# WHEN CO-TRAINING BECOMES REDUNDANT: UPDATE GEOMETRY IN UNIFIED MULTIMODAL REINFORCEMENT LEARNING

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

## ABSTRACT

Unified multimodal models (UMMs) combine multimodal understanding and generation in one backbone through separate experts. In image-generation reinforcement learning (RL), the reward signal backpropagates through both experts, so it can update the understanding expert as well as the generation expert, yet the role of the former's updates remains unclear as prior work typically keeps it frozen. In this work, we show that the understanding expert's contribution is front-loaded. Two velocity-space probes including *compensability* (measuring whether its updates carry content the generator cannot locally reproduce) and *transport* (measuring how strongly its updates align with directions the generator realizes efficiently), reveal that the early updates of the understanding expert are both distinct and efficient, while both signals decay as training proceeds. This finding is also mirrored at the reward level, where later understanding training steps add no measurable gain. These observations motivate a staged protocol: a geometry-based freeze rule that reads the two probes relative to their running peak, and freezes the understanding expert at a predicted step. Across two backbones, three reward models, and three RL algorithms, staged training consistently outperforms naive joint training, improving $\text{GenEval2}_{\text{GM}}$ by up to $+11.1\%$ while reaching higher rewards in substantially fewer training hours. Our results suggest the understanding expert's updates are best treated as a trajectory-dependent, front-loaded resource rather than a fixed co-training partner.

## 1 INTRODUCTION

Unified multimodal models (UMMs) combine multimodal understanding and generation within a single backbone. Recent UMMs, including BAGEL (Deng et al., 2025), SenseNova-U1 (Diao et al., 2026), and Tuna-2 (Liu et al., 2026c), organize these capabilities with separate understanding and generation experts that interact through shared attention. In image-generation reinforcement learning (RL), the gradient of the RL objective flows through a computation graph involving both experts, so the same signal can update UND and GEN[1] when both are trainable. Their roles in generation tasks, however, are asymmetric: GEN directly modifies the image-generation dynamics, whereas UND only shapes the conditioning representations and shared features on which GEN operates. Prior work settles this asymmetry by convention, typically updating GEN alone and keeping UND frozen (Liu et al., 2026b), while recent joint post-training asks how to train both capabilities together (Jiang et al., 2025; Liu et al., 2026a). However, these studies have not explored whether every update to UND remains useful as training proceeds. Should the understanding expert be kept frozen, updated jointly with the generation expert, or assigned a different role at different points in training? To answer these questions, our investigation proceeds in three steps, trying to sharpen the picture of what UND's updates actually do during image-generation RL.

**Updating UND redirects and shrinks GEN's updates, yet improves generation (§3).** In a controlled comparison between training with UND frozen and training with both experts trainable, unfreezing UND redirects the optimization path of GEN: GEN's updates become nearly orthogonal to those obtained with UND frozen and smaller in norm, as if the two experts were bidding for a fixed

[1]For conciseness, we use UND and GEN to denote the understanding and generation experts, respectively.



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.

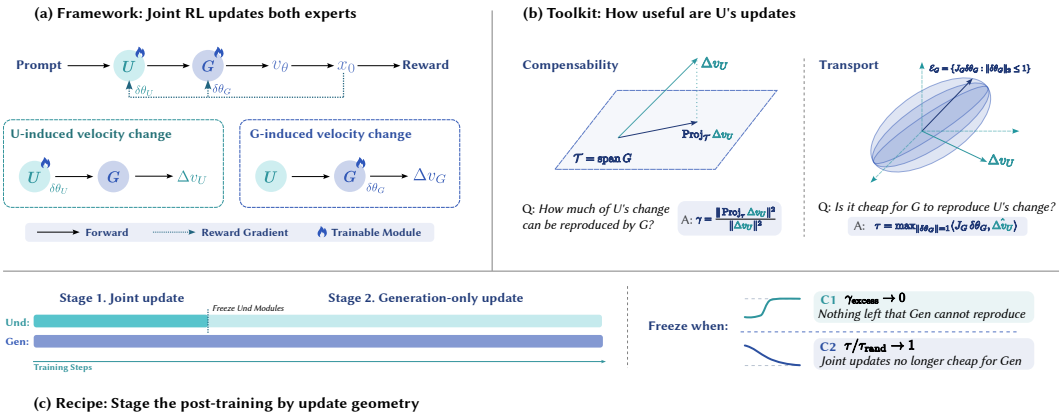



Figure 1: **Overview of Our Approach. (a)** Joint RL on UMMs updates both UND and GEN from the reward feedback, and the resulting velocity update can be decoupled into an UND-induced part $\Delta v_U$ and a GEN-induced part $\Delta v_G$. **(b)** We evaluate UND's contribution with two velocity-space measures, *compensability* $\gamma$ and *transport* $\tau$, which ask whether UND writes changes GEN cannot reproduce and whether the shared changes are cheap for GEN to realize. **(c)** UND's updates are distinct and useful early but become redundant during the transition phase, which motivates our staged protocol: UND and GEN are updated jointly at first, and UND is frozen once either signal decays to a small fraction of its own peak acquisition rate (Eq. 5), leaving the full update budget to GEN.

update budget. We also show that joint UND-GEN training improves the generation reward while leaving understanding performance unchanged or slightly improved. Therefore, UND's updates are not wasted, even though they divert and slow down GEN's own learning.

**UND's generation-relevant contribution is front-loaded (§4).** We call an update to UND *redundant* when GEN could have produced the same effect by itself. This concept is made precise in the velocity space of the flow-matching generator with two probes: *compensability*, the fraction of UND's induced velocity change that lies in GEN's local tangent space, and *transport*, how strongly that change aligns with GEN's high-gain directions, i.e., how cheaply GEN could realize it. In our experiments across five settings on BAGEL, spanning three RL algorithms and three reward models, both probes show that UND's contribution is front-loaded rather than uniform. Early on, UND's updates write velocity changes that GEN cannot locally reproduce and that GEN can realize efficiently. As training proceeds, the compensability excess collapses toward zero and the transport advantage decays toward the random baseline, indicating that UND's later updates become redundant. At the reward level, checkpoint composition on the same training trajectory also mirrors this pattern: early UND updates improve the reward of fixed GEN checkpoints, whereas later UND updates provide no measurable additional gain.

