# Reward Is Not a Universal Interface for Generative Reinforcement Learning

**Victor Shea-Jay Huang** **Benjin Zhu** **Hongsheng Li**[†]
Multimedia Laboratory (MMLab), The Chinese University of Hong Kong
jeix782@gmail.com hsli@ee.cuhk.edu.hk
[†]Corresponding author.

## Abstract

A reward is not a universal RL interface: it becomes a valid update only through the probability object exposed by the policy branch. Autoregressive post-training works cleanly because token ratios are exact, but parallel discrete policies hide within-step dependence and flow/diffusion policies often lack cheap density ratios. We introduce MINDRL, a reward-to-update interface controller that translates a shared reward through branch-native score objects while turning factorization, tractability, drift, and smoothness barriers into budgets over block size, serialization, anchors, clipping, reranking, and branch weights. As a closed-loop controller, MINDRL makes the shared reward actionable for AR, parallel discrete, flow/diffusion, and AR+flow policies by selecting branch-native update objects and adapting their control budgets. Across language-side AR/parallel-discrete tests and matched flow/diffusion and AR+flow evaluations, MINDRL improves task-quality or reward-risk tradeoffs while preserving each branch's native structure.

## 1 Introduction

A reward is not a universal RL interface. Autoregressive (AR) language models make this easy to miss: each sampled token has an exact softmax probability, so likelihood ratios, KL penalties, and token credit are cheap [Schulman et al., 2017, Rafailov et al., 2023, Shao et al., 2024]. The same scalar reward is now attached to masked diffusion language models and dVLM/dVLA variants that update many discrete positions in parallel [Sahoo et al., 2024, Nie et al., 2025, Wen et al., 2025], as well as embodied and generative systems that emit continuous chunks through diffusion or flow-matching heads [Ho et al., 2020, Lipman et al., 2022, Ren et al., 2024]. The central question is which exposed probability object lets reward become a valid branch update.

Two interface failures make the mismatch visible. Parallel discrete generators reduce sequential depth but become fragile when dependent tokens are committed independently; Conditional Total Correlation (CTC, not Connectionist Temporal Classification) measures this within-block dependence [Watanabe, 1960, Graves et al., 2006]. Flow and diffusion heads model smooth coupled outputs but often lack the cheap exact density ratio assumed by token-style GRPO, so reward proxies can raise task reward while increasing drift, jerk, collisions, or feasibility failures; recent flow-RL work builds flow-native likelihood paths or surrogates for this tractability barrier [Liu et al., 2025b].

We propose MINDRL, a controller that first identifies the score object exposed by each branch and then adjusts block size, anchors, clipping, reranking, or branch weights when the measured barrier rises. High dependence shrinks or serializes blocks; rising flow drift strengthens anchors or clips surrogate steps; surrogate risk adjusts reranking and branch weights. MINDRL keeps branch-native optimizers as update objects while allocating reward budgets across policy decomposition, barrier probes, barrier-to-action control, granularity/anchor dynamics, and branch-wise composite optimization.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

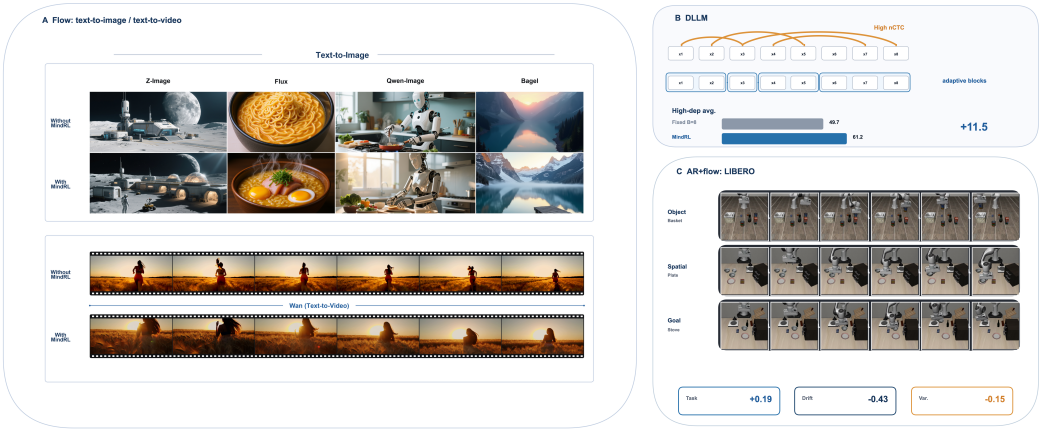


Figure 1: **Barrier-controlled updates across flow, dLLM, and embodied AR+flow interfaces.** MINDRL applies rewards through branch-native update objects: flow scores, dependence-aware dLLM blocks, and exact AR credit with anchored flow surrogates.

The theory gives the control rule. KL-regularized class restrictions induce reverse projection gaps; parallel product updates expose CTC as a measurable zero-barrier proxy; flow/diffusion branches require path-density evaluation or controlled surrogates. Together these results replace "apply the reward everywhere" with: expose the valid score object, measure the branch barrier, and adjust granularity or trust region before updating. The empirical goal is to test whether reward passes through the branch-native score object and whether the controller spends the right budget under interface stress.

**Contributions:**

- We formulate the reward-interface barrier: a shared reward becomes a valid update only through the probability or surrogate object exposed by each policy branch.
- We make the parallel-discrete barrier measurable by using normalized CTC as a dependence signal for block granularity and selective serialization.
- We characterize the flow/diffusion tractability barrier through path-density cost, surrogate bias, anchors, drift, and branch-level trust regions.
- We instantiate and evaluate the interface-control view across dLLM decoding, text-to-image/text-to-video flow generation, and embodied AR+flow control, showing how barrier budgets preserve branch structure while controlling drift and surrogate risk.

