# HOLDING THE PER-STEP KL: WHY CLIPPING CANNOT, BUT CONTROL CAN

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

GRPO-based reinforcement learning (RL) aligns flow-matching models with task rewards, but ratio clipping does not directly control the KL divergence of policy updates. We analyze this gap through an exact Gaussian decomposition of the per-step log-ratio, in which the per-step KL sets both its mean and variance. This reveals a distinction between calibrating the clipping frequency and controlling policy displacement: for the centered threshold rules considered, standardization removes the first-order dependence of the clipping gate on the overall displacement scale. Building on this distinction, we develop the method in three layers. F-Clip clips the centered log-ratio in KL-scaled coordinates; Z-Clip adapts the clipping range to the batch spread, so that its threshold corresponds to a target clipping fraction; and K-Hold uses learning-rate feedback to keep the floor-corrected mean KL over selected denoising steps near a prescribed target. Our analysis provides a unified characterization of ratio clipping in flow-matching RL, connecting the per-step log-ratio to the KL quantity that must ultimately be controlled. Experiments on OCR-based text rendering with SD3.5-M and FLUX.2-klein validate the predicted clipping and KL behavior. K-Hold improves held-out reward-model scores over the baselines, while preserving the gain under matched OCR performance.

## 1 INTRODUCTION

Flow-matching models have become a widely used backbone for image and video generation (Lipman et al., 2023; Esser et al., 2024), while RL provides a way to align them with rewards not captured during pretraining, such as rendered-text accuracy or learned preference scores. This approach builds on methods developed for language models. GRPO (Shao et al., 2024) samples a group of outputs for each prompt, assigns them rewards, and updates the policy using advantages normalized within the group. Flow-GRPO and DanceGRPO (Liu et al., 2025a; Xue et al., 2025) extend this framework to flow models by replacing the deterministic sampler with a marginal-preserving stochastic sampler, making each denoising step a Gaussian policy. They retain the clipped PPO objective (Schulman et al., 2017), under which a sample stops contributing to the gradient once its per-step importance ratio leaves a window around one. This clipping window serves as the only trust-region mechanism in these methods and operates independently at each denoising step.

In flow models, the clipping window behaves differently from its counterpart in language models. The per-step ratio has a mean below one, while its spread varies by more than an order of magnitude across denoising steps. As a result, positive-advantage samples rarely reach the upper clipping boundary, and the policy over-optimizes the training reward: the training score continues to increase while image quality and prompt adherence deteriorate (Wang et al., 2026). GRPO-Guard addresses this issue by centering the log-ratio at each step, rescaling it by the standard deviation of that step's policy, and reweighting the loss across steps, which delays over-optimization. However, the term removed by centering is the KL divergence between the policy that generated the samples and the policy being trained at that step. This leads to two cases. If the divergence remains in the ratio, the same clipping window acts more tightly at some steps than at others because the divergence varies across steps. If it is normalized away, the window instead measures displacement in the raw coordinates of the latent space. In either case, the magnitude of the per-step policy update is not set explicitly and is only observed after the update.



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Rather than measuring the displacement only after the update, we characterize its effect on the log-ratio directly. At each step, the old and current policies differ only in their means, which gives an exact decomposition of the log-ratio into the negative per-step KL and a single Gaussian term scaled by the same KL. The clipping window thus depends only on this quantity and unit Gaussian noise (Sec. 4.1). Building on this identity, we develop the method in three layers. *(i)* We first examine the effect of the existing normalization by tracing its rescaling through the gradient, leading to **F-Clip**, which uses a centered ratio while retaining the same reweighting (Sec. 4.2). *(ii)* We then set the clipping range according to the batch spread, giving **Z-Clip**, whose single parameter specifies the target clipping fraction consistently across steps and systems (Sec. 4.3). *(iii)* Finally, to hold the per-step KL at a chosen level, we move enforcement from the clipping window to the step size and introduce **K-Hold**, a learning-rate controller that maintains the KL at a target specified in nats (Sec. 4.4). Extensive experiments validate the theoretical analysis (Sec. 5), including the predicted behavior of F-Clip, Z-Clip, and K-Hold, as well as their held-out rewards relative to the baselines.

## 2 Related Work

TRPO constrains each policy update through the KL divergence from the policy that generated the samples (Schulman et al., 2015). PPO preserves this goal but replaces the explicit KL constraint with clipping of the per-sample importance ratio (Schulman et al., 2017). Later analyses showed that this surrogate can be loose: a single-sample ratio provides limited information about the divergence between two policies, and clipping does not directly prevent that divergence from growing (Engstrom et al., 2020; Wang et al., 2020; 2019). GRPO adopts the same clipped objective while replacing the critic with advantages normalized within a group of samples (Shao et al., 2024). Its use in language models has motivated several refinements of the ratio, including asymmetric clipping (Yu et al., 2025), corrected normalization (Liu et al., 2025b), and clipping the importance weight rather than the gradient (Chen et al., 2025). In these methods, the policy divergence remains a quantity inferred only indirectly from the ratio. In flow models, however, the per-step policy is Gaussian, making the divergence available in closed form. This work takes that closed form as its starting point.

**RL for diffusion and flow models.** Reward-driven fine-tuning of diffusion models generally follows one of two approaches. The first avoids explicit likelihood-ratio optimization (Domingo i Enrich et al., 2025; Zheng et al., 2026; McAllister et al., 2026). For instance, Diffusion-DPO directly optimizes preference pairs (Wallace et al., 2024), while DRaFT backpropagates a differentiable reward through the sampler (Clark et al., 2024). The second approach treats the reverse process as a Markov decision process and applies policy gradients to the per-step likelihood (Black et al., 2024; Fan et al., 2023). Flow-GRPO and DanceGRPO extend this approach to flow models by replacing the ODE sampler with a marginal-preserving SDE, making each denoising step a Gaussian policy (Liu et al., 2025a; Xue et al., 2025). Subsequent work along this line improves sample efficiency and credit assignment across denoising steps (Li et al., 2025; He et al., 2026; Li et al., 2026; Deng et al., 2026; Savani et al., 2026). Across these methods, clipping of the per-step importance ratio is largely inherited unchanged. This work focuses on what that clipping rule actually constrains.

**Regulating the per-step ratio in flow GRPO.** The works most closely related to ours modify the per-step ratio directly. TempFlowGRPO reweights the gradient at each step using a noise-aware factor to reduce imbalance across steps (He et al., 2026). Flow-DPPO argues that a single-sample ratio is a poor proxy for the policy divergence and replaces ratio clipping with a mask based on the exact per-step KL (Ping et al., 2026). GRPO-Guard, the direct predecessor of this work, observes that the ratio distribution is shifted below one and has a spread that varies across steps, attributing the failure of fixed clipping to this shift (Wang et al., 2026). By centering the log-ratio at each step and rescaling it by the policy scale, GRPO-Guard balances the clipping fraction across steps and delays over-optimization. We identify the corrected shift as the per-step KL and show that a clipping window can either remain uniform across steps or respond to this quantity, but not both.

## 3 Preliminaries

**Flow matching.** Let $x_0$ be a sample from the data distribution and $x_1 \sim \mathcal{N}(0, I_d)$ be Gaussian noise in a $d$-dimensional latent space. Rectified flow defines the interpolant $x_\tau = (1-\tau)x_0 + \tau x_1$

for $\tau \in [0,1]$ and trains a velocity network $v_\theta$ to predict $v = x_1 - x_0$ by minimizing $\mathbb{E}_{\tau,x_0,x_1}\|v - v_\theta(x_\tau,\tau)\|^2$. Sampling integrates the learned velocity from $\tau = 1$ to $\tau = 0$ over a grid $\tau_0 = 1 > \tau_1 > \cdots > \tau_T$. We index the intervals by $t \in \{0,\dots,T-1\}$ and write $v_\theta(x,t)$ accordingly. The step size $dt = \tau_t - \tau_{t+1} > 0$ may vary with $t$; throughout the paper, $dt$ denotes this discrete step size rather than a differential.

**SDE sampling and the per-step policy.** Flow-GRPO (Liu et al., 2025a) and DanceGRPO (Xue et al., 2025) replace the deterministic ordinary differential equation (ODE) sampler with a marginal-preserving stochastic differential equation (SDE), making each step a Gaussian policy to which policy-gradient methods can be applied. We use the following form throughout the paper:

$$x_{t+1} = \mu_{\theta,t}(x_t) + s_t\,\varepsilon, \quad \mu_{\theta,t}(x) = \gamma_t(x) - \beta_t\,dt\,v_\theta(x,t), \quad s_t = \sigma_t\sqrt{dt}, \quad \varepsilon \sim \mathcal{N}(0, I_d), \tag{1}$$

with the velocity and state coefficients

$$\beta_t = 1 + \frac{\sigma_t^2(1-\tau_t)}{2\tau_t}, \qquad \gamma_t(x) = x\Big(1 - \frac{\sigma_t^2\,dt}{2\tau_t}\Big). \tag{2}$$

Each step defines the Gaussian policy $\pi_{\theta,t}(\cdot \mid x_t) = \mathcal{N}(\mu_{\theta,t}(x_t), s_t^2 I_d)$. The two frameworks differ only in the noise scale: Flow-GRPO uses $\sigma_t = a\sqrt{\tau_t/(1-\tau_t)}$ and DanceGRPO uses $\sigma_t = a$, with noise level $a > 0$.

**GRPO for flow models.** For a prompt $c$, a group of $G$ trajectories $\{x^i\}_{i=1}^G$ is sampled from the current policy $\pi_{\text{old}}$, with each trajectory assigned a scalar reward based on its final image. The advantage $\hat{A}^i$ is standardized within the group, clamped, and shared across all steps of trajectory $i$. Define the per-step importance ratio as $r_t^i(\theta) = \pi_{\theta,t}(x_{t+1}^i \mid x_t^i)/\pi_{\text{old},t}(x_{t+1}^i \mid x_t^i)$, the policy is updated by maximizing the clipped objective

$$\mathcal{J}(\theta) = \frac{1}{G}\sum_{i=1}^{G}\frac{1}{T}\sum_{t=0}^{T-1}\min\Big(r_t^i(\theta)\hat{A}^i,\ \mathrm{clip}\big(r_t^i(\theta), 1-\epsilon_{\mathrm{c}}, 1+\epsilon_{\mathrm{c}}\big)\hat{A}^i\Big), \tag{3}$$

where $\epsilon_{\mathrm{c}}$ is the clip range. In the released Flow-GRPO codebase, the log-probability of a step is computed as a mean over latent dimensions. Hence, $\log r_t(\theta) = \ell_t = L_t/d$, where $L_t$ is the total log-ratio, and Eq. (3) exponentiates $\ell_t$, giving $e^{\ell_t}$.

**Policy displacement and the per-step KL.** At the same state, the sampling policy and the current policy differ only in their means, so their divergence has a closed-form expression. Let $\Delta\mu_t = \mu_{\text{old},t}(x_t) - \mu_{\theta,t}(x_t)$ denote the displacement between the policy means. Then

$$\kappa_t = D_{\mathrm{KL}}\big(\pi_{\text{old},t}\,\|\,\pi_{\theta,t}\big) = \frac{\|\Delta\mu_t\|^2}{2s_t^2}, \qquad k_t = \kappa_t/d, \tag{4}$$

where $\kappa_t$ is the per-step KL and $k_t$ is its per-dimension value. The deterministic term to which GRPO-Guard (Wang et al., 2026) traces the shift of the ratio distribution away from one is $-\kappa_t$.

## 4 METHOD

The per-step KL $\kappa_t$ of Eq. (4) is what a trust region should bound, yet the clipped objective does not constrain it directly; $\kappa_t$ enters only implicitly through the log-ratio distribution. Leaving that shift in place makes the clip range act unevenly across steps, while taking it out leaves the clip rule with nothing that responds to the displacement. Sec. 4.1 makes this dependence explicit and separates the problem into three layers: the clipped variable, the clip range, and the update. The sections below address them in turn: Sec. 4.2 examines which variable the clip should act on, leading to **F-Clip**; Sec. 4.3 derives a clip range that maintains the same clipping fraction at every step, yielding **Z-Clip**; Sec. 4.4 shows that no clip range can both remain uniform across steps and respond to the displacement, and moves this role outside the objective, resulting in **K-Hold**; Sec. 4.5 presents the three components in implementation-ready form. App. A defines the notation, App. B states and proves the theory in full, and App. C justifies the assumptions. **Numbered statements use the same numbering as in the appendix.**

### 4.1 The Exact Per-Step Decomposition

The log-ratio of a step is where the shift arises, so we derive an exact decomposition. Since $\pi_{\theta,t}$ and $\pi_{\mathrm{old},t}$ share the covariance $s_t^2 I_d$, the log-ratio is affine in a single coordinate of the sampling noise.

**Theorem 1** (Three-term decomposition)**.** *Let $x_{t+1} = \mu_{\mathrm{old},t} + s_t\varepsilon$ be drawn from the sampling policy and $L_t = \log \pi_{\theta,t}(x_{t+1} \mid x_t) - \log \pi_{\mathrm{old},t}(x_{t+1} \mid x_t)$. With $z = \Delta\mu_t \cdot \varepsilon / \|\Delta\mu_t\| \sim \mathcal{N}(0,1)$,*

$$L_t = -\kappa_t - \sqrt{2\kappa_t}\, z. \tag{5}$$

*Consequently, (i) $\mathbb{E}[e^{L_t}] = 1$; (ii) $\mathbb{E}[L_t] = -\kappa_t$ and $\mathrm{Var}[L_t] = 2\kappa_t$; (iii) $\mathbb{E}[L_t] = -1/2\,\mathrm{Var}[L_t]$.*

In Eq. (5), $\kappa_t$ indicates both the mean of the log-ratio and the scale of its stochastic term. A $d$-dimensional step therefore carries only one scalar source of randomness, and this single number fixes both where the log-ratio is centered and how widely it varies. The objective, however, exponentiates the per-dimension mean $\ell_t = L_t/d$ rather than $L_t$. Cor. 1 gives the corresponding distribution by dividing Eq. (5) by $d$,

$$\ell_t = -k_t - \frac{\sqrt{2\kappa_t}}{d}\, z. \tag{6}$$

Eq. (6) reveals three nested layers, moving from the variable being clipped, to the range of clipping, and finally to the update that sets the policy displacement. The first layer is the clipped variable, $\ell_t$ or $\ell_t + k_t$, which decides whether the mean shift is retained in the clipping window. The second is the clip range $\epsilon_c$ relative to the spread $\sqrt{2\kappa_t}/d$, which determines the fraction of samples clipped. The third layer lies outside the clipping rule and concerns $\kappa_t$ itself, which the first two layers cannot directly change. By Eq. (4), $\kappa_t$ is set by $\Delta\mu_t$, and $\Delta\mu_t$ changes only through $\theta$, its level is therefore set by the update. The subsections below follow these three layers in turn.

### 4.2 Clip Geometry and Gradient Equivalence

The first layer concerns the variable on which clipping acts. Removing the mean shift in Eq. (6) yields the debiased variable $\tilde{u}_t = \ell_t + k_t$. Rescaling it by the policy scale then gives $\hat{u}_t = s_t\tilde{u}_t$, which GRPO-Guard exponentiates. We compare the samples clipped under these choices and examine whether the rescaling changes the gradients of the samples that remain unclipped.

**Proposition 2** (Three geometries)**.** *Conditionally on $x_t$, with $z$ as in Thm. 1,*

$$\tilde{u}_t = -\frac{\sqrt{2\kappa_t}}{d}\, z, \quad \hat{u}_t = -\frac{\|\Delta\mu_t\|}{d}\, z, \quad \frac{\tilde{u}_t}{\mathrm{std}(\tilde{u}_t)} = -z. \tag{7}$$

*A fixed threshold on $\tilde{u}_t$ corresponds to a threshold on $\kappa_t z^2$, while a fixed threshold on $\hat{u}_t$ corresponds to one on $\|\Delta\mu_t\|\,|z|$. The standardized form removes $\sqrt{2\kappa_t}$ entirely. Clipping $\hat{u}_t$ at $\epsilon_c$ is equivalent to clipping $\tilde{u}_t$ at $\epsilon_c/s_t$.*

In each form the clipped variable is $z$ times a scale that the sample does not choose, so a fixed threshold bounds that scale against $|z|$, and the three scales are different quantities. For $\tilde{u}_t$, the resulting bound is on $\kappa_t z^2$, a per-sample trust region in the KL of the policy pair. For $\hat{u}_t$, the bound is on $\|\Delta\mu_t\|\,|z|$, a Euclidean length that depends on the coordinates used by the latent representation (Rem. 6). In the standardized form, the scale factor is removed, leaving a bound on $|z|$ alone.

The next question is how each choice affects the gradient.

**Lemma 1** (Debiasing cancels the displacement in the gradient)**.** *Under Ass. 2, with $J_t = \partial\mu_{\theta,t}/\partial\theta$,*

$$\nabla_\theta \ell_t = \frac{(\Delta\mu_t + s_t\varepsilon)^\top J_t}{d\, s_t^2}, \quad \nabla_\theta k_t = -\frac{\Delta\mu_t^\top J_t}{d\, s_t^2}, \quad \nabla_\theta \tilde{u}_t = \frac{\varepsilon^\top J_t}{d\, s_t} = -\frac{\beta_t\sqrt{dt}}{d\,\sigma_t}\,\varepsilon^\top \frac{\partial v_{\theta,t}}{\partial\theta}. \tag{8}$$

The displacement enters the first two gradients with opposite signs and therefore cancels in their sum, leaving $\nabla_\theta\tilde{u}_t$ dependent only on the sampling noise. We next use the pair notation below to examine whether the rescaling by $s_t$ has a similar effect.

We represent each variant by a pair $(u, \delta_t)$, where $u$ is the clipped variable and $\delta_t$ is the per-step loss weight. An unclipped sample then contributes $\mathcal{L} = -\hat{A}\,\delta_t e^u$.

**Theorem 2** (Net-multiplier identity)**.** *For any unclipped sample, the per-sample gradients of* $(\hat{u}_t, 1/dt)$ *and of* $(\tilde{u}_t, \sigma_t/\sqrt{dt})$ *satisfy*

$$\nabla_\theta \mathcal{L}_{(\hat{u}_t,1/dt)} = e^{(s_t-1)\tilde{u}_t}\, \nabla_\theta \mathcal{L}_{(\tilde{u}_t,\sigma_t/\sqrt{dt})}, \quad \nabla_\theta \mathcal{L} = \hat{A}\, e^{u}\, \beta_t\, \varepsilon^\top \big(\partial v_{\theta,t}/\partial\theta\big)/d \tag{9}$$

*with the first factor* $1 + O(\tilde{u}_t)$ *under Ass. 1.*

For an unclipped sample, $s_t$ therefore acts as a per-step loss weight rather than as a normalization factor. The two variants differ in which samples they clip (Rem. 8), as characterized by the geometry in Prop. 2. The prefix in Eq. (9) captures the explicit schedule dependence, but does not characterize how $\varepsilon^\top \partial v_{\theta,t}/\partial\theta$ varies across steps.

Prop. 2 places the trust region on $\tilde{u}_t$, and Thm. 2 shows that replacing $\hat{u}_t$ with $\tilde{u}_t$ leaves the gradients of unclipped samples unchanged up to the stated factor. We therefore propose **F-Clip**, which clips $\tilde{u}_t$ while retaining the framework's original $\delta_t$. This completes the first layer; the clipping range remains fixed.