**Update geometry signals when joint training becomes redundant (§5).** UND's front-loaded role during the RL process probes motivate our staged training protocol, as sketched in Figure 1(c). We propose to update UND and GEN jointly while UND's signals are still strong, and freeze UND once they have decayed, leaving the full update budget to GEN for the remainder of training. Since the absolute scale of both signals varies across settings, we compare each against its own running peak and freeze UND at the first window in which either ratio falls below a fixed threshold. The thresholds are calibrated once on BAGEL, and can be transferred unchanged to a different backbone.

We evaluate our staged protocol on two UMM backbones (BAGEL and SenseNova-U1), three reward models (GenEval, GenEval2, and HPSv3), and three RL algorithms (DiffusionNFT, Flow-GRPO, and AWM). Staged training outperforms both GEN-only training and naive joint training in every setting. On BAGEL with DiffusionNFT, it raises GenEval from 83.89 (GEN-only) and 90.85 (joint) to 91.55, and relative to joint training it improves $\text{GenEval2}_{\text{AM}}$, $\text{GenEval2}_{\text{GM}}$, and HPSv3 by $+1.28\%$, $+11.1\%$, and $+1.46\%$; with Flow-GRPO and AWM it improves GenEval by $+2.70\%$ and $+3.00\%$. On SenseNova-U1, where the BAGEL thresholds are applied unchanged, the largest relative gain is $+5.94\%$ on $\text{GenEval2}_{\text{GM}}$. These gains also come at a lower training cost: staged training reaches higher rewards in substantially fewer training hours (Figure 8).

Together, these findings suggest that the understanding expert does not have a fixed role throughout image-generation post-training. Its early updates can provide distinct generation-relevant changes, while later optimization can be concentrated on GEN after this contribution declines. This update geometry perspective turns the choice between freezing and jointly updating UND into a trajectory-dependent decision, achieving better performance and efficiency in a simple, interpretable, and transferable way.

## 2 RELATED WORK

### 2.1 UNIFIED MULTIMODAL MODELS (UMMS)

Unified multimodal model (UMM) unifies multimodal understanding and generation into a single model. A class of UMMs performs both tasks end-to-end within one backbone, either by autoregressively modeling images as discrete tokens (Team et al., 2024; Wang et al., 2024), or by attaching a diffusion head to the shared transformer (Zhou et al., 2024; Xie et al., 2024; Ma et al., 2025a). Because the two tasks share parameters, the interference between understanding and generation is typically mitigated with decoupled visual encoders (Wu et al., 2025a; Chen et al., 2025b) or modality-specific experts (Deng et al., 2025; Fu et al., 2026). The second family stitches an LLM to an external diffusion decoder: the LLM predicts visual representations, such as CLIP features or learnable query tokens, that condition a separately trained image generator (Sun et al., 2024; Ge et al., 2024; Chen et al., 2025a; Pan et al., 2025). A recent line of work removes the pretrained tokenizer altogether, either by generating without vector quantization through masked autoregression (Wu et al., 2025b), or by generating directly from raw image patches with no visual encoder (Diao et al., 2026; Liu et al., 2026c; Wang et al., 2026).

### 2.2 POST-TRAINING OF UNIFIED MULTIMODAL MODELS

Reward-guided RL has become a standard post-training stage for diffusion and flow-matching generators (Liu et al., 2025a; Zheng et al., 2025; Wallace et al., 2023; Liu et al., 2025b; Xue et al., 2025), and recent work extends it to UMMs by jointly optimizing understanding and generation within a single policy: CoRL (Jiang et al., 2025) co-reinforces both capabilities through a unified RL stage followed by task-specific refinement, UniGRPO (Liu et al., 2026a) formulates reasoning-driven generation as a joint Markov decision process over text and image policies, UnifiedGRPO (Nie et al., 2026) further unlocks interleaved generation, and self-rewarding schemes let the model's own understanding supervise its generation (Jin et al., 2025; Hong et al., 2025; Shen et al., 2026). Beyond RL, reconstruction- or understanding-supervised post-training aligns the two without reward (Xie et al., 2025; Liu et al., 2026b). These works ask how to train both capabilities together, but not whether every update to every expert remains useful over the course of training. A parallel line of work diagnoses the interference between the two capabilities at the level of optimization objectives (Ye et al., 2026), representations (Lu et al., 2026), gradient statistics (Hao et al., 2025), or native UMM synergy (Wu et al., 2026). In contrast, we measure redundancy in the generator's velocity space, and turn this condition into a stopping rule that freezes the understanding expert at a predicted step without per-setting tuning.

## 3 WHAT DOES UND'S UPDATE CONTRIBUTE TO JOINT TRAINING?

Prior work on applying reinforcement learning to UMM generation tasks typically updates only the weights of the *generation expert* (GEN), while keeping the *understanding expert* (UND) frozen (Liu et al., 2026b). We revisit this choice by studying the structure and role of UND's updates in flow-matching RL on UMMs, asking whether they provide additional improvements, and whether they affect the model's understanding capabilities. All analyses in this section are conducted with BAGEL-MoT-7B, using DiffusionNFT as the RL algorithm and GenEval as the reward, unless otherwise specified.