## 2 Related Work

RL post-training for AR language models builds on tractable per-token log-probabilities via PPO, DPO, and GRPO [Schulman et al., 2017, Rafailov et al., 2023, Shao et al., 2024]; recent VLA systems extend these gains to embodied policies [Tan et al., 2025, Zang et al., 2025]. Those methods are strong when the policy exposes the AR interface they assume. Reducing AR sequential depth has a long history in non-AR translation, masked refinement, and diffusion-style LMs [Gu et al., 2017, Ghazvininejad et al., 2019, Sahoo et al., 2024, Nie et al., 2025]; concurrent work characterizes limits on parallel factorization [Zhang et al., 2026, Zhong et al., 2026]. Flow and diffusion policies require flow-native RL adapters to handle the density tractability barrier: diffusion PPO [Ren et al., 2024], stochastic path training [Zhang et al., 2025], flow-matching policy gradients [McAllister et al., 2025], and energy-guided flow objectives [Alles et al., 2025]. Concurrent generative RL work—Flow-GRPO [Liu et al., 2025b], UniGRPO [Liu et al., 2026], CoDD [Li et al., 2026]—targets individual families or barriers. MINDRL asks the cross-family interface question: once a reward is shared across branches, which score object, granularity, anchor, and branch weight make the update valid? For AR+flow hybrids, HybridVLA and DiffusionVLA [Liu et al., 2025a, Wen et al., 2024] combine AR and diffusion for robot control; MINDRL specifies the mixed reward-to-update interface by assigning

exact AR credit and flow-native surrogates to their respective branches.

## 3 Two Barriers in Generative RL

We formalize both barriers for a conditional policy $\pi(a \mid c)$, where $c$ is any prompt, state, observation history, or multimodal context and $a$ may be discrete, continuous, or mixed. The central object is not the scalar reward alone, but the probability object through which that reward becomes an update. We call an update *branch-valid* when the branch exposes either the exact KL-RL score required by the policy family or a surrogate whose error is explicitly controlled by an anchor, clipping rule, or branch weight.

### 3.1 KL-regularized RL as projection

For a fixed context $c$, reference policy $\pi_0(\cdot \mid c)$, reward $r(a, c)$, and temperature $\beta > 0$, consider

$$J_c(\pi) = \mathbb{E}_{a\sim\pi(\cdot|c)}[r(a, c)] - \beta\,\mathrm{KL}(\pi(\cdot \mid c) \,\|\, \pi_0(\cdot \mid c))\,. \tag{1}$$

Assume $Z_\beta(c) < \infty$ and that all policies under comparison are absolutely continuous with respect to $\pi_0(\cdot \mid c)$. The unrestricted optimizer is the soft-optimal policy

$$\pi^\star_\beta(a \mid c) = \frac{\pi_0(a \mid c)\exp(r(a, c)/\beta)}{Z_\beta(c)}. \tag{2}$$

**Proposition 3.1** (Projection-regret identity)**.** *Let $\mathcal{F}$ be any restricted conditional policy class whose members are absolutely continuous with respect to $\pi_0(\cdot \mid c)$. Then, for each context $c$,*

$$\max_\pi J_c(\pi) - \max_{\pi\in\mathcal{F}} J_c(\pi) = \beta \min_{q\in\mathcal{F}} \mathrm{KL}\big(q(\cdot \mid c) \,\|\, \pi^\star_\beta(\cdot \mid c)\big)\,. \tag{3}$$

Substitution gives $J_c(\pi) = \beta \log Z_\beta(c) - \beta\,\mathrm{KL}(\pi\|\pi^\star_\beta)$. This
is the first link in the argument: once the reward induces a soft-optimal target, a branch that cannot expose the right probability object pays a projection cost or a surrogate-error cost. The next two subsections identify the two ways this happens in generative policies: missing joint factors in parallel discrete updates and missing cheap density ratios in continuous flow updates.

*Remark* 3.2 (Prior art). Proposition 3.1 follows from standard KL-RL results [Levine, 2018, Korbak et al., 2022, Ziegler et al., 2019]. The contribution here is not the variational identity itself, but its use as an interface test: a reward update is valid only after the policy branch supplies the distributional object on the right side of Eq. (3), or an explicitly controlled replacement.

### 3.2 Barrier I: factorization in parallel discrete policies

Let $M$ be a set of positions updated in one parallel step, $X_M = \{X_i\}_{i\in M}$, and $C$ the visible context. A parallel discrete update typically produces per-position token distributions and commits the block independently:

$$q(x_M \mid C) = \prod_{i\in M} q_i(x_i \mid C), \qquad \mathcal{Q}_{\mathrm{fact}} = \{q :\ q(x_M \mid C) = \prod_{i\in M} q_i(x_i \mid C)\}. \tag{4}$$

Applying Proposition 3.1, the exact RL class regret of this family is

$$\Delta^{\mathrm{fact}}_{\mathrm{RL}}(C) = \beta \min_{q\in\mathcal{Q}_{\mathrm{fact}}} \mathrm{KL}\big(q(x_M \mid C) \,\|\, \pi^\star_\beta(x_M \mid C)\big)\,. \tag{5}$$

This reverse projection is the exact class cost of the factorized branch. It is the quantity the reward would care about, but it is hard to measure offline because it depends on $\pi^\star_\beta$ and a reverse product-family projection.