### 4.3 A Clip Fraction that Does not Move

The second layer concerns the clip range. F-Clip still compares a fixed $\epsilon_c$ against the spread $\sqrt{2\kappa_t}/d$ in Eq. (6). We examine the resulting clipping fraction.

**Proposition 3** (Clip fraction of a fixed threshold)**.** *Conditionally on* $x_t$ *and neglecting the* $O(\epsilon_c^2)$ *difference between clipping* $e^u$ *and clipping* $u$*, a threshold* $\epsilon_c$ *on* $\hat{u}_t$ *clips a fraction*

$$\varphi_t(\epsilon_c) = 2\Big(1 - \Phi\big(d\,\epsilon_c/\|\Delta\mu_t\|\big)\Big) \tag{10}$$

*of the samples, while the corresponding fraction for the debiased variable is* $2(1 - \Phi(d\epsilon_c/\sqrt{2\kappa_t}))$*. Both depend on the realized displacement scale.*

A fixed $\epsilon_c$ corresponds to a threshold $z_t^\star = d\,\epsilon_c/\|\Delta\mu_t\|$ on $z$. Since this threshold varies with the displacement, $\epsilon_c$ does not fix the clipping fraction (Cor. 3). This fraction controls how many samples from each step contribute to the gradient, so the effective weighting across steps varies implicitly.

A range scaled by the spread of the clipped variable removes this dependence. Concretely, let $\hat{s}_t = \mathrm{std}_B(\tilde{u}_t)$ over the batch at step $t$ and set $\epsilon_t = z^\star \hat{s}_t$, with $z^\star$ fixed throughout training. The extent to which a single $z^\star$ corresponds to a common clipping fraction depends on the variation of $\kappa_t$ across states in the batch.

**Proposition 4** (Invariance of the standardized threshold)**.** *In the large-*$n$ *limit, the following hold.*

- *(i) If* $\tilde{R}_t = 0$*, then* $\varphi_t = 2(1 - \Phi(z^\star))$ *for every* $t$*, and a target fraction* $\varphi$ *is obtained with by* $z^\star = \Phi^{-1}(1 - \varphi/2)$*.*
- *(ii) In general,* $\varphi_t$ *departs from* $2(1 - \Phi(z^\star))$ *by a functional of the law of* $w = \kappa_{t,x}/\mathbb{E}_x\kappa_t$ *alone, linear in* $\tilde{R}_t = \mathrm{Var}[w]$ *at leading order.*

*At fixed relative spread,* $\varphi_t$ *is independent of* $\kappa_t$*,* $\|\Delta\mu_t\|$ *and* $s_t$*.*

Here, $\tilde{R}_t = \mathrm{Var}_x[\kappa_t]/(\mathbb{E}_x[\kappa_t])^2$ is the relative spread of the per-state KL across the batch (Cor. 2), and $\hat{s}_t$ can be obtained from the batch mean of $\ell_t$ alone (Rem. 10). The same parameter can thus be used across different backbones, schedulers, and noise levels without recalibration. Prop. 4(ii) accounts for heterogeneity within the batch: the resulting deviation is small and measurable, and its sign changes at one value of $z^\star$ (Rem. 11).

We therefore propose **Z-Clip**, which replaces the fixed range of F-Clip with $\epsilon_t = z^\star \hat{s}_t$ while leaving the clipped variable and $\delta_t$ unchanged. The range scales with $\hat{s}_t$, while $z^\star$ remains fixed, so the remaining parameter specifies a clipping fraction rather than an absolute range.

### 4.4 Enforcing the Level

A rule with a fixed $z^\star$ has no mechanism to respond to $\kappa_t$, whereas allowing $z^\star$ to vary with the displacement breaks uniformity across steps. To see whether this trade-off is specific to these two choices or holds for any threshold rule, we examine the gradient that remains after clipping.

Fix $\{x_t, J_t, s_t\}$ and the joint law of $(A, \varepsilon)$, which are set by $\pi_{\mathrm{old},t}$ and the reward independently of $\theta$. A clipping rule can be represented by the set $U$ of samples it retains in noise coordinates. Its first-order expected surviving gradient is $G_t(U) = \mathbb{E}[\mathbf{1}_U(\varepsilon, A)A\varepsilon^\top]J_t/(ds_t)$. The rule responds to the magnitude of the displacement at step $t$ when $G_t$ depends on $\|\Delta\mu_t\|$; dependence on its direction is present for every rule. The PPO maximum retains $U(z^\star) = \{\mathrm{sign}(A)z \geq -z^\star\}$. We further define $c_t(z^\star) = \mathbb{E}[(Az)^-\mathbf{1}\{|z| > z^\star\}] \geq 0$ and let $\varepsilon_\perp$ denote the component of $\varepsilon$ orthogonal to $\widehat{\Delta\mu}_t$.

**Proposition 5** (Uniformity and enforcement are exclusive for threshold rules)**.** *Under Ass. 1 and 2, the clipped policy loss has expected gradient $-\delta_t G_t(U)$ up to $O(\tilde{u}_t)$, where $\delta_t$ is the variant's loss weight, and*

$$G_t\big(U(z^\star)\big) = G_t(\mathbb{R}^d) + \frac{c_t(z^\star)}{d\, s_t}\,\widehat{\Delta\mu}_t^\top J_t - \frac{1}{d\, s_t}\,\mathbb{E}\big[\mathbf{1}_{U^c} A\, \varepsilon_\perp^\top\big] J_t, \tag{11}$$

*with $c_t$ non-increasing in $z^\star$. Any dependence of $G_t$ on $\|\Delta\mu_t\|$ enters through $z^\star$. A threshold rule thus responds to the displacement magnitude exactly when $z^\star$ depends on $\|\Delta\mu_t\|$ and remains uniform when it does not.*

Prop. 5 applies to rules defined by a threshold on $z$, which includes the clipping rules considered here. A fixed $\epsilon_c$ pulls back on the displacement through the second term of Eq. (11), with a stronger effect as the range narrows, whereas Z-Clip does not. Lem. 1 removes $\Delta\mu_t$ from $\nabla_\theta\tilde{u}_t$, leaving the clipping indicator as the only remaining source of displacement dependence. Under Ass. 3, Cor. 4 places the resulting equilibrium at a displacement with a Euclidean scale that varies across steps and changes with the schedule and sampling law. It therefore cannot be specified directly in nats.

Thus, enforcement must instead act through the update. Under Ass. 4, an update $\Delta\theta$ induces a displacement $J_t\Delta\theta$ at step $t$, giving $\kappa_t(\Delta\theta) = \|J_t\Delta\theta\|^2/(2s_t^2)$. Lem. 4 then gives a simple scaling law: scaling the update by $\alpha$ scales every $\kappa_t$ by $\alpha^2$, with the same factor at all steps. Whether the profile across steps can also change depends on how the steps share the update.

**Proposition 6** (Profile rigidity under row-degenerate, aligned coupling)**.** *Let $g_b$ be the gradient direction contributed by step $b$ and $C[b][t] = \|J_t g_b\|/s_t$. Assume (a) row degeneracy, $C[b][t] = c_b\rho_t$ with scalars $c_b > 0$ and a profile $\rho_t > 0$, and (b) alignment, such that for each $t$, the vectors $J_t g_b$ are positively collinear. Then for every update $\Delta\theta = \sum_b \delta_b g_b$ with $\delta_b \geq 0$,*

$$\kappa_t(\Delta\theta) = \tfrac{1}{2}\Big(\sum_b \delta_b c_b\Big)^2 \rho_t^2, \quad \textit{so} \quad \frac{\kappa_t(\Delta\theta)}{\kappa_{t'}(\Delta\theta)} = \frac{\rho_t^2}{\rho_{t'}^2}, \tag{12}$$

*and a gradient-scale-invariant optimizer leaves $(\sum_b \delta_b c_b)^2$ unchanged when the $\delta_b$ are rescaled.*

The profile shape cannot be adjusted independently, whereas its overall scale can (Rem. 14). The minimum achievable scale is limited by finite-precision arithmetic. Under reduced precision, a nonzero update can produce a displacement floor that no longer decreases as the update is reduced. The measured quantity can thus be written as $\kappa_t^{\mathrm{meas}} = \kappa_t(\Delta\theta) + \kappa_t^{\mathrm{floor}}$, where $\kappa_t^{\mathrm{floor}} \geq 0$ is a constant associated with the backbone, numerical format, and step. Any target below this floor cannot be reached simply by reducing the update magnitude (Rem. 15).

There are three possible points at which to act on the aggregate scale, two of which are limited by quantities that their own control variables cannot change. **Actuator A** (Def. 8) rescales the realized update by the largest $\alpha \in [\alpha_{\min}, 1]$, with $\alpha_{\min} > 0$, that keeps $\kappa_t^{\mathrm{meas}}$ within a prescribed budget. If the budget lies below the floor, $\alpha$ remains at $\alpha_{\min}$. **Actuator B** (Def. 9) adds the closed-form KL relative to the epoch-start policy to the loss. Even as its coefficient increases, the update magnitude remains set by the step size, so $\bar{\kappa}$ remains of the order that the step size fixes (Lem. 6). Both actuators are ultimately limited by the step size, which is where the third actuator operates.

**Actuator C.** Fix a target $\kappa^\star > 0$ in nats, control steps $\mathcal{T}_{\mathrm{ctrl}}$, a floor table, a gain $g > 0$, a smoothing constant $\eta \in (0, 1]$ and bounds $0 < m_{\min} < m_{\max}$. Once per epoch, let $\bar{\kappa}$ denote the mean floor-subtracted per-step KL over $\mathcal{T}_{\mathrm{ctrl}}$, measured after the final update of the epoch. We update the smoothed log-KL as $\log\bar{\kappa}_s \leftarrow (1-\eta)\log\bar{\kappa}_s + \eta\log\bar{\kappa}$ and set

$$m \;\leftarrow\; \mathrm{clip}\Big(m\,(\kappa^\star/\bar{\kappa}_s)^{g/2},\, [m_{\min}, m_{\max}]\Big), \tag{13}$$

where $m$ scales the optimizer's learning rate for the next epoch. The exponent $g/2$ is fixed by the loop gain rather than chosen independently; Lem. 7 characterizes its effect.

**Lemma 7** (Log-gain of the step-size actuator)**.** *Under Ass. 4, for an optimizer whose update is homogeneous of degree one in the learning rate at fixed gradient statistics, $\partial \log \bar{\kappa} / \partial \log m = 2$, so Eq.* (13) *then gives*

$$\log \bar{\kappa}' = (1 - g) \log \bar{\kappa} + g \log \kappa^\star \tag{14}$$

*at fixed gradient statistics. The log-error $\log(\bar{\kappa}/\kappa^\star)$ is reduced by a factor of $1 - g$ per epoch and vanishes in one epoch when $g = 1$. The profile shape $(\kappa_t/\kappa_{t'})$ remains unchanged across epochs.*

The exponent $g/2$ accounts for the gain of two, so each epoch changes $\log \bar{\kappa}$ by $g$ times the log-error and reduces the error geometrically. Because the controller changes only $m$, Prop. 6 applies the same scaling across steps and preserves the existing profile.

We finally propose **K-Hold**, which trains Z-Clip with Actuator C. The first two layers remain unchanged, preserving the clipping fraction from Sec. 4.3, while the step size acts on the third layer. The target does not require an externally specified absolute value: it can be set as $\kappa^\star = \varrho\, \bar{\kappa}_{\mathrm{ref}}$, where $\bar{\kappa}_{\mathrm{ref}}$ is the open-loop level at $m = 1$ over a reference window. Ass. 5 specifies the range of targets for which the actuator remains unsaturated, while drift in the open-loop level introduces a steady-state offset that cannot be eliminated by any finite gain (Rem. 16).

### 4.5 The Three Layers in Closed Form

We now write the three layers in implementation form. Starting from GRPO-Guard, each layer changes a single part of the preceding formulation.

***F-Clip*: Simplification.** Each variant follows Eq. (3) with a pair $(u, \delta_t)$, where $u$ is the variable whose exponential is clipped and $\delta_t$ is the per-step loss weight,

$$\mathcal{J}_{(u,\delta)}(\theta) = \frac{1}{G} \sum_{i=1}^{G} \frac{1}{T} \sum_{t=0}^{T-1} \delta_t \min\Big( r_t^i \hat{A}^i,\ \mathrm{clip}\big(r_t^i, 1 - \epsilon_{\mathrm{c}}, 1 + \epsilon_{\mathrm{c}}\big) \hat{A}^i \Big), \quad r_t^i = e^{u_t^i}. \tag{15}$$

GRPO-Guard uses $u = \hat{u}_t$, with $\hat{u}_t = s_t \tilde{u}_t$, whereas F-Clip uses $u = \tilde{u}_t$; the two differ only by the factor $s_t$. Thm. 2 shows that this factor does not affect the gradient of unclipped samples, so removing it only changes the geometry of the clipping window to the Fisher metric. F-Clip is the minimal form equivalent to GRPO-Guard (Rem. 12), with a prefix $\beta_t / s_t$ that leaves the $\kappa_t$ profile unchanged (Cor. 5).

***Z-Clip*: Anchor.** A fixed $\epsilon_{\mathrm{c}}$ specifies an absolute range. Cor. 3 maps this range to a threshold $d\epsilon_{\mathrm{c}} / \|\Delta \mu_t\|$ in $z$, which varies across steps and training. As a result, the same value of $\epsilon_{\mathrm{c}}$ does not correspond to a fixed clipping fraction. Z-Clip instead sets the clipping window from the batch,

$$\mathcal{J}_{\mathrm{Z}}(\theta) = \mathcal{J}_{(\tilde{u}_t, 1/dt)}(\theta)\big|_{\epsilon_{\mathrm{c}} \to z^\star \hat{s}_t}, \tag{16}$$

so the clipping fraction itself becomes the parameter (Prop. 4), without recalibration across backbones, schedulers, or noise levels.

***K-Hold*: Control.** F-Clip and Z-Clip are both threshold rules, and Prop. 5 shows that no such rule can be both uniform and responsive to the displacement. K-Hold keeps $\mathcal{J}_{\mathrm{Z}}$ and moves this response to the update,

$$\Delta\theta = m\, \mathcal{O}\big(\nabla_\theta \mathcal{J}_{\mathrm{Z}}\big), \tag{17}$$

where $\mathcal{O}$ denotes the update returned by the optimizer at its base learning rate, and $m$ is updated once per epoch according to Eq. (13). The resulting controller has a known gain and contraction rate while preserving the profile shape (Lem. 7). K-Hold can then set the per-step KL in nats, rather than measuring it only after the update.

## 5 Experiments

### 5.1 Experimental Settings

**Setup.** The task is visual text rendering, and the training reward is the OCR accuracy of the text rendered in the image. All runs build on the official Flow-GRPO codebase[1] and its prompt set.

[1] https://github.com/yifan123/flow_grpo

Table 1: Main results on the OCR task at the last checkpoint, averaged over two seeds. The last column is the mean standardized gain over the base model. Best per column in bold.

| Method | OCR | PickScore | HPSv2 | ImageReward | UnifiedReward | Average |
|---|---|---|---|---|---|---|
| Base (SD3.5-M) | 0.134 | 20.24 | 0.188 | −0.238 | 2.365 | +0.00 |
| +Flow-GRPO | 0.885 | 20.23 | 0.165 | −1.039 | 2.000 | −0.70 |
| +GRPO-Guard | **0.974** | 20.53 | 0.172 | −0.736 | 2.280 | −0.38 |
| **+K-Hold (Ours)** | 0.959 | **21.11** | **0.204** | **+0.013** | **2.835** | **+0.47** |
| Base (FLUX.2-klein) | 0.133 | 20.15 | 0.198 | −0.133 | 2.415 | +0.00 |
| +Flow-GRPO | 0.910 | 19.73 | 0.128 | −1.973 | 1.098 | −1.99 |
| +GRPO-Guard | **0.922** | 20.77 | 0.195 | −0.414 | 2.611 | −0.04 |
| **+K-Hold (Ours)** | 0.912 | **21.17** | **0.213** | **+0.122** | **3.024** | **+0.52** |

*Note.* All images are sampled without classifier-free guidance, following the no-guidance setting of the Flow-GRPO. The results reported in GRPO-Guard paper use a guidance scale of 4.5 and are not comparable here.

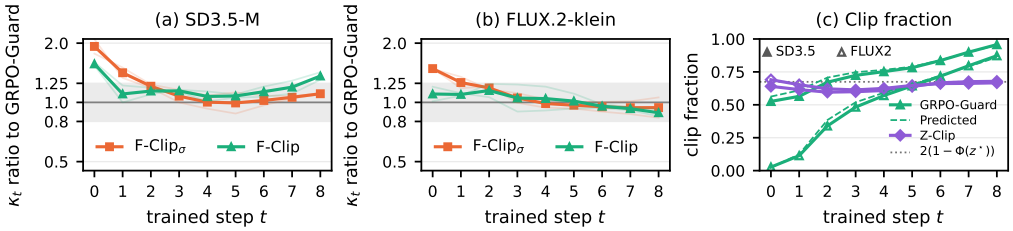


Figure 1: The two clipping layers on SD3.5-M and FLUX.2-klein, epochs 16–239. (a, b) Per-step KL ratio to GRPO-Guard of F-Clip$_\sigma$ and F-Clip, thin lines per seed with band $[0.8, 1.25]$. (c) Per-step clip fraction of GRPO-Guard and Z-Clip.

We fine-tune SD3.5-M[2] and FLUX.2-klein-base-4B[3] with LoRA on a single NVIDIA B200. We evaluate K-Hold along with the two components it builds on, F-Clip and Z-Clip (Sec. 4), and with Flow-GRPO (Liu et al., 2025a), DanceGRPO (Xue et al., 2025) and GRPO-Guard (Wang et al., 2026) as baselines. App. D gives the full configuration.

**Evaluation Metrics.** We report the OCR accuracy, which is the training reward, and four held-out reward models evaluated on 256 test prompts. PickScore-v1 (Kirstain et al., 2023) and HPS-v2.1 (Wu et al., 2023) are CLIP-based human preference models. ImageReward-v1.0 (Xu et al., 2023) is a BLIP-based one. UnifiedReward-7B-v1.5 (Wang et al., 2025) is a vision-language model that rates alignment and quality from 1 to 5.

### 5.2 Main Results

Tab. 1 compares K-Hold with the base model, Flow-GRPO, and GRPO-Guard at the final checkpoint on both backbones. Flow-GRPO and GRPO-Guard improve the OCR reward, but underperform the base model on the Average. In contrast, K-Hold outperforms the base model on both backbones, with a slightly lower OCR reward than GRPO-Guard. Due to page limit, DanceGRPO results are reported in App. E, where K-Hold exhibits the same pattern.