**Updating UND re-orients and slows GEN's updates.** Let $\Delta\theta_G^{\text{joint}}$ and $\Delta\theta_G^{\text{frozen}}$ denote the updates to GEN's weights with and without unfreezing UND, respectively. We calculate the cosine similarity $\cos(\Delta\theta_G^{\text{joint}}, \Delta\theta_G^{\text{frozen}})$ at matched training steps. As shown in Figure 2 (Left), the two updates are

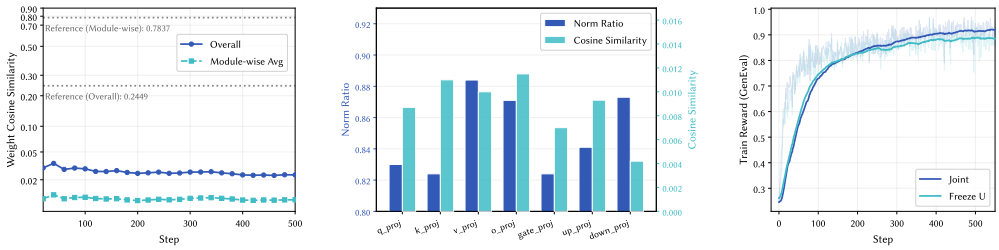



Figure 2: **Joint Training Changes GEN's Update Dynamics.** (**Left**) Cosine similarity between the updates to GEN with and without unfreezing UND, computed both on the full $\Delta\theta_G$ (*Overall*) and on the updates to each module (*Module-wise*). (**Middle**) Module-wise norm ratio ($\|\Delta\theta_G^{\text{joint}}\|_2/\|\Delta\theta_G^{\text{frozen}}\|_2$) and cosine similarities at step 500. (**Right**) Training reward curves for joint training v.s. freezing UND.

nearly orthogonal compared to the reference temporal similarity $\cos(\Delta\theta_G^{\text{joint}}(t), \Delta\theta_G^{\text{joint}}(t+\Delta t))$ within the same run of training. The norm ratio $\|\Delta\theta_G^{\text{joint}}\|_2/\|\Delta\theta_G^{\text{frozen}}\|_2 < 1$ in Figure 2 (Middle) also indicates that joint training slows GEN's updates down. Together, joint training behaves as though UND and GEN were bidding for a fixed update budget: UND's presence re-orients GEN's updates to a different path and leaves a smaller update budget.

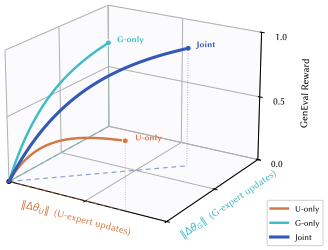



Figure 3: Joint training achieves higher generation reward than training GEN or UND alone.

Table 1: Reinforcement learning with UND updates generally maintains UMM's understanding capabilities.

| Benchmark | Baseline | RL on GenEval | RL on GenEval2 |
|---|---|---|---|
| MMMU | 51.33 | **52.11** | 51.30 |
| MMVP | 70.67 | **72.67** | 70.67 |
| MME-P | **1654** | 1649 | 1650 |
| MME-C | **695** | 694 | **695** |

**Updating UND improves generation without hurting understanding.** Despite re-orienting and slowing GEN's updates, unfreezing UND yields a modest but consistent improvement in generation reward, as shown in Figure 2 (Right) and Figure 3. Moreover, the visual understanding benchmark results in Table 1 also show that this improvement in generation does not come at the cost of understanding. Results with other reward models and UMM backbones are reported in Appendix C.3.

## 4 WHEN DOES CO-TRAINING BECOME REDUNDANT?

To determine when UND's contribution becomes redundant, we first measure it in velocity space through two complementary probes. We then track these signals over training and test whether the same transition is reflected in the reward-level value of UND's updates.

### 4.1 MEASURING UND'S CONTRIBUTION IN VELOCITY SPACE

We measure UND's contribution from two complementary aspects: whether GEN can locally reproduce its velocity change, and whether the shared component aligns with directions that GEN can realize efficiently. An overview of these measurements is shown in Figure 1.

**Setup.** Let $\theta = [\theta_U; \theta_G]$ denote the model parameters, with $\theta_U$ and $\theta_G$ the parameters of UND and GEN respectively, and let $v_\theta(z) \in \mathbb{R}^d$ be the velocity prediction at an input $z = (x_t, t, c)$. We write $\Delta v_U(z) = v_{[\theta_U+\delta\theta_U;\theta_G]}(z) - v_\theta(z)$ for the change induced by updating only UND's parameters, and $\Delta v_{\text{rand}}(z)$ for the change induced by a random weight perturbation of the same norm. For the

generation expert we use the weight-space Jacobian $J_G(z) = \partial v_\theta(z)/\partial \theta_G \in \mathbb{R}^{d\times|\theta_G|}$, that maps weight perturbations of GEN to the induced velocity changes.

**Definition 4.1** (***Compensability***). We define the *compensability* of an update $\Delta v$ as the fraction of its velocity change that lies in GEN's tangent space, which GEN could locally reproduce:

$$\gamma(\Delta v) = \frac{\|QQ^\top \Delta v\|_2^2}{\|\Delta v\|_2^2}, \tag{1}$$

where $Q \in \mathbb{R}^{d\times m}$ has orthonormal columns spanning a sampled tangent space of GEN.

$\gamma$ is a fraction in $[0, 1]$ measuring how much of an update GEN could locally reproduce, so we report the excess $\gamma_{\text{excess}} := \gamma(\Delta v_U) - \gamma(\Delta v_{\text{rand}})$ against the same random baseline. A negative excess means the real update is reproduced by GEN less than a random one would be, indicating that UND realizes a distinct update that GEN cannot. See Appendix B for detailed construction.

**Definition 4.2** (***Transport***). We define the *transport* of an update $\Delta v$ through GEN as the amplification of a velocity change along the top singular directions of $J_G$ (i.e., the high-gain modes of GEN):

$$\tau(\Delta v) = \frac{\|J_G^\top \Delta v\|_2}{\|\Delta v\|_2}. \tag{2}$$

While compensability measures how much of UND's velocity change GEN could reproduce, transport measures how cheaply GEN could realize it. We report the ratio $\tau(\Delta v_U)/\tau(\Delta v_{\text{rand}})$, where $\tau(\Delta v_{\text{rand}})$ serves as the noise baseline. A ratio above one means UND's update is amplified along GEN's high-gain modes beyond what a structureless update of the same norm would achieve. Equivalently, the dual form $\tau(\Delta v) = \max_{\|\delta\theta_G\|=1} \langle J_G\, \delta\theta_G, \hat{\Delta v}\rangle$ also indicates that high transport means GEN can realize UND's changes with little parameter cost.