The measurable companion is the forward projection of the same target joint onto the same product family:

$$\Delta^{\mathrm{fact}}_{\mathrm{F}}(C) = \min_{q\in\mathcal{Q}_{\mathrm{fact}}} \mathrm{KL}\big(\pi^\star_\beta(x_M \mid C) \,\|\, q(x_M \mid C)\big)\,. \tag{6}$$

By the classical information-projection identity [Csiszár, 1975, Watanabe, 1960, Amari, 2016]

$$\Delta_{\mathrm{F}}^{\mathrm{fact}}(C) = \mathrm{CTC}_{\pi_\beta^\star}(X_M \mid C) = \mathrm{KL}\left(\pi_\beta^\star(x_M \mid C) \,\middle\|\, \prod_{i\in M} \pi_\beta^\star(x_i \mid C)\right). \tag{7}$$

Equivalently, whenever the conditional entropies are finite,

$$\mathrm{CTC}(X_M \mid C) = \sum_{i\in M} H(X_i \mid C) - H(X_M \mid C). \tag{8}$$

*Remark* 3.3 (Novelty scope). Equation (7) is the classical fact that forward KL projection onto a product family equals total correlation [Watanabe, 1960, Csiszár, 1975, Amari, 2016]. Our contribution is to instantiate that identity for parallel discrete RL and use normalized CTC (nCTC) as a practical dependence diagnostic.

**CTC as a zero-barrier diagnostic.** The reverse regret in Eq. (5) and forward CTC in Eq. (7) assign different positive prices to dependence, but under common support they share the same zero-barrier boundary: both vanish exactly when the soft-optimal block distribution factorizes. We use CTC as a measurable dependence signal and granularity-control diagnostic for this boundary.

**Proposition 3.4** (Granularity controls the forward factorization barrier)**.** *Let $\mathcal{P}$ be a partition of positions into parallel decision blocks and define the within-block forward factorization barrier*

$$G_{\mathrm{F}}(\mathcal{P}; C) = \sum_{M\in\mathcal{P}} \mathrm{CTC}(X_M \mid C). \tag{9}$$

*If $\mathcal{P}'$ refines $\mathcal{P}$ by splitting one or more blocks, then $G_{\mathrm{F}}(\mathcal{P}'; C) \le G_{\mathrm{F}}(\mathcal{P}; C)$. The singleton partition has $G_{\mathrm{F}}(\mathcal{P}_{\mathrm{single}}; C) = 0$. Thus shrinking a parallel block or routing a high-dependence block to a sequential AR factor removes that block's measurable within-step factorization barrier, at the cost of more decision steps.*

This is the discrete control knob: low nCTC permits large blocks, while high nCTC triggers block shrinking, selective serialization, or AR fallback.

**AR as the discrete escape.** An AR policy writes

$$\pi_{\mathrm{AR}}(x_{1:T} \mid C) = \prod_{t=1}^{T} \pi(x_t \mid x_{<t}, C). \tag{10}$$

Each factor conditions on prior decisions, so the within-step CTC barrier appears only when multiple unconditioned positions are grouped into one decision.

### 3.3 Barrier II: tractability in flow and diffusion policies

Continuous flow matching policies generate an action chunk $a \in \mathbb{R}^d$ by sampling $x_0 \sim p_0$ and integrating an ODE

$$\frac{dx_t}{dt} = v_\theta(x_t, t, c), \qquad a = x_1. \tag{11}$$

Exact density evaluation follows the instantaneous change-of-variables formula:

$$\log \pi_\theta(a \mid c) = \log p_0(x_0) - \int_0^1 \mathrm{div}_x\, v_\theta(x_t, t, c)\, dt. \tag{12}$$

Thus PPO/GRPO-style ratios require path-density evaluation for old and new policies, unlike softmax token ratios. The continuous branch may still be trainable, but the update must pass through a flow-native object rather than pretending that endpoint samples expose token-like likelihoods.

*Remark* 3.5 (Flow likelihood tractability barrier). For an ODE-based flow policy with state dimension $d$ and $N$ numerical integration steps, exact log-density evaluation requires tracking the divergence term in Eq. (12) along the trajectory. Naive exact Jacobian evaluation scales quadratically in $d$ per integration step, while stochastic trace estimators reduce cost at the price of estimator variance. Consequently, standard token-style PPO/GRPO is not a native update rule for such policies unless the model exposes a tractable likelihood path or a controlled surrogate.

Existing continuous generative RL methods are adapters for this barrier: ReinFlow uses stochastic paths [Zhang et al., 2025], FPO builds clipped flow-matching surrogates [McAllister et al., 2025], and FlowQ learns anchored energy-guided paths [Alles et al., 2025]. All require flow-native machinery rather than token log-prob wrappers.

**Proposition 3.6** (Surrogate and anchor decomposition for flow branches)**.** *For a flow branch, let $\ell_\theta(a, c)$ denote the exact log-ratio term required by a PPO/GRPO-style update and let $\widehat{\ell}_\theta(a, c)$ be the ratio or surrogate exposed by a flow-native adapter. All expectations are taken under the update sampling distribution, and assume $\mathbb{E}[|A(a, c)|\,|\widehat{\ell}_\theta(a, c) - \ell_\theta(a, c)|] < \infty$. For any advantage value $A(a, c)$,*

$$\left|\mathbb{E}\Big[A(a, c)\widehat{\ell}_\theta(a, c)\Big] - \mathbb{E}[A(a, c)\ell_\theta(a, c)]\right| \le \mathbb{E}\Big[|A(a, c)|\,|\widehat{\ell}_\theta(a, c) - \ell_\theta(a, c)|\Big]. \tag{13}$$

*If the branch anchor $B_\theta(c)$ upper bounds $\mathrm{KL}(\pi_\theta(\cdot \mid c)\|\pi_0(\cdot \mid c))$, then*

$$\mathbb{E}_c[\mathrm{TV}(\pi_\theta(\cdot \mid c), \pi_0(\cdot \mid c))] \le \sqrt{\frac{1}{2}\mathbb{E}_c[B_\theta(c)]}. \tag{14}$$

Eq. (13) isolates surrogate error in the update score; Eq. (14) states what a KL-type anchor controls. The two terms are deliberately separate: a low-bias surrogate can still drift away from the reference policy, and a strong anchor does not make an inaccurate surrogate exact.

