := R_{\text{soft}}(p) - R_{\text{hard}}(\mathcal{D}(p)). \quad (27)$$

The discrepancy $e(p)$ measures how much the relaxed guidance reward differs from the decoded-text reward at the same terminal prediction. It is introduced only to make the transfer issue explicit.

B.2 A diagnostic transfer decomposition

Consider a guidance update δx_s at time s . Define the terminal relaxed distributions before and after the update as

$$\begin{aligned} p_1 &= \sigma_\Delta(\Phi_{s \rightarrow 1}(x_s)), \\ p_1^+ &= \sigma_\Delta(\Phi_{s \rightarrow 1}(x_s + \delta x_s)). \end{aligned} \quad (28)$$

The relaxed reward change is

$$\Delta R_{\text{soft}} := R_{\text{soft}}(p_1^+) - R_{\text{soft}}(p_1), \quad (29)$$

whereas the decoded-text reward change is

$$\Delta R_{\text{hard}} := R_{\text{hard}}(\mathcal{D}(p_1^+)) - R_{\text{hard}}(\mathcal{D}(p_1)). \quad (30)$$

By the definition of $e(p)$, $R_{\text{hard}}(\mathcal{D}(p)) = R_{\text{soft}}(p) - e(p)$. Substituting this identity for $p = p_1^+$ and $p = p_1$ gives

$$\begin{aligned} \Delta R_{\text{hard}} &= [R_{\text{soft}}(p_1^+) - e(p_1^+)] \\ &\quad - [R_{\text{soft}}(p_1) - e(p_1)] \\ &= \Delta R_{\text{soft}} - \Delta e, \end{aligned} \quad (31)$$

where $\Delta e := e(p_1^+) - e(p_1)$.

Therefore,

$$\Delta R_{\text{hard}} > 0 \iff \Delta R_{\text{soft}} > \Delta e. \quad (32)$$

This condition shows that increasing the differentiable relaxed reward is not, by itself, sufficient to guarantee an improvement after hard decoding. A relaxed reward gain transfers to the decoded-text reward only when it exceeds the increase in the soft-hard discrepancy. This does not invalidate relaxed reward guidance; it only identifies the additional transfer term that raw soft-reward ascent does not control directly.

B.3 Scope of raw terminal pullback

The terminal-pullback direction for the relaxed reward is

$$g_{\text{TP}}(x_s) = \nabla_{x_s} R_{\text{soft}}(\sigma_\Delta(\Phi_{s \rightarrow 1}(x_s))). \quad (33)$$

For clarity, consider a raw infinitesimal update $x_s^+ = x_s + \eta g_{\text{TP}}$. Using a first-order expansion of the relaxed terminal objective around x_s , we obtain

$$\begin{aligned} \Delta R_{\text{soft}} &= R_{\text{soft}}(\sigma_\Delta(\Phi_{s \rightarrow 1}(x_s + \eta g_{\text{TP}}))) \\ &\quad - R_{\text{soft}}(\sigma_\Delta(\Phi_{s \rightarrow 1}(x_s))) \\ &= \eta \langle \nabla_{x_s} R_{\text{soft}}(\sigma_\Delta(\Phi_{s \rightarrow 1}(x_s))), g_{\text{TP}} \rangle \\ &\quad + O(\eta^2) \\ &= \eta \|g_{\text{TP}}\|_2^2 + O(\eta^2). \end{aligned} \quad (34)$$

Thus, for sufficiently small η and $g_{\text{TP}} \neq 0$, raw terminal pullback increases the relaxed terminal reward locally.

Combining Equation (34) with the decomposition in Equation (31) gives

$$\begin{aligned}\Delta R_{\text{hard}} &= \Delta R_{\text{soft}} - \Delta e \\ &= \eta \|g_{\text{TP}}\|_2^2 - \Delta e + O(\eta^2).\end{aligned}\quad (35)$$

Therefore, the local relaxed-reward improvement transfers to decoded-text reward only when

$$\eta \|g_{\text{TP}}\|_2^2 + O(\eta^2) > \Delta e. \quad (36)$$

This is a scope statement for raw terminal pullback: the direction is locally valid for the differentiable relaxed reward, but the decoded-text outcome also depends on how the update changes the soft-hard discrepancy.

B.4 Sources of transfer discrepancy

The discrepancy $e(p)$ can arise from structural differences between relaxed categorical guidance and decoded-text evaluation.

Representation mismatch. The relaxed reward is evaluated on a distributional input, whereas the decoded reward is evaluated on a token sequence. Even if both quantities target the same attribute, they operate on different representations.

Soft embedding mismatch. When the reward model is evaluated on relaxed text, the input is often formed by expected token embeddings. Let $W \in \mathbb{R}^{V \times d}$ be the token embedding table. At position ℓ , the relaxed embedding is $\bar{z}_\ell = \sum_{v=1}^V p_{\ell v} W_v$, whereas hard decoding uses $z_\ell^{\text{hard}} = W_{v_\ell^*}$, $v_\ell^* = \arg \max_v p_{\ell v}$. Their difference is

$$\bar{z}_\ell - z_\ell^{\text{hard}} = \sum_{v \neq v_\ell^*} p_{\ell v} (W_v - W_{v_\ell^*}). \quad (37)$$

This term can be non-negligible when the terminal distribution assigns substantial probability mass to several competing tokens.

Hard-decoding discontinuity. The hard decoder is discontinuous on argmax boundaries. A small change in the relaxed terminal distribution may leave the decoded sequence unchanged inside an argmax cell, or may abruptly change a decoded token when it crosses a decision boundary. The boundary-level behavior is analyzed separately in Appendix D.

B.5 Implication for reliability-calibrated guidance

The discrepancy $e(p)$ and its change Δe are generally unavailable during sampling, because they depend on the decoded sequence and the decoded-text evaluation. Therefore, RCFG does not attempt to estimate or optimize Δe directly.

Instead, this diagnostic analysis motivates reliability calibration around conditions where relaxed reward improvement is less predictive of decoded-text behavior. In the main method, terminal pullback remains the primary reward primitive. The additional calibration terms address several reliability factors that can make few-step categorical guidance unstable: saturation scaling limits unnecessary update magnitude when the target confidence is already high, reward-quality correction removes local first-order conflicts with a quality surrogate, and the simplex-aware residual branch injects categorical-geometry information without replacing the primary reward direction.

Thus, the method does not solve the soft-hard gap as a standalone objective. Rather, it uses the transfer analysis to motivate a sampling-time update that is more reliable for few-step categorical flow-map generation.

C Simplex Geometry of Categorical Flow-Map Guidance

This appendix summarizes the simplex geometry that is directly relevant to the simplex-aware branch of RCFG. In categorical flow-map generation, guidance is computed through relaxed vocabulary distributions. Therefore, terminal reward-pullback directions should be interpreted with respect to categorical simplex structure, rather than only as unconstrained Euclidean directions.

C.1 Terminal categorical predictions are simplex-valued

For each token position ℓ , a categorical distribution lies in the vocabulary simplex:

$$p_\ell \in \Delta^V := \left\{ p \in \mathbb{R}^V : p_v \geq 0, \sum_{v=1}^V p_v = 1 \right\}. \quad (38)$$

Thus a relaxed token sequence is represented as a product of vocabulary simplexes:

$$p = (p_1, \dots, p_L) \in \Delta^{L \times V}. \quad (39)$$

In RCFG, the terminal relaxed prediction used by the reward model is

$$p_1 = \sigma_{\Delta}(\Phi_{s \rightarrow 1}(x_s)), \quad (40)$$

where σ_{Δ} denotes a differentiable simplex normalization, implemented as softmax normalization in our experiments. Therefore, the reward gradient at the terminal endpoint is a gradient over a relaxed categorical distribution.

This differs from unconstrained continuous-space guidance. A categorical distribution must preserve normalization and non-negativity at each token position. These constraints motivate constructing terminal corrections through valid simplex distributions.

C.2 Euclidean directions and simplex feasibility

Consider a single token distribution $p \in \Delta^V$ and a finite Euclidean update $p^+ = p + \eta g$, $\eta > 0$. For p^+ to remain in the simplex, the update direction and step size must satisfy

$$\mathbf{1}^\top g = 0, \quad p_v + \eta g_v \geq 0, \quad \forall v \in \{1, \dots, V\}. \quad (41)$$

The first condition is the affine normalization constraint. It requires the first-order perturbation to lie in the tangent space

$$T\Delta^V = \left\{ \delta \in \mathbb{R}^V : \mathbf{1}^\top \delta = 0 \right\}. \quad (42)$$

A generic Euclidean direction does not automatically satisfy this condition. Its affine tangent component is

$$\Pi_T(g) = g - \frac{\mathbf{1}^\top g}{V} \mathbf{1}. \quad (43)$$

For the product simplex, the same operation is applied position-wise:

$$[\Pi_T^{\text{prod}}(g)]_\ell = g_\ell - \frac{\mathbf{1}^\top g_\ell}{V} \mathbf{1}. \quad (44)$$

However, tangent feasibility only accounts for the sum-to-one constraint. It does not by itself guarantee non-negativity for a finite step. If $g_v < 0$ for some coordinate v , then $p_v + \eta g_v \geq 0 \iff \eta \leq \frac{p_v}{-g_v}$. Thus, for a tangent direction g , the largest feasible positive step size is

$$\eta_{\max}(p, g) = \min_{v: g_v < 0} \frac{p_v}{-g_v}, \quad (45)$$

with the convention that the minimum is $+\infty$ if there is no negative component.

This shows two concrete limitations of a raw Euclidean direction in categorical coordinates. First, the direction may fail to preserve normalization. Second, even if it is tangent to the simplex, a finite step can violate non-negativity when it exceeds the boundary-limited radius in Equation (45).

C.3 Boundary sensitivity under few-step guidance

The non-negativity constraint becomes more restrictive near the boundary of the vocabulary simplex. Suppose $g \in T\Delta^V$ and there exists a coordinate v such that $g_v < 0$, $p_v \leq \rho$, $-g_v \geq \gamma > 0$. Then any feasible update along g must satisfy

$$\eta \leq \frac{p_v}{-g_v} \leq \frac{\rho}{\gamma}. \quad (46)$$

Thus, a single low-probability coordinate with a sufficiently strong negative guidance component can sharply limit the feasible step size.