### 5.3 Components Analysis

To evaluate the two clipping layers against the theory, Fig. 1 compares them with GRPO-Guard on both backbones. Panels (a) and (b) show the per-step KL profile. Thm. 2 pairs GRPO-Guard with the variant that carries $s_t$ in the weight instead of in the clipped variable, written F-Clip$_\sigma$, so their profiles should agree. They do at all but the first two steps, where the excess comes from the per-step weight and not from the clip set. F-Clip keeps GRPO-Guard's weight and differs from it by the prefix $1/s_t$, and it too stays within the band, except near the image end on SD3.5-M. Panel (c) shows the clip fraction. The fixed window increases across denoising steps, as predicted by Prop. 3

[2] https://huggingface.co/spaces/stabilityai/stable-diffusion-3.5-medium
[3] https://huggingface.co/black-forest-labs/FLUX.2-klein-base-4B

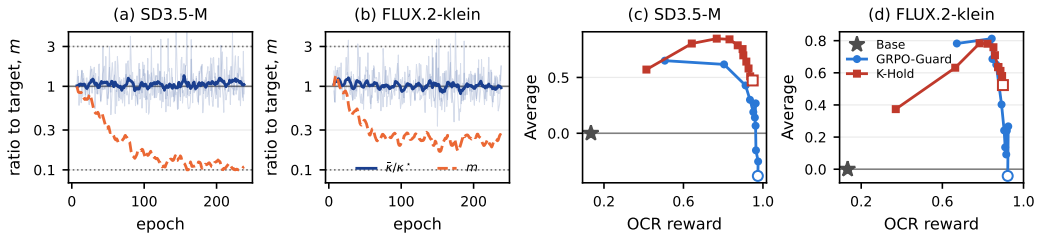


Figure 2: Actuator C at the main setting $\varrho = 1$. (a, b) Ratio of the measured to the target per-epoch KL (solid) and the multiplier $m$ (dashed). (c, d) Average against the OCR reward at every checkpoint, the base model as a star and end points as open markers.

from measured KL alone. In contrast, the standardized window in Z-Clip keeps the clip fraction approximately constant.

To isolate the effect of Actuator C, Fig. 2 tracks K-Hold under the main setting on both backbones. Panels (a) and (b) show that the measured per-epoch KL remains near its target throughout training. This demonstrates that the feedback loop controls a quantity that the objective alone does not. The multiplier continues to adapt over training, indicating that the controller actively maintains the target. App. F shows the same behavior across all target ratios and under saturated settings. Panels (c) and (d) trace held-out quality against the OCR reward rather than against training time, so a method that simply advances more slowly ends further left instead of higher.

### 5.4 Ablations

We compare K-Hold with four fixed multipliers drawn from the range its controller visits, which separates the effect of adapting $m$ to the measured KL from the effect of $m$ itself. The smallest of them also tests whether K-Hold keeps held-out quality by learning more slowly. Although it maintains a relatively high Average score, it substantially slows OCR improvement. K-Hold reaches much higher OCR while retaining a higher Average score, showing that its behavior is not explained by a uniformly small learning rate alone. The layer-wise pattern is consistent on FLUX.2-klein: GRPO-Guard, F-Clip, and Z-Clip remain within $0.1$ of one another, while K-Hold is the only variant above the base model. App. G reports results for the other two actuators.

Table 2: Left, SD3.5-M, a fixed multiplier $m$ on Z-Clip against K-Hold. Right, FLUX.2-klein, the three layers.

| SD3.5-M | OCR | Avg. |
|---|---|---|
| $m = 0.1$ | 0.886 | $+0.22$ |
| $m = 0.3$ | 0.951 | $-0.36$ |
| $m = 0.5$ | 0.965 | $-0.34$ |
| $m = 1.0$ | 0.977 | $-1.58$ |
| K-Hold | 0.959 | $+0.47$ |

| FLUX.2-klein | OCR | Avg. |
|---|---|---|
| Flow-GRPO | 0.910 | $-1.99$ |
| GRPO-Guard | 0.922 | $-0.04$ |
| F-Clip | 0.928 | $-0.14$ |
| Z-Clip | 0.922 | $-0.08$ |
| K-Hold | 0.912 | $+0.52$ |

## 6 Conclusion and Limitations

In this paper, we show that the per-step KL determines both the mean shift and variance of the log-ratio in flow-matching RL, and use this characterization to separate clipping calibration from KL control. F-Clip expresses clipping in KL-scaled coordinates, while Z-Clip standardizes the clipping range to equalize the clip fraction across steps. Because clipping alone cannot regulate the underlying policy displacement, K-Hold instead controls the per-step KL through learning-rate feedback and maintains it near a prescribed target. Experiments on two backbones confirm the predicted clipping and KL behavior. At matched OCR performance, K-Hold also improves held-out reward over the baselines.

Our framework has two main limitations. First, the controller regulates the overall scale of the per-step KL profile rather than each denoising step independently. The assumption that these steps move together is supported empirically, but not derived theoretically. Second, the framework does not prescribe how to choose the KL target, nor does it establish that maintaining a fixed KL necessarily improves training over allowing it to drift. The target-ratio sweep and downstream performance gains should therefore be viewed as empirical results.

## AI USE STATEMENT

In this work, we used generative AI tools for "assisting in the writing of proofs, proposing or refining hypotheses, implementing methods, and providing feedback on research methodology or experiments". We have not used generative AI tools for "developing theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, designing research methodology or experiments, or interpreting results", and "generating synthetic data sets, assisting with translation, cleaning and reformatting datasets, and supporting qualitative and thematic data analysis" are not applicable to this work. Additionally, we used generative AI tools for "drafting parts of the paper, editing the paper to improve readability, and suggesting a structure for the paper". We have reviewed all AI-assisted work. "AI assistance on proofs was limited to checking the consistency of notation and verifying derivations written by the authors, and every hypothesis proposed or refined and every methodological suggestion made with AI assistance was derived or tested by the authors before it was adopted". We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

## REPRODUCIBILITY STATEMENT

Anonymous code to reproduce the methods of this paper is provided in the supplementary material. Detailed experimental setup is given in Appendix D. All theoretical results are stated with their assumptions and proved in full in Appendix B, with the notation collected in Appendix A and the dependencies among the results summarized in Appendix B.10. Every assumption used in these proofs is justified in Appendix C, either by construction of the training pipeline or by a direct measurement.

## REFERENCES


Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations*, 2024.

Aili Chen, Aonian Li, Bangwei Gong, Binyang Jiang, Bo Fei, Bo Yang, Boji Shan, Changqing Yu, Chao Wang, Cheng Zhu, et al. Minimax-m1: Scaling test-time compute efficiently with lightning attention. *arXiv preprint arXiv:2506.13585*, 2025.

Kevin Clark, Paul Vicol, Kevin Swersky, and David J. Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *International Conference on Learning Representations*, 2024.

Haoyou Deng, Keyu Yan, Chaojie Mao, Xiang Wang, Yu Liu, Changxin Gao, and Nong Sang. DenseGRPO: From sparse to dense reward for flow matching model alignment. In *The Fourteenth International Conference on Learning Representations*, 2026.

Carles Domingo i Enrich, Michal Drozdzal, Brian Karrer, and Ricky TQ Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. In *International Conference on Learning Representations*, 2025.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep rl: A case study on ppo and trpo. In *International Conference on Learning Representations*, 2020.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning*, 2024.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36:79858–79885, 2023.

Xiaoxuan He, Siming Fu, Yuke Zhao, Wanli Li, Jian Yang, Dacheng Yin, Fengyun Rao, and Bo Zhang. Tempflow-grpo: When timing matters for grpo in flow models. In *International Conference on Learning Representations*, 2026.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:36652–36663, 2023.

Junzhe Li, Yutao Cui, Tao Huang, Weijie Kong, Yiming Cheng, Chuxuan Zeng, Yinping Ma, Chun Fan, Miles Yang, Zhao Zhong, et al. Mixgrpo: Unlocking flow-based grpo efficiency with mixed ode-sde. *arXiv preprint arXiv:2507.21802*, 2025.

Yuming Li, Yikai Wang, Zhongyu Zhao, Ming Lu, Qi She, Shanghang Zhang, et al. Branchgrpo: Stable and efficient grpo with structured branching in diffusion models. In *International Conference on Learning Representations*, 2026.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *Advances in Neural Information Processing Systems*, 38:40783–40818, 2025a.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. In *Conference on Language Modeling*, 2025b.

David McAllister, Songwei Ge, Brent Yi, Chung Min Kim, Ethan Weber, Hongsuk Choi, Haiwen Feng, and Angjoo Kanazawa. Flow matching policy gradients. In *International Conference on Learning Representations*, 2026.

Bowen Ping, Xiangxin Zhou, Penghui Qi, Minnan Luo, Liefeng Bo, and Tianyu Pang. Flow-dppo: Divergence proximal policy optimization for flow matching models. *arXiv preprint arXiv:2606.11025*, 2026.

Yash Savani, Branislav Kveton, Yuchen Liu, Yilin Wang, Jing Shi, Subhojyoti Mukherjee, Nikos Vlassis, and Krishna Kumar Singh. Stepwise credit assignment for grpo on flow-matching models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 42007–42017, 2026.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pp. 1889–1897, 2015.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Conference on Computer Vision and Pattern Recognition*, pp. 8228–8238, 2024.

Jing Wang, Jiajun Liang, Jie Liu, Henglin Liu, Gongye Liu, Jun Zheng, Wanyuan Pang, Ao Ma, Zhenyu Xie, Xintao Wang, et al. Grpo-guard: Mitigating implicit over-optimization in flow matching via regulated clipping. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5988–5998, 2026.

Yibin Wang, Yuhang Zang, Hao Li, Cheng Jin, and Jiaqi Wang. Unified reward model for multimodal understanding and generation. *arXiv preprint arXiv:2503.05236*, 2025.

Yuhui Wang, Hao He, Xiaoyang Tan, and Yaozhong Gan. Trust region-guided proximal policy optimization. *Advances in Neural Information Processing Systems*, 32, 2019.

Yuhui Wang, Hao He, and Xiaoyang Tan. Truly proximal policy optimization. In *Uncertainty in Artificial Intelligence*, pp. 113–122, 2020.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, and Ping Luo. Dancegrpo: Unleashing grpo on visual generation, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2025.

Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. Diffusionnft: Online diffusion reinforcement with forward process. In *International Conference on Learning Representations*, 2026.

## APPENDIX GUIDE

The appendix provides the notation, definitions, statements, and proofs supporting the main text, together with the empirical measurements corresponding to each theoretical result.


**Appendix A** Notations . . . . . 14

**Appendix B** Theory: definitions, statements, proofs . . . . . 15

B.1 Setting . . . . . 15

B.2 Exact decomposition of the log-ratio . . . . . 15

B.3 The schedule factor . . . . . 16

B.4 Classification of clipped variables . . . . . 17

B.5 Gradient theorem and the net-multiplier identity . . . . . 18

B.6 Clip fractions: fixed thresholds versus a standardized threshold . . . . . 19

B.7 Loss-side control of the per-step KL is structurally weak . . . . . 23

B.8 Enforcement: actuators on the per-epoch KL . . . . . 24

B.9 Framework covariance: DanceGRPO . . . . . 25

B.10 Summary . . . . . 26

**Appendix C** Justification of the assumptions . . . . . 27

**Appendix D** Experimental settings . . . . . 29

**Appendix E** Results on the DanceGRPO sampler . . . . . 30

**Appendix F** Behaviour of the target ratio . . . . . 32

**Appendix G** Actuators A and B . . . . . 33

## A NOTATIONS

| Symbol | Meaning |
|---|---|
| *Sampler and schedule* | |
| $t$, $T$ | trained step index in sampling order, number of trained steps |
| $d$ | latent dimension |
| $\tau_t$, $dt$ | flow-matching time ($\tau_0 = 1$, decreasing), step size $\tau_t - \tau_{t+1}$ |
| $a$, $\sigma_t$ | SDE noise level, SDE noise scale ($a\sqrt{\tau_t/(1-\tau_t)}$ or $a$) |
| $s_t = \sigma_t\sqrt{dt}$ | per-step policy standard deviation |
| $\beta_t$, $\gamma_t(x)$ | velocity and state coefficients ($\bar{\beta} = 1 + a^2/2$ for $t \geq 1$, $\beta_0 = 1$) |
| $v_\theta$, $\mu_{\theta,t}$ | velocity network, policy mean $\gamma_t(x) - \beta_t\, dt\, v_\theta(x,t)$ |
| $\varepsilon$ | Gaussian noise vector $\mathcal{N}(0, I_d)$ of a sample |
| $\Delta\mu_t$, $\Delta v_t$ | displacement $\mu_{\mathrm{old},t} - \mu_{\theta,t} = -\beta_t\, dt\, \Delta v_t$, velocity difference |
| $J_t$ | $\partial\mu_{\theta,t}/\partial\theta$, a $d \times p$ matrix at $x_t$ |
| $F_t$ | schedule factor $\beta_t^2 dt/(2\sigma_t^2)$, so that $\kappa_t = F_t\|\Delta v_t\|^2$ |
| *Log-ratios and per-step KL* | |
| $L_t$, $\ell_t = L_t/d$ | total and per-dimension log-ratio against the sampling policy |
| $\kappa_t$, $k_t = \kappa_t/d$ | per-state KL $\|\Delta\mu_t\|^2/(2s_t^2)$ (nats), its per-dimension value |
| $z$ | noise coordinate $\Delta\mu_t \cdot \varepsilon/\|\Delta\mu_t\| \sim \mathcal{N}(0,1)$ |
| $\tilde{R}_t$ | relative spread of the per-state KL, $\mathrm{Var}_x\kappa_t/(\mathbb{E}_x\kappa_t)^2$ |
| $w$ | relative per-state KL $\kappa_{t,x}/\mathbb{E}_x\kappa_t$ |
| $\mathbb{E}_x$, $\mathbb{E}_B$, $\mathrm{Var}_B$ | expectation over states, batch mean, batch variance |
| *Clipped variables, windows and variants* | |
| $\tilde{u}_t = \ell_t + k_t$ | debiased per-dimension log-ratio (per-state constant) |
| $\hat{u}_t = s_t\tilde{u}_t$ | RatioNorm variable, $\log \hat{r}_t(\theta)$ |
| $u$, $\delta_t$ | a variant's clipped variable and per-step loss weight |
| $\epsilon_{\mathrm{c}}$, $\epsilon_t$ | fixed clip range, standardized clip range $z^\star \hat{s}_t$ |
| $\hat{s}_t$ | batch standard deviation of $\tilde{u}_t$ |
| $z^\star$ | standardized threshold parameter |
| $z_t^\star$ | implied parameter of a fixed threshold, $d\epsilon_{\mathrm{c}}/\|\Delta\mu_t\|$ |
| $\varphi_t$ | two-sided clip fraction at step $t$ |
| F-Clip, F-Clip$_\sigma$ | clip window in the Fisher metric (on $\tilde{u}_t$), weight $1/dt$ and $\sigma_t/\sqrt{dt}$ |
| Z-Clip, K-Hold | F-Clip with the standardized window, and Z-Clip under Actuator C |
| *Gate and feedback* | |
| $A$ | group-normalized, clamped advantage of a sample |
| $U$, $G_t(U)$ | unclipped set in noise coordinates, expected surviving gradient |
| $c_t(z^\star)$ | gate pull-back coefficient $\mathbb{E}[(Az)^-\mathbf{1}\{\|z\| > z^\star\}]$ |
| $\varepsilon_\perp$ | component of $\varepsilon$ orthogonal to $\widehat{\Delta\mu}_t$ |
| $z^\star_{\mathrm{bal}}$, $\kappa_t^*$ | balance point and set point of a fixed threshold |
| *Profile, floor and actuators* | |
| $g_b$ | per-step gradient direction |
| $C[b][t]$ | coupling matrix $\|J_t g_b\|/s_t$ |
| $c_b$, $\rho_t$ | row-degenerate factors of $C$ |
| $\kappa_t^{\mathrm{floor}}$, $\kappa_t^{\mathrm{meas}}$ | bf16 measurement floor, measured per-step KL |
| $\alpha$, $\alpha_{\min}$, $b$ | Actuator A, projection scale, its lower bound, per-dimension budget |
| $\lambda$, $\mathcal{T}_{\mathrm{pen}}$, $c$ | Actuator B, penalty coefficient, penalized steps, dead-band constant |
| $m$, $g$, $\eta$ | Actuator C, learning-rate multiplier, gain, log-smoothing constant |
| $\kappa^\star$ | KL target (nats) |
| $\varrho$ | relative target ratio |
| $\bar{\kappa}$, $\mathcal{T}_{\mathrm{ctrl}}$ | control signal, control steps |
| $\nu$ | per-epoch log drift of the open-loop level |

# B THEORY: DEFINITIONS, STATEMENTS, PROOFS

## B.1 SETTING

Throughout, $d$ is the latent dimension, $t \in \{0, \dots, T-1\}$ indexes the trained sampling steps in sampling order, $G$ is the group size, $i$ the sample index and $\hat{A}_t^i$ the group-normalized, clamped advantage. $\tau_t \in (0, 1]$ is the flow-matching time of the scheduler at step $t$, with $\tau_0 = 1$ and $\tau_t$ decreasing; $dt = \tau_t - \tau_{t+1} > 0$ is the step size, a number depending on $t$, and $(dt)^2$ its square. $a > 0$ is the SDE noise level and $\sigma_t$ the SDE noise scale, with $s_t = \sigma_t\sqrt{dt}$ the per-step policy standard deviation. $\Delta\mu_t$ is the displacement of the policy mean, $\beta_t$ the drift coefficient and $\delta_t$ the per-step reweight scale; $r_t(\theta)$ and $\hat{r}_t(\theta)$ are the raw and normalized ratios, $\varepsilon$ the Gaussian noise vector and $\epsilon_c$ the clip range. Log-probabilities are evaluated as per-dimension means: $\log r_t(\theta) = \ell_t = L_t/d$ with $L_t$ the total log-ratio, and the correction term $\|\Delta\mu_t\|^2/(2\sigma_t^2 dt)$ is read in the same per-dimension units, where it equals $k_t = \kappa_t/d$ (Definition 2); the normalized ratio is $\log \hat{r}_t(\theta) = \hat{u}_t$ (Definition 3). All statements are per step $t$ and conditional on the current state $x_t$ unless stated otherwise; the subscript $t$ is dropped when no ambiguity arises.

**Definition 1** (Per-step Gaussian policy of the SDE sampler)**.** Let $\sigma_t > 0$ be the SDE noise scale and define the velocity coefficient and the state coefficient

$$\beta_t \;=\; 1 + \frac{\sigma_t^2(1-\tau_t)}{2\tau_t}, \qquad \gamma_t(x) \;=\; x\Big(1 - \frac{\sigma_t^2\, dt}{2\tau_t}\Big). \tag{18}$$

With velocity network $v_\theta(x, t)$ the sampler draws

$$x_{t+1} = \mu_{\theta,t}(x_t) + s_t\,\varepsilon, \quad \mu_{\theta,t}(x) = \gamma_t(x) - \beta_t\, dt\, v_\theta(x, t), \quad s_t = \sigma_t\sqrt{dt}, \quad \varepsilon \sim \mathcal{N}(0, I_d), \tag{19}$$

i.e. the per-step policy is $\pi_{\theta,t}(\cdot \mid x_t) = \mathcal{N}(\mu_{\theta,t}(x_t), s_t^2 I_d)$. The two frameworks differ only in $\sigma_t$:

$$\text{Flow-GRPO: } \sigma_t = a\sqrt{\tfrac{\tau_t}{1-\tau_t}}, \qquad \text{DanceGRPO: } \sigma_t = a. \tag{20}$$

*Remark* 1. Eq. (19) is the Euler–Maruyama step of the marginal-preserving SDE of both frameworks written in one form, and substituting the DanceGRPO score term expressed through the rectified-flow velocity yields the same expression, so the two published samplers are one formula with two noise schedules. The $t = 0$ convention of Eq. (20) is part of the definition, since the substitution $\tau_{\max}$ for $\tau_0$ enters $\sigma_0$ only, while $\beta_0$ and $\gamma_0$ are evaluated at $\tau_0 = 1$, so $\beta_0 = 1$. The log-probability is taken as a *per-dimension mean*, $\ell_t = L_t/d$ for that quantity and $L_t$ for the total log-ratio.