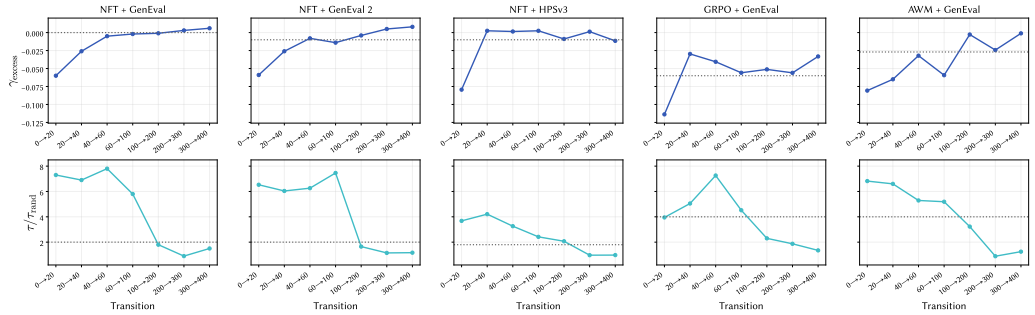



Figure 4: **UND's distinctness and transport ratio decay over training.** Compensability excess $\gamma(\Delta v_U) - \gamma(\Delta v_{\text{rand}})$ and transport ratio $\tau(\Delta v_U)/\tau(\Delta v_{\text{rand}})$ for five settings that span 3 RL algorithms (DiffusionNFT, Flow-GRPO, AWM) and 3 reward models (GenEval, GenEval2, HPSv3). The compensability excess is negative early on, and collapses toward zero during RL. The transport advantage starts high and gradually decays toward the random baseline as training progresses.

### 4.2 CONDITION 1: UND'S UPDATE STOPS CARRYING CONTENT GEN CANNOT REPRODUCE

The first question is whether UND writes anything GEN could not produce on its own, which is indicated by the compensability excess $\gamma_{\text{excess}} = \gamma(\Delta v_U) - \gamma(\Delta v_{\text{rand}})$. When it is negative, UND's realized velocity change lies outside GEN's sampled tangent space, so the update carries content GEN cannot locally reproduce.

Figure 4 (top) shows $\gamma_{\text{excess}}$ over training. In the first transition it is relatively negative in all five settings, meaning UND's early updates write predominantly outside GEN's span. By the second window it jumps abruptly, and afterwards it stays near zero under the three NFT settings or stabilizes at a moderate negative value under GRPO. The abrupt transition of $\gamma_{\text{excess}}$ marks the point at which UND's distinct contribution has been written. After this transition, UND's updates no longer

systematically produce content that GEN cannot reproduce (NFT), or do so at a much reduced rate (GRPO).

### 4.3 CONDITION 2: THE TRANSPORT OF UND'S UPDATE THROUGH GEN DECLINES

While the first condition concerned the part of UND's update that only UND can realize, this one concerns the part that both experts can achieve. The question is whether these shared velocity changes are cheap for GEN to realize. The transport ratio $\tau(\Delta v_U)/\tau(\Delta v_{\text{rand}})$ measures how much more strongly UND's update is aligned with GEN's high-gain modes than a structureless update of the same norm. A ratio above one means GEN can realize the same changes at low parameter cost, so the two experts realize them jointly and efficiently.

Figure 4 (bottom) shows that the transport efficiency is high early on and declines smoothly toward the random baseline in all five settings. In the GRPO case, the compensability excess remains negative after its initial jump, showing that UND still writes content GEN cannot reproduce. However, the transport ratio has already decayed: UND's updates still affect the output, but no longer along GEN's high-gain modes. Therefore, the two experts no longer realize the same velocity changes jointly at low parameter cost, indicating the efficiency advantage of joint training is gone.

### 4.4 THE REWARD-LEVEL VALUE OF UND'S UPDATE ALSO DECLINES

The two conditions above are measured in velocity space, locating when the velocity updates produced by UND become redundant. In this section we will show that UND's reward contribution also follows the similar trajectory, i.e., early UND updates contribute more to the reward score while the later ones contribute less.

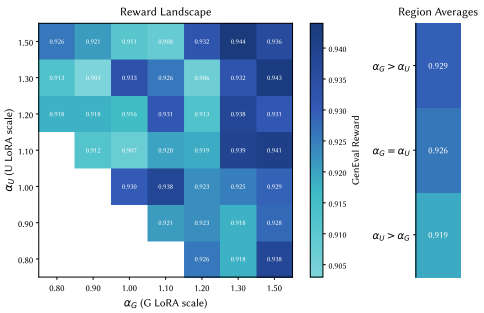



Figure 5: **Scaling up GEN's updates more than UND's produces higher reward.** The heatmap shows the reward landscape with different UND/GEN LoRA scaling factors $\alpha_U$ and $\alpha_G$. Reward favors scaling GEN more than UND: the region $\alpha_G > \alpha_U$ dominates the diagonal $\alpha_G = \alpha_U$, and the opposite region lags.

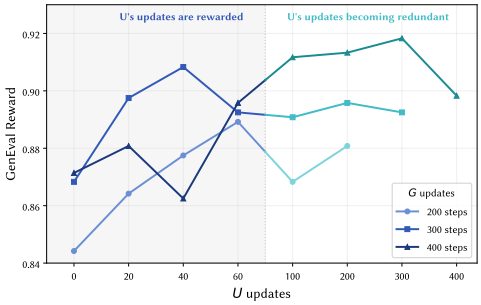



Figure 6: **UND's later updates are not rewarded.** From a single joint run, each GEN checkpoint is composed with UND checkpoints up to the same training step and evaluated on GenEval. For every fixed GEN, early UND checkpoints raise the reward; while later updates beyond the dashed line no longer produce a measurable gain.