*Remark* 3.7 (Anchor precondition). Flow-matching and BC anchors are computable proxies for the KL condition; increasing $\eta_{\mathrm{FM/BC}}$ suppresses prior drift but does not by itself certify the premise of Eq. (14). The paper therefore reports drift-control evidence separately from surrogate-error control.

### 3.4 MindRL composite view

MINDRL treats the policy as a set of branches $\mathcal{B}$. Each branch $b$ exposes either an exact ratio or an adapter surrogate $S_b$, an anchor $B_b$ (corresponding to $B_\theta(c)$ of Proposition 3.6 for flow branches), and a structure cost $C_b$ such as nCTC, density-estimator variance, prior drift, or jerk:

$$\mathcal{L}_{\mathrm{MindRL}}(\theta) = \sum_{b\in\mathcal{B}} \lambda_b \mathbb{E}[S_b] - \sum_{b\in\mathcal{B}} \eta_b \mathbb{E}[B_b] - \sum_{b\in\mathcal{B}} \gamma_b \mathbb{E}[C_b]. \tag{15}$$

Eq. (15) is barrier-budgeted RL: reward acts through branch-native scores, while $\eta_b$ and $\gamma_b$ price factorization, tractability, drift, or smoothness costs. The objective keeps exact-ratio branches exact and allocates approximation budgets only to branches that need controlled surrogates.

**Proposition 3.8** (Branch-native composite update)**.** *Assume each branch adapter supplies either the exact KL-RL ratio for its policy family or a surrogate with branch error $\epsilon_b = \widehat{\ell}_b - \ell_b$. Let $\mathcal{B}_{\mathrm{sur}}$ denote the surrogate branches and let $\mathcal{B}_{\mathrm{exact}}$ denote the exact-ratio branches. Assume further that for each $b \in \mathcal{B}_{\mathrm{sur}}$, the advantage estimator $A_b$ and surrogate error $\epsilon_b$ satisfy $\mathbb{E}[|A_b|\,|\epsilon_b|] < \infty$ under the update sampling distribution. Then the discrepancy between the* MINDRL *composite score and the exact branch-wise score is bounded by*

$$|\Delta_{\mathrm{MindRL}}| \le \sum_{b\in\mathcal{B}_{\mathrm{sur}}} \lambda_b \mathbb{E}[|A_b|\,|\epsilon_b|]\,, \tag{16}$$

*while exact-ratio branches contribute no surrogate error. Consequently, AR and tractable discrete branches should be optimized with exact token ratios, whereas flow/diffusion branches should be optimized only through a controlled native surrogate and anchor.*

Thus MINDRL keeps exact-ratio branches exact and localizes approximation to branches that require surrogates. For a surrogate branch, large surrogate error or advantage magnitude should increase anchor strength $\eta_b$, tighten clipping, or reduce branch weight $\lambda_b$, trading update speed for fidelity.

*Remark* 3.9 (Advantage boundedness). Advantage clipping and bounded surrogate families are sufficient for this integrability condition, but the bound itself only requires the stated first moment to be finite.

**Connecting the two barriers to MindRL control actions.** The five MindRL components implement these results: decompose branches, probe nCTC and density/surrogate risk, choose native scores plus control actions, update granularity and anchors, and combine the branch terms with the surrogate-error bound above.

## 4 Method: Measuring and Exploiting Within-Block Dependence

This section turns $\mathrm{CTC}(X_M \mid C)$ into a practical dependence signal (nCTC) and a teacher-free block-size policy. Offline measurement uses a reference conditional scorer; inference-time control uses uncertainty proxies because the target joint is unavailable during decoding.

Direct access to the target joint $p(x_M \mid C)$ is intractable, so we measure dependence with a reference scorer $p_{\mathrm{ref}}$ that returns conditional token probabilities under the same visibility context $C$. In experiments this is the paired AR model; it is used only offline, never during decoding. For each benchmark example, we teacher-force completion $x$, sample a block $M$, and compare the chain-rule joint score to marginals under the same $C$:

$$\widehat{D}(x; M, C) = \log p_{\mathrm{ref}}(x_M \mid C) - \sum_{i \in M} \log p_{\mathrm{ref}}(x_i \mid C). \tag{17}$$

Under a perfect oracle $p_{\mathrm{ref}} = p$, $\mathbb{E}[\widehat{D}] = \mathrm{CTC}(X_M \mid C)$. With a learned paired scorer, $\widehat{D}$ is a reference-dependent measurement of the same dependence functional, not direct access to the unknown soft-optimal target. We report the pair-normalized metric

$$\widehat{\mathrm{nCTC}}_{\mathrm{pair}}(x; M, C) \triangleq \frac{\widehat{D}(x; M, C)}{\binom{B}{2}}. \tag{18}$$

The offline scorer overhead does not affect decoding-time compute.