**Definition 2** (Displacement and per-step KL)**.** For fixed $x_t$ let $\Delta\mu_t = \mu_{\text{old},t}(x_t) - \mu_{\theta,t}(x_t)$ and $\Delta v_t = v_{\text{old}}(x_t, t) - v_\theta(x_t, t)$, so that $\Delta\mu_t = -\beta_t\, dt \Delta v_t$. The per-state KL divergence of the sampling policy from the current one is

$$\kappa_t = D_{\text{KL}}\big(\pi_{\text{old},t} \,\|\, \pi_{\theta,t}\big) = \frac{\|\Delta\mu_t\|^2}{2s_t^2}, \qquad k_t = \kappa_t/d. \tag{21}$$

**Assumption 1** (Near on-policy regime)**.** *Every update within an epoch is taken at a policy whose per-dimension log-ratio against the sampling policy satisfies $|\tilde{u}_t| \ll 1$ (Definition 3), equivalently a single pass over the sampled batch per epoch with a small number of optimizer steps, and likewise $|\hat{u}_t| = s_t|\tilde{u}_t| \ll 1$. Statements marked "first-order" below use $e^u = 1 + u + O(u^2)$ for $u \in \{\tilde{u}_t, \hat{u}_t\}$ and for their difference $(s_t - 1)\tilde{u}_t = \hat{u}_t - \tilde{u}_t$, which is then small as well.*

## B.2 EXACT DECOMPOSITION OF THE LOG-RATIO

**Theorem 1** (Three-term decomposition)**.** *Let $x_{t+1} = \mu_{\text{old},t} + s_t\varepsilon$ be drawn from the sampling policy and $L_t = \log \pi_{\theta,t}(x_{t+1} \mid x_t) - \log \pi_{\text{old},t}(x_{t+1} \mid x_t)$. Then, with $z = \Delta\mu_t \cdot \varepsilon/\|\Delta\mu_t\| \sim \mathcal{N}(0, 1)$,*

$$L_t \;=\; -\kappa_t - \sqrt{2\kappa_t}\, z. \tag{22}$$

*Consequently, (i) $\mathbb{E}[e^{L_t}] = 1$; (ii) $\mathbb{E}[L_t] = -\kappa_t$ and $\text{Var}[L_t] = 2\kappa_t$; (iii) $\mathbb{E}[L_t] = -1/2\,\text{Var}[L_t]$.*

*Proof.* Both policies share the covariance $s_t^2 I_d$, so $L_t = \big(\|x_{t+1} - \mu_{\text{old}}\|^2 - \|x_{t+1} - \mu_\theta\|^2\big)/(2s_t^2)$. Substituting $x_{t+1} - \mu_{\text{old}} = s_t\varepsilon$ and $x_{t+1} - \mu_\theta = \Delta\mu_t + s_t\varepsilon$ gives $L_t = -\big(\|\Delta\mu_t\|^2 + 2s_t\Delta\mu_t \cdot \varepsilon\big)/(2s_t^2) = -\kappa_t - \|\Delta\mu_t\|\,(\Delta\mu_t \cdot \varepsilon/\|\Delta\mu_t\|)/s_t$, and $\|\Delta\mu_t\|/s_t = \sqrt{2\kappa_t}$. Since $\varepsilon$ is isotropic, $z \sim \mathcal{N}(0, 1)$, which gives (ii) and (iii). (i) is the likelihood-ratio identity $\mathbb{E}_{\pi_{\text{old}}}[\pi_\theta/\pi_{\text{old}}] = 1$, or directly $\mathbb{E}[e^{-\kappa - \sqrt{2\kappa}z}] = e^{-\kappa+\kappa} = 1$. □

*Remark* 2 (What the decomposition says and does not say). Two levels must be kept apart. At the level of the total log-ratio $L_t$, the mean shift $-\kappa_t$ is the Jensen gap forced by (i), so the true ratio $e^{L_t}$ has expectation exactly 1, its median is $e^{-\kappa_t} < 1$ and $\mathbb{P}(e^{L_t} < 1) = \Phi(\sqrt{\kappa_t/2}) > 1/2$. Removing the shift from $L_t$ would break $\mathbb{E}[e^L] = 1$, so at this level "the ratio is biased below one" describes skew, not bias. The objective, however, does not exponentiate $L_t$ but the per-dimension mean $\ell_t = L_t/d$, and the resulting quantity $e^{\ell_t} = (e^{L_t})^{1/d}$ is not protected by the likelihood-ratio identity (Corollary 1). The quantity that varies across steps by orders of magnitude is the variance $2\kappa_t$. Nothing in Eq. (22) concerns the learning dynamics of $\kappa_t$.

**Corollary 1** (The per-dimension ratio, exact offset and recentering constant)**.** *Conditionally on $x_t$, $\ell_t \sim \mathcal{N}(-k_t,\, 2k_t/d)$, and for an additive constant $c$*

$$\mathbb{E}\big[e^{\ell_t + c}\big] = \exp\Big(c - k_t\big(1 - \tfrac{1}{d}\big)\Big), \qquad \mathbb{E}\big[\ell_t + c\big] = c - k_t. \tag{23}$$

*Hence $\mathbb{E}[e^{\ell_t+c}] = 1$ iff $c = k_t(1 - 1/d)$, and $\mathbb{E}[\ell_t + c] = 0$ iff $c = k_t$. At $c = 0$ the mean of the ratio is $e^{-k_t(1-1/d)} < 1$.*

*Proof.* $\ell_t = L_t/d$ with Eq. (22) gives the law, and the moment-generating function of a Gaussian gives $\mathbb{E}[e^{\ell_t+c}] = \exp(c - k_t + \frac{1}{2} \cdot 2k_t/d)$. Setting it to one gives the first constant, and the mean gives the second. □

*Remark* 3 (The constant the objective needs, and when the offset matters). The objective compares the ratio with the window $[1 - \epsilon_c, 1 + \epsilon_c]$, which in the log variable is $[\log(1 - \epsilon_c), \log(1 + \epsilon_c)]$, symmetric about 0 up to $O(\epsilon_c^2)$, and the ratio enters nowhere else except through the factor $e^u = 1 + O(u)$ of Assumption 1. The constant the objective needs is therefore the one that centres the log variable, $c = k_t$, which is GRPO-Guard's. The constant $k_t(1 - 1/d)$ that would set the ratio's mean to 1 differs from it by $k_t/d$, a shift of $\sqrt{k_t/(2d)}$ standard deviations of $\tilde{u}_t$ with no effect on any clip decision or gradient. What decides whether the offset itself is visible is its size against the window. Since it equals $k_t$ in the units of a window on $\ell_t$ and $s_t k_t$ on $\hat{u}_t$, a step whose offset exceeds the half-width has its whole batch clipped on one side irrespective of $\varepsilon$.

*Remark* 4 (Scope: a surrogate, not an importance ratio). $e^{\ell_t} = (e^{L_t})^{1/d}$ is not a valid importance weight, and neither is $\hat{r}_t = e^{\hat{u}_t}$. The objective analysed in this appendix is the implemented surrogate, in which the ratio enters only through a clipped factor $e^u$ with $|u| \ll 1$ (Assumption 1). No claim of unbiased importance sampling is made anywhere below, and the mean-shift observation that motivates GRPO-Guard is, by Eq. (23), a property of the $1/d$ power, not a violation of $\mathbb{E}[e^{L_t}] = 1$.

**Corollary 2** (State spread of the per-step KL)**.** *For the raw log-ratio $L_t$ at a single $\theta$, averaging over the states $x_t$ of a batch,*

$$\mathbb{E}[L_t] = -\mathbb{E}_x[\kappa_t], \qquad \mathrm{Var}[L_t] = 2\,\mathbb{E}_x[\kappa_t] + \mathrm{Var}_x[\kappa_t], \tag{24}$$

*so that*

$$\tilde{R}_t \;\equiv\; \frac{\mathrm{Var}_x[\kappa_t]}{(\mathbb{E}_x[\kappa_t])^2} \;=\; \frac{\mathrm{Var}[L_t] + 2\,\mathbb{E}[L_t]}{\mathbb{E}[L_t]^2} \;\geq\; 0, \tag{25}$$

*with equality iff $\kappa_t$ is constant across states. In the per-dimension units of the objective the same bound reads $\mathrm{Var}[\ell_t]/(-2\,\mathbb{E}[\ell_t]) \geq 1/d$. $\tilde{R}_t$ is invariant under $\kappa_t \mapsto c\kappa_t$ and under the choice of $d$.*

*Proof.* The laws of total expectation and total variance applied to Eq. (22) give $\mathbb{E}[L] = \mathbb{E}_x[\mathbb{E}[L \mid x]] = -\mathbb{E}_x\kappa$ and $\mathrm{Var}[L] = \mathbb{E}_x[\mathrm{Var}[L \mid x]] + \mathrm{Var}_x[\mathbb{E}[L \mid x]] = 2\mathbb{E}_x\kappa + \mathrm{Var}_x\kappa$. Subtracting twice the first from the second gives the numerator of Eq. (25), a variance, hence non-negative and zero iff $\kappa_t$ is a.s. constant. Dividing the two identities by $d$ and $d^2$ converts them to the per-dimension units, whence $\mathrm{Var}[\ell_t]/(-2\,\mathbb{E}[\ell_t]) = \big(1 + \mathrm{Var}_x\kappa_t/(2\mathbb{E}_x\kappa_t)\big)/d \geq 1/d$. □

### B.3 The schedule factor

**Proposition 1** (Schedule form of the per-step KL)**.** *For fixed $x_t$,*

$$\kappa_t \;=\; F_t\,\|\Delta v_t\|^2, \qquad F_t \;=\; \frac{\beta_t^2\,dt}{2\sigma_t^2}. \tag{26}$$

*Under Flow-GRPO, $\beta_t = \bar{\beta} \equiv 1 + a^2/2$ for every $t \geq 1$ and $\beta_0 = 1$. Under DanceGRPO, $\beta_t = 1 + a^2(1-\tau_t)/(2\tau_t)$ varies with t.*

*Proof.* From Definition 2, $\|\Delta\mu_t\|^2 = \beta_t^2(dt)^2\|\Delta v_t\|^2$ and $s_t^2 = \sigma_t^2\, dt$; substitute into Eq. (21). For Flow-GRPO insert $\sigma_t^2 = a^2\tau_t/(1-\tau_t)$ into Eq. (18): $\beta_t = 1+a^2\tau_t(1-\tau_t)/(2\tau_t(1-\tau_t)) = 1+a^2/2$. At $t = 0$, by the convention of Definition 1, $\beta_0$ is evaluated at $\tau_0 = 1$, where the factor $(1-\tau_0)$ in Eq. (18) vanishes, so $\beta_0 = 1$ exactly while $\sigma_0 = a/\sqrt{1-\tau_1}$ is finite. □

*Remark* 5 (Three consequences). (a) At a fixed realized displacement $\Delta\mu_t$, $\kappa_t = \|\Delta\mu_t\|^2/(2\sigma_t^2 dt)$ scales as $a^{-2}$ under both schedules of Eq. (20), since the noise level enters the per-step KL only through $\sigma_t^2$. (b) The functional form of $F_t$ in $t$ is fixed by the scheduler alone (through $\tau_t$, $dt$), so a change of scheduler shift changes $F_t$ in a way computable before training. (c) $F_t$ separates a *schedule* factor, exact and precomputable, from a *dynamics* factor $\|\Delta v_t\|^2$, about which the theory makes no claim. Suppose $\kappa_t$ and $F_t$ attain their extremes at the same pair of steps $(t_{\min}, t_{\max})$, as they do when both are monotone in $t$. Then the span $\kappa_{t_{\max}}/\kappa_{t_{\min}}$ is the span of $F_t$ multiplied by $\|\Delta v_{t_{\max}}\|^2/\|\Delta v_{t_{\min}}\|^2$, so the two spans agree only when $\|\Delta v_t\|$ takes the same value at both extremal steps.

### B.4 Classification of clipped variables

**Definition 3** (Debiased and RatioNorm variables)**.** The debiased per-dimension log-ratio and the RatioNorm variable are

$$\tilde{u}_t \;=\; \ell_t + k_t \;=\; \frac{L_t + \kappa_t}{d}, \qquad \hat{u}_t \;=\; s_t\,\tilde{u}_t. \tag{27}$$

$\tilde{u}_t$ is GRPO-Guard's "mean-revised" variable with the per-state constant $k_{t,x}$ (the released implementation adds the batch mean of $k_{t,x}$ instead, while the analysis uses the per-state constant), and $\hat{u}_t = \log \hat{r}_t(\theta)$ is its normalized ratio, the quantity that the GRPO-Guard objective exponentiates.

**Proposition 2** (Three geometries)**.** *Conditionally on $x_t$, with $z$ as in Theorem 1,*

$$\textit{(A) debias only} \quad \tilde{u}_t = -\frac{\sqrt{2\kappa_t}}{d}\, z, \qquad \mathrm{Var}[\tilde{u}_t] = \frac{2\kappa_t}{d^2}, \tag{28}$$

$$\textit{(B) RatioNorm} \quad \hat{u}_t = -\frac{\Delta\mu_t \cdot \varepsilon}{d} = -\frac{\|\Delta\mu_t\|}{d}\, z, \qquad \mathrm{Var}[\hat{u}_t] = \frac{\|\Delta\mu_t\|^2}{d^2}, \tag{29}$$

$$\textit{(C) full whitening} \quad \tilde{u}_t/\mathrm{std}(\tilde{u}_t) = -z, \qquad \mathrm{Var} = 1. \tag{30}$$

*A fixed threshold on (A) gates the realized quantity $\kappa_t z^2$, one on (B) gates the Euclidean displacement $\|\Delta\mu_t\|\,|z|$, and (C) removes the factor $\sqrt{2\kappa_t}$ altogether. Rules (A) and (B) generate the same family of clip sets and differ by a step-dependent rescaling of the threshold, and they define the same rule up to a global relabelling of $\epsilon_c$ iff $s_t$ does not depend on t.*

*Proof.* Divide Eq. (22) by $d$ and add $k_t$ to get Eq. (28). Multiplying by $s_t$ and using $s_t\sqrt{2\kappa_t} = \|\Delta\mu_t\|$ gives Eq. (29), and Eq. (30) is immediate. The gated quantities follow by squaring, the event $|\tilde{u}_t| \leq \epsilon_c$ being $\kappa_t z^2 \leq d^2\epsilon_c^2/2$ and $|\hat{u}_t| \leq \epsilon_c$ being $\|\Delta\mu_t\|\,|z| \leq d\,\epsilon_c$. Since $|\hat{u}_t| \leq \epsilon_c$ is $|\tilde{u}_t| \leq \epsilon_c/s_t$, the two rules share the family of clip sets, and the rescaling $\epsilon_c/s_t$ is a global relabelling iff $s_t$ is constant in $t$. □

*Remark* 6 (Two senses of "inert"). Proposition 2 is a statement about *which samples are clipped*. The clip set of (A) is a per-sample trust region in the KL of the policy pair, that is in the Fisher metric of the mean parameter $\mathcal{I} = I_d/s_t^2$, under which the KL is exactly the quadratic form Eq. (21). It bounds $\kappa_t z^2$ and not $\kappa_t$ itself, which is a property of the state and not of the sample. The clip set of (B) is Euclidean, and the difference is not one of units. As a functional of the two policies, $\kappa_t$ is invariant under a bijective reparametrisation of the state applied to both policies, whereas $\|\Delta\mu_t\|$ is coordinate-dependent. The expression Eq. (21) itself holds in the Gaussian coordinates. The factor $s_t$ is therefore not inert for which samples are clipped (Corollary 3), while for the net gradient multiplier of unclipped samples it is, which is the content of Theorem 2. The clip-set statement determines the clip fractions of the two rules (Section B.6).

### B.5 Gradient theorem and the net-multiplier identity

**Assumption 2** (The correction term is differentiated)**.** *The mean-revision term $k_t$ is evaluated from $\mu_{\theta,t}$ with gradient flow, i.e. $\Delta\mu_t$ is a function of $\theta$ inside the loss and is not detached. This is a property of the objective, stated here as an assumption because an implementation may choose either.*

**Lemma 1** (Debiasing cancels the displacement term in the gradient)**.** *Under Assumption 2, with $J_t = \partial\mu_{\theta,t}/\partial\theta$ (a $d \times p$ matrix evaluated at $x_t$),*

$$\nabla_\theta \ell_t = \frac{(\Delta\mu_t + s_t\varepsilon)^\top J_t}{d\, s_t^2}, \quad \nabla_\theta k_t = -\frac{\Delta\mu_t^\top J_t}{d\, s_t^2}, \quad \nabla_\theta \tilde{u}_t = \frac{\varepsilon^\top J_t}{d\, s_t}. \tag{31}$$

*In terms of the velocity network, $J_t = -\beta_t\, dt\, \partial v_\theta/\partial\theta$, hence $\nabla_\theta \tilde{u}_t = -\frac{\beta_t\sqrt{dt}}{d\,\sigma_t}\,\varepsilon^\top \frac{\partial v_\theta}{\partial\theta}$.*

*Proof.* $\ell_t = -\|x_{t+1} - \mu_\theta\|^2/(2ds_t^2) + \text{const}$ and $x_{t+1} - \mu_\theta = \Delta\mu_t + s_t\varepsilon$ give the first identity; $k_t = \|\Delta\mu_t\|^2/(2ds_t^2)$ with $\nabla_\theta\Delta\mu_t = -J_t$ gives the second; adding cancels the $\Delta\mu_t$ term. The last form uses Eq. (19) and $s_t = \sigma_t\sqrt{dt}$. □

*Remark* 7 (If the correction were detached). Without Assumption 2, $\nabla_\theta k_t = 0$ and $\nabla_\theta\tilde{u}_t = \nabla_\theta\ell_t = (\Delta\mu_t + s_t\varepsilon)^\top J_t/(ds_t^2)$ retains the displacement term. The identity Eq. (32) below is unaffected, because it compares two variants through whichever $\nabla_\theta\tilde{u}_t$ they share. What fails is the reduction of the common per-step weight to $\beta_t$ (Theorem 2, second sentence) and with it the reading of GRPO-Guard's Table 2 (Remark 9), because the surviving term $\Delta\mu_t^\top J_t/(ds_t^2)$ has relative magnitude $\|\Delta\mu_t\|/(s_t\|\varepsilon\|) \approx \sqrt{2k_t}$ against the noise term, small under Assumption 1, but systematic in direction where the noise term averages out. An implementation that detaches the correction would therefore see a per-step weight different from the one derived here.