This boundary sensitivity is relevant for few-step guidance. When the sampler uses only a small number of flow-map evaluations, each guidance update carries a larger terminal effect. Aggressive reward-pullback steps can therefore interact more strongly with low-probability coordinates and simplex boundaries.

C.4 Simplex-native terminal correction in RCFG

The observations above motivate the simplex-aware branch of RCFG. The method does not use the tangent projection in Equation (43) as its update rule. Instead, RCFG keeps terminal reward pullback as the primary signal and constructs an additional terminal correction through a valid simplex-valued target.

The sampling-time update in RCFG is

$$x_s^+ = x_s + \eta(t - s)a_{\text{sat}}(u_s)\|v_s\|_2 d_{\text{RCFG}}, \quad (47)$$

where

$$v_s = \frac{\Phi_{s \rightarrow t}(x_s) - x_s}{t - s} \quad (48)$$

is the base flow-map velocity. The factor $\|v_s\|_2$ makes the guidance magnitude relative to the natural scale of the current flow-map transition, and $a_{\text{sat}}(u_s)$ reduces the update magnitude when the target confidence is already saturated.

The final normalized direction is

$$d_{\text{RCFG}} = u(u(g_P) + \lambda_M u(g_{\text{gap}})), \quad (49)$$

where g_P is the primary reward-quality corrected direction and g_{gap} is the residual component derived from the simplex-aware correction branch.

The simplex-specific correction is constructed at the terminal relaxed distribution. Let

$$r_p = \nabla_{p_1} r(p_1) \quad (50)$$

be the reward gradient in terminal categorical coordinates. RCFG forms the mirror target

$$q = \sigma_{\Delta} (\log(p_1 + \epsilon) + \eta_m [r_p - \text{mean}(r_p)]) . \quad (51)$$

Because q is obtained through σ_{Δ} , it remains a valid vocabulary distribution. The correction is then obtained by pulling the terminal prediction toward this simplex-native target:

$$g_M = -\nabla_{x_s} D_{\text{KL}}(q \| \sigma_{\Delta}(\Phi_{s \rightarrow 1}(x_s))) . \quad (52)$$

This construction uses simplex geometry at the terminal distribution level. It does not replace the main reward-quality direction. Instead, the simplex-aware correction is used as a residual component:

$$g_{\text{gap}} = g_C - \frac{\langle g_C, g_P \rangle}{\|g_P\|_2^2 + \epsilon} g_P . \quad (53)$$

Thus, the simplex-aware branch injects categorical-geometry information while preserving the primary terminal reward-quality guidance direction.

C.5 Takeaway

The simplex geometry above explains why raw Euclidean reward pullback is structurally incomplete for categorical terminal predictions. A Euclidean direction may fail the affine normalization condition, and even a tangent direction can violate non-negativity under finite steps near the simplex boundary.

RCFG uses this geometry in a targeted way. It does not turn the sampler into a projected-gradient method. Instead, it constructs a simplex-valid terminal mirror target, pulls the corresponding correction back through the remaining flow map, and injects this correction only as an orthogonal residual to the primary reward-quality direction. Together with transition-relative magnitude calibration, this provides the geometric motivation for the simplex-aware part of RCFG.

D Hard Decoding Boundaries in Categorical Flow-Map Guidance

This appendix analyzes the boundary structure induced by hard decoding. In categorical flow-map

generation, guidance is computed on relaxed terminal distributions, while the final text is obtained by a deterministic decoder. The decoder is locally constant inside argmax regions and changes only when the relaxed distribution crosses an argmax boundary. This explains why a smooth change in relaxed token probabilities may have no immediate effect on decoded text, while a small change near a boundary may abruptly change a decoded token.

D.1 Argmax cells on the vocabulary simplex

For a single token position, let $p \in \Delta^V$ be a relaxed vocabulary distribution. The hard decoder selects the largest-probability token:

$$\mathcal{D}(p) = \arg \max_{v \in \{1, \dots, V\}} p_v . \quad (54)$$

We assume a fixed deterministic tie-breaking rule when ties occur. A tie boundary is a point where two or more tokens have the same largest probability; for example, $p_i = p_j \geq p_k$ for all other tokens k . Away from such boundaries, the selected token is unique.

For each token i , define the argmax cell

$$\mathcal{C}_i = \{p \in \Delta^V : p_i \geq p_j, \forall j \neq i\} . \quad (55)$$

This is the set of all relaxed token distributions that decode to token i . Inside the relative interior of $\mathcal{C}_i$, small changes in the probabilities do not change the decoded token.

The boundary between two competing tokens i and j is contained in

$$\mathcal{B}_{ij} = \{p \in \Delta^V : p_i = p_j \geq p_k, \forall k \notin \{i, j\}\} . \quad (56)$$

This is the region where tokens i and j are tied for the largest probability. Crossing such a boundary can change the decoded token from i to j , or from j to i , depending on the direction of the perturbation.

For a sequence of length L , the decoder acts position-wise:

$$\mathcal{D}(p_1, \dots, p_L) = (\mathcal{D}(p_1), \dots, \mathcal{D}(p_L)) , \quad (57)$$

$$p_{\ell} \in \Delta^V .$$

Thus, if a decoded sequence is $y = (i_1, \dots, i_L)$, the corresponding decoding region is the product cell $\mathcal{C}_{i_1} \times \dots \times \mathcal{C}_{i_L}$. This product form means that each token position contributes its own argmax cell, and the whole decoded sequence is fixed only when all positions stay in their corresponding cells.

Therefore, each token-position simplex is partitioned into argmax regions. Within the same region, small probability changes do not change the decoded token; a token change can occur only after crossing a region boundary.

D.2 Margin-based decoding stability

Assume that $p \in \Delta^V$ has a unique top-1 token. Let $i = \arg \max_v p_v$, $j = \arg \max_{v \neq i} p_v$, and define the top-two margin

$$m(p) = p_i - p_j. \quad (58)$$

When $m(p) > 0$, the decoded token is locally stable. The following elementary bound gives a sufficient radius for preserving the decoded token.

Lemma 1 (Argmax margin stability). *Let $p \in \Delta^V$ have a unique top-1 token i and top-two margin $m(p) > 0$. If a perturbation $\delta \in \mathbb{R}^V$ satisfies*

$$\|\delta\|_\infty < \frac{m(p)}{2}, \quad (59)$$

then the decoded token is preserved:

$$\arg \max_v (p_v + \delta_v) = \arg \max_v p_v. \quad (60)$$

Proof. Let i be the top-1 token and j be the top-2 token. For any $v \neq i$,

$$\begin{aligned} (p_i + \delta_i) - (p_v + \delta_v) &= p_i - p_v + \delta_i - \delta_v \\ &\geq p_i - p_j - |\delta_i| - |\delta_v| \\ &\geq m(p) - 2\|\delta\|_\infty. \end{aligned} \quad (61)$$

If $\|\delta\|_\infty < m(p)/2$, then the right-hand side is positive for every $v \neq i$, so i remains the unique argmax. $\square$

This lemma shows that if the top-1 token is much more likely than the top-2 token, a small perturbation will not change the decoded token; if the two are close, a small perturbation may change the decoded token.

D.3 Sequence-level boundary crossing

For a sequence distribution $p = (p_1, \dots, p_L) \in \Delta^{L \times V}$, assume that every position has a unique top-1 token. Let $m_\ell(p) = p_{\ell,(1)} - p_{\ell,(2)}$ be the top-two margin at position ℓ , where $p_{\ell,(1)}$ and $p_{\ell,(2)}$ denote the largest and second-largest token probabilities.

Applying Lemma 1 position-wise, the decoded sequence is unchanged whenever

$$\|\delta_\ell\|_\infty < \frac{m_\ell(p)}{2}, \quad \forall \ell \in \{1, \dots, L\}. \quad (62)$$

Equivalently,

$$\mathcal{D}(p + \delta) = \mathcal{D}(p). \quad (63)$$

Define the set of positions that are not guaranteed to preserve their decoded tokens:

$$\mathcal{U}(p, \delta) = \left\{ \ell : \|\delta_\ell\|_\infty \geq \frac{m_\ell(p)}{2} \right\}. \quad (64)$$

By the margin lemma, every position outside $\mathcal{U}(p, \delta)$ is guaranteed to keep its decoded token. Therefore, token changes can occur only inside $\mathcal{U}(p, \delta)$. Since the Hamming distance counts the number of positions whose decoded tokens change, we have

$$d_H(\mathcal{D}(p + \delta), \mathcal{D}(p)) \leq |\mathcal{U}(p, \delta)|. \quad (65)$$

This is a conservative bound: positions in $\mathcal{U}(p, \delta)$ may or may not change, but positions outside this set cannot change under the stated margin condition.

D.4 Piecewise-constant decoded rewards

Let R_{hard} be a reward or metric evaluated on decoded text. The corresponding function over relaxed distributions is

$$\mathcal{R}_{\text{hard}}(p) = R_{\text{hard}}(\mathcal{D}(p)). \quad (66)$$

Because $\mathcal{D}$ is constant inside each argmax cell, $\mathcal{R}_{\text{hard}}$ is also constant inside each region where the decoded sequence is unchanged.

Thus, if a perturbation satisfies $\mathcal{D}(p + \delta) = \mathcal{D}(p)$, then

$$R_{\text{hard}}(\mathcal{D}(p + \delta)) - R_{\text{hard}}(\mathcal{D}(p)) = 0. \quad (67)$$

This does not mean that the relaxed update is useless. It means that changes in the relaxed distribution and changes in the decoded sequence occur on different geometries. A relaxed reward can change smoothly inside an argmax cell, while the decoded text remains unchanged. Once a perturbation crosses an argmax boundary, the decoded token may change abruptly.