At inference, CTC is unavailable, so the controller uses dLLM uncertainty proxies: smaller blocks for likely high dependence, larger blocks otherwise. From Eq. (8), total marginal uncertainty upper-bounds the barrier for the same visibility context:

$$\mathrm{CTC}(X_M \mid C) \le \sum_{i \in M} H(X_i \mid C). \tag{19}$$

For candidate pool $\widetilde{U}_t$ at step $t$, define

$$\bar{u}_t = \frac{1}{|\widetilde{U}_t|} \sum_{i \in \widetilde{U}_t} H(X_i \mid C_t). \tag{20}$$

The bound is conservative; it supplies a safe scheduling direction rather than a calibrated regret value. We use an inverse teacher-free schedule

$$B_t = \mathrm{clip}\left( \left\lfloor \frac{\alpha}{\bar{u}_t + \epsilon} \right\rfloor,\ 1,\ B_{\max} \right). \tag{21}$$

For compute neutrality, $\alpha$ matches the average forward passes of fixed-$B$ decoding.

## 5 MINDRL: Model-Interface-Native RL

The two barriers imply a constructive method. MINDRL starts from the interface constraint: reward can only update a branch through an exact score or a controlled surrogate that the branch actually exposes. It treats each barrier profile as live training state and converts it into score selection, discrete granularity, continuous anchors, clipping, reranking, and branch weights (Figure 2).

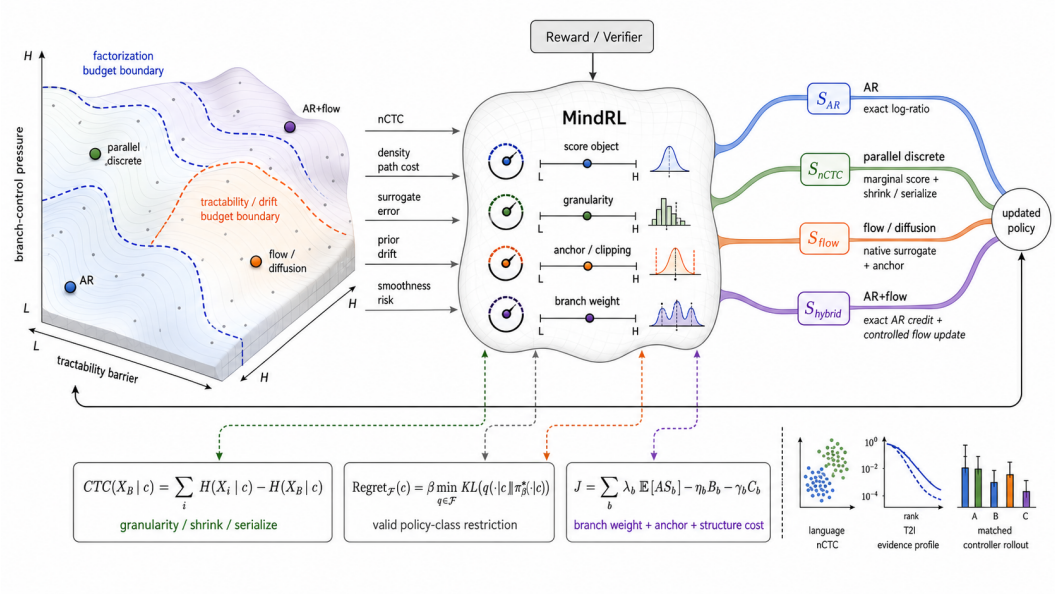


Figure 2: MINDRL **overview.** A barrier-budget manifold produces probes for nCTC, density-path cost, surrogate error, drift, and smoothness. The controller maps these probes and a shared reward/verifier into branch-valid update objects with adaptive score objects, granularity, anchors/clipping, and branch weights.

### 5.1 Policy decomposition and branch state

MINDRL represents a policy by a branch graph:

$$\pi_\theta(a \mid c) = \prod_{b \in \mathcal{B}} \pi_{\theta_b}(a_b \mid c, a_{\mathrm{pa}(b)}), \tag{22}$$

where $a_b$ is a token sequence, parallel token block, continuous action chunk, image/video latent, or another branch output. Each branch carries a POLICYSPEC:

$$s_b = (\text{modality}_b, \text{structure}_b, \text{parents}_b, \text{score availability}_b, \text{anchor}_b). \tag{23}$$

This covers AR, parallel discrete, flow/diffusion, and mixed AR+flow policies.

For each batch, MINDRL records a BARRIERPROFILE

$$z_b = \phi_b(\pi_{\theta_b}, a_b, c) = \left(\widehat{\mathrm{nCTC}}_b,\ \xi_b^{\mathrm{dens}},\ \nu_b^{\mathrm{sur}},\ \delta_b^{\mathrm{drift}},\ \sigma_b^{\mathrm{smooth}}\right), \tag{24}$$

Inactive coordinates are set by branch type: AR uses exact token likelihoods; parallel discrete reports nCTC and block size; flow reports density availability, surrogate variance, drift, and smoothness.

### 5.2 Barrier-to-action control and branch objective

The controller maps a branch specification, its current barrier profile, and optional controller memory $m_b^t$ to an ADAPTERDECISION:

$$d_b^t = \rho(s_b, z_b^t, m_b^t) = (\mathcal{A}_b,\ u_b,\ \eta_b,\ \gamma_b,\ \lambda_b)\,, \tag{25}$$

Here $\mathcal{A}_b$ is the adapter, $u_b$ the granularity, $\eta_b$ the anchor strength, $\gamma_b$ the structure-cost weight, and $\lambda_b$ the branch weight. Adapter choice is only one coordinate: high nCTC can reduce $u_b$ or serialize variables; high surrogate variance can reduce $\lambda_b$ or tighten clipping; high drift or jerk can increase $\eta_b$ and $\gamma_b$. These actions restore a valid reward interface: smaller discrete blocks restore conditioning, stronger flow anchors keep a surrogate near a faithful region, lower branch weights prevent unreliable scores from dominating, and the adapter exposes sampling, score/surrogate, anchor, and structure-metric hooks.