**Definition 4** (Loss-side variants)**.** For advantage $A$, the per-sample policy loss of a variant is $\mathcal{L} = -A\,\delta_t\exp(u)$ on unclipped samples, where $u$ is the variant's clipped variable and $\delta_t$ its per-step loss weight: baseline $(u, \delta_t) = (\ell_t, 1)$ with a fixed window; GRPO-Guard $(\hat{u}_t, 1/dt)$ with a fixed window; F-Clip $(\tilde{u}_t, 1/dt)$ with a fixed window, with the factor $s_t$ removed from the clipped variable; F-Clip$_\sigma$ $(\tilde{u}_t, \sigma_t/\sqrt{dt})$ with a fixed window, F-Clip with the weight that Theorem 2 shows to be guard's; Z-Clip $(\tilde{u}_t, 1/dt)$ with the standardized window of Definition 5; GRPO-Guard on DanceGRPO as published $(\hat{u}_t, \beta_t/dt)$. K-Hold is Z-Clip trained under Actuator C (Definition 10).

**Theorem 2** (Net-multiplier identity)**.** *On any unclipped sample the per-sample gradients of the GRPO-Guard variant and F-Clip$_\sigma$ satisfy*

$$\nabla_\theta\mathcal{L}_{\text{guard}} \;=\; e^{(s_t-1)\tilde{u}_t}\,\nabla_\theta\mathcal{L}_{\text{fisher}}, \tag{32}$$

*and for either variant, with $u$ its own log variable,*

$$\nabla_\theta\mathcal{L} \;=\; A\,e^{u}\,\beta_t\,\varepsilon^\top\big(\partial v_\theta/\partial\theta\big)/d. \tag{33}$$

*Under Flow-GRPO the prefix $\beta_t$ is the constant $\bar{\beta}$ for $t \geq 1$ and 1 at $t = 0$.*

*Proof.* For guard, $u = \hat{u}_t = s_t\tilde{u}_t$ and $\delta_t = 1/dt$, so $\nabla\mathcal{L}_{\text{guard}} = -A\,e^{s_t\tilde{u}_t}\,s_t\nabla\tilde{u}_t/dt$, while F-Clip$_\sigma$ gives $\nabla\mathcal{L}_{\text{fisher}} = -A\,e^{\tilde{u}_t}\,\nabla\tilde{u}_t\,\sigma_t/\sqrt{dt}$. The net multiplier acting on $\nabla_\theta\tilde{u}_t$ is $s_t/dt$ for the first and $\sigma_t/\sqrt{dt}$ for the second, and these agree because $s_t = \sigma_t\sqrt{dt}$, leaving the ratio $e^{(s_t-1)\tilde{u}_t}$. Substituting $\nabla_\theta\tilde{u}_t = -\beta_t\sqrt{dt}\,\varepsilon^\top(\partial v_\theta/\partial\theta)/(d\sigma_t)$ from Lemma 1 into either variant cancels the two signs and the factor $\sigma_t/\sqrt{dt}$, which gives Eq. (33). □

*Remark* 8 (What remains after the identity). Under Assumption 1 the factor $e^{(s_t-1)\tilde{u}_t}$ in Eq. (32) is $1 + O(\tilde{u}_t)$, so the two variants have the same per-sample gradient to first order. They differ *only* in which samples are clipped (Proposition 2), and under the PPO $\max(\cdot,\cdot)$ form clipping is a one-sided gate. The metric choice is therefore inert for the gradients of unclipped samples, so F-Clip$_\sigma$ cannot reshape the $\kappa_t$ profile relative to guard through the multiplier, only through the clip set.

*Remark* 9 (GRPO-Guard's Table 2 recovered). The "gradient scale" column of GRPO-Guard's ablation table is Lemma 1 composed with Definition 4, up to the per-dimension factor $1/d$ and sign, where mean-revised $(\tilde{u}_t, 1)$ gives $\beta\sqrt{dt}\,\varepsilon/\sigma_t$, RatioNorm $(\hat{u}_t, 1)$ gives $\beta\,dt\,\varepsilon$, and GRPO-Guard

$(\hat{u}_t, 1/dt)$ gives $\beta\varepsilon$. Their table thus already states, in their own symbols, that the per-step gradient weight of the full method is $\beta$, constant under Flow-GRPO, which is the content of Theorem 2. Eq. (33) fixes the explicit schedule prefix only, and makes no claim about the magnitude of $\varepsilon^\top \partial v_\theta / \partial \theta$ across $t$.

**Lemma 2** (Schedule covariance of the reweighting)**.** *With the per-step gradient weight $\beta_t$ of Theorem 2, GRPO-Guard's $\delta_t = 1/dt$ under Flow-GRPO equalizes the steps $t \geq 1$ up to a global constant and under-weights the first step by $1/\bar{\beta}$. Under DanceGRPO, GRPO-Guard's $\delta_t = \beta_t/dt$ gives the per-step weight $\beta_t^2$, which increases in $t$ from $\beta_0^2 = 1$ as $(1 + a^2(1-\tau_t)/(2\tau_t))^2$. There the weight $1/dt$ gives $\beta_t$, and step equalization requires $\delta_t = 1/(\beta_t\, dt)$.*

*Proof.* Replacing $\delta_t = 1/dt$ by $\beta_t/dt$ in the proof of Theorem 2 makes the net multiplier $\beta_t\sigma_t/\sqrt{dt}$, whose product with the factor of Lemma 1 is $\beta_t^2$, and $\delta_t = 1/(\beta_t dt)$ makes the product one. The monotonicity is Eq. (18) with $\sigma_t = a$ and $\tau_t$ decreasing. □

### B.6 Clip fractions: fixed thresholds versus a standardized threshold

Throughout this section the *clip fraction* $\varphi_t$ is the two-sided exceedance fraction $\mathbb{P}(|e^u - 1| > \epsilon_c)$, and $z^\star$ is the threshold rewritten in the $z$ units of Theorem 1, which is always possible because every clipped variable of Proposition 2 is proportional to $z$. A fixed $\epsilon_c$ on $\hat{u}_t$ gives $z^\star = d\epsilon_c/\|\Delta\mu_t\|$ (Proposition 3), whereas for the standardized rule $z^\star$ is the parameter itself (Definition 5), and at a fixed state $\varphi_t = 2(1 - \Phi(z^\star))$. Under the PPO objective $\max(-A\, e^u, -A\,\mathrm{clip}(e^u))$ a sample loses its gradient only when it exceeds on the side favoured by its advantage, so the zero-gradient fraction is $\mathbb{P}(A > 0,\, z < -z^\star) + \mathbb{P}(A \leq 0,\, z > z^\star)$, counting the samples with $A = 0$ that carry no gradient in any case. It equals $\varphi_t/2$ when $A$ is independent of $z$, and in general its relation to $\varphi_t$ follows from one structural premise, used again in Corollary 4.

**Assumption 3** (Negative quadrant dependence of advantage and noise)**.** *At each state, the advantage $A$ and the noise coordinate $z = \varepsilon \cdot \widehat{\Delta\mu_t}$ satisfy $\mathbb{P}(A \leq a,\, z \leq b) \leq \mathbb{P}(A \leq a)\,\mathbb{P}(z \leq b)$ for all $a, b$, that is, samples displaced toward the current policy ($z < 0$, ratio above one) are favoured by the advantage at least as often as chance ($\mathbb{P}(A > 0 \mid z \leq 0) \geq \mathbb{P}(A > 0)$), as when the policy has moved toward where the advantage is.*

**Lemma 3** (Two consequences of Assumption 3)**.** *(i) The zero-gradient fraction of the PPO gate is at least $\varphi_t/2$; (ii) $\mathbb{E}[Az] = \mathrm{Cov}(A, z) \leq 0$, exactly and without any condition on $\mathbb{E}[A]$.*

*Proof.* (i) The inequality of Assumption 3 is equivalent to $\mathbb{P}(A > a,\, z \leq b) \geq \mathbb{P}(A > a)\mathbb{P}(z \leq b)$ and to $\mathbb{P}(A \leq a,\, z > b) \geq \mathbb{P}(A \leq a)\mathbb{P}(z > b)$; with $a = 0$ and $b = \mp z^\star$, and $\mathbb{P}(z \leq -z^\star) = \mathbb{P}(z > z^\star) = \varphi_t/2$ for $z \sim \mathcal{N}(0,1)$, the two gated probabilities sum to at least $[\mathbb{P}(A > 0) + \mathbb{P}(A \leq 0)]\varphi_t/2 = \varphi_t/2$. (ii) Hoeffding's covariance identity $\mathrm{Cov}(A, z) = \iint [\mathbb{P}(A \leq a, z \leq b) - \mathbb{P}(A \leq a)\mathbb{P}(z \leq b)]\, da\, db \leq 0$ (finite second moments: $A$ is clamped, $z$ Gaussian), and $\mathbb{E}[Az] = \mathrm{Cov}(A, z) + \mathbb{E}[A]\,\mathbb{E}[z]$ with $\mathbb{E}[z] = 0$. □

**Proposition 3** (Clip fraction of a fixed threshold)**.** *Conditionally on $x_t$ and neglecting the $O(\epsilon_c^2)$ difference between clipping $e^u$ and clipping $u$, a threshold $\epsilon_c$ on the RatioNorm variable clips a fraction*

$$\varphi_t(\epsilon_c) = 2\Big(1 - \Phi\big(d\,\epsilon_c/\|\Delta\mu_t\|\big)\Big) \tag{34}$$

*of the samples, and on the debiased variable a fraction $2(1 - \Phi(d\epsilon_c/\sqrt{2\kappa_t}))$. Both depend on the realized displacement scale.*

*Proof.* By Eq. (29), $\hat{u}_t \sim \mathcal{N}(0, \|\Delta\mu_t\|^2/d^2)$, so the clip event $|\hat{u}_t| > \epsilon_c$ has probability $2(1 - \Phi(d\epsilon_c/\|\Delta\mu_t\|))$. The debiased case uses Eq. (28). □

**Definition 5** (Standardized (z-score) threshold)**.** Given a batch $B$ at step $t$, let $\hat{s}_t = \mathrm{std}_B(\tilde{u}_t)$ and clip at $\epsilon_t = z^\star \hat{s}_t$ for a fixed $z^\star > 0$.

*Remark* 10 (The batch variance of $\tilde{u}_t$ is the batch mean of $\ell_t$). Since $\tilde{u}_t \mid x_t \sim \mathcal{N}(0, 2\kappa_t/d^2)$ has conditional mean zero, the law of total variance gives, exactly and for any spread of $\kappa_t$ across states,

$$\mathrm{Var}_B[\tilde{u}_t] = \frac{2\,\mathbb{E}_x[\kappa_t]}{d^2} = \frac{-2\,\mathbb{E}_B[\ell_t]}{d}, \tag{35}$$

the second equality by Corollary 2. Three consequences. First, the standardized threshold has the closed form $\epsilon_t = z^\star\sqrt{\max(-2\,\mathbb{E}_B[\ell_t], 0)/d}$, which replaces the sample second moment of $\tilde{u}_t$ by the sample first moment of $\ell_t$. It is still a batch estimate (both forms carry an $O(1/n)$ correction as in Proposition 4, with different constants), with lower finite-$n$ variance only where $\kappa_t$ is above about one nat (relative standard errors $\sqrt{2/(n\kappa_t)}$ against $\sqrt{2(1+\tilde{R}_t)/(n-1)}$, crossing at $\kappa_t = (n-1)/(n(1+\tilde{R}_t))$), and the max guards the epoch start, where $\kappa_t \equiv 0$ and the sample mean can be negative by noise. Second, Eq. (35) is an equality between two batch statistics of the per-state variable $\tilde{u}_t = \ell_t + k_{t,x}$ that must hold in any correct implementation of the sampler and of the per-dimension log-probability (a batch-mean constant leaves $\mathrm{Var}_B$ at $\mathrm{Var}_B[\ell_t]$ and does not satisfy it), a sharper self-check than the non-negativity of $\mathrm{Var}[L_t] + 2\mathbb{E}[L_t]$, because the cross-state term $\mathrm{Var}_x\kappa_t$ enters $\mathrm{Var}_B[\ell_t]$ but not $\mathrm{Var}_B[\tilde{u}_t]$. Third, that very difference estimates Eq. (25) from the same statistics,

$$\tilde{R}_t = \frac{\mathrm{Var}_B[\ell_t] - \mathrm{Var}_B[\tilde{u}_t]}{\mathbb{E}_B[\ell_t]^2}, \tag{36}$$

since $\mathrm{Var}_B[\ell_t] - \mathrm{Var}_B[\tilde{u}_t] = \mathrm{Var}_x\kappa_t/d^2$ and $\mathbb{E}_B[\ell_t]^2 = (\mathbb{E}_x\kappa_t)^2/d^2$.

**Proposition 4** (Invariance of the standardized threshold)**.** *Under Definition 5, in the large-$n$ limit, the following hold.*

- *(i) If $\tilde{R}_t = 0$, then $\varphi_t = 2(1-\Phi(z^\star))$ for every $t$, and a target fraction $\varphi$ is reached by $z^\star = \Phi^{-1}(1-\varphi/2)$.*
- *(ii) In general, $\varphi_t$ departs from $2(1-\Phi(z^\star))$ by a functional of the law of $w = \kappa_{t,x}/\mathbb{E}_x\kappa_t$ alone, linear in $\tilde{R}_t = \mathrm{Var}[w]$ at leading order with the coefficient of Remark 11.*

*At fixed relative spread, $\varphi_t$ is independent of $\kappa_t$, $\|\Delta\mu_t\|$ and $s_t$.*

*Proof.* For (i), $\tilde{R}_t = 0$ makes $\kappa_{t,x}$ a.s. constant, so $\tilde{u}_t$ is Gaussian within the batch with standard deviation $\hat{s}_t$ and the threshold rescales with the variable. For (ii), $\hat{s}_t^2 = 2\mathbb{E}_x\kappa_t/d^2$ by Eq. (35) and $\tilde{u}_t \mid x \sim \mathcal{N}(0, 2\kappa_{t,x}/d^2)$ give $\varphi_t = \mathbb{E}_x\big[2(1-\Phi(z^\star\sqrt{\mathbb{E}_x\kappa_t/\kappa_{t,x}}))\big]$, a functional of the law of $w$, which has $\mathbb{E}[w] = 1$ and $\mathrm{Var}[w] = \tilde{R}_t$, and the expansion is Remark 11. With $\hat{s}_t$ estimated from the same batch of $n$ samples the law of $\tilde{u}_t/\hat{s}_t$ is Student-like and $\varphi_t$ carries a further $O(1/n)$ correction. □

**Corollary 3** (A fixed threshold is a moving standardized threshold)**.** *A fixed $\epsilon_c$ on $\hat{u}_t$ is the standardized rule of Definition 5 with the step- and time-dependent parameter $z_t^\star = d\,\epsilon_c/\|\Delta\mu_t\|$. Consequently the clip fraction (i) is uniform in $t$ only if $\|\Delta\mu_t\|$ is, (ii) drifts during training exactly as $\|\Delta\mu_t\|$ does, in either direction, and (iii) transfers across schedules and backbones only if $d\,\epsilon_c/\|\Delta\mu_t\|$ does. Under Assumption 3 the fraction of samples that keep their gradient through the one-sided gate is at most $1-\varphi_t/2$, with equality when $A$ is independent of $z$, so the non-uniformity of (i) is an implicit per-step loss weight.*

*Proof.* By Eq. (29), $|\hat{u}_t| > \epsilon_c$ is $|z| > d\,\epsilon_c/\|\Delta\mu_t\|$, which is Definition 5 at $z_t^\star$. Items (i)–(iii) follow from the strict monotonicity of $z^\star \mapsto 2(1-\Phi(z^\star))$, with $\Delta\mu_t = -\beta_t\,dt\,\Delta v_t$ of Definition 2 for the $t$-dependence in (i). The surviving fraction is one minus the zero-gradient fraction of Lemma 3(i). □

*Remark* 11 (The sign of the residual). With $w = \kappa_{t,x}/\mathbb{E}_x\kappa_t$ and $f(w) = 2\big(1-\Phi(z^\star w^{-1/2})\big)$, the standardized clip fraction is $\varphi_t = \mathbb{E}_w[f(w)]$, and a second-order expansion about $w = 1$ gives

$$\varphi_t \;=\; 2\big(1-\Phi(z^\star)\big) + \frac{z^\star\phi(z^\star)\,(z^{\star 2}-3)}{4}\,\tilde{R}_t + O\big(\mathbb{E}|w-1|^3\big). \tag{37}$$

The correction changes sign at $z^\star = \sqrt{3}$, that is at a target fraction of $8.3\%$, and involves $\tilde{R}_t$ alone. For a fixed threshold the spread acts through the displacement instead, and $x = \|\Delta\mu_t\| \mapsto 2(1-\Phi(c/x))$ is concave for $c/x < \sqrt{2}$, so by Jensen's inequality the batch clip fraction lies below Eq. (34) evaluated at the mean displacement. The two rules are therefore separable in a single measurement, the residual of the standardized rule tracking $\tilde{R}_t$ with sign set by $z^\star \gtrless \sqrt{3}$ and that of a fixed threshold tracking $\|\Delta\mu_t\|$ with sign set by $c/x \gtrless \sqrt{2}$.

*Proof.* $f'(w) = z^\star\phi(z^\star w^{-1/2})\, w^{-3/2}$ and, using $\phi'(y) = -y\phi(y)$, $f''(w) = z^\star\phi(z^\star w^{-1/2})\big[\frac{1}{2}z^{\star 2}w^{-7/2} - \frac{3}{2}w^{-5/2}\big]$, so $f''(1) = z^\star\phi(z^\star)(z^{\star 2}-3)/2$; Taylor's theorem with $\mathbb{E}[w-1]=0$ and $\mathrm{Var}[w]=\tilde{R}_t$ gives Eq. (37). For the fixed threshold, $h(x) = 2(1-\Phi(c/x))$ has $h''(x) = 2\phi(c/x)\, c\, x^{-3}(c^2/x^2 - 2)$, negative iff $c/x < \sqrt{2}$. □

**Definition 6** (Feedback of a clip rule on the displacement)**.** Fix $x_t$, $J_t$, $s_t$ and the joint law of the advantage $A$ and the noise $\varepsilon$ of a sample. This law is set by the sampling policy $\pi_{\mathrm{old},t}$ and the reward, and does not depend on $\theta$. A clip rule is specified by its unclipped set $U \subset \mathbb{R}^d$ in the noise coordinates, which may depend on the sample's advantage (one-sided gates). Its *first-order expected surviving gradient* is

$$G_t(U) \;=\; \frac{1}{d\, s_t}\, \mathbb{E}\big[\mathbf{1}_U(\varepsilon, A)\, A\, \varepsilon^\top\big]\, J_t, \tag{38}$$

and the rule *feeds back on the displacement* at step $t$ if $G_t$ depends on the magnitude $\|\Delta\mu_t\|$ (dependence on the direction $\widehat{\Delta\mu}_t = \Delta\mu_t/\|\Delta\mu_t\|$, through $z = \varepsilon\cdot\widehat{\Delta\mu}_t$, is present in every rule and is not feedback). The rule retained by the PPO maximum at a threshold $z^\star$ is $U(z^\star) = \{\mathrm{sign}(A)\, z \geq -z^\star\}$, and we write

$$c_t(z^\star) = \mathbb{E}\big[(Az)^-\, \mathbf{1}\{|z| > z^\star\}\big] \;\geq\; 0, \tag{39}$$

$\varepsilon_\perp$ for the component of $\varepsilon$ orthogonal to $\widehat{\Delta\mu}_t$.