**We need fewer UND updates compared to GEN updates.** Starting from a jointly trained LoRA checkpoint $\Delta\theta = [\Delta\theta_U, \Delta\theta_G]$ on the base model $\theta_0 = [\theta_U; \theta_G]$, we scale the two updates independently, $\theta(\alpha_U, \alpha_G) = [\theta_U + \alpha_U\Delta\theta_U; \theta_G + \alpha_G\Delta\theta_G]$, and evaluate GenEval reward over the grid of Figure 5. The landscape is asymmetric around the diagonal: the region $\alpha_G > \alpha_U$ consistently yields higher reward than symmetric scaling $\alpha_G = \alpha_U$, and the region $\alpha_G < \alpha_U$ yields lower. This indicates that the naive joint training (i.e., $\alpha_U = \alpha_G$) produces a suboptimal update budget allocation, and a better choice is to learn less from UND and more from GEN.

**Updating UND is no longer rewarded once the two conditions fail.** The scaling experiment shows that fewer updates to UND should be retained compared with those to GEN, while this one further investigates the temporal point at which UND's updates cease to be beneficial. From the same joint run, we fix a GEN checkpoint at later steps and compose it with early UND checkpoints from step 0 up to the same training step, evaluating GenEval reward for every combination. Figure 6 shows that for every fixed GEN, early UND checkpoints raise the reward, while UND checkpoints beyond

the dashed line (between steps 60 and 100) add no measurable gain. Notably, this trend aligns with the velocity-level probes on the same run. In the first column of Figure 4 (NFT with GenEval), the compensability excess $\gamma_{\text{excess}}$ is near zero beyond step 60, so UND's update no longer carries content GEN cannot reproduce, and the transport ratio $\tau/\tau_{\text{rand}}$ drops sharply at the same point, so realizing UND's velocity changes jointly with GEN is no longer cheap. The redundancy signal in velocity space is thus mirrored at the reward level.

**Takeaways.** Updating UND is rewarded when both conditions hold in the velocity space:

**C1** *Part of* UND*'s update is outside* GEN*'s span, so that* UND *can realize changes that* GEN *cannot.*

**C2** *For the part of* UND*'s update that* GEN *can also realize, it should align with* GEN*'s high-gain modes to keep the shared changes cheap during joint training.*

## 5 LEVERAGING UPDATE REDUNDANCY FOR STAGED TRAINING

### 5.1 A GEOMETRY-BASED FREEZE CRITERION

The two conditions in Section 4 suggest a straightforward staged training strategy: train UND and GEN jointly first, and then freeze UND's update once the conditions stop holding. We now formalize a single stopping rule that reads the same $\gamma_{\text{excess}}$ and $\tau/\tau_{\text{rand}}$ probes and produces a freeze step.

**Acquisition rates.** Let probe window $k$ span steps $(s_k, s_{k+1}]$ with length $L_k = s_{k+1} - s_k$, and let $D$ be a reference window length[2]. We define the acquisition rate in each window as the per-unit-step excess distinctness, rescaled to the reference spacing:

$$\dot{\gamma}_k = \frac{D}{L_k} \max\big(0,\ -\gamma^{(k)}_{\text{excess}}\big), \qquad \dot{\tau}_k = \frac{D}{L_k} \max\big(0,\ \tau^{(k)}/\tau^{(k)}_{\text{rand}} - 1\big). \tag{3}$$

Here $\dot{\gamma}_k$ is the C1 rate that measures how fast UND writes content GEN cannot reproduce, and $\dot{\tau}_k$ is the C2 rate indicating how fast UND's update is amplified through GEN beyond the random baseline. Normalizing by $L_k$ makes windows of different lengths comparable, and the $\max(0, \cdot)$ clipping removes directionless noise (values where $\gamma_{\text{excess}} > 0$ or $\tau/\tau_{\text{rand}} < 1$ carry no distinctness signal).

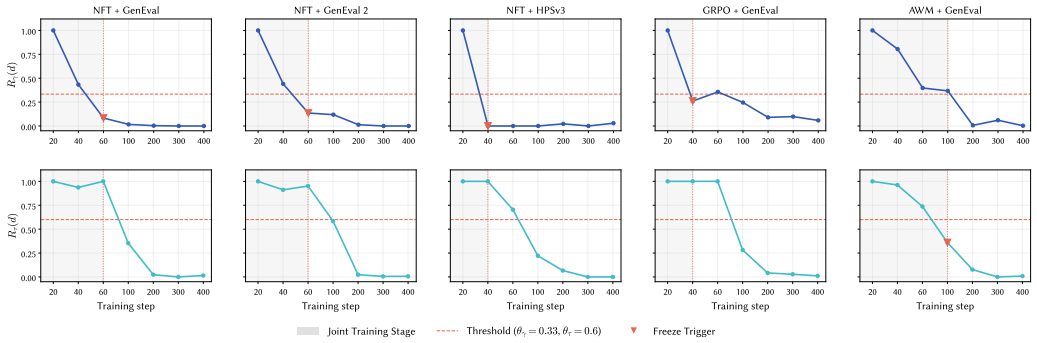



Figure 7: **Acquisition ratios.** Visualization of $R_\gamma(d)$ and $R_\tau(d)$ for each of the five settings. The thresholds are indicated by horizontal lines, and the vertical dashed lines mark the freeze points.