MINDRL maximizes the branch-wise composite objective

$$\mathcal{J}_{\mathrm{MindRL}}(\theta) = \sum_{b\in\mathcal{B}} \lambda_b \mathbb{E}\Big[\widehat{A}(a,c)\, S_b(a_b, c, a_{\mathrm{pa}(b)})\Big] - \sum_{b\in\mathcal{B}} \eta_b \mathbb{E}\big[B_b(\theta_b;\theta_b^0)\big] - \sum_{b\in\mathcal{B}} \gamma_b \mathbb{E}[C_b(u_b, z_b)]\,. \tag{26}$$

Here $S_b$ is the exact score or native surrogate, $B_b$ the anchor, and $C_b$ a structure cost such as nCTC, density-estimator variance, drift, or jerk. The reward is shared, but score objects and control actions remain branch-specific. This separation is the main design invariant: MINDRL never asks a branch to optimize a reward through a probability object that the branch does not expose. The method therefore follows the theoretical chain: measure the barrier, select a branch-native score, and spend control budget only on the branch where the exact interface fails.

### 5.3 Policy classes and native adapters

AR branches expose exact sequence factors and can use token log-ratio PPO/GRPO. Parallel discrete branches expose cheap marginals but not a sufficient joint, so MINDRL controls block size, critical-token serialization, and optional AR fallback. Flow/diffusion branches preserve continuous coupling but usually lack cheap exact density ratios, so MINDRL uses a flow-native surrogate together with anchor, clipping, and drift control. AR+flow branches combine the first and third cases: exact AR credit for plan tokens and an anchored flow surrogate for continuous action chunks. Thus the policy family does not determine the update by name; the exposed score object and measured barrier determine the update. This is why score routing alone is incomplete: after the score object is selected, the controller still has to regulate granularity, anchor strength, clipping, and branch weight.

### 5.4 AR+flow as the canonical mixed policy

An AR+flow policy factors into an AR plan $y$ and a continuous action chunk $a$. MINDRL applies exact prefix-conditioned ratios to $y$ and a native flow surrogate with behavior, smoothness, and drift anchors to $a$. Thus AR+flow gives a concrete branch-valid discrete-continuous RL interface under MINDRL.

## 6 Experiments

The experiments ask whether the measured barriers choose the right control action in three settings: language-side nCTC and dLLM granularity, flow/diffusion measurements, and matched AR+flow controller evaluations. The language block tests Proposition 3.4; the remaining paragraphs evaluate continuous-branch behavior, specify the mixed AR+flow interface, test Propositions 3.1–3.6, and compare score routing with the full controller.

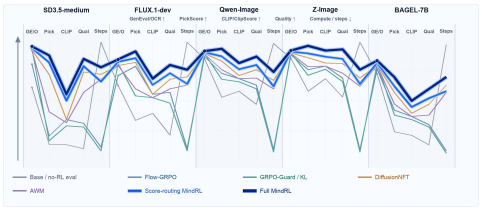


Figure 3: **Text-to-image evaluation summary.** Public and Flow-Factory measurements are separated; compute is lower-is-better.

### 6.1 Language-side evidence: AR, dLLM, and MindRL-discrete

The language evaluation asks whether the measured factorization barrier selects the right discrete control action. It matches AR and parallel-discrete policies under the same SFT data, sampling budget, and evaluation protocol, then tests whether high nCTC predicts fixed-block sensitivity and whether MINDRL-discrete improves accuracy by reducing dependence-sensitive parallelism (Table 1). The Qwen3-8B name denotes the open-source AR prototype/base [Yang et al., 2025]; we convert it into a paired diffusion-style LM using the open-source dLLM framework [Zhou et al., 2026] and the continuation-SFT protocol.

nCTC uses block size $B = 16$ and the paired AR scorer with the same SFT data; following Eq. (18), we report the pair-normalized measurement $\widehat{\mathrm{nCTC}}_{\mathrm{pair}} = \widehat{D}(x; M, C)/\binom{B}{2}$ averaged over the validation split.

Table 1: **Language-side dependence and granularity control.** Paired dLLM/AR nCTC uses $B{=}16$; scores compare fixed $B{=}8$ decoding with adaptive MINDRL-discrete on standard benchmarks [Cobbe et al., 2021, Hendrycks et al., 2021, Chen et al., 2021, Zellers et al., 2019, Paperno et al., 2016].

| Task | Type | $\text{nCTC}_{\text{pair}}$ | Fixed $B{=}8$ | MINDRL adaptive | Gain | Fixed gap (pp) |
|---|---|---|---|---|---|---|
| GSM8K | High | 0.46 | 65.4 | 76.8 | +11.4 | -15.8 |
| MATH-500 | Very High | 0.50 | 25.4 | 39.2 | +13.8 | -15.0 |
| HumanEval | High | 0.44 | 58.5 | 67.7 | +9.3 | -13.7 |
| HellaSwag | Low | 0.09 | 86.2 | 86.4 | +0.2 | -1.7 |
| LAMBADA | Low | 0.06 | 77.5 | 77.8 | +0.3 | -1.5 |
| High-dep avg | – | 0.47 | 49.7 | 61.2 | **+11.5** | -14.8 |
| Low-dep avg | – | 0.08 | 81.9 | 82.1 | **+0.3** | -1.6 |

**Flow/diffusion and mixed-interface evidence.** Figure 3 combines public reference values with Flow-Factory measurements [Ping et al., 2026]; the figure separates public reference values from project measurements.