D.5 Effect of magnitude calibration on boundary crossing

The margin bound also explains the role of controlling update magnitude. For a normalized direction $\bar{g}$, consider a scaled perturbation $\delta(a) = \eta a \bar{g}$, $a \in [0, 1]$. At position ℓ ,

$$\|\delta_\ell(a)\|_\infty = \eta a \|\bar{g}_\ell\|_\infty. \quad (68)$$

This equality follows from positive homogeneity of the ℓ_∞ norm, since $\delta_\ell(a) = \eta a \bar{g}_\ell$ and $\eta a \geq 0$. Therefore, the decoded token at position ℓ is guaranteed to remain unchanged if

$$\eta a \|\bar{g}_\ell\|_\infty < \frac{m_\ell(p)}{2}. \quad (69)$$

Using the unstable set in Equation (64), we obtain

$$\begin{aligned} d_H(\mathcal{D}(p + \eta a \bar{g}), \mathcal{D}(p)) \\ \leq |\{\ell : m_\ell(p) \leq 2\eta a \|\bar{g}_\ell\|_\infty\}|. \end{aligned} \quad (70)$$

Reducing the magnitude factor a can shrink the set of positions that may cross argmax boundaries. This is a boundary-stability statement, not a claim that smaller updates always improve decoded reward. If the update is too small, the decoded sequence may remain unchanged.

D.6 Connection to RCFG

The hard-decoding boundary analysis mainly supports the magnitude-calibration aspect of RCFG. It shows that decoded tokens are locally stable inside argmax cells, while positions with small top-two margins can change under small relaxed perturbations. Therefore, overly large guidance steps can cause uncontrolled boundary crossings, whereas overly small steps may leave the decoded sequence unchanged.

RCFG does not directly optimize margins or solve a hard-decoding boundary problem. Instead, it uses the magnitude-calibrated update in Equation (5), where the factor $\|v_s\|_2$ makes the guidance magnitude relative to the current base flow-map transition and $a_{\text{sat}}(u_s)$ reduces unnecessary update magnitude when the target confidence is already high.

The hard-decoding boundary structure explains why such magnitude calibration is needed: if the relaxed perturbation is too large, it may cause uncontrolled token changes by crossing argmax boundaries; if it is too small, the decoded sequence may remain unchanged. Thus, this boundary analysis directly motivates calibrating the guidance step size, rather than using an update magnitude determined only by the raw reward-gradient scale.

E Reliability-Calibrated Flow Guidance Update

This appendix instantiates the reliability-calibrated update used by RCFG. The purpose is not to introduce a separate objective, but to make explicit how

the sampling-time correction in the main method combines three reliability controls: transition-relative magnitude calibration, local reward-quality conflict correction, and simplex-aware residual correction.

At time s , let $x_s \in \Delta^{L \times V}$ be the current relaxed categorical state. The terminal relaxed token distribution is

$$p_1 = \sigma_\Delta(\Phi_{s \rightarrow 1}(x_s)), \quad (71)$$

where σ_Δ is the differentiable simplex normalization, implemented as softmax normalization in our experiments. The differentiable guidance reward is the target-class log-probability evaluated on the relaxed endpoint:

$$r(p_1) = \log P_\psi(c | p_1). \quad (72)$$

The reward-pullback direction is

$$g_R = \nabla_{x_s} r(\sigma_\Delta(\Phi_{s \rightarrow 1}(x_s))). \quad (73)$$

For any tensor z , define the per-sample normalized direction

$$u(z) = \frac{z}{\max(\|z\|_2, \epsilon)}. \quad (74)$$

Thus $\|u(z)\|_2 \leq 1$.

E.1 Sampling-time calibrated update

RCFG applies a state correction before the next base flow-map transition. The guided state is

$$x_s^+ = x_s + \gamma_s a_{\text{sat}}(u_s) \|v_s\|_2 d_{\text{RCFG}}, \quad (75)$$

where

$$v_s = \frac{\Phi_{s \rightarrow t}(x_s) - x_s}{t - s} \quad (76)$$

is the base flow-map velocity. With the constant guidance schedule used in our experiments,

$$\gamma_s = \eta(t - s). \quad (77)$$

The next state is then obtained by applying the base flow map:

$$x_t = \Phi_{s \rightarrow t}(x_s^+). \quad (78)$$

This form makes the guidance magnitude relative to the base flow-map displacement. Since the update is scaled by $\|v_s\|_2$, the guidance step is kept commensurate with how far the current flow-map transition naturally moves, instead of being determined by an absolute gradient norm independent of the current step.

E.2 Saturation-calibrated step scale

Let $u_s = P_\psi(c \mid p_1) \in [0, 1]$ be the target confidence predicted by the reward model at the relaxed endpoint. The soft guidance reward in Equation (72) uses the target log-probability, while the saturation scale is controlled by the corresponding probability u_s .

The remaining target-confidence headroom is

$$h_s = 1 - u_s. \quad (79)$$

For any updated confidence $u_s^+ \in [0, 1]$,

$$u_s^+ - u_s \leq 1 - u_s = h_s. \quad (80)$$

Thus, when u_s is already close to one, the remaining confidence gain for the same target objective is small. We use the state-dependent scale

$$a_{\text{sat}}(u_s) = a_{\min} + (1 - a_{\min})(1 - u_s)^\rho, \quad (81)$$

where $a_{\min} \in [0, 1]$ and $\rho > 0$. It satisfies $a_{\text{sat}}(0) = 1, a_{\text{sat}}(1) = a_{\min}$, for $u \in [0, 1]$,

$$\frac{da_{\text{sat}}}{du} = -\rho(1 - a_{\min})(1 - u)^{\rho-1} \leq 0. \quad (82)$$

Therefore, the scale is large when the target is not yet satisfied and becomes conservative when the target confidence is saturated.

Since $\|d_{\text{RCFG}}\|_2 \leq 1$, the guided displacement satisfies

$$\begin{aligned} \|x_s^+ - x_s\|_2 &\leq \gamma_s a_{\text{sat}}(u_s) \|v_s\|_2 \\ &\leq \gamma_s \|v_s\|_2. \end{aligned} \quad (83)$$

As $u_s \rightarrow 1$, the scale approaches $a_{\min}$ and saturated states receive the more conservative bound

$$\|x_s^+ - x_s\|_2 \leq \gamma_s a_{\min} \|v_s\|_2. \quad (84)$$

This is a step-magnitude calibration property. It does not imply a global guarantee on decoded reward improvement.

E.3 Local reward-quality conflict correction

The quality surrogate used by RCFG is

$$Q(p_1) = \alpha_H H(p_1) - \alpha_{\text{KL}} D_{\text{KL}}(p_1 \| p_1^{\text{base}}), \quad (85)$$

where $H(p_1)$ is the token entropy and p_1^{base} is the detached base endpoint distribution. Its pullback direction is

$$g_Q = \nabla_{x_s} Q(\sigma_\Delta(\Phi_{s \rightarrow 1}(x_s))). \quad (86)$$

A local reward-quality conflict occurs when

$$\langle g_R, g_Q \rangle < 0. \quad (87)$$

To see this, define the terminal reward and quality objectives viewed from the current state as

$$\begin{aligned} \mathcal{J}_R(x_s) &= r(\sigma_\Delta(\Phi_{s \rightarrow 1}(x_s))), \\ \mathcal{J}_Q(x_s) &= Q(\sigma_\Delta(\Phi_{s \rightarrow 1}(x_s))). \end{aligned} \quad (88)$$

Then $g_R = \nabla_{x_s} \mathcal{J}_R(x_s)$, $g_Q = \nabla_{x_s} \mathcal{J}_Q(x_s)$. For a small raw reward update $x_s^+ = x_s + \eta g_R$, Taylor expansion gives

$$\mathcal{J}_Q(x_s^+) - \mathcal{J}_Q(x_s) = \eta \langle g_Q, g_R \rangle + O(\eta^2). \quad (89)$$

Thus, when $\langle g_R, g_Q \rangle < 0$, the raw reward direction decreases the quality surrogate to first order.

RCFG therefore removes the component of g_R that is opposed to g_Q :

$$g_R^{\text{oq}} = \begin{cases} g_R - \frac{\langle g_R, g_Q \rangle}{\max(\|g_Q\|_2^2, \epsilon)} g_Q, & \langle g_R, g_Q \rangle < 0, \\ g_R, & \langle g_R, g_Q \rangle \geq 0. \end{cases} \quad (90)$$

When $\langle g_R, g_Q \rangle < 0$, let $M_Q = \max(\|g_Q\|_2^2, \epsilon)$. Then

$$\langle g_R^{\text{oq}}, g_Q \rangle = \langle g_R, g_Q \rangle \left(1 - \frac{\|g_Q\|_2^2}{M_Q}\right). \quad (91)$$

Thus, when the quality direction is non-degenerate ($\|g_Q\|_2^2 \geq \epsilon$), the conflicting first-order quality component is removed. When the quality direction is numerically small, the stabilizer keeps the correction bounded.

The primary quality-corrected direction is

$$g_P = u(g_R^{\text{oq}}) + \lambda_Q u(g_Q). \quad (92)$$

The first term keeps reward pullback as the primary component after conflict removal, while the second term explicitly restores a quality-improving component. This is a local first-order correction, not a global multi-objective optimality guarantee.

E.4 Simplex-mirror residual correction

The previous correction operates on the pullback directions at the current state. RCFG further constructs a simplex-native residual correction at the terminal relaxed distribution. Let

$$r_p = \nabla_{p_1} r(p_1) \quad (93)$$

be the terminal reward gradient in simplex coordinates. We form the centered and clipped reward direction

$$\tilde{r}_p = \text{clip}_{[-c_{\max}, c_{\max}]}(\text{sg}[r_p - \text{mean}(r_p)]), \quad (94)$$

where $\text{sg}(\cdot)$ denotes stop-gradient and $\text{clip}(\cdot)$ denotes element-wise clipping. The corresponding mirror target is

$$q = \sigma_{\Delta}(\log(\text{sg}(p_1) + \epsilon) + \eta_m \tilde{r}_p). \quad (95)$$

Because q is produced by simplex normalization, it remains a valid terminal token distribution.