**Peak-relative acquisition.** Each rate is then expressed relative to the best rate seen so far, forming the acquisition ratio:

$$R_\gamma(d) = \frac{\dot{\gamma}_d}{\max_{j \le d} \dot{\gamma}_j}, \qquad R_\tau(d) = \frac{\dot{\tau}_d}{\max_{j \le d} \dot{\tau}_j}. \tag{4}$$

Both ratios start at 1.0 by construction, which measure how much of UND's peak acquisition capacity remains in the current window. For example, $R_\gamma(d) = 1$ means UND is still writing novel content at its peak rate, and $R_\gamma(d) \to 0$ means the C1 signal has collapsed.

[2] $D$ cancels in the peak-relative ratios, so its exact value does not affect the stopping decision.

**The Freeze rule.** We now present the rule that reads $R_\gamma$ and $R_\tau$ and produces a freeze step $t_{\text{freeze}}$:

**Criterion *(UND's Freeze Rule).*** We freeze UND at the end of the first probe window $d \geq 1$ in which either acquisition ratio drops below its threshold:

$$d^* = \min\Big\{d \geq 1 \;:\; R_\gamma(d) < \theta_\gamma \;\text{ or }\; R_\tau(d) < \theta_\tau\Big\}, \qquad t_{\text{freeze}} = s_{d^*+1}. \tag{5}$$

### 5.2 EXPERIMENT SETUP

We validate our method on BAGEL across three reward models (GenEval (Ghosh et al., 2023), GenEval2 (Kamath et al., 2025), and HPSv3 (Ma et al., 2025b)) and three RL algorithms (DiffusionNFT, Flow-GRPO, and AWM), and further test generalization to a second UMM backbone (SenseNova-U1) with DiffusionNFT. All experiments are run on 8 GPUs with 80G VRAM each. We use fixed thresholds $\theta_\gamma = \frac{1}{3}$ and $\theta_\tau = 0.6$ obtained empirically on BAGEL, and apply them unchanged to SenseNova-U1. This shows that our thresholds are insensitive to the model, so no per-model tuning is required. All experiments are run on Flow-Factory (Ping et al., 2026), with BAGEL using its existing implementation and SenseNova-U1 using our own implementation. More details see Appendix A.

Table 2: **Evaluation Results on BAGEL.** Our staged training strategy outperforms GEN-only and naive UND-GEN joint training across reward models and RL algorithms. Numbers in $(\cdot)$ denote the relative improvement over UND-GEN Joint.

| RL Algorithm | Reward Model | GEN Only | Naive UND-GEN Joint | Ours |
|---|---|---|---|---|
| DiffusionNFT | GenEval | 83.89 | 90.85 | **91.55** (+0.77%) |
| | $\text{GenEval2}_{\text{AM}}$ | 82.12 | 82.88 | **83.94** (+1.28%) |
| | $\text{GenEval2}_{\text{GM}}$ | 42.85 | 44.08 | **48.97** (+11.1%) |
| | HPSv3 | 13.42 | 13.69 | **13.89** (+1.46%) |
| Flow-GRPO | GenEval | 89.06 | 90.49 | **92.93** (+2.70%) |
| AWM | GenEval | 83.13 | 84.37 | **86.90** (+3.00%) |

### 5.3 MAIN RESULTS

**Training Efficiency.** Figure 8 compares our staged training strategy with naive joint training as a function of GPU hours. On BAGEL with DiffusionNFT and GenEval, our method improves substantially faster and maintains a clear advantage throughout training. The behavior is different for AWM: joint training achieves a higher reward during the early stage, but its reward later degrades, whereas our method continues improving and eventually reaches a substantially higher and more stable reward. The same benefit transfers to SenseNova-U1. With HPSv3, our method quickly reaches a higher reward and maintains this gap throughout training. With GenEval2, joint training performs better initially, but its reward eventually declines, while our method continues improving and surpasses it in the later stage.

**Evaluation Results.** Table 2 reports the final results on BAGEL. Our method consistently outperforms naive joint training across all reward models and RL algorithms. With DiffusionNFT, our method improves GenEval, GenEval2, and HPSv3, with particularly large gains on $\text{GenEval2}_{\text{GM}}$ (+11.1%). The improvement also holds across different RL algorithms: our method achieves gains of +2.70% with Flow-GRPO and +3.00% with AWM on GenEval. These results demonstrate that the benefit of staged training and the proposed freeze criterion generalizes across reward models and RL algorithms. Moreover, Table 3 further shows our method transfers to a second UMM backbone.

Table 3: **Evaluation Results on SenseNova-U1.**

| Reward | GEN Only | Joint | Ours |
|---|---|---|---|
| $\text{GenEval2}_{\text{AM}}$ | 88.36 | 88.13 | **89.35** (+1.38%) |
| $\text{GenEval2}_{\text{GM}}$ | 61.52 | 62.49 | **66.20** (+5.94%) |
| HPSv3 | 13.25 | 13.38 | **13.86** (+3.59%) |

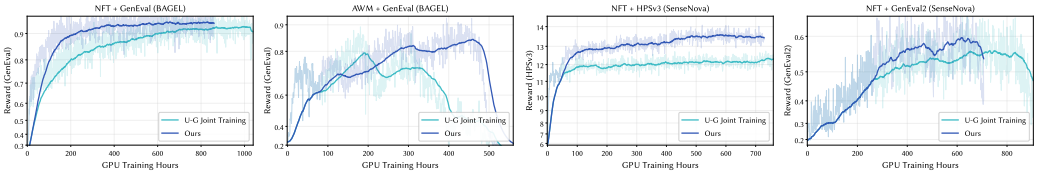



Figure 8: **Training Efficiency.** Reward score v.s. GPU hours on BAGEL and SenseNova-U1. Our staged training strategy achieves better reward with less training time.

### 5.4 Ablations

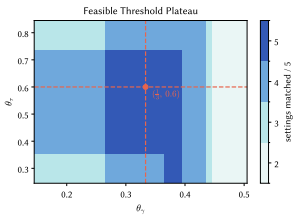



Figure 9: **Threshold calibration.** A broad plateau of $(\theta_\gamma, \theta_\tau)$ values yields a 5/5 recovery rate on BAGEL.