For AR+flow action policies, matched evaluations compare an implemented adapter across six evaluation conditions: SFT/BC, interface-mismatched token-logprob proxy, reward-driven $K$-sample reranking, anchored flow adapter, score-routing-only MINDRL, and full MINDRL-ARFlow. EO1 uses the upstream Qwen2.5-VL backbone and flow-matching action head [Qu et al., 2025, Bai et al., 2025] on LIBERO [Liu et al., 2023]; FLUX.1-dev uses Flow-Factory under fixed prompts, seeds, sampler steps, reward/verifier, and evaluator budget [Black Forest Labs, 2024]. Compared with score routing alone, full MINDRL adds adaptive anchors, clipping, branch weights, and drift/surrogate control; under the matched evaluation protocols, it raises task metric by $+0.19/+0.09$ on EO1/FLUX while accepting lower reward by $-0.12/-0.06$ and reducing drift by $0.43/0.21$ and surrogate variance by $0.15/0.06$.

**Controlled verification and controller components.** Controlled checks are consistent with the proposed mechanisms: dependence raises nCTC/regret under fixed marginals, MINDRL-flow lowers Gaussian drift from 0.952 to 0.500, and adaptive granularity improves high-dependence averages from 49.7 to 61.2 while low-dependence remains stable. Together, matched evaluations and component checks indicate that score selection alone is insufficient; granularity, anchors, clipping, and branch weights keep updates controlled under the branch interface.

## 7 Conclusion

We introduced MINDRL, a reward-to-update interface controller for discrete-continuous generative policies. It makes reward updates branch-valid by selecting exact AR ratios, factorized parallel-discrete products, or tractable/surrogate flow-density paths, then setting granularity, anchors, clipping, and branch weights. The resulting rule is direct: exact-ratio branches stay exact, parallel discrete branches spend granularity, and flow branches spend surrogate and anchor budgets before reward updates the policy.

**Limitations and scale-up.** The evidence is strongest for language-side dependence, decode-time granularity control, and matched controller evaluations on the evaluated branches. nCTC still requires a paired AR scorer, FM/BC anchors are proxies for KL control, and future work should scale the evaluation to more VLA families, more flow generators, and larger matched online evaluations. The main claim is therefore local: reward updates are valid only through the branch object that supplies an exact score or a controlled surrogate.

## Acknowledgments and Disclosure of Funding

This work received no third-party funding or support. The authors declare no competing interests.

## References


Marvin Alles, Nutan Chen, Patrick van der Smagt, and Botond Cseke. Flowq: Energy-guided flow policies for offline reinforcement learning. *arXiv preprint arXiv:2505.14139*, 2025.

Shun-ichi Amari. *Information geometry and its applications*. Springer, 2016.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025. URL `https://arxiv.org/abs/2502.13923`.

Black Forest Labs. FLUX.1-dev, 2024. URL `https://huggingface.co/black-forest-labs/FLUX.1-dev`.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D Lee, Deming Chen, and Tri Dao. Medusa: Simple llm inference acceleration framework with multiple decoding heads. *arXiv preprint arXiv:2401.10774*, 2024.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. *Advances in Neural Information Processing Systems*, 35:28266–28279, 2022.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11315–11325, 2022.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Imre Csiszár. I-divergence geometry of probability distributions and minimization problems. *The annals of probability*, pages 146–158, 1975.

Yichao Fu, Peter Bailis, Ion Stoica, and Hao Zhang. Break the sequential dependency of llm inference using lookahead decoding. *arXiv preprint arXiv:2402.02057*, 2024.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 6112–6121, 2019.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and LingPeng Kong. Diffuseq: Sequence to sequence text generation with diffusion models. *arXiv preprint arXiv:2210.08933*, 2022.

Google DeepMind. Gemini diffusion. `https://deepmind.google/en/models/gemini-diffusion/`, 2025. Accessed: 2026-04-24.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. Non-autoregressive neural machine translation. *arXiv preprint arXiv:1711.02281*, 2017.

Jiatao Gu, Changhan Wang, and Junbo Zhao. Levenshtein transformer. *Advances in neural information processing systems*, 32, 2019.

Zhengfu He, Tianxiang Sun, Qiong Tang, Kuanning Wang, Xuan-Jing Huang, and Xipeng Qiu. Diffusionbert: Improving generative masked language models with diffusion models. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 4521–4534, 2023.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in neural information processing systems*, 34:12454–12465, 2021.

Inception Labs, Samar Khanna, Siddhant Kharbanda, Shufan Li, Harshit Varma, Eric Wang, Sawyer Birnbaum, Ziyang Luo, Yanis Miraoui, Akash Palrecha, Stefano Ermon, Aditya Grover, and Volodymyr Kuleshov. Mercury: Ultra-fast language models based on diffusion. *arXiv preprint arXiv:2506.17298*, 2025. URL `https://arxiv.org/abs/2506.17298`.

Tomasz Korbak, Ethan Perez, and Christopher Buckley. Rl with kl penalties is better viewed as bayesian inference. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1083–1091, 2022.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pages 19274–19286. PMLR, 2023.

Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

Ian Li, Zilei Shao, Benjie Wang, Rose Yu, Guy Van den Broeck, and Anji Liu. Breaking the factorization barrier in diffusion language models. *arXiv preprint arXiv:2603.00045*, 2026.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in neural information processing systems*, 35: 4328–4343, 2022.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

Jiaming Liu, Hao Chen, Pengju An, Zhuoyang Liu, Renrui Zhang, Chenyang Gu, Xiaoqi Li, Ziyu Guo, Sixiang Chen, Mengzhen Liu, et al. Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model. *arXiv preprint arXiv:2503.10631*, 2025a.

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. *arXiv preprint arXiv:2505.05470*, 2025b.