The mirror correction pulls the predicted endpoint toward this simplex-native target:

$$g_M = -\nabla_{x_s} D_{\text{KL}}(\text{sg}(q) \| p_1). \quad (96)$$

To avoid directly opposing reward pullback, the implementation checks whether g_M conflicts with g_R . If $\cos(g_M, g_R) < 0$, the component of g_M parallel to g_R is removed:

$$g_M^{\perp R} = g_M - \frac{\langle g_M, g_R \rangle}{\max(\|g_R\|_2^2, \epsilon)} g_R. \quad (97)$$

The selected mirror direction is

$$\tilde{g}_M = \begin{cases} g_M^{\perp R}, & \cos(g_M, g_R) < 0, \\ g_M, & \cos(g_M, g_R) \geq 0. \end{cases} \quad (98)$$

The confidence-gated correction direction is

$$g_C = u(g_R) + \beta_s u(\tilde{g}_M), \quad (99)$$

$$\beta_s = \lambda_C \text{sigmoid}\left(\frac{u_s - \tau}{\kappa}\right).$$

The gate activates the simplex-mirror correction more strongly when the target confidence is above the threshold τ , where additional raw reward ascent is more likely to be saturated.

E.5 Orthogonal residual composition

The final RCFG direction combines the primary quality-corrected direction g_P and the simplex-mirror correction g_C . Rather than adding g_C directly, RCFG keeps only the component of g_C orthogonal to g_P :

$$g_{\text{gap}} = g_C - \frac{\langle g_C, g_P \rangle}{\max(\|g_P\|_2^2, \epsilon)} g_P. \quad (100)$$

The unnormalized calibrated direction is

$$h_{\text{RCFG}} = u(g_P) + \lambda_M u(g_{\text{gap}}). \quad (101)$$

The normalized direction used in the state update is

$$d_{\text{RCFG}} = u(h_{\text{RCFG}}). \quad (102)$$

The residual g_{gap} injects simplex-aware information while avoiding direct cancellation of the primary reward-quality direction. In our implementation, λ_C and λ_M are controlled by the same mixing coefficient unless otherwise specified.

E.6 Local soft-objective sanity check

The calibration terms should not be interpreted as a guarantee of decoded reward improvement. Nevertheless, it is useful to record a local condition under which the implemented direction still increases the terminal soft objective.

Let $\rho_s = \gamma_s a_{\text{sat}}(u_s) \|v_s\|_2$, $x_s^+ = x_s + \rho_s d_{\text{RCFG}}$. Let

$$\mathcal{J}_s(x_s) = r(\sigma_{\Delta}(\Phi_{s \rightarrow 1}(x_s))) \quad (103)$$

be the terminal soft reward objective. If $\mathcal{J}_s$ is α -smooth around x_s , then

$$\mathcal{J}_s(x_s^+) - \mathcal{J}_s(x_s) \geq \rho_s \langle g_R, d_{\text{RCFG}} \rangle - \frac{\alpha}{2} \rho_s^2. \quad (104)$$

Therefore, when $\langle g_R, d_{\text{RCFG}} \rangle > 0$ and ρ_s is sufficiently small, the update increases the terminal soft reward locally. This result shows that the calibration terms can preserve terminal soft-reward ascent when the final direction remains aligned with g_R and the calibrated step is sufficiently small. Decoded-text reward is outside this local differentiable statement: its transfer from the relaxed objective is governed by the diagnostic discrepancy in Appendix B and by the hard-decoding boundary behavior analyzed in Appendix D.

E.7 Relation to fixed-scale reward pullback

A fixed-scale reward-pullback baseline uses

$$x_s^+ = x_s + \gamma_s \alpha \|v_s\|_2 u(g_R), \quad \alpha \in [0, 1]. \quad (105)$$

In contrast, RCFG uses

$$x_s^+ = x_s + \gamma_s a_{\text{sat}}(u_s) \|v_s\|_2 d_{\text{RCFG}}, \quad (106)$$

where both the scale and the direction depend on the current state:

$$a_{\text{sat}}(u_s) = a_{\min} + (1 - a_{\min})(1 - u_s)^{\rho}, \quad (107)$$

$$d_{\text{RCFG}} = u(u(g_P) + \lambda_M u(g_{\text{gap}})).$$

Algorithm 1 Reliability-Calibrated Flow Guidance

Require: Current relaxed state x_s , flow map Φ , target reward model P_ψ , guidance scale η

- 1: Predict endpoint $p_1 = \sigma_\Delta(\Phi_{s \rightarrow 1}(x_s))$
 - 2: Compute terminal reward direction g_R by Eq. (4)
 - 3: Compute target confidence $u_s = P_\psi(c \mid p_1)$ and scale $a_{\text{sat}}(u_s)$
 - 4: Compute quality direction g_Q and primary direction g_P
 - 5: Compute simplex-aware correction g_C and residual g_{res}
 - 6: Form d_{RCFG} by Eq. (18)
 - 7: Compute base velocity $v_s = (\Phi_{s \rightarrow t}(x_s) - x_s)/(t - s)$
 - 8: Update relaxed state $x_s^+ = x_s + \eta(t - s)a_{\text{sat}}(u_s)\|v_s\|_2 d_{\text{RCFG}}$
 - 9: Advance $x_t = \Phi_{s \rightarrow t}(x_s^+)$ **return** x_t
-

The scale is state-dependent because it changes with the terminal target confidence. If two states satisfy $u^{(1)} < u^{(2)}$, then, for $a_{\min} < 1$ and $\rho > 0$,

$$a_{\text{sat}}(u^{(1)}) > a_{\text{sat}}(u^{(2)}). \quad (108)$$

Thus, the calibrated update keeps stronger guidance for less satisfied states and becomes more conservative for saturated states. In addition, the direction is modified by local reward-quality conflict correction and simplex-mirror residual composition. This distinction is useful when interpreting fixed-scale or component-removal ablations.

E.8 Algorithmic Description of RCFG

Algorithm 1 summarizes the sampling-time implementation of RCFG. At each step, the method first predicts the terminal relaxed categorical endpoint, computes the reliability-calibrated guidance direction, applies the calibrated relaxed-state update, and then advances the state using the original flow map. The base flow map is kept fixed throughout sampling.

F Categorical Simplex Structure of Flow-Map Guidance

This appendix summarizes the categorical structures that make few-step flow-map guidance for text different from unconstrained continuous-space guidance. We focus on three interfaces: the relaxed categorical endpoint used by the reward model, the simplex-native mirror target used by RCFG, and

the flow-map consistency issue that becomes more visible under large few-step updates. Detailed analyses of simplex feasibility, soft-hard transfer, and hard decoding boundaries are given in Appendix C, Appendix B, and Appendix D, respectively.

F.1 Categorical endpoint and reward interface

In categorical flow-map language generation, the terminal prediction is interpreted as a sequence of vocabulary distributions:

$$p_1 = \sigma_\Delta(\Phi_{s \rightarrow 1}(x_s)) \in \Delta^{L \times V}, \quad p_{1,\ell} \in \Delta^V. \quad (109)$$

Here σ_Δ denotes a differentiable simplex normalization, implemented as softmax normalization in our experiments. The guidance reward is computed on this relaxed endpoint:

$$r(p_1) = \log P_\psi(c \mid p_1). \quad (110)$$

This setting introduces three categorical interfaces:

$$\begin{aligned} \text{simplex endpoint:} & \quad p_{1,\ell} \in \Delta^V, \\ \text{relaxed reward input:} & \quad p_1 \mapsto r(p_1), \\ \text{hard decoding:} & \quad p_1 \mapsto \mathcal{D}(p_1). \end{aligned} \quad (111)$$

The first interface motivates simplex-aware terminal correction, the second explains why the reward can be differentiated during guidance, and the third explains why the relaxed endpoint must eventually be interpreted through hard tokens.

F.2 Soft embedding interface

A differentiable reward model for relaxed text often receives expected token embeddings. Let $W \in \mathbb{R}^{V \times d}$ be the token embedding table. At position ℓ , the relaxed embedding is

$$\bar{z}_\ell = \sum_{v=1}^V p_{\ell v} W_v, \quad (112)$$

whereas the embedding of the hard decoded token is

$$z_\ell^{\text{hard}} = W_{v_\ell^*}, \quad v_\ell^* = \arg \max_v p_{\ell v}. \quad (113)$$

Their difference is

$$\bar{z}_\ell - z_\ell^{\text{hard}} = \sum_{v \neq v_\ell^*} p_{\ell v} (W_v - W_{v_\ell^*}). \quad (114)$$

Thus, the relaxed reward model and the hard decoded text can receive different input representations even when they use the same embedding table. When the terminal distribution is close to one-hot, the relaxed embedding is close to the hard token embedding. When probability mass is spread over several competing tokens, the two inputs can differ noticeably. Therefore, differences between the relaxed reward and the hard decoded evaluation are natural in categorical text guidance.

F.3 Mirror target as simplex reward ascent

The categorical simplex also motivates the mirror-style terminal target used in RCFG. Let

$$r_p = \nabla_{p_1} r(p_1) \quad (115)$$

be the reward gradient at the terminal relaxed distribution. Ignoring stop-gradient, clipping, and numerical stabilizers, the ideal mirror target has the form

$$q = \sigma_{\Delta} (\log(p_1 + \epsilon) + \eta_m r_p). \quad (116)$$

This construction can be understood as a first-order reward-ascent step on the simplex. Around the current terminal distribution p_1 , the reward is locally linearized by $\langle q', r_p \rangle$, where $r_p = \nabla_{p_1} r(p_1)$. To avoid moving the terminal distribution too far from the current endpoint, we penalize the KL deviation from p_1 . This gives the KL-regularized simplex update

$$q = \arg \max_{q' \in \Delta^{L \times V}} \left\{ \langle q', r_p \rangle - \frac{1}{\eta_m} D_{\text{KL}}(q' \| p_1) \right\}. \quad (117)$$

Since both the linear reward term and the KL penalty decompose over token positions, it suffices to consider one position. Dropping the position index, the update solves

$$\max_{q' \in \Delta^V} \left\{ \sum_{v=1}^V q'_v r_{p,v} - \frac{1}{\eta_m} \sum_{v=1}^V q'_v \log \frac{q'_v}{p_v} \right\}. \quad (118)$$

Using a Lagrange multiplier for the constraint $\sum_v q'_v = 1$, the stationary condition is

$$r_{p,v} - \frac{1}{\eta_m} \left(\log \frac{q'_v}{p_v} + 1 \right) + \lambda = 0. \quad (119)$$

Rearranging gives

$$q'_v \propto p_v \exp(\eta_m r_{p,v}). \quad (120)$$

After normalization over the vocabulary, this yields

$$q' = \sigma_{\Delta} (\log p + \eta_m r_p). \quad (121)$$

In the implementation, this target is stabilized by stop-gradient, mean-centering, gradient clipping, and an ϵ term inside the logarithm:

$$q = \sigma_{\Delta} (\log(p_1 + \epsilon) + \eta_m [r_p - \text{mean}(r_p)]). \quad (122)$$

These stabilizers do not change the main interpretation: q is a valid terminal categorical distribution that represents a simplex-native reward ascent target.