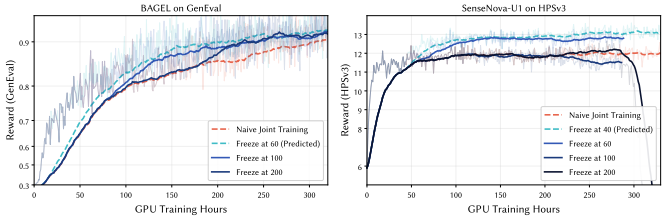



Figure 10: **Freeze-step sensitivity.** Training reward under different freeze steps on BAGEL evaluated by GenEval (left) and SenseNova-U1 evaluated by HPSv3 (right).

**Threshold calibration.** The thresholds $\theta_\gamma = \frac{1}{3}$ and $\theta_\tau = 0.6$ are chosen as the geometric midpoints of a broad feasible region. A grid search on BAGEL shows a wide plateau of $(\theta_\gamma, \theta_\tau)$ pairs that all give a perfect $5/5$ match with the empirical freeze optima (Figure 9), and the selected values sit at the center of this plateau as simple round numbers. The five settings are those in Figure 4, spanning three RL algorithms and three reward models. We directly apply the same thresholds when transferring to SenseNova-U1 without any per-model tuning, confirming that the rule does not require fine-grained cross-model calibration.

**Sensitivity of the freeze step.** Our rule need not freeze at exactly the predicted step to be effective. Figure 10 compares training reward under different freeze steps on BAGEL with GenEval and SenseNova-U1 with HPSv3. In both settings, freeze points near the predicted step do not substantially reduce the reward of our method, which maintains a clear advantage over naive joint training. However, when the freeze point is delayed too far, as seen with the 200-step freeze on HPSv3, training becomes unstable and the reward degrades.

**Fixed learning-rate schedule.** We test a fixed linear decay of UND's learning rate without additional probe computation, and the decayed run stays close to naive joint training rather than matching our gains. This indicates the benefit comes from when UND's updates stop, not from shrinking them on a fixed schedule. More details are in Appendix C.1.

## 6 Conclusion

We studied the role of the understanding expert's updates in image-generation RL for unified multimodal models. Through two velocity-space probes, we showed that its generation-relevant contribution is front-loaded: early updates carry content the generator cannot reproduce and align with the generator's efficient directions, while both signals decay as training proceeds. This motivates a staged training protocol in which a geometry-based freeze rule stops updating the understanding expert once its marginal contribution becomes redundant. Across two backbones, three reward models, and three RL algorithms, our staged training consistently outperforms naive joint training while reaching higher rewards in fewer training hours and leaving understanding intact. Our results suggest that the understanding expert's role should be decided by trajectory-dependent update geometry rather than fixed a priori.

**Limitations.** Both backbones in our work are built upon the MoT architecture, which separates the two capabilities into explicit experts. Exploration of other architectures is left to future work.

### AI USE STATEMENT

In this work, we used generative AI tools for several tasks with required disclosure: implementing parts of our method in the experimental code, providing feedback on our research methodology and experimental design, and organizing and analyzing training logs. We did not use generative AI tools for the remaining tasks with required disclosure. We additionally used them for tasks with recommended disclosure: editing the manuscript to improve readability and creating or editing software code. We reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

### ETHICS STATEMENT

This work studies training protocols for unified multimodal models and uses only publicly available backbones, reward models, and evaluation benchmarks. It does not involve human subjects, personal data, or the release of new datasets.

### REPRODUCIBILITY STATEMENT

Appendix A lists the hyperparameters, trainable modules, and optimization settings shared by naive joint training and our staged protocol, which differ only in the freeze decision rule, and Appendix B gives the detailed construction of the compensability and transport probes, including the finite-difference JVP estimator and the cost per transition.

## REFERENCES

Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, Le Xue, Caiming Xiong, and Ran Xu. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset. *arXiv preprint arXiv:2505.09568*, 2025a.

Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025b.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

Haiwen Diao, Penghao Wu, Hanming Deng, Jiahao Wang, Shihao Bai, Silei Wu, Weichen Fan, Wenjie Ye, Wenwen Tong, Xiangyu Fan, Yan Li, Yubo Wang, Zhijie Cao, Zhiqian Lin, Zhitao Yang, Zhongang Cai, Yuwei Niu, Yue Zhu, Bo Liu, Chengguang Lv, Haojia Yu, Haozhe Xie, Hongli Wang, Jianan Fan, Jiaqi Li, Jiefan Lu, Jingcheng Ni, Junxiang Xu, Kaihuan Liang, Lianqiang Shi, Linjun Dai, Linyan Wang, Oscar Qian, Peng Gao, Pengfei Liu, Qingping Sun, Rui Shen, Ruisi Wang, Shengnan Ma, Shuang Yang, Siyi Xie, Siying Li, Tianbo Zhong, Xiangli Kong, Xuanke Shi, Yang Gao, Yongqiang Yao, Yves Wang, Zhengqi Bai, Zhengyu Lin, Zixin Yin, Wenxiu Sun, Ruihao Gong, Quan Wang, Lewei Lu, Lei Yang, Ziwei Liu, and Dahua Lin. Sensenova-u1: Unifying multimodal understanding and generation with neo-unify architecture. *arXiv preprint arXiv:2605.12500*, 2026.

Fengyi Fu, Mengqi Huang, Shaojin Wu, Yunsheng Jiang, Yufei Huo, Hao Li, Yinghang Song, Fei Ding, Jianzhu Guo, Qian He, Zheren Fu, Zhendong Mao, and Yongdong Zhang. Lance: Unified multimodal modeling by multi-task synergy. *arXiv preprint arXiv:2605.18678*, 2026.