**Proposition 5** (Uniformity and enforcement are exclusive for threshold rules)**.** *Under Assumptions 2 and 1, the expected gradient of the clipped policy loss of a variant with loss weight $\delta_t$ is $-\delta_t\, G_t(U)$ up to $O(\tilde{u}_t)$, and*

$$G_t\big(U(z^\star)\big) = G_t(\mathbb{R}^d) + \frac{c_t(z^\star)}{d\, s_t}\, \widehat{\Delta\mu}_t^{\,\top}\, J_t - \frac{1}{d\, s_t}\, \mathbb{E}\big[\mathbf{1}_{U^c} A\, \varepsilon_\perp^\top\big] J_t, \tag{40}$$

*with $c_t$ non-increasing in $z^\star$. Any dependence of $G_t$ on $\|\Delta\mu_t\|$ enters through $U$ and hence through $z^\star$. A threshold rule therefore feeds back in the sense of Definition 6 exactly when its $z^\star$ depends on $\|\Delta\mu_t\|$, and it is uniform exactly when it does not, so no threshold rule is both.*

*Proof.* On an unclipped sample the loss is $-A\,\delta_t e^u$ with $\nabla_\theta e^u = e^u\nabla_\theta u$. By Lemma 1, which uses Assumption 2, $\nabla_\theta\tilde{u}_t = \varepsilon^\top J_t/(ds_t)$ contains no $\Delta\mu_t$, and under Assumption 1 $e^u = 1+O(\tilde{u}_t)$. On a clipped sample the gradient is zero. Taking the expectation over the sample, whose law $(A,\varepsilon)$ is that of the sampling policy and hence $\theta$-independent, gives $-\delta_t G_t(U) + O(\tilde{u}_t)$, and the only remaining dependence on $\theta$, and so on $\|\Delta\mu_t\|$, is the indicator $\mathbf{1}_U$. The retained set is $U(z^\star)$ because by Eqs. (28)–(29) both $\tilde{u}_t$ and $\hat{u}_t$ decrease in $z$, so the ratio exceeds $1+\epsilon_c$ iff $z < -z^\star$, and the PPO maximum discards the sample iff this happens on the side favoured by $A$. Writing $\varepsilon = z\widehat{\Delta\mu}_t + \varepsilon_\perp$ and $G_t(U) = G_t(\mathbb{R}^d) - \mathbb{E}[\mathbf{1}_{U^c}A\varepsilon^\top]J_t/(ds_t)$, on $U^c$ one has $Az = |A|\,\mathrm{sign}(A)z < -|A|z^\star < 0$, so $-\mathbb{E}[\mathbf{1}_{U^c}Az] = c_t(z^\star)$, which is Eq. (40), and $c_t$ is non-increasing in $z^\star$ because the event $\{|z|>z^\star\}$ shrinks with $z^\star$. Since $\|\Delta\mu_t\|$ enters $G_t$ only through $z^\star$, the rule feeds back iff $z^\star$ depends on $\|\Delta\mu_t\|$ and is uniform iff it does not. For the standardized rule $z^\star$ is fixed, so the rule is uniform and does not feed back. For a fixed $\epsilon_c$, $z_t^\star$ decreases as the displacement grows (Corollary 3) and $c_t$ increases with it, so the rule feeds back negatively and is not uniform. □

**Corollary 4** (Force balance: fixed threshold, standardized rule, penalty)**.** *Write each mechanism's coefficient along $J_t^\top\widehat{\Delta\mu}_t$ in the common units of Eq.* (38)*. For the PPO gate it is $\mathbb{E}[Az\,\mathbf{1}_U]/(ds_t)$*

$$\mathbb{E}\big[Az\,\mathbf{1}_{U(z^\star)}\big] = \mathbb{E}[Az] + c_t(z^\star), \tag{41}$$

*increasing continuously from $\mathbb{E}[Az]$ at $z^\star=\infty$ to $\mathbb{E}[(Az)^+]$ at $z^\star=0$, and for the penalty of Definition 9 it is $\lambda\sqrt{2\kappa_t}/s_t$ (Lemma 6), unbounded in the displacement. Since $\mathbb{E}[Az]\leq 0$ (Lemma 3(ii)), the following hold:*

- *(a)* Fixed threshold. *For $\mathbb{E}[Az]<0$ a unique $z^\star_{\mathrm{bal}}$ solves $c_t(z^\star_{\mathrm{bal}}) = -\mathbb{E}[Az]$, and the balance is stable at $\kappa_t^* = d^2\epsilon_c^2/(2s_t^2 z^{\star 2}_{\mathrm{bal}})$, which varies across steps through $s_t$ and $z^\star_{\mathrm{bal},t}$.*
- *(b)* Standardized rule. *The coefficient does not vary with $\|\Delta\mu_t\|$, so there is no equilibrium in the batch-mean displacement. Within batch, (a) applies per state with $\epsilon_c = z^\star\hat{s}_t$, restoring the dispersion of displacements toward the batch level with strength governed by $\tilde{R}_t$.*

*(c)* Penalty. *A set point exists for any bounded drive and $\lambda$ calibrates $\kappa_t$ directly, the ratio of penalty to gate coefficient being $\lambda d\sqrt{2\kappa_t}/c_t$.*

*Proof.* On $U(z^\star)^c = \{\mathrm{sign}(A)z < -z^\star\}$ one has $Az < 0$ and $|z| > z^\star$, and conversely, so $\mathbb{E}[Az\,\mathbf{1}_{U^c}] = -c_t(z^\star)$ and $\mathbb{E}[Az\,\mathbf{1}_U] = \mathbb{E}[Az] - \mathbb{E}[Az\,\mathbf{1}_{U^c}]$ gives Eq. (41); monotonicity and the limits are those of the event $\{|z| > z^\star\}$, and continuity follows from $z$ having a density. The penalty coefficient is $\lambda\nabla_\theta\kappa_t = -\lambda(\sqrt{2\kappa_t}/s_t)J_t^\top\widehat{\Delta\mu}_t$. The inequality $\mathbb{E}[Az] \le 0$ also follows without Assumption 3 when the displacement was produced by ascent on the same signal, since an ascent step $\Delta\theta \propto J_t^\top\mathbb{E}[A\varepsilon]$ gives $\Delta\mu_t \propto -J_tJ_t^\top\mathbb{E}[A\varepsilon]$ with $J_tJ_t^\top$ positive semidefinite, and for several steps under the alignment premise (b) of Proposition 6; that route makes $\mathbb{E}[Az]$ a property of the realized displacement rather than of the sampling law, and (a) then holds only where $\mathbb{E}[Az]$ can be treated as fixed. (a): since $-\mathbb{E}[Az] \le \mathbb{E}[(Az)^-] = c_t(0)$ the equation has a solution, unique by strict monotonicity, and stability is that monotonicity composed with the decreasing map $\|\Delta\mu_t\| \mapsto z_t^\star$. (b) and (c) are immediate; the batch statement is Proposition 3 with $\epsilon_c = z^\star\hat{s}_t$ read per state. The cross term follows from $\varepsilon = z\widehat{\Delta\mu}_t + \varepsilon_\perp$ in Eq. (38). All three statements concern the $\widehat{\Delta\mu}_t$-component alone. Under $\Delta\theta \propto G_t(U)^\top$ the change $\delta\kappa_t$ also contains the cross term $\mathbb{E}[\mathbf{1}_U A\varepsilon_\perp^\top]J_tJ_t^\top\widehat{\Delta\mu}_t$, which is not sign-definite. □

*Remark* 12 (The minimal equivalent form). Z-Clip is Definition 5 applied to the debiased variable of Eq. (28), with the published loss weight $\delta_t = 1/dt$ retained. Debiasing is fixed by Corollary 1 and the threshold rule by Proposition 4, which also fixes the meaning of the remaining parameter, since $2(1-\Phi(z^\star))$ is the clip fraction in every cell, so $z^\star$ is a target fraction in disguise and, unlike $\epsilon_c$, need not be recalibrated across systems (Corollary 5(ii)). For the comparison in the main text we set it by inverting $2(1-\Phi(z^\star)) = \bar{\varphi}$ with $\bar{\varphi}$ the mean clip fraction of the published configuration, so that the two rules are compared at a matched clip fraction rather than at a matched threshold. Two points are stated precisely so as not to overreach. First, Theorem 2 makes $(\hat{u}_t, 1/dt)$ equivalent to $(\tilde{u}_t, \sigma_t/\sqrt{dt})$, not to $(\tilde{u}_t, 1/dt)$, so Z-Clip's net per-step multiplier is $\beta_t/s_t$ rather than $\beta_t$. That difference is retained deliberately as the authors' component. It leaves the profile unchanged under the hypotheses of Proposition 6 (Corollary 5(iii)) and is an empirical question outside them. Second, dropping $s_t$ is justified by Proposition 2, which restores the KL geometry of the clip set, not by Theorem 2.

**Corollary 5** (What removing the factor $s_t$ does and does not change)**.** *Fix the debiased variable, the loss weight $\delta_t = 1/dt$ and a threshold rule, and compare clipping $\hat{u}_t = s_t\tilde{u}_t$ with clipping $\tilde{u}_t$. The removal changes*

*(i) the clip probability at a common fixed threshold from $2(1-\Phi(d\epsilon_c/(s_t\sqrt{2\kappa_t})))$ to $2(1-\Phi(d\epsilon_c/\sqrt{2\kappa_t}))$, a function of $\kappa_t$ alone;*

*(ii) the implied threshold parameter from $z_t^\star = d\epsilon_c/\|\Delta\mu_t\|$ to the fixed $z^\star$ of Definition 5;*

*(iii) the net per-step gradient multiplier from $\beta_t$ to $\beta_t/s_t$, which leaves the profile $\kappa_t/\kappa_{t'}$ unchanged under Assumption 4 with (a) and (b).*

*Proof.* (i) By Proposition 3 read on $\tilde{u}_t$ and on $\hat{u}_t = s_t\tilde{u}_t$, whose standard deviations differ by $s_t$. (ii) is Corollary 3 against Definition 5. (iii) Theorem 2 with $\delta_t = 1/dt$ on $\tilde{u}_t$ gives the multiplier $(1/dt)\cdot\beta_t\sqrt{dt}/\sigma_t = \beta_t/s_t$ by Lemma 1, against $\beta_t$ for guard; under Assumption 4 with (a) and (b), Proposition 6 makes $\kappa_t/\kappa_{t'}$ independent of the per-step weights, and a gradient-scale-invariant optimizer absorbs the remaining global constant. □

*Remark* 13 (Two words for two claims). Item (i) of Corollary 5 makes the clip set schedule-free but not uniform in $t$, and uniformity is Definition 5. Item (ii) is why $\epsilon_c$ must be recalibrated per system while $z^\star$ need not. Item (iii) is the one difference from guard that Z-Clip retains. It is systematic rather than random, so the main text reports the span of $1/s_t$ together with a direct check that the $\kappa_t$ profile does not move. Throughout, *identity* refers to Theorem 2, which makes guard and F-Clip$_\sigma$ agree per sample, whereas *empirical equivalence* refers to agreement of Z-Clip with guard on $\kappa_t$ profiles and clip fractions and on held-out rewards at matched proxy level, which is measured and not derived.

### B.7 Loss-side control of the per-step KL is structurally weak

**Assumption 4** (Shared parameters, first-order update)**.** *The parameters $\theta$ are shared by all steps $t$, and for a realized update $\Delta\theta$ the displacement at step $t$ is $\Delta\mu_t = J_t\Delta\theta$ up to the rounding-induced displacement of Remark 15, which does not scale with $\Delta\theta$. The linearisation is used only for the statement of the profile, and the drift probe measures $\Delta\mu_t$ without it.*

**Lemma 4** ($\kappa_t$ as a quadratic form in the update)**.** *Under Assumption 4, $\kappa_t(\Delta\theta) = \|J_t\Delta\theta\|^2/(2s_t^2)$ is quadratic and homogeneous of degree two, $\kappa_t(\alpha\Delta\theta) = \alpha^2\kappa_t(\Delta\theta)$. Hence any global rescaling of the update rescales all $\kappa_t$ by the same factor and leaves the profile $(\kappa_t/\kappa_{t'})_{t,t'}$ unchanged.*

*Proof.* Immediate from the definition. □

**Proposition 6** (Profile rigidity under row-degenerate, aligned coupling)**.** *Let $g_b$ be the gradient direction contributed by step $b$ and define the coupling matrix $C[b][t] = \|J_t g_b\|/s_t$. Assume*

*(a) row degeneracy: $C[b][t] = c_b\,\rho_t$ with scalars $c_b > 0$ and a profile $\rho_t > 0$;*

*(b) alignment: for each $t$, the vectors $J_t g_b$, $b = 0, \dots, T-1$, are positively collinear.*

*Then, for every update $\Delta\theta = \sum_b \delta_b g_b$ with $\delta_b \geq 0$,*

$$\kappa_t(\Delta\theta) = \tfrac{1}{2}\Big(\sum_b \delta_b c_b\Big)^2 \rho_t^2, \qquad \text{so} \qquad \frac{\kappa_t(\Delta\theta)}{\kappa_{t'}(\Delta\theta)} = \frac{\rho_t^2}{\rho_{t'}^2}, \tag{42}$$

*and a gradient-scale-invariant optimizer leaves $(\sum_b \delta_b c_b)^2$ unchanged under a rescaling of the $\delta_b$.*

*Proof.* $\kappa_t(\Delta\theta) = \|\sum_b \delta_b J_t g_b\|^2/(2s_t^2)$. Under (b) the norm of the sum equals the sum of the norms, so by (a) $\|\sum_b \delta_b J_t g_b\| = \big(\sum_b \delta_b c_b\big)\rho_t s_t$, which gives Eq. (42); the factor $\big(\sum_b \delta_b c_b\big)^2$ carries the whole dependence on $\delta$ and cancels in the ratio. Without (b) the inequality gives the same expression as an upper bound, and the realized profile may lie anywhere below it, so shape rigidity requires (b). An optimizer that is scale-invariant in the gradient, such as Adam, satisfies $\Delta\theta(c\delta) = \Delta\theta(\delta)$, so a common rescaling of the weights does not move the remaining factor either. □

*Remark* 14 (What is and is not claimed). Row degeneracy (a) and alignment (b) are empirical premises about a given system, and the proposition itself does not depend on the data. What it claims is that under Assumption 4 with (a) and (b) the loss side offers no first-order handle on the profile shape, not that no such handle exists. Regularities of the profile shape across systems, if observed, are observations and not consequences of this proposition, which says nothing about where $\rho_t$ comes from.

**Definition 7** (Measurement floor)**.** On a reduced-precision training stack a nonzero parameter update produces, in addition to the displacement $J_t\Delta\theta$, an rounding-induced displacement whose magnitude does not scale with the update. We model the measured per-step KL as

$$\kappa_t^{\mathrm{meas}} = \kappa_t(\Delta\theta) + \kappa_t^{\mathrm{floor}} \;\; (\Delta\theta \neq 0), \qquad \kappa_t^{\mathrm{meas}} = 0 \;\; (\Delta\theta = 0), \tag{43}$$

with $\kappa_t^{\mathrm{floor}} \geq 0$ a constant of the backbone, the numerical format and the step, independent of $\|\Delta\theta\|$.

**Lemma 5** (Transport of the floor across noise schedules)**.** *If the epoch displacement lives in the velocity, $\Delta v_t^{\mathrm{floor}}$ with $\|\Delta v_t^{\mathrm{floor}}\|$ independent of the noise schedule, then on the same timestep grid two schedules with drift coefficients $\beta_t, \beta_t'$ and policy scales $s_t, s_t'$ have floors related by*

$$\kappa_t'^{\mathrm{floor}} = \kappa_t^{\mathrm{floor}} \Big(\frac{\beta_t'}{\beta_t}\Big)^2 \Big(\frac{s_t}{s_t'}\Big)^2. \tag{44}$$

*Proof.* $\kappa_t^{\mathrm{floor}} = \beta_t^2 (dt)^2 \|\Delta v_t^{\mathrm{floor}}\|^2/(2s_t^2)$ by Definition 2; take the ratio at $dt$ and $\|\Delta v_t^{\mathrm{floor}}\|$. □

*Remark* 15 (Consequences for control). The floor is identified on updates scaled by $\alpha \to 0$, where $\kappa_t(\Delta\theta) = O(\alpha^2)$ by Lemma 4 while the floor is $O(1)$, and physically sub-ULP weight changes are quantized into activation changes of fixed size. A quantity built from $\kappa_t^{\mathrm{meas}}$ is therefore informative only where $\kappa_t(\Delta\theta) \gg \kappa_t^{\mathrm{floor}}$ or after subtracting the floor, and a target below $\kappa_t^{\mathrm{floor}}$ cannot be reached by shrinking the update at all, since rescaling $\Delta\theta$ moves only the first term of Eq. (43).

A post-hoc projection onto a KL budget below the floor therefore shrinks the update without moving the observable and training stalls, and the lesson concerns the observable and the target, not Lemma 4. Subtracting the floor does not remove the constraint either, since an attainable target for a controller on $\bar{\kappa} = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}}(\kappa_t^{\mathrm{meas}} - \kappa_t^{\mathrm{floor}})$ must exceed the uncertainty of what was subtracted, of order $\frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}}\kappa_t^{\mathrm{floor}}$ when the floor is known only to a constant factor.

### B.8 Enforcement: actuators on the per-epoch KL

By Corollary 4, under Assumption 3, a fixed threshold does have a stable set point, but it is a Euclidean one that differs across steps, drifts with the schedule and the sampling law and cannot be assigned a KL target, while the standardized rule has none at all. By Proposition 5 the feedback a fixed threshold provides is paid for with uniformity, and Proposition 6 restricts what an actuator may be: under shared parameters only the aggregate scale of the profile is a lever. Three actuators on that scale are defined below, in the order in which they act on the update: after the optimizer (projection), in the loss (penalty), and on the optimizer's step size. Their statements are exact, and whether a given actuator regulates a given system is an empirical question governed by Assumption 5 and decided in the main text.

**Definition 8** (Actuator A: projection of the realized update)**.** Fix a per-dimension budget $b > 0$. After each optimizer step the realized update $\Delta\theta$ is replaced by $\alpha\Delta\theta$ with the largest $\alpha \in [\alpha_{\min}, 1]$ such that $\max_t \kappa_t^{\mathrm{meas}}(\alpha\Delta\theta)/d \le b$ on a probe batch, and $\alpha = \alpha_{\min}$ if no such $\alpha$ exists (which is the case whenever $b < \min_t \kappa_t^{\mathrm{floor}}/d$, the stall of Remark 15).

By Lemma 4, $\kappa_t(\alpha\Delta\theta) = \alpha^2\kappa_t(\Delta\theta)$, so the projection is exact on the floor-free part of the observable and inert on the floor (Remark 15); and because it acts after a per-coordinate optimizer, it is the one actuator whose effect the next optimizer step does not see.