F.4 Flow-map consistency under guidance

A learned flow map is expected to satisfy an approximate consistency relation:

$$\Phi_{t \rightarrow 1}(\Phi_{s \rightarrow t}(x_s)) \approx \Phi_{s \rightarrow 1}(x_s). \quad (123)$$

This relation matters in few-step generation because the sampler uses a small number of large transitions. A guidance perturbation can move the state away from the region where the learned flow maps compose accurately.

Define the local flow-map consistency error

$$\mathcal{E}_{s,t}(x) = \|\Phi_{t \rightarrow 1}(\Phi_{s \rightarrow t}(x)) - \Phi_{s \rightarrow 1}(x)\|_2. \quad (124)$$

Let $A(x) = \Phi_{t \rightarrow 1}(\Phi_{s \rightarrow t}(x))$, $B(x) = \Phi_{s \rightarrow 1}(x)$. If A and B are locally Lipschitz around x_s with constants L_A and L_B , then for a perturbation δ ,

$$\begin{aligned} \mathcal{E}_{s,t}(x_s + \delta) &= \|A(x_s + \delta) - B(x_s + \delta)\|_2 \\ &\leq \mathcal{E}_{s,t}(x_s) + (L_A + L_B)\|\delta\|_2. \end{aligned} \quad (125)$$

This bound does not claim that every guidance perturbation harms flow-map consistency. It only states that large off-trajectory perturbations can enlarge the local discrepancy between the composed prediction and the direct prediction.

When the perturbation is the guidance step δ_s , the same bound gives

$$\mathcal{E}_{s,t}(x_s + \delta_s) \leq \mathcal{E}_{s,t}(x_s) + (L_A + L_B)\|\delta_s\|_2. \quad (126)$$

Thus, this consistency view supports controlling the guidance displacement in few-step sampling: each transition spans a larger interval, and an overly large state correction can have a stronger effect on subsequent flow-map predictions.

F.5 Takeaway

The categorical structure above identifies two roles that are important for few-step flow-map guidance in text. First, the terminal endpoint is a vocabulary simplex distribution, so the correction should be expressed through a valid categorical target. Second, few-step sampling relies on large flow-map transitions, so the strength of the correction should be tied to the transition scale.

RCFG follows these two roles directly. The mirror target provides a simplex-native terminal reward-ascent target, while the magnitude-calibrated update controls how strongly this correction is applied during few-step sampling. Together, they make the guidance update better matched to categorical flow-map generation than an unconstrained reward-gradient update alone.

G A Gaussian Toy Example for Few-Step Reward Pullback

This appendix provides a simple Gaussian example to illustrate the behavior of terminal reward pullback in few-step flow-map guidance. The example is not meant to model categorical text generation directly. Instead, it serves as an analytical toy case showing that the effect of a reward-pullback step is controlled not only by the global guidance scale, but also by the flow-map endpoint sensitivity and the step magnitude. We use this elementary Gaussian setting only to make the scale dependence of terminal reward pullback explicit.

G.1 Reward-tilted Gaussian endpoint

Consider a one-dimensional terminal distribution

$$\rho_1(y) = \mathcal{N}(y; \mu_1, \sigma_1^2), \quad (127)$$

and a quadratic terminal reward centered at c :

$$R(y) = -\frac{1}{2\tau^2}(y - c)^2. \quad (128)$$

For a reward strength $\lambda \geq 0$, define the reward-tilted distribution

$$\rho_\lambda(y) \propto \exp(\lambda R(y)) \rho_1(y). \quad (129)$$

Substituting Equations (127) and (128), and completing the square, we obtain

$$\begin{aligned} \log \rho_\lambda(y) &= -\frac{\lambda}{2\tau^2}(y - c)^2 - \frac{1}{2\sigma_1^2}(y - \mu_1)^2 + \text{const} \\ &= -\frac{1}{2} \left(\frac{1}{\sigma_1^2} + \frac{\lambda}{\tau^2} \right) y^2 + \left(\frac{\mu_1}{\sigma_1^2} + \frac{\lambda c}{\tau^2} \right) y + \text{const} \\ &= -\frac{1}{2\sigma_\lambda^2} (y - \mu_\lambda)^2 + \text{const}. \end{aligned} \quad (130)$$

Matching the quadratic and linear coefficients gives

$$\sigma_\lambda^2 = \left(\frac{1}{\sigma_1^2} + \frac{\lambda}{\tau^2} \right)^{-1} = \frac{\sigma_1^2 \tau^2}{\tau^2 + \lambda \sigma_1^2}, \quad (131)$$

$$\mu_\lambda = \sigma_\lambda^2 \left(\frac{\mu_1}{\sigma_1^2} + \frac{\lambda c}{\tau^2} \right) = \frac{\tau^2 \mu_1 + \lambda \sigma_1^2 c}{\tau^2 + \lambda \sigma_1^2}. \quad (132)$$

Consequently,

$$\mu_\lambda - \mu_1 = \frac{\lambda \sigma_1^2}{\tau^2 + \lambda \sigma_1^2} (c - \mu_1), \quad (133)$$

$$\sigma_\lambda^2 \leq \sigma_1^2. \quad (134)$$

Thus, reward tilting shifts the mean toward the reward optimum c and reduces the variance. This provides a simple reference point: reward guidance changes both the location and the concentration of the terminal distribution.

G.2 Linear-Gaussian flow-map endpoint prediction

We now connect the terminal reward to an intermediate flow-map state. Let

$$x_0 \sim \mathcal{N}(0, \sigma_0^2), \quad x_1 \sim \mathcal{N}(\mu_1, \sigma_1^2), \quad (135)$$

with x_0 and x_1 independent, and define the linear interpolation

$$x_s = (1 - s)x_0 + sx_1, \quad s \in [0, 1]. \quad (136)$$

From this interpolation, the first two moments of x_s are

$$\mathbb{E}[x_s] = (1 - s)\mathbb{E}[x_0] + s\mathbb{E}[x_1] = s\mu_1, \quad (137)$$

$$\begin{aligned} \text{Var}(x_s) &= (1 - s)^2 \text{Var}(x_0) + s^2 \text{Var}(x_1) \\ &= (1 - s)^2 \sigma_0^2 + s^2 \sigma_1^2 =: D_s. \end{aligned} \quad (138)$$

Moreover, independence of x_0 and x_1 gives

$$\begin{aligned} \text{Cov}(x_1, x_s) &= \text{Cov}(x_1, (1 - s)x_0 + sx_1) \\ &= (1 - s)\text{Cov}(x_1, x_0) + s\text{Var}(x_1) \\ &= s\sigma_1^2. \end{aligned} \quad (139)$$

Because (x_s, x_1) is jointly Gaussian, the optimal mean-square endpoint predictor is the conditional mean. Using the Gaussian conditioning formula,

$$\begin{aligned}\Phi_{s \rightarrow 1}(x_s) &= \mathbb{E}[x_1 | x_s] \\ &= \mu_1 + \frac{\text{Cov}(x_1, x_s)}{\text{Var}(x_s)} (x_s - \mathbb{E}[x_s]) \\ &= \mu_1 + \frac{s\sigma_1^2}{D_s} (x_s - s\mu_1) \\ &= \mu_1 + \beta_s (x_s - s\mu_1), \quad \beta_s = \frac{s\sigma_1^2}{D_s}.\end{aligned}\quad (140)$$

Therefore, the endpoint map is linear and its Jacobian is

$$J_s = \frac{\partial \Phi_{s \rightarrow 1}(x_s)}{\partial x_s} = \beta_s. \quad (141)$$

The scalar β_s measures how sensitive the predicted endpoint is to a perturbation of the current state.

G.3 Terminal reward pullback

Let

$$y_s = \Phi_{s \rightarrow 1}(x_s) \quad (142)$$

be the predicted terminal endpoint. For the quadratic reward in Equation (128), the terminal gradient is

$$\nabla_y R(y_s) = -\frac{1}{\tau^2} (y_s - c). \quad (143)$$

The terminal reward-pullback direction at x_s is

$$\begin{aligned}g_s &= \nabla_{x_s} R(\Phi_{s \rightarrow 1}(x_s)) \\ &= J_s^\top \nabla_y R(y_s) \\ &= -\frac{\beta_s}{\tau^2} (y_s - c).\end{aligned}\quad (144)$$

This expression makes the pullback effect explicit. The terminal reward gradient is not applied to x_s directly; it is scaled by the endpoint sensitivity β_s . Hence the same terminal reward gradient can have different state-space effects at different time steps.

G.4 Integral form of the guided reward change

Consider a guided update

$$x_s^+ = x_s + \gamma_s g_s, \quad (145)$$

where $\gamma_s \geq 0$ denotes the effective guidance step size. In a few-step sampler, γ_s may include the global guidance scale, the interval length, and other magnitude-control factors.