Jie Liu, Zilyu Ye, Linxiao Yuan, Shenhan Zhu, Yu Gao, Jie Wu, Kunchang Li, Xionghui Wang, Xiaonan Nie, Weilin Huang, et al. Unigrpo: Unified policy optimization for reasoning-driven visual generation. *arXiv preprint arXiv:2603.23500*, 2026.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

David McAllister, Songwei Ge, Brent Yi, Chung Min Kim, Ethan Weber, Hongsuk Choi, Haiwen Feng, and Angjoo Kanazawa. Flow matching policy gradients. *arXiv preprint arXiv:2507.21053*, 2025.

William J McGill. Multivariate information transmission. *Psychometrika*, 19(2):97–116, 1954.

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

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. *arXiv preprint arXiv:2406.03736*, 2024.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc-Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 1525–1534, 2016.

Bowen Ping, Chengyou Jia, Minnan Luo, Hangwei Qian, and Ivor Tsang. Flow-factory: A unified framework for reinforcement learning in flow-matching models. *arXiv preprint arXiv:2602.12529*, 2026.

Delin Qu, Haoming Song, Qizhi Chen, Zhaoqing Chen, Xianqiang Gao, Xinyi Ye, Qi Lv, Modi Shi, Guanghui Ren, Cheng Ruan, et al. Eo-1: Interleaved vision-text-action pretraining for general robot control. *arXiv preprint arXiv:2508.21112*, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Allen Z Ren, Justin Lidard, Lars L Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion policy policy optimization. *arXiv preprint arXiv:2409.00588*, 2024.

Subham S Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

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, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Mitchell Stern, William Chan, Jamie Kiros, and Jakob Uszkoreit. Insertion transformer: Flexible sequence generation via insertion operations. In *International Conference on Machine Learning*, pages 5976–5985. PMLR, 2019.

Haoran Sun, Lijun Yu, Bo Dai, Dale Schuurmans, and Hanjun Dai. Score-based continuous-time discrete diffusion models. *arXiv preprint arXiv:2211.16750*, 2022.

Shuhan Tan, Kairan Dou, Yue Zhao, and Philipp Krähenbühl. Interactive post-training for vision-language-action models. *arXiv preprint arXiv:2505.17016*, 2025.

Chunqi Wang, Ji Zhang, and Haiqing Chen. Semi-autoregressive neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 479–488, 2018.

Satosi Watanabe. Information theoretical analysis of multivariate correlation. *IBM Journal of research and development*, 4(1):66–82, 1960.

Junjie Wen, Minjie Zhu, Yichen Zhu, Zhibin Tang, Jinming Li, Zhongyi Zhou, Chengmeng Li, Xiaoyu Liu, Yaxin Peng, Chaomin Shen, et al. Diffusion-vla: Generalizable and interpretable robot foundation model via self-generated reasoning. *arXiv preprint arXiv:2412.03293*, 2024.

Junjie Wen, Minjie Zhu, Jiaming Liu, Zhiyuan Liu, Yicun Yang, Linfeng Zhang, Shanghang Zhang, Yichen Zhu, and Yi Xu. dvla: Diffusion vision-language-action model with multimodal chain-of-thought. *arXiv preprint arXiv:2509.25681*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Hongzhi Zang, Mingjie Wei, Si Xu, Yongji Wu, Zhen Guo, Yuanqing Wang, Hao Lin, Liangzhi Shi, Yuqing Xie, Zhexuan Xu, et al. Rlinf-vla: A unified and efficient framework for reinforcement learning of vision-language-action models. *arXiv preprint arXiv:2510.06710*, 2025.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 4791–4800, 2019.

Shaorong Zhang, Longxuan Yu, Rob Brekelmans, Luhan Tang, Salman Asif, and Greg Ver Steeg. Generation order and parallel decoding in masked diffusion models: An information-theoretic perspective. *arXiv preprint arXiv:2602.00286*, 2026.

Tonghe Zhang, Chao Yu, Sichang Su, and Yu Wang. Reinflow: Fine-tuning flow matching policy with online reinforcement learning. *arXiv preprint arXiv:2505.22094*, 2025.

Yangyang Zhong, Yanmei Gu, Zhengqing Zang, Xiaomeng Li, Yuqi Ding, Xibei Jia, Yuting Shen, Zhenzhong Lan, Liwang Zhu, Weiping Liu, et al. Parallelism and generation order in masked diffusion language models: Limits today, potential tomorrow. *arXiv preprint arXiv:2601.15593*, 2026.

Zhanhui Zhou, Lingjie Chen, Hanghang Tong, and Dawn Song. dllm: Simple diffusion language modeling. *arXiv preprint arXiv:2602.22661*, 2026.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and original introduction state the interface-control claims and scope; the original conclusion discusses limitations.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The original conclusion includes a Limitations and scale-up paragraph.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [No]

Justification: The main text states the theoretical results and assumptions, but full derivations are not included in this appendix-free version.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [No]

Justification: The main text describes the method and reported evaluations, but does not provide all details needed for complete reproduction.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The authors have not publicly released the code or reproduction artifacts.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [No]

   Justification: The original main text does not provide all training, data and evaluation settings.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The reported tables give point estimates without complete per-result uncertainty information.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: A complete per-experiment and project-wide compute account is not provided.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The authors report conformity with the NeurIPS Code of Ethics; the work does not involve human subjects or crowdsourcing.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: A dedicated analysis of possible societal impacts is not included.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: No pretrained weights, new datasets or scraped content are released with this paper.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: The references credit existing assets, but all asset licenses, versions and terms are not enumerated.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: No new dataset, model weights or public code package is released with this version.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The work does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The work does not involve research with human participants requiring IRB approval.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: The original experiments identify AR and dLLM model backbones and paired scorers as research objects.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.