**Definition 9** (Actuator B: adaptive Fisher–KL penalty to the epoch-start policy)**.** Fix a target $\kappa^\star > 0$ (nats), a set of penalized steps $\mathcal{T}_{\mathrm{pen}}$ and control steps $\mathcal{T}_{\mathrm{ctrl}}$, and a floor table $\kappa_t^{\mathrm{floor}}$. The penalized per-sample loss is

$$\mathcal{L}_\lambda = \mathcal{L} + \lambda\,\frac{1}{|\mathcal{T}_{\mathrm{pen}}|}\sum_{t\in\mathcal{T}_{\mathrm{pen}}}\kappa_t(\theta), \qquad \kappa_t(\theta) = \frac{\|\mu_{\theta,t}(x_t) - \mu_{\mathrm{old},t}(x_t)\|^2}{2s_t^2}, \tag{45}$$

with $\mathcal{L}$ the clipped policy loss of Z-Clip. Once per epoch, with $\bar{\kappa}$ the mean over $\mathcal{T}_{\mathrm{ctrl}}$ of the floor-subtracted per-step KL measured after the epoch's last update, $\lambda \leftarrow 2\lambda$ if $\bar{\kappa} > c\,\kappa^\star$, $\lambda \leftarrow \lambda/2$ if $\bar{\kappa} < \kappa^\star/c$, else unchanged, within $[\lambda_{\min}, \lambda_{\max}]$, for a dead-band constant $c > 1$.

This is the penalty form of the trust region (the form PPO proposed and then replaced by clipping), applied to the closed-form KL of Definition 2 rather than to a sampled estimate. It is distinct from the $\beta$-term of Flow-GRPO, which penalizes the KL to the *base* model (an anchor); Eq. (45) penalizes the KL to the *epoch-start* policy (a trust region). Clipping is retained as noise masking (Theorem 1); enforcement lives in the penalty.

**Lemma 6** (The penalty gradient is the natural pull-back)**.** *With $\Delta\mu_t = \mu_{\mathrm{old},t} - \mu_{\theta,t}$,*

$$\nabla_\theta\kappa_t = -\frac{J_t^\top\Delta\mu_t}{s_t^2} = -\frac{\sqrt{2\kappa_t}}{s_t}\,J_t^\top\widehat{\Delta\mu}_t, \tag{46}$$

*which vanishes exactly at $\Delta\mu_t = 0$ and in particular at the epoch start. For an optimizer that is scale-invariant in the gradient, $\lambda$ changes the update, but $\|\Delta\theta\|$ is set by the step size $m$, so $\bar{\kappa}(\lambda) = O(m^2)$ and $\lambda \to \infty$ does not drive $\bar{\kappa} \to 0$.*

*Proof.* Differentiate $\kappa_t = \|\Delta\mu_t\|^2/(2s_t^2)$ with $\nabla_\theta\Delta\mu_t = -J_t$ (Lemma 1), and at the epoch start $\theta = \theta_{\mathrm{old}}$ gives $\Delta\mu_t = 0$. Scale invariance acts on the total gradient and not on its decomposition into two terms, so $\nabla\mathcal{L} + \lambda\nabla\bar{\kappa}$ and $\nabla\mathcal{L} + \lambda'\nabla\bar{\kappa}$ give different updates for $\lambda \neq \lambda'$. Writing that update as $\Delta\theta = -m\,\mathcal{N}(g + \lambda\nabla\bar{\kappa})$ with $\mathcal{N}$ positively homogeneous of degree zero, as $\lambda \to \infty$ the direction tends to $-\mathcal{N}(\nabla\bar{\kappa})$ while $\|\Delta\theta\|$ stays of order $m$, so $\bar{\kappa}$ stays of order $m^2$ by Lemma 4. For an optimizer returning the exact minimiser $\Delta\theta = -(2\lambda H)^{-1}g$ of the penalized quadratic model, $\nabla(g^\top\Delta\theta + \lambda\Delta\theta^\top H\Delta\theta) = 0$ gives $\bar{\kappa} \propto \lambda^{-2}$, unbounded below, and monotonicity in $\lambda$ puts the fixed points of Definition 9 in $[\kappa^\star/c,\, c\,\kappa^\star]$. □

**Definition 10** (Actuator C: global step-size control)**.** Fix a target $\kappa^\star > 0$ (nats), control steps $\mathcal{T}_{\mathrm{ctrl}}$, a floor table $\kappa_t^{\mathrm{floor}}$, a gain $g > 0$, a smoothing constant $\eta \in (0, 1]$ and bounds $0 < m_{\min} < m_{\max}$. The loss is the clipped policy loss of Z-Clip, and the resulting variant is K-Hold. Once per epoch, with $\bar{\kappa}$ the mean over $\mathcal{T}_{\mathrm{ctrl}}$ of the floor-subtracted per-step KL measured after the epoch's last update (cumulative from the epoch start), form the log-smoothed estimate $\log \bar{\kappa}_s \leftarrow (1-\eta) \log \bar{\kappa}_s + \eta \log \bar{\kappa}$ and set

$$m \;\leftarrow\; \mathrm{clip}\Big(m\,(\kappa^\star/\bar{\kappa}_s)^{g/2},\, [m_{\min}, m_{\max}]\Big), \tag{47}$$

where $m$ multiplies the optimizer's learning rate for the next epoch. The target may be absolute or relative, $\kappa^\star = \varrho\,\bar{\kappa}_{\mathrm{ref}}$ with $\bar{\kappa}_{\mathrm{ref}}$ the open-loop level at $m = 1$ over a reference window, and a relative target needs no external number.

**Lemma 7** (Log-gain of the step-size actuator)**.** *Under Assumption 4, for an optimizer whose update is homogeneous of degree one in the learning rate at fixed gradient statistics, $\partial \log \bar{\kappa}/\partial \log m = 2$, so Eq.* (47) *gives*

$$\log \bar{\kappa}' = (1 - g) \log \bar{\kappa} + g \log \kappa^\star \tag{48}$$

*at fixed gradient statistics, so the log-error $\log(\bar{\kappa}/\kappa^\star)$ contracts by the factor $1 - g$ per epoch and vanishes in one epoch at $g = 1$. The profile shape $(\kappa_t/\kappa_{t'})$ is unchanged at every epoch.*

*Proof.* $\Delta\theta(m) = m\,\Delta\theta(1)$ by homogeneity and $\kappa_t(m\Delta\theta) = m^2\kappa_t(\Delta\theta)$ by Lemma 4, which gives the log-gain and, with the floor subtracted before the logarithm, shape invariance. Taking logarithms of Eq. (47), $\log m' = \log m + \frac{g}{2}(\log \kappa^\star - \log \bar{\kappa})$, and substituting $\log \bar{\kappa}' = \log \bar{\kappa} + 2(\log m' - \log m)$ gives Eq. (48). □

*Remark* 16 (Steady-state offset against a drifting open loop). If the open-loop level $\bar{\kappa}_{\mathrm{ol}}$ changes by a factor $e^\nu$ per epoch, Eq. (48) acquires a $+\nu$ term and the steady-state error is $\log(\bar{\kappa}/\kappa^\star) \to \nu/g$, proportional to the drift rate and inverse to the gain. No finite gain removes it, since the rule integrates in $\log m$ while a per-epoch drift is a ramp. Unlike Actuator B, this actuator does not change the gradient seen by the optimizer, and unlike Actuator A, it does not have to be reapplied at every update, because it acts on the step size for the whole epoch.

**Assumption 5** (Conditions under which an actuator is expected to regulate)**.**

- *(A1)* Observable. *On the steps used for the signal $\kappa_t(\Delta\theta)$ dominates $\kappa_t^{\mathrm{floor}}$, or the floor is subtracted and the target exceeds its uncertainty (Remark 15).*
- *(A2)* Attainable. *The set point lies inside the actuator's bounds, $[\alpha_{\min}, 1]$, $[\lambda_{\min}, \lambda_{\max}]$ or $[m_{\min}, m_{\max}]$, and for Actuator B also above the step-size-determined value of Lemma 6.*

*Proposition 6 is assumed throughout, so that regulating the aggregate of the control steps regulates the profile.*

*Remark* 17 (Attribution of failures). The failures separate by assumption. An actuator variable pinned at a bound contradicts (A2). A settled actuator with the signal off target contradicts (A1) if the signal is floor-dominated, and Proposition 6 if the late steps do not follow the early ones. A stalled policy with the signal on target indicates a target below the attainable floor (Remark 15). None of these outcomes bears on Sections B.2–B.7, and whether the per-epoch KL set by $\kappa^\star$ governs held-out quality at matched proxy level is an empirical question outside this appendix.

### B.9 Framework covariance: DanceGRPO

**Proposition 7** (One drift, two schedules)**.** *Definitions 1–2, Theorem 1, Corollary 2, Propositions 1, 2, 3, 4, 5, Corollaries 3, 4 and Lemma 4 hold verbatim under both noise schedules of Eq.* (20)*. Theorem 2 holds under both, but the resulting per-step weight is the constant $\bar{\beta}$ only under Flow-GRPO; under DanceGRPO it is $\beta_t$ (or $\beta_t^2$ for $\delta_t = \beta_t\,dt$, Lemma 2).*

*Proof.* None of the listed statements uses the specific form of $\sigma_t$ beyond $\sigma_t > 0$; only Proposition 1's evaluation of $\beta_t$ and the constancy claim in Theorem 2 do, and those are stated per framework. □

*Remark* 18 (What the covariance predicts). Holding the backbone fixed and changing the noise schedule changes $F_t$ by a factor computable from the scheduler alone (Proposition 1); by Remark 5(c), under its alignment premise, the span of the $\kappa_t$ profile is predicted to change by the

ratio of the two $F_t$ spans, up to the ratio of $\|\Delta v_t\|^2$ at the extremal steps between the two schedules, which is not predicted. The fixed threshold's implied parameter is $z_t^\star = d\epsilon_c/(s_t\sqrt{2\kappa_t})$ (Corollary 3), with $s_t = a\sqrt{dt}$ under DanceGRPO and $a\sqrt{\tau_t\, dt/(1-\tau_t)}$ under Flow-GRPO: the Flow-GRPO scale carries the extra factor $\sqrt{\tau_t/(1-\tau_t)}$, which decreases in $t$, so whenever $\kappa_t$ increases in $t$ the two factors of $s_t\sqrt{2\kappa_t}$ oppose each other under Flow-GRPO and, on grids where $dt$ increases in $t$, reinforce each other under DanceGRPO. Under those conditions the fixed threshold's per-step non-uniformity is predicted to be larger under DanceGRPO, while the standardized rule is unaffected (Proposition 4); the monotonicity of $s_t$ itself on a given grid is a scheduler number, not a consequence.

### B.10 Summary

The table below lists the statements of this appendix in order, with what each gives and what its proof draws on. The third column records direct dependencies only, and corollaries are indented under the statement they follow from.

Theorem 1, Corollaries 2 and 1, Propositions 1, 2, 3, 4 and 5, Corollaries 3, 4 and 5(i)–(ii), Lemmas 1, 2 and 4 are exact under the stated definitions. Corollary 5(iii), Proposition 6 and Lemmas 6 and 7 are conditional on the named premises, which are empirical. No statement in this appendix concerns held-out rewards, and no downstream superiority of any variant is derived here.

| **Statement** | **What it gives** | **Uses** |
|---|---|---|
| *§B.2. Exact decomposition of the log-ratio* | | |
| Thm. 1 | $L_t = -\kappa_t - \sqrt{2\kappa_t}z$ | Def. 1, 2 |
| ↪ Cor. 1 | offset of $e^{\ell_t}$, exact constant $k_t$ | Thm. 1 |
| ↪ Cor. 2 | batch moments, $\tilde{R}_t$ | Thm. 1 |
| *§B.3. The schedule factor* | | |
| Prop. 1 | $\kappa_t = F_t\|\Delta v_t\|^2$ | Def. 1, 2 |
| *§B.4. Classification of clipped variables* | | |
| Prop. 2 | three clip geometries | Thm. 1, Def. 3 |
| *§B.5. Gradient theorem and the net-multiplier identity* | | |
| Lem. 1 | $\nabla_\theta \tilde{u}_t = \varepsilon^\top J_t/(ds_t)$ | Def. 3, Ass. 2 |
| Thm. 2 | guard $\equiv$ F-Clip$_\sigma$; prefix $\beta_t$ | Lem. 1, Def. 4, Ass. 1 |
| ↪ Lem. 2 | per-step weight under each $\delta_t$ | Thm. 2, Prop. 1 |
| *§B.6. Clip fractions* | | |
| Lem. 3 | zero-gradient fraction; $\mathbb{E}[Az] \le 0$ | Ass. 3, Thm. 1 |
| Prop. 3 | clip fraction of a fixed threshold | Prop. 2 |
| Prop. 4 | invariance of the standardized rule | Def. 5, Rmk. 10 |
| ↪ Cor. 3 | fixed threshold is a moving $z^\star$ | Prop. 2, Def. 5, Lem. 3 |
| Prop. 5 | uniformity and feedback exclusive | Def. 6, Lem. 1, Ass. 1 |
| ↪ Cor. 4 | force balance, three mechanisms | Prop. 5, Lem. 3, 6 |
| ↪ Cor. 5 | what removing $s_t$ changes | Prop. 3, 4, Thm. 2, Prop. 6 |
| *§B.7. Loss-side control is structurally weak* | | |
| Lem. 4 | $\kappa_t$ quadratic in $\Delta\theta$ | Ass. 4, Def. 2 |
| Prop. 6 | profile shape is rigid | Lem. 4 |
| Lem. 5 | floor across schedules | Def. 7, 2 |
| *§B.8. Actuators on the per-epoch KL* | | |
| Lem. 6 | pull-back, epoch-start, reach | Lem. 1, 4, Def. 9 |
| Lem. 7 | $\partial \log \bar{\kappa}/\partial \log m = 2$ | Lem. 4, Def. 10 |
| *§B.9. Framework covariance* | | |
| Prop. 7 | covariance across schedules | Def. 1, Prop. 1 |

## C JUSTIFICATION OF THE ASSUMPTIONS

**Assumption 1 (near on-policy).** The assumption requires $|\tilde{u}_t| \ll 1$ for every sample at every update, and enters only through the first-order expansion $e^u = 1 + u + O(u^2)$. Table 3 reports the standard deviation and the largest absolute value of $\tilde{u}_t$ per trained step over all samples of 32 logged epochs on each backbone.

Table 3: Magnitude of the mean-revised log-ratio $\tilde{u}_t$ per trained step over all samples.

| | SD3 | | FLUX | |
|---|---|---|---|---|
| $t$ | $\mathrm{std}(\tilde{u}_t)$ | $\max\|\tilde{u}_t\|$ | $\mathrm{std}(\tilde{u}_t)$ | $\max\|\tilde{u}_t\|$ |
| 0 | $1.7 \times 10^{-6}$ | $1.6 \times 10^{-5}$ | $5.0 \times 10^{-7}$ | $3.4 \times 10^{-6}$ |
| 1 | $2.4 \times 10^{-6}$ | $2.7 \times 10^{-5}$ | $8.8 \times 10^{-7}$ | $8.6 \times 10^{-6}$ |
| 2 | $4.9 \times 10^{-6}$ | $7.1 \times 10^{-5}$ | $2.0 \times 10^{-6}$ | $2.2 \times 10^{-5}$ |
| 3 | $8.2 \times 10^{-6}$ | $1.3 \times 10^{-4}$ | $3.5 \times 10^{-6}$ | $4.8 \times 10^{-5}$ |
| 4 | $1.1 \times 10^{-5}$ | $1.1 \times 10^{-4}$ | $5.1 \times 10^{-6}$ | $4.8 \times 10^{-5}$ |
| 5 | $1.6 \times 10^{-5}$ | $1.6 \times 10^{-4}$ | $6.9 \times 10^{-6}$ | $5.8 \times 10^{-5}$ |
| 6 | $2.3 \times 10^{-5}$ | $1.9 \times 10^{-4}$ | $9.4 \times 10^{-6}$ | $7.1 \times 10^{-5}$ |
| 7 | $3.8 \times 10^{-5}$ | $3.6 \times 10^{-4}$ | $1.4 \times 10^{-5}$ | $8.3 \times 10^{-5}$ |
| 8 | $7.7 \times 10^{-5}$ | $6.7 \times 10^{-4}$ | $2.4 \times 10^{-5}$ | $1.3 \times 10^{-4}$ |

**Assumption 2 (the correction term is differentiated).** In the GRPO-Guard release (`scripts/train_sd3_GRPO_Guard.py`, in line 917) the correction `ratio_mean_bias` is formed from `prev_sample_mean` of the current forward without `.detach()`, and we keep the line verbatim. The same line reduces over every axis including the batch, so the constant added is the batch mean $\bar{k}_t$ rather than the per-state $k_{t,x}$. The analysis uses the per-state constant, and the size of the difference is measured below (Assumption 3, last item).

**Assumption 3 (negative quadrant dependence).** The assumption is a statement about expectations. Its defining inequality reads $\omega_t = \mathbb{E}[\mathbf{1}\{A > 0\} \mid z \le 0] - \mathbb{E}[\mathbf{1}\{A > 0\}] \ge 0$, and it enters the analysis through $\mathbb{E}[Az] \le 0$ (Lemma 3). Both are tested as such, pooled over the samples of each trained step with $z$ recovered per sample as $z = -\tilde{u}_t / \sqrt{2k_{t,x}/d}$. Figure 3 follows $\omega_t$ and $\mathbb{E}[Az]$ over 32 logged epochs on both backbones. Both carry the sign of the assumption throughout at the high-noise and middle steps and are within one standard error of zero at the image-end steps, and no step has a mean of the opposite sign beyond noise on either backbone. The dependence is indirect by construction. The first update of an epoch has ratio one at every sample and is never clipped, and the samples of the second update did not enter the gradient that produced the displacement, so $A$ and $z$ are linked only through the policy having moved toward higher reward across prompts. The assumption therefore holds as an expectation at every trained step.

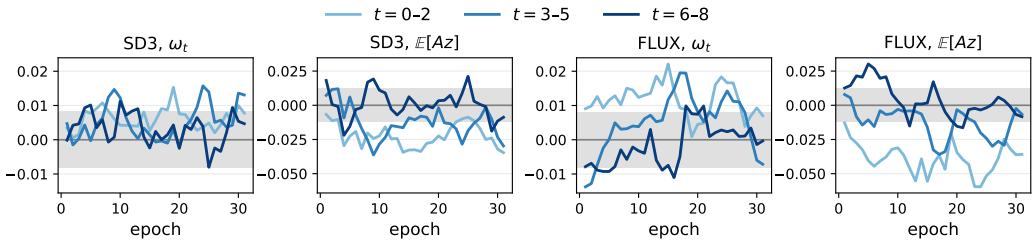


Figure 3: $\omega_t$ and $\mathbb{E}[Az]$ along training on SD3 and FLUX. $t$ is the trained denoising step, $t = 0$ the highest-noise step and $t = 8$ the step nearest the image, grouped into three bands with the samples of each band pooled. The grey band is one standard error of a plotted point about zero.

**Assumption 4 (shared parameters, first-order update).** All trained steps update the same LoRA parameters through one backward pass and one optimizer step, so $\theta$ is shared by construction. The second clause is tested by scaling a realized update. With $\Delta\theta$ the fifth effective update of training, $\kappa_t(\alpha\Delta\theta)$ is measured for $\alpha \in \{1/8, 1/4, 1/2, 1\}$ through the fp32 forward and fitted per step as $a_t\alpha^2 + f_t$. Table 4 reports the linear share $a_t/\kappa_t(\Delta\theta)$ and the constant $f_t$ against the floor of Remark 15, measured independently on the tiny-update probes. The constant reproduces that floor within a factor $1.3$ on SD3 and $2.5$ on FLUX across three to four decades, and the remainder is quadratic in $\alpha$ with residuals below $12\%$ on SD3 and $42\%$ on FLUX. The displacement is therefore the linear response $J_t\Delta\theta$ plus the rounding-induced term, and the assumption holds as stated.