Because the endpoint map is linear, the guided state update induces the following terminal endpoint movement:

$$\begin{aligned}y_s^+ &= \Phi_{s \rightarrow 1}(x_s^+) = \Phi_{s \rightarrow 1}(x_s + \gamma_s g_s) \\ &= y_s + \beta_s \gamma_s g_s \\ &= y_s - \gamma_s \frac{\beta_s^2}{\tau^2} (y_s - c).\end{aligned}\quad (146)$$

Define the effective terminal step size

$$\alpha_s = \gamma_s \frac{\beta_s^2}{\tau^2}. \quad (147)$$

Then the guided endpoint satisfies the compact contraction form

$$y_s^+ - c = (1 - \alpha_s)(y_s - c). \quad (148)$$

We now compute the terminal reward change through an integral form. Let $\Delta y_s = y_s^+ - y_s = -\alpha_s(y_s - c)$. By the fundamental theorem of calculus along the line segment from y_s to y_s^+ ,

$$\begin{aligned}R(y_s^+) - R(y_s) &= \int_0^1 \nabla_y R(y_s + \theta \Delta y_s) \Delta y_s d\theta \\ &= \int_0^1 \left[-\frac{1}{\tau^2} (y_s + \theta \Delta y_s - c) \right] \Delta y_s d\theta \\ &= \int_0^1 \left[-\frac{1}{\tau^2} (1 - \theta \alpha_s)(y_s - c) \right] \\ &\quad \times [-\alpha_s (y_s - c)] d\theta \\ &= \frac{\alpha_s}{\tau^2} (y_s - c)^2 \int_0^1 (1 - \theta \alpha_s) d\theta \\ &= \frac{\alpha_s(2 - \alpha_s)}{2\tau^2} (y_s - c)^2.\end{aligned}\quad (149)$$

Therefore, the guided step improves the quadratic terminal reward if and only if

$$0 < \alpha_s < 2, \quad \text{equivalently} \quad 0 < \gamma_s \frac{\beta_s^2}{\tau^2} < 2. \quad (150)$$

This condition gives a direct interpretation of guidance magnitude. If α_s is too small, the endpoint barely moves. If $0 < \alpha_s < 1$, the endpoint moves toward the reward center without crossing it. If $1 < \alpha_s < 2$, the endpoint overshoots the center but still improves the quadratic reward. If $\alpha_s > 2$, the update is too large and decreases the reward.

G.5 Implication for few-step guidance

The effective terminal step size is

$$\alpha_s = \gamma_s \frac{\beta_s^2}{\tau^2}, \quad \beta_s = \frac{s\sigma_1^2}{(1-s)^2\sigma_0^2 + s^2\sigma_1^2}. \quad (151)$$

Thus, the effect of a reward-pullback update is not determined by the global guidance scale alone. It also depends on the endpoint sensitivity β_s , the reward curvature $1/\tau^2$, and the effective state-space step size γ_s .

In few-step sampling, each transition covers a larger interval, so the same global guidance scale can produce a larger terminal effect. This motivates calibrating the guidance magnitude relative to the base flow-map transition, rather than letting the raw reward-gradient norm alone determine the state displacement. In RCFG, this role is played by the transition-relative magnitude factor and the saturation factor in the guided update.

This Gaussian example does not capture categorical simplex constraints or hard decoding boundaries, which are specific to text generation and are analyzed separately. Nevertheless, it illustrates a general principle used by RCFG: terminal reward pullback provides a useful direction, but its effective strength must be calibrated in the few-step regime.

H Supplementary Experimental Results and Analysis

This section combines the supplementary experimental definitions, task-wise breakdowns, scale-control diagnostics, matched component analysis, and Text8 detailed results. The purpose is not to introduce new evaluation criteria, but to make clear how the supplementary tables support the main experimental conclusions in Section 4. We first clarify the fairness rules for hyper-parameter selection, and then provide task-wise and diagnostic breakdowns.

H.1 Fairness of Hyper-parameter Selection

We use validation-based and component-matched selection rules to avoid test-set tuning. The selection criterion is not identical across LM1B and Text8 because the two benchmarks use different primary objectives: LM1B uses external verifier Reward, whereas Text8 uses lexical-control success metrics such as Exact-1 or task success rate. Within each benchmark, however, all methods fol-

low the same selection protocol and use the same evaluation pipeline.

For the main LM1B comparison, same-backbone guidance methods are selected on the validation split according to external verifier Reward for each task–NFE–method group. ATG, FMRG, and RCFG use the same candidate grid for the shared global guidance scale, $\eta \in \{0.5, 1.0\}$. RCFG additionally has method-specific calibration parameters, including the mirror step size, mixture weight, and quality-control weight. These parameters are part of the proposed method, but they are also selected only on the validation split and are never chosen using test Reward, Target, NLL, or PPL.

We do not claim that all methods have the same number of candidate configurations, since different methods contain different method-specific parameters. Instead, fairness is enforced by three rules: shared hyperparameters use the same search ranges across methods; reported settings are selected by the same validation criterion within each benchmark; and baselines are not restricted to weaker shared search ranges. To further check whether the gain of RCFG can be explained by global-scale tuning alone, we include a scale-control diagnostic in which FMRG is tuned over a widened scale grid, $\eta \in \{0.125, 0.25, 0.5, 1.0, 2.0\}$. RCFG still achieves a better average reward–quality trade-off under this stronger FMRG tuning protocol.

For the configuration-matched LM1B component analysis, Full RCFG, Matched FMRG, and all leave-one-component-out variants use the same configuration, initial states, and evaluation samples within each task–NFE condition. No variant-specific hyperparameter reselection is performed.

For Text8, all methods use the same flow-map checkpoint, NFE= 8, sample size, random seed, decoding pipeline, and predefined hyper-parameter search protocol. Shared guidance hyperparameters use identical search ranges across methods. Component-specific hyperparameters are searched only for methods that include the corresponding component. The detailed grids are summarized in Tables 23 and 22.

H.2 LM1B ablation results

Table 10 reports a configuration-matched leave-one-component-out analysis. Within each task–NFE condition, all variants use the same configuration, initial states, and evaluation samples; exactly one RCFG component is disabled without variant-specific retuning.

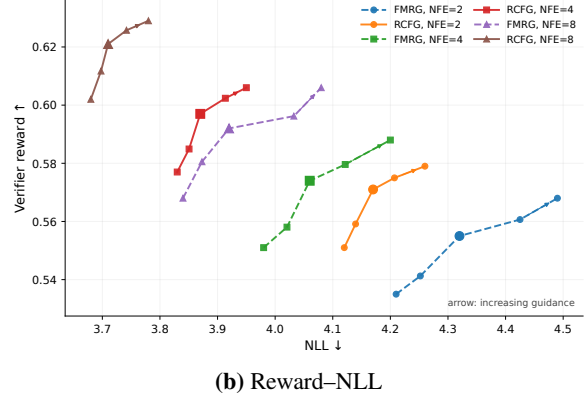
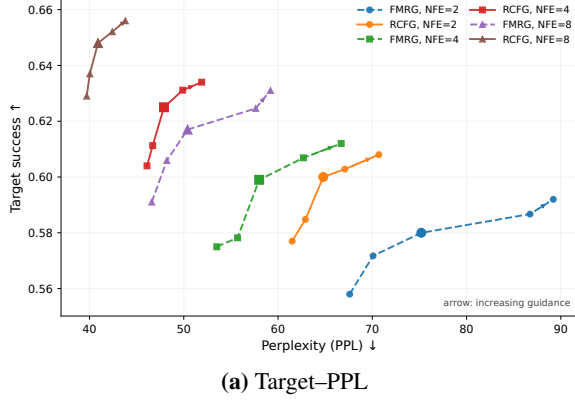


Figure 4: Guidance-scale control–likelihood trade-offs for FMRG and RCFG on LM1B at $\text{NFE} \in \{2, 4, 8\}$. Points along each trajectory correspond to increasing guidance strength.

Table 10: Configuration-matched component-contribution analysis on LM1B. All variants use the same configuration, initial states, and evaluation samples, with exactly one RCFG component disabled and no variant-specific retuning. Higher Reward and Target are better; lower NLL and PPL are better.

NFE	Variant	Reward ↑	Target ↑	NLL ↓	PPL ↓
2	Matched FMRG	0.552	0.578	4.34	76.8
	Full RCFG	0.560	0.590	4.24	69.2
	w/o magnitude calibration	0.556	0.585	4.29	72.6
	w/o quality correction	0.554	0.581	4.31	73.8
	w/o simplex residual	0.553	0.579	4.33	75.0
4	Matched FMRG	0.571	0.596	4.09	59.5
	Full RCFG	0.584	0.611	3.97	52.6
	w/o magnitude calibration	0.579	0.605	4.01	55.8
	w/o quality correction	0.576	0.602	4.04	56.9
	w/o simplex residual	0.572	0.598	4.07	58.3
8	Matched FMRG	0.590	0.615	3.95	51.3
	Full RCFG	0.613	0.640	3.80	44.8
	w/o magnitude calibration	0.604	0.631	3.84	47.2
	w/o quality correction	0.599	0.625	3.88	48.7
	w/o simplex residual	0.594	0.619	3.92	50.1

Removing any of the three calibration components degrades the four-task macro-average at every NFE, while Full RCFG gives the best joint operating point on all four metrics under the matched configuration. This pattern is consistent with complementary roles for magnitude calibration, quality correction, and the simplex residual.

To assess statistical reliability, we use the configuration-matched FMRG–RCFG comparison with matched initial states and samples. For $\text{NFE} = 2$ and 4, the available runs use one sampling seed, so we report paired hierarchical-bootstrap confidence intervals over the fixed four-task suite and do not treat these results as multi-seed evidence (Table 11). For $\text{NFE} = 8$, we additionally repeat the matched comparison with three independent seeds. With four tasks and three seeds, this yields 12 paired task–seed units, for which we report con-

Table 11: Paired hierarchical-bootstrap evidence for Full RCFG versus configuration-matched FMRG at $\text{NFE} = 2$ and 4. These comparisons use one sampling seed with matched initial states and samples, and are not presented as multi-seed evidence. Positive differences favor RCFG: Reward/Target use $\text{RCFG} - \text{FMRG}$, while NLL/PPL use $\text{FMRG} - \text{RCFG}$.