Table 4: Linear share and fitted floor of the per-step KL under scaling of a realized update.

| | SD3 | | FLUX | |
|---|---|---|---|---|
| $t$ | $a_t/\kappa_t$ | $f_t/\kappa_t^{\mathrm{floor}}$ | $a_t/\kappa_t$ | $f_t/\kappa_t^{\mathrm{floor}}$ |
| 0 | 0.76 | 1.24 | 0.78 | 1.30 |
| 1 | 0.75 | 1.21 | 0.90 | 1.25 |
| 2 | 0.72 | 1.12 | 0.97 | 1.28 |
| 3 | 0.74 | 1.30 | 0.97 | 2.46 |
| 4 | 0.74 | 0.95 | 0.98 | 0.96 |
| 5 | 0.70 | 0.86 | 0.94 | 2.16 |
| 6 | 0.64 | 0.82 | 0.90 | 1.78 |
| 7 | 0.62 | 0.78 | 0.84 | 1.74 |
| 8 | 0.60 | 0.81 | 0.73 | 1.57 |

## D EXPERIMENTAL SETTINGS

**Task and backbones.** The task is visual text rendering with the OCR accuracy of the rendered text as the reward, on the prompt set released with Flow-GRPO (Liu et al., 2025a). The backbones are SD3.5-M[4] and FLUX.2-klein-base-4B[5] at $512 \times 512$, fine-tuned through LoRA on the attention projections of the transformer. Both are trained under the Flow-GRPO sampler, and SD3.5-M also under the DanceGRPO sampler (Xue et al., 2025), which differs in the noise scale $\sigma_t$ alone. Table 5 lists the training settings, with the entries that depend on the sampler marked. The KL penalty to the base model is off in every run, since it would suppress the quantity $\kappa_t$ that the paper measures, and the advantage is normalized by the group standard deviation as in GRPO-Guard (Wang et al., 2026).

Table 5: Training parameters.

| | SD3.5-M | FLUX.2-klein |
|---|---|---|
| Noise level $a$ | 0.7 | 0.7 |
| Resolution | 512 | 512 |
| Sampling steps, trained | 10, 9 | 10, 9 |
| Guidance scale | 1.0 | 1.0 |
| Scheduler shift | 3 | empirical $\mu$ |
| Prompts per epoch | 48 | 48 |
| Images per prompt | 24 | 24 |
| Sampling batch | 48 | 24 |
| Updates per epoch | 2 | 2 |
| LoRA $(r, \alpha)$ | (32, 64) | (32, 64) |
| Learning rate | $3 \times 10^{-4}$ | $3 \times 10^{-4}$ |
| Adam $(\beta_1, \beta_2)$ | (0.9, 0.999) | (0.9, 0.999) |
| Weight decay | $10^{-4}$ | $10^{-4}$ |
| Gradient clip | 1.0 | 1.0 |
| Epochs | 240 | 240 |
| Checkpoint interval | 20 | 20 |
| GPU | 1×B200 | 1×B200 |

**Variants.** Table 6 lists, for each variant, the clipped variable $u$ and the per-step weight $\delta_t$ of its loss $\mathcal{L} = -A\,\delta_t e^u$ (Definition 4), and the clip range. Flow-GRPO, DanceGRPO and GRPO-Guard use their own settings, F-Clip the range $\epsilon_c/\bar{s}$ with $\epsilon_c = 2 \times 10^{-6}$, and Z-Clip $z^\star = 0.42$ in every configuration. K-Hold runs Actuator C (Definition 10) with $\mathcal{T}_{\text{ctrl}} = \{0, \dots, 4\}$, $g = 0.75$, $\eta = 0.35$, $[m_{\min}, m_{\max}] = [0.1, 3]$, four uncontrolled epochs, four reference epochs $\varrho\,\bar{\kappa}_{\text{ref}}$ with a relative target $\varrho\,\bar{\kappa}_{\text{ref}}$ at $\varrho = 1$ on both backbones. The floor table is measured per configuration by an 8-epoch run under Actuator A with $b = 10^{-7}$.

Table 6: Per-variant settings. $\tilde{u}_t$ is the debiased log-ratio and $\hat{u}_t = s_t \tilde{u}_t$ the normalized one. Weights marked $^{(1)}$ and $^{(2)}$ hold under the Flow-GRPO and the DanceGRPO sampler.

| Variant | Clipped variable | Clip range | Weight $\delta_t$ |
|---|---|---|---|
| Flow-GRPO | $\ell_t$ | fixed, $10^{-4}$ | 1 |
| DanceGRPO | $\ell_t$ | fixed, $10^{-4}$ | 1 |
| GRPO-Guard | $\hat{u}_t$ | fixed, $2 \times 10^{-6}$ | $1/dt^{(1)}$, $\beta_t/dt^{(2)}$ |
| F-Clip | $\tilde{u}_t$ | fixed, $2 \times 10^{-6}/\bar{s}$ | $1/dt^{(1)}$, $1/dt^{(2)}$ |
| Z-Clip | $\tilde{u}_t$ | $z^\star\,\mathrm{std}_B(\tilde{u}_t)$, $z^\star = 0.42$ | $1/dt^{(1)}$, $1/dt^{(2)}$ |
| K-Hold | $\tilde{u}_t$ | as Z-Clip, Actuator C | $1/dt^{(1)}$, $1/dt^{(2)}$ |

[4] https://huggingface.co/spaces/stabilityai/stable-diffusion-3.5-medium
[5] https://huggingface.co/black-forest-labs/FLUX.2-klein-base-4B

## E RESULTS ON THE DANCEGRPO SAMPLER

All runs of this section use SD3.5-M with the DanceGRPO noise schedule implemented in the Flow-GRPO codebase, and otherwise the protocol of Appendix D. The sampler changes one thing that the main text does not exercise, the per-step loss weight. GRPO-Guard introduces its weight $\delta_t$ to equalize the gradient magnitudes between the denoising steps and sets $\delta_t = \beta_t/dt$ in DanceGRPO. By Lemma 2 this weight does not equalize there. The net per-step weight it produces is $\beta_t^2$, which rises from one at the noisiest step to $\beta_8^2$ at the image end, the weight $1/dt$ gives $\beta_t$, and equalization needs $\delta_t = 1/(\beta_t\, dt)$. The lemma corrects GRPO-Guard's account of what the weight does. However, it does not claim the equalizing weight the right choice, and it says nothing about how any of the three trains, which the table at the end of this section reports as measured.

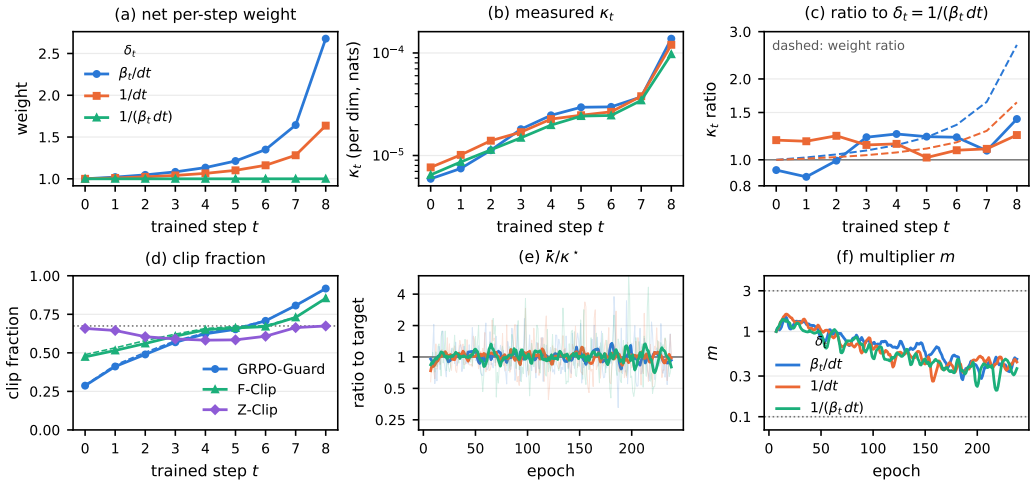


Figure 4: DanceGRPO on SD3.5-M. (a) Net per-step weight of the three $\delta_t$ on the dance grid. (b) Per-step KL of the Z-Clip runs with each weight. (c) Their ratio to the step-equalizing run, with the weight ratios $\beta_t^2$ and $\beta_t$ dashed. (d) Per-step clip fraction with the predicted fraction dashed and the nominal value $2(1-\Phi(z^\star))$ dotted. (e, f) Ratio of the measured to the target per-epoch KL and the multiplier $m$ of K-Hold under each weight, with the bounds dotted.

Table 7: DanceGRPO on SD3.5-M at the last checkpoint, averaged over two seeds. The upper block runs at $\delta_t = 1/dt$. The last column is the mean standardized gain over the base model across the four held-out scores.

| Method | OCR | PickScore | HPSv2 | ImageReward | UnifiedReward | Average |
|---|---|---|---|---|---|---|
| Base | 0.134 | 20.24 | 0.188 | −0.238 | 2.366 | +0.00 |
| DanceGRPO | 0.930 | 19.72 | 0.121 | −1.601 | 1.423 | −1.60 |
| GRPO-Guard | 0.933 | 20.63 | 0.165 | −0.765 | 2.219 | −0.49 |
| F-Clip | **0.957** | 20.66 | 0.172 | −0.698 | 2.207 | −0.41 |
| Z-Clip | 0.935 | 20.73 | 0.182 | −0.454 | 2.320 | −0.16 |
| K-Hold | 0.919 | **21.00** | **0.197** | **−0.039** | **2.576** | **+0.26** |
| Z-Clip, $\delta_t = \beta_t/dt$ | 0.928 | 20.64 | 0.173 | −0.669 | 2.151 | −0.41 |
| Z-Clip, $\delta_t = 1/(\beta_t\, dt)$ | 0.932 | 20.39 | 0.162 | −0.982 | 1.990 | −0.71 |
| K-Hold, $\delta_t = \beta_t/dt$ | 0.935 | 20.85 | 0.182 | −0.363 | 2.371 | −0.10 |
| K-Hold, $\delta_t = 1/(\beta_t\, dt)$ | 0.939 | 20.87 | 0.182 | −0.371 | 2.523 | −0.03 |

Figure 4 collects the readings. Panel (a) is computed from the scheduler alone, and the three weights differ by up to $\beta_t^2$ across the grid. Panel (b) measures three Z-Clip runs that differ in $\delta_t$ and in nothing else, and their profiles share one shape, ordered as Lemma 2 gives and separated by far less. Panel (c) divides the two non-equalized profiles by the equalized one and draws $\beta_t^2$ and $\beta_t$ beside

them, and the measured ratio stays below the weight ratio at every step. Proposition 6 accounts for the gap, since the steps share parameters and a per-step weight changes the composition of one update rather than the size of an update of its own. Two things follow. No $\delta_t$ flattens the profile that Lemma 2 equalizes, and $\delta_t$ still moves that profile, so it remains a design choice, only not for the reason it was introduced. Panels (d) to (f) repeat the second and third layers on this sampler. The fixed windows sweep across the steps and the standardized one does not, as on Flow-GRPO but wider (Remark 18), and K-Hold holds the per-epoch KL at the target under all three weights, since the weight redistributes that KL over the steps and does not decide whether its sum can be held.

Table 7 reads the three weights on held-out reward. The OCR reward is reached under every weight, and the four held-out scores separate them. Under Z-Clip and under K-Hold alike the best of the three is $\delta_t = 1/dt$, and the weight that Lemma 2 singles out as the one equalizing the steps, $\delta_t = 1/(\beta_t\, dt)$, is the worst for Z-Clip and gains nothing for K-Hold. Equalizing the per-step weight is therefore not what the held-out scores reward, which is the same conclusion the profiles gave and is one more reason to read $\delta_t$ as a design choice rather than as a correction.

## F BEHAVIOUR OF THE TARGET RATIO

K-Hold sets its target relative to the run's own open-loop level, $\kappa^\star = \varrho\,\bar{\kappa}_{\mathrm{ref}}$ (Definition 10). Table 8 gives the two readings of Assumption 5 per setting and seed, and Figure 5 further presents the trajectories behind them.

Table 8: Per control setting (seed 1 / seed 2). The (A1) margin is the resolved target over twice the floor total, and the (A2) reading is the fraction of epochs with the multiplier saturated at a bound.

| backbone | ratio | target (nats) | (A1) margin | (A2) epochs saturated |
|---|---|---|---|---|
| SD3 | 0.5× | 0.038/0.026 | 0.59/0.41 | 53%/80% |
| SD3 | 1.0× | 0.061/0.050 | 0.95/0.77 | 5%/34% |
| SD3 | 2.0× | 0.145/0.098 | 2.26/1.52 | 0%/0% |
| FLUX | 0.5× | 0.034/0.024 | 2.25/1.57 | 0%/51% |
| FLUX | 1.0× | 0.049/0.055 | 3.20/3.59 | 0%/0% |
| FLUX | 2.0× | 0.095/0.055 | 6.27/3.59 | 0%/0% |

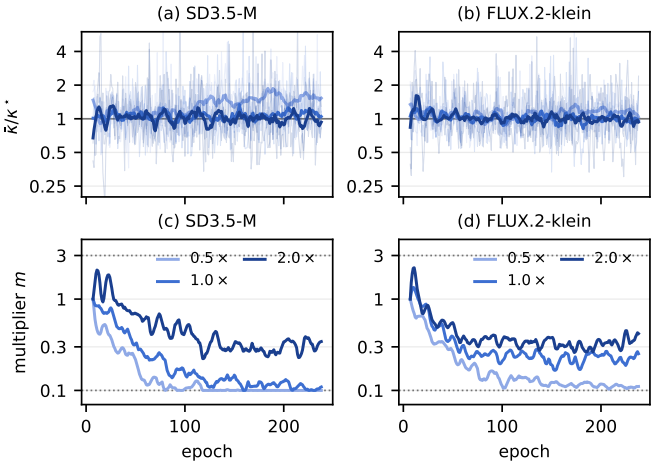


Figure 5: The actuator at every target ratio. Top, the ratio $\bar{\kappa}/\kappa^\star$ per epoch; bottom, the multiplier $m$ with its bounds $0.1$ and $3$.

Table 9: Scores at the last checkpoint against the target ratio, mean over two seeds.

| ratio | OCR | PickScore | HPSv2 | ImageReward | UnifiedReward | Average |
|---|---|---|---|---|---|---|
| SD3.5-M | | | | | | |
| 0.5× | 0.913 | 21.27 | 0.211 | 0.137 | 3.009 | +0.66 |
| 1.0× | 0.959 | 21.11 | 0.204 | 0.013 | 2.835 | +0.47 |
| 2.0× | 0.975 | 20.80 | 0.188 | −0.382 | 2.544 | +0.02 |
| FLUX.2-klein | | | | | | |
| 0.5× | 0.887 | 21.41 | 0.228 | 0.343 | 3.179 | +0.83 |
| 1.0× | 0.912 | 21.17 | 0.213 | 0.122 | 3.024 | +0.52 |
| 2.0× | 0.907 | 20.94 | 0.195 | −0.172 | 2.833 | +0.15 |

Table 9 gives the scores of the same runs at the last checkpoint. The trade-off is monotone in the ratio on both backbones, OCR falling and every gold score rising as $\varrho$ decreases, and continues through the saturated settings. The main text uses $\varrho = 1$, which sits between the two ends of this trade-off on both backbones.

## G ACTUATORS A AND B

The projection of Definition 8 budgets $\kappa_t^{\mathrm{meas}}$, which carries the floor of Remark 15, and no rescaling of the update removes that floor. The penalty of Definition 9 reaches only as far as the step size allows, so $\bar{\kappa}$ stays of order $m^2$ however large $\lambda$ grows (Lemma 6). The runs on SD3.5-M with the settings of the main experiments are consistent with both statements.

**Actuator A.** The projection runs at four per-dimension budgets spanning two decades, and once with the budget on the cumulative displacement of the epoch (Table 10). The scale falls from $1.00$ to $0.02$, which by Lemma 4 divides the floor-free part of the KL by $2500$. The observed KL moves by a factor of $1.5$ at the last step and $4$ at the noisiest, and at the last step it stays one to three decades above what the projected update can produce. The budget acts on a quantity that the floor dominates, and the clip fraction at the last step is the same in all five runs.

Table 10: Actuator A on SD3.5-M, 32 epochs, mean over epochs 4–31, per-dimension quantities. The last row caps the cumulative KL of the epoch rather than each update.

| budget $b$ | $\alpha$ | $\max_t \kappa_t/d$ before projection | $\kappa_0/d$ | $\kappa_8/d$ | clip fraction, $t = 8$ |
|---|---|---|---|---|---|
| $3 \times 10^{-6}$ | 0.02 | 2.2e-04 | 1.7e-08 | 5.8e-05 | 0.44 |
| $10^{-5}$ | 0.05 | 2.6e-04 | 1.8e-08 | 6.3e-05 | 0.44 |
| $3 \times 10^{-5}$ | 0.16 | 2.4e-04 | 2.4e-08 | 6.6e-05 | 0.45 |
| $3 \times 10^{-4}$ | 1.00 | 1.7e-04 | 7.2e-08 | 8.5e-05 | 0.46 |
| $3 \times 10^{-5}$, epoch ref. | 0.16 | 2.4e-04 | 2.3e-08 | 6.4e-05 | 0.45 |

**Actuator B.** The penalty runs at two targets, $0.029$ and $0.057$ nats, with $\lambda$ doubled or halved against a dead band of $1.5$ within $[10^{-2}, 10^{3}]$ (Table 11). The signal at the first epoch is $0.71$ nats, twelve to twenty times either target, and $\lambda$ sits at $\lambda_{\max}$ from the first epoch on, the wind-up that Lemma 6 gives for a target beyond the reach of the step size at that moment. The signal then falls through the target to zero, $\lambda$ returns to $\lambda_{\min}$ within a dozen epochs, and the policy does not move again, with the OCR flat near $0.1$ against $0.3$ to $0.5$ for the open-loop run of the same protocol. The stall is a property of the optimizer state after the wind-up rather than of the penalty, and the run does not reach the regime in which the penalty could be assessed.

Table 11: Actuator B on SD3.5-M, 32 epochs, $\lambda \in [10^{-2}, 10^{3}]$, $\bar{\kappa}$ in nats. The open-loop run of the same protocol has evaluation OCR 0.3–0.5 over the same epochs.

| | | $\lambda$ reaches at epoch | | $\bar{\kappa}$ at epoch | | | eval OCR at epoch | |
|---|---|---|---|---|---|---|---|---|
| $\kappa^\star$ | $\lambda$ at epoch 0 | $\lambda_{\max}$ | $\lambda_{\min}$ | 0 | 10 | last | 8 | 24 |
| 0.029 | $10^3$ | 0 | 24 | 0.71 | 0.05 | 0.000 | 0.087 | 0.082 |
| 0.057 | $10^3$ | 0 | 12 | 0.71 | 0.042 | 0.000 | 0.097 | 0.132 |