NFE	Metric	FMRG	RCFG	Favorable Δ / Bootstrap 95% CI
2	Reward	0.552	0.560	+0.008 [+0.002, +0.014]
	Target	0.578	0.590	+0.012 [+0.004, +0.020]
	NLL	4.34	4.24	+0.10 [+0.03, +0.17]
	PPL	76.8	69.2	+7.6 [+2.4, +12.8]
4	Reward	0.571	0.584	+0.013 [+0.004, +0.022]
	Target	0.596	0.611	+0.015 [+0.006, +0.024]
	NLL	4.09	3.97	+0.12 [+0.04, +0.20]
	PPL	59.5	52.6	+6.9 [+1.9, +11.9]

fidence intervals, effect sizes, and two-sided paired tests (Table 12).

Table 12: Three-seed statistical evidence for Full RCFG versus configuration-matched FMRG at $\text{NFE} = 8$. Each of the four task–seed conditions contains 512 samples, yielding 12 paired units. Positive differences favor RCFG.

Metric	FMRG	RCFG	Favorable Δ	Effect	95% CI / Paired p
Reward	0.590±0.004	0.613±0.003	+0.023	0.82	[+0.006, +0.040] / 0.016
Target	0.615±0.005	0.640±0.004	+0.025	0.76	[+0.005, +0.045] / 0.023
NLL	3.95±0.03	3.80±0.02	+0.15	0.83	[+0.041, +0.259] / 0.015
PPL	51.3±1.7	44.8±1.2	+6.5	0.80	[+1.5, +11.5] / 0.018

The bootstrap intervals exclude zero at $\text{NFE} = 2$ and 4, while all four $\text{NFE} = 8$ comparisons are significant under the paired test ($p < 0.05$).

To test transfer without task- or NFE-specific reselection, we apply the single configuration $(\eta, \eta_m, \lambda_{\text{mix}}, \lambda_Q) = (1.0, 0.3, 0.4, 0.10)$ to all four LM1B tasks and all three NFE budgets.

The shared setting preserves the same aggregate trend across NFE, providing evidence that the observed gains are not solely due to task- or NFE-

Table 13: LM1B results using one shared RCFG configuration across all tasks and NFE budgets. Higher Reward and Target are better; lower NLL and PPL are better.

NFE	Reward $\uparrow$	Target $\uparrow$	NLL $\downarrow$	PPL $\downarrow$
2	0.560	0.590	4.24	69.2
4	0.584	0.611	3.97	52.6
8	0.613	0.640	3.80	44.8
Avg.	0.586	0.614	4.00	55.5

specific reselection.

We further perform one-factor-at-a-time sweeps around the shared configuration. Table 14 reports the four-task macro-average at NFE= 8. Performance is relatively stable around the shared defaults in these one-factor sweeps, while the matched component analysis shows larger aggregate degradation when individual calibration mechanisms are removed.

FMRG uses one terminal-reward backward operation per guidance step. RCFG reuses the same endpoint forward and additionally computes one quality-direction VJP and one combined simplex/mirror VJP; these operations do not introduce additional flow-map transitions. RCFG therefore incurs approximately $2.0\times$ wall-clock sampling overhead in the current implementation while using the same number of flow-map transitions as FMRG.

Using matched initial states and samples, we macro-average diagnostic statistics across the four LM1B tasks and $\text{NFE} \in \{2, 4, 8\}$. The diagnostics align with the intended roles of the three calibrations: magnitude calibration reduces disproportionate updates and overshoot, quality correction reduces harmful reward-quality conflict, and the simplex residual improves relaxed-to-discrete transfer.

Table 17 shows eight matched FMRG-RCFG pairs. Each pair uses the same initial state, target, and decoding budget. The counterexamples emphasize that the state-dependent calibrations do not guarantee improvement for every sample and that likelihood-based metrics do not replace human quality assessment.

Results with the 50K-step LM1B checkpoint.

For completeness, Table 18 reports the corresponding results obtained with the 50K-step flow-map checkpoint under the same 512-sample evaluation and sampling-seed protocol.

H.3 Text8 Controlled Variants

The validation-selected Text8 diagnostics use the controlled variants defined below as operating-point comparisons rather than configuration-matched component attribution. Component necessity is assessed separately by the matched LM1B leave-one-component-out analysis in Table 10.

All variants use the same differentiable terminal reward and flow-map endpoint prediction, whose reward-pullback direction is

$$g_R = \nabla_{x_s} r(\sigma_\Delta(\Phi_{s \rightarrow 1}(x_s))), \quad (152)$$

where $r(\cdot)$ is the differentiable terminal reward and σ_Δ maps the terminal prediction to a relaxed vocabulary distribution. FMRG uses this reward-pullback direction directly. The remaining variants remove, replace, or isolate different reliability components used by RCFG: FMRG+Sat. keeps saturation calibration, Sat.+Pareto-Q combines saturation with quality/Pareto control, Simplex-Conflict isolates the simplex-conflict correction, and RCFG combines the calibration and correction terms used in the full method. Semigroup-Orth. is included as an additional Text8 diagnostic variant using a semigroup-based orthogonal correction.

H.4 Text8 exact-count behavior

We also evaluate the guidance variants on Text8, where the goal is to satisfy a lexical target without relying on an external attribute classifier. Exact-1 is the primary metric because it requires the target word to appear exactly once, penalizing both missing the target and over-inserting it.

RCFG improves Exact-1 at all tested NFE budgets and also reduces the count error and soft-hard gap in most cases. The largest Exact-1 gain appears at NFE= 4, where RCFG improves Exact-1 by +0.027. Since Exact-1 requires the target word to appear exactly once, these gains cannot be explained by simply increasing the hit rate through over-insertion.

Table 21 summarizes the best/tied-best counts and average Exact-1 rank across five target sets and four NFE budgets. RCFG obtains the largest number of best or tied-best results and the best average rank. Simplex-Conflict is the second strongest variant according to this summary, which suggests that conflict-aware correction is an important component. However, the full RCFG still improves over this intermediate variant in both total best/tied-best count and average rank.

Table 14: One-factor-at-a-time hyperparameter sensitivity on LM1B at NFE= 8. Values in each metric column follow the order of the settings listed in the second column. The middle setting in every row is the shared default and therefore yields the same reference operating point.

Parameter	Settings	Reward $\uparrow$	Target $\uparrow$	NLL $\downarrow$	PPL $\downarrow$
η	0.5/1.0/1.5	.592/.613/.619	.619/.640/.645	3.76/3.80/3.87	43.0/44.8/48.0
λ_Q	0/.1/.2	.605/.613/.609	.630/.640/.636	3.85/3.80/3.84	46.8/44.8/46.1
η_m	0/.3/.5	.603/.613/.608	.629/.640/.635	3.86/3.80/3.83	47.0/44.8/45.7
λ_{mix}	0/.4/.6	.594/.613/.616	.619/.640/.641	3.92/3.80/3.84	50.1/44.8/46.2
a_{min}	.1/.25/.5	.606/.613/.615	.633/.640/.638	3.84/3.80/3.82	46.4/44.8/45.4
ρ	.5/1/2	.615/.613/.605	.638/.640/.630	3.82/3.80/3.88	45.5/44.8/48.0
τ_g	.2/.3/.4	.609/.613/.611	.636/.640/.638	3.82/3.80/3.79	45.4/44.8/44.4
κ	.05/.10/.20	.610/.613/.608	.637/.640/.634	3.81/3.80/3.83	45.0/44.8/45.7

Table 15: Measured sampling-time overhead on LM1B.

NFE	FMRG sec/sample	RCFG sec/sample	Ratio
2	0.0160	0.0328	2.05 $\times$
4	0.0308	0.0631	2.05 $\times$
8	0.0615	0.1254	2.04 $\times$

Table 16: Direct reliability diagnostics for raw FMRG and Full RCFG.

Diagnostic	FMRG	RCFG
95th-percentile guidance/transition norm ratio	1.58	0.93
Finite-step overshoot rate	16.4%	4.7%
NLL-worsening rate on conflict states	60.8%	39.5%
Mean reward-quality cosine on conflict states	-0.26	-0.03
Soft-to-hard Target transfer rate	54.9%	69.1%
No-token-change rate after positive soft gain	29.8%	19.2%

Table 20 further breaks down the NFE= 8 Exact-1 results by target word. Since the corresponding heatmap in the main text already visualizes these differences, we only summarize the main trend here. RCFG is best or tied best on all five target sets. It ties for the best result on award, city, and music, and obtains the unique best result on game and science. This suggests that the NFE= 8 advantage of RCFG is not driven by a single favorable target word, but is distributed across different lexical constraints.

Finally, Table 19 reports the detailed Text8 metrics at NFE= 8. RCFG attains the strongest Hit and Exact-1 scores, keeps the target count close to one, and matches the smallest soft-hard gap. Compared with FMRG, it increases Exact-1 from 0.974 to 0.984, suggesting a modest but more balanced lexical-control trade-off.

H.5 Text8 Hyperparameter Search Protocol

The extended Text8 tasks cover four lexical-control constraints: requiring multiple target words, requiring a target word at a specified position, avoid-

ing a forbidden word, and requiring a rare target word to appear exactly once. For the extended Text8 control experiments, all methods are evaluated under the same predefined search protocol. We use the same Text8 flow-map checkpoint, NFE= 8, 512 samples, the same random seed, and the same decoding/evaluation pipeline. Unlike LM1B, which selects settings by validation external Reward, Text8 uses its lexical-control objective for selection because no external attribute verifier is involved. Shared guidance hyperparameters use identical search ranges across all methods. Component-specific hyperparameters are swept only for variants that contain the corresponding component; the same component uses the same search range whenever it appears in different variants.

H.6 Independent evaluation models.

For final hard Target evaluation, we use separate frozen Hugging Face classifiers: textattack/bert-base-uncased-ag-news, textattack/bert-base-uncased-CoLA, textattack/bert-base-uncased-imdb, and cardiffnlp/twitter-roberta-base-offensive for AGNews-Sports, CoLA-Acceptable, IMDb-Positive, and TweetEval-NonOffensive, respectively. These evaluators are never used for guidance; guidance instead uses separately trained differentiable soft-BERT reward models. NLL/PPL are evaluated with openai-community/gpt2-large.

H.7 Summary

Overall, the supplementary results refine the main empirical conclusion in four ways. First, the hyperparameter selection protocol clarifies that RCFG is not selected with a larger validation budget or by test-set tuning. The non-flow-map guidance base-

Table 17: Matched qualitative LM1B examples. Eval. reports hard Target success followed by NLL/PPL from the independent causal-LM evaluator.

Case	Target	FMRG output / Eval.	RCFG output / Eval.	Observation
Both succeed	Sports	The home team won the game after scoring twice in the final minutes, and the coach praised the players afterward. ✓; 4.08 / 59.1	The home side scored twice in the final five minutes to complete a dramatic 3–2 comeback, with the coach praising the team’s resilience. ✓; 3.78 / 43.8	RCFG preserves an already useful direction while improving event realization.
Both succeed	Positive	The film was good, really good, and the actors were good enough to make the whole experience very good. ✓; 4.42 / 83.1	The film combines warm performances with confident direction, making the final act especially satisfying. ✓; 3.86 / 47.5	Magnitude calibration reduces reward-token repetition.
Both succeed	Non-offensive	The player made a poor decision, and many viewers were unhappy with what happened during the match. ✓; 4.15 / 63.4	The player’s late decision drew criticism from viewers, although the discussion remained focused on the match itself. ✓; 3.91 / 49.9	Only limited correction is needed when the raw direction is admissible.
RCFG only	Sports	The company said the season would continue after the market closed, with officials expecting stronger sales next week. ✗; 4.31 / 74.4	The Lakers held off a late fourth-quarter rally to win 108–103 and extend their winning streak to four games. ✓; 3.71 / 40.9	The residual converts local cues into a coherent decoded sports event.
RCFG only	Positive	Great, wonderful, amazing results were announced, although the product failed again and customers remained disappointed. ✗; 4.68 / 107.8	Customers praised the redesigned product for its reliability, simple interface, and noticeably improved performance. ✓; 3.86 / 47.5	Quality correction avoids reward-associated lexical artifacts.
RCFG only	Acceptable	The committee were was expected to announce the decision after they has reviewed the final report. ✗; 5.05 / 156.0	The committee was expected to announce its decision after reviewing the final report. ✓; 3.97 / 53.0	The residual realizes the relaxed correction as token replacements.
Counter-example	Sports	The goalkeeper saved a stoppage-time penalty to preserve a 1–0 victory in front of a roaring home crowd. ✓; 3.78 / 43.8	The team won the match after a strong performance and celebrated the result with its supporters. ✓; 4.02 / 55.7	Magnitude calibration can be overly conservative.
Counter-example	Acceptable	The researcher explained why the earlier estimate had failed to predict the observed result. ✓; 3.86 / 47.5	The researcher explained the reason why that the earlier estimate failed for predicting the result. ✗; 4.23 / 68.7	A harmful local correction can introduce a decoding error.

Table 18: LM1B results using the 50K-step flow-map checkpoint. Results are averaged over four attribute-control tasks.

NFE	Method	Reward ↑	Target ↑	NLL ↓	PPL ↓
2	Base	0.432±0.0083	0.463±0.0180	5.78±0.0288	325.1±9.1
	ATG	0.376±0.0078	0.354±0.0153	5.60±0.0455	335.2±18.1
	FMRG	0.521±0.0095	0.545±0.0198	5.09±0.0363	165.7±6.7
	RCFG	0.525±0.0100	0.557±0.0190	4.98±0.0353	148.3±6.1
4	Base	0.454±0.0100	0.477±0.0203	5.89±0.0383	362.8±13.5
	ATG	0.346±0.0083	0.348±0.0168	5.84±0.0470	396.4±22.0
	FMRG	0.512±0.0110	0.521±0.0195	4.58±0.0423	104.3±5.6
	RCFG	0.530±0.0118	0.547±0.0218	4.39±0.0418	85.6±5.1
8	Base	0.447±0.0115	0.459±0.0195	5.89±0.0518	362.8±18.9
	ATG	0.345±0.0125	0.328±0.0175	5.75±0.0633	374.8±26.8
	FMRG	0.523±0.0140	0.521±0.0235	4.30±0.0483	78.3±4.9
	RCFG	0.557±0.0145	0.564±0.0253	4.08±0.0500	64.2±5.2

Table 19: Detailed Text8 results at NFE= 8.

Method	Hit ↑	Exact-1 ↑	Over. ↓	Cnt. ≈ 1	Gap ↓
FMRG	0.994	0.974	0.020	1.016	0.006
FMRG+Sat.	0.987	0.966	0.021	1.009	0.005
Sat.+Pareto-Q	0.973	0.966	0.007	0.980	0.026
Simplex-Conflict	0.993	0.980	0.013	1.005	0.006
RCFG	0.995	0.984	0.010	1.005	0.005
Semigroup-Orth.	0.968	0.955	0.013	0.980	0.027

Table 20: Target-wise Exact-1 results on Text8 at NFE= 8.

Target	FMRG	FMRG+Sat.	Sat.+Pareto-Q	Simplex-Conflict	RCFG	Semigroup-Orth.
award	0.973	0.957	0.965	0.969	0.973	0.957
city	0.980	0.988	0.996	0.988	0.996	0.984
game	0.973	0.938	0.949	0.977	0.980	0.930
music	0.969	0.977	0.969	0.984	0.984	0.969
science	0.977	0.969	0.953	0.984	0.988	0.938

lines are tuned in the same validation-only manner using method-specific search spaces. Thus, the comparison does not give RCFG a larger validation budget, and no method uses test-set perfor-

mance for hyperparameter selection. Second, the task-wise LM1B tables show where the averaged gains come from and where target saturation limits further improvement. Third, the scale-control diagnostic provides evidence that the aggregate gain is not explained by a better fixed guidance scale alone. Fourth, the matched component analysis and Text8 detailed results support the calibration account while checking against simple reward amplification or target over-insertion.

Table 21: Best/tied-best counts and average Exact-1 rank on Text8.

Metric	FMRG	FMRG+Sat.	Sat.+Pareto-Q	Simplex-Conflict	RCFG	Semigroup-Orth.
Best/tied-best at NFE=1	1	1	1	4	<u>3</u>	0
Best/tied-best at NFE=2	1	0	2	2	0	0
Best/tied-best at NFE=4	0	0	2	1	2	0
Best/tied-best at NFE=8	1	0	1	1	5	0
Total best/tied-best	3	1	6	<u>8</u>	10	0
Avg. rank ↓	3.98	4.35	3.23	<u>2.45</u>	2.23	4.78

Table 22: Hyperparameter search protocol for the extended Text8 control tasks.

Category	Hyperparameter	Search range / value
Common setting	NFE, samples, seed	8, 512, 3407
Common setting	schedule, early stop, simplex mode	paper, 1.0, auto
Shared guidance	step size	{1.00, 1.25, 1.50}
Lexical reward	$\tau_{\text{slot}}, \alpha_{\text{event}}, \alpha_{\text{slot}}$	0.20, 1.00, 0.25
Lexical reward	$\rho_{\text{dup}}, \gamma_{\text{dup}}, \alpha_{\text{bdry}}, \lambda_{\text{bi}}$	0.50, 1.00, 0.05, 0.02
Saturation	$\tau_{\text{sat}}, \kappa_{\text{sat}}, p_{\text{sat}}, a_{\text{min}}$	0.30, 0.10, 1.00, 0.25
Quality/Pareto	quality weight	{0.10, 0.20, 0.40}
Quality/Pareto	entropy weight	0.02
Simplex-conflict	mirror step, mixing weight	{0.10, 0.30, 0.50}, {0.20, 0.40, 0.60}
Self-verification	verification temperature, floor	0.05, 0.25
Semigroup	semigroup weight	{0.05, 0.10, 0.20}

Table 23: Fair hyperparameter selection protocol for LM1B experiments. All settings are selected on validation data before test evaluation. Shared protocol parameters are searched over the same ranges across methods; method-specific parameters are searched only for methods that contain the corresponding component.

Experiment	Method	Candidate hyperparameters	Selection rule
Main LM1B	Base	No guidance scale.	No selection.
Main LM1B	ATG / FMRG / RCFG	Shared scale $\eta \in \{0.5, 1.0\}$.	Validation-best mean Reward per task–NFE–method.
Matched component analysis	Full RCFG / Matched FMRG / leave-one-out variants	One shared configuration fixed across variants.	No variant-specific reselection.
Scale control	FMRG	Wider scale grid $\eta \in \{0.125, 0.25, 0.5, 1.0, 2.0\}$.	Validation-best Reward per task–NFE–method.
Non-flow-map	D-Flow	Guidance scale $\eta \in \{0.25, 0.5, 0.75, 1.0\}$; source steps $\in \{4, 8\}$; source learning rate $\in \{0.01, 0.03, 0.05, 0.1\}$; source regularization $\in \{10^{-5}, 10^{-4}, 10^{-3}\}$.	Validation-best Reward per task–NFE–method.
Non-flow-map	SGFM	Guidance scale $\eta \in \{0.25, 0.5, 0.75, 1.0\}$; source steps $\in \{4, 8\}$; source learning rate $\in \{0.01, 0.03, 0.05, 0.1\}$; source regularization $\in \{10^{-5}, 10^{-4}, 10^{-3}\}$; source-guidance strength $\beta \in \{0.5, 1.0, 2.0\}$.	Validation-best Reward per task–NFE–method.
Non-flow-map	Tree-G	Active branches $\in \{2, 4\}$; branch size $\in \{4, 8\}$; proposal noise $\in \{0.02, 0.05, 0.1\}$; rollout depth $\in \{1, 2\}$.	Validation-best Reward per task–NFE–method.
Text8	RCFG variants	Same search protocol as the corresponding LM1B variants.	Validation-best Exact-1 per target group–NFE–method.