Yuying Ge, Sijie Zhao, Jinguo Zhu, Yixiao Ge, Kun Yi, Lin Song, Chen Li, Xiaohan Ding, and Ying Shan. Seed-x: Multimodal models with unified multi-granularity comprehension and generation. *arXiv preprint arXiv:2404.14396*, 2024.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36: 52132–52152, 2023.

Jitai Hao, Hao Liu, Xinyan Xiao, Qiang Huang, and Jun Yu. Uni-x: Mitigating modality conflict with a two-end-separated architecture for unified multimodal models. *arXiv preprint arXiv:2509.24365*, 2025.

Jixiang Hong, Yiran Zhang, Guanzhong Wang, Yi Liu, Ji-Rong Wen, and Rui Yan. Suder: Self-improving unified large multimodal models for understanding and generation with dual self-rewards. *arXiv preprint arXiv:2506.07963*, 2025.

Jingjing Jiang, Chongjie Si, Jun Luo, Hanwang Zhang, and Chao Ma. Co-reinforcement learning for unified multimodal understanding and generation. *arXiv preprint arXiv:2505.17534*, 2025.

Weiyang Jin, Yuwei Niu, Jiaqi Liao, Chengqi Duan, Aoxue Li, Shenghua Gao, and Xihui Liu. Srum: Fine-grained self-rewarding for unified multimodal models. *arXiv preprint arXiv:2510.12784*, 2025.

Amita Kamath, Kai-Wei Chang, Ranjay Krishna, Luke Zettlemoyer, Yushi Hu, and Marjan Ghazvininejad. Geneval 2: Addressing benchmark drift in text-to-image evaluation. *arXiv preprint arXiv:2512.16853*, 2025.

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*, 2025a.

Jie Liu, Gongye Liu, Jiajun Liang, Ziyang Yuan, Xiaokun Liu, Mingwu Zheng, Xiele Wu, Qiulin Wang, Menghan Xia, Xintao Wang, Xiaohong Liu, Fei Yang, Pengfei Wan, Di Zhang, Kun Gai, Yujiu Yang, and Wanli Ouyang. Improving video generation with human feedback. *arXiv preprint arXiv:2501.13918*, 2025b.

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

Zeyu Liu, Zanlin Ni, Yang Yue, Cheng Da, Huan Yang, Di Zhang, Kun Gai, and Gao Huang. Steering visual generation in unified multimodal models with understanding supervision. *arXiv preprint arXiv:2605.05781*, 2026b.

Zhiheng Liu, Weiming Ren, Xiaoke Huang, Shoufa Chen, Tianhong Li, Mengzhao Chen, Yatai Ji, Sen He, Jonas Schult, Belinda Zeng, Tao Xiang, Wenhu Chen, Ping Luo, Luke Zettlemoyer, and Yuren Cong. Tuna-2: Pixel embeddings beat vision encoders for multimodal understanding and generation. *arXiv preprint arXiv:2604.24763*, 2026c.

Renjie Lu, Xulong Zhang, Xiaoyang Qu, Shangfei Wang, and Jianzong Wang. Diva: Harnessing the representation divergence in unified multimodal models for mutual reinforcement. *arXiv preprint arXiv:2605.25328*, 2026.

Yiyang Ma, Xingchao Liu, Xiaokang Chen, Wen Liu, Chengyue Wu, Zhiyu Wu, Zizheng Pan, Zhenda Xie, Haowei Zhang, Xingkai Yu, Liang Zhao, Yisong Wang, Jiaying Liu, and Chong Ruan. Janusflow: Harmonizing autoregression and rectified flow for unified multimodal understanding and generation. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7739–7751, 2025a.

Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. Hpsv3: Towards wide-spectrum human preference score. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15086–15095. IEEE, 2025b.

Ming Nie, Chunwei Wang, Jianhua Han, Hang Xu, and Li Zhang. Towards unified multimodal interleaved generation via group relative policy optimization. *arXiv preprint arXiv:2603.09538*, 2026.

Xichen Pan, Satya Narayan Shukla, Aashu Singh, Zhuokai Zhao, Shlok Kumar Mishra, Jialiang Wang, Zhiyang Xu, Jiuhai Chen, Kunpeng Li, Felix Juefei-Xu, Ji Hou, and Saining Xie. Transfer between modalities with metaqueries. *arXiv preprint arXiv:2504.06256*, 2025.

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.

Zijun Shen, Sihan Yang, Ruichuan An, Ziyu Guo, Hao Liang, Ming Lu, Renrui Zhang, and Wentao Zhang. Uni-synergy: Bridging understanding and generation for personalized reasoning via co-operative reinforcement learning. *arXiv preprint arXiv:2605.10445*, 2026.

Quan Sun, Yufeng Cui, Xiaosong Zhang, Fan Zhang, Qiying Yu, Zhengxiong Luo, Yueze Wang, Yongming Rao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative multimodal models are in-context learners. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14398–14409, 2024.

Chameleon Team, Mingda Chen, Jacob Kahn, and Shang-Wen Li. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 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. *arXiv preprint arXiv:2311.12908*, 2023.

Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, Yingli Zhao, Yulong Ao, Xuebin Min, Tao Li, Boya Wu, Bo Zhao, Bowen Zhang, Liangdong Wang, Guang Liu, Zheqi He, Xi Yang, Jingjing Liu, Yonghua Lin, Tiejun Huang, and Zhongyuan Wang. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024.

Yuqing Wang, Zhijie Lin, Ceyuan Yang, Yang Zhao, Fei Xiao, Hao He, Qi Zhao, Zihan Ding, Fuyun Wang, Shuai Wang, Youliang Zhang, Haoqi Fan, and Xihui Liu. Representation forcing for bottleneck-free unified multimodal models. *arXiv preprint arXiv:2605.31604*, 2026.

Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, and Ping Luo. Janus: Decoupling visual encoding for unified multimodal understanding and generation. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12966–12977, 2025a.

Penghao Wu, Haiwen Diao, Weichen Fan, Lewei Lu, Dahua Lin, and Ziwei Liu. Uncovering understanding–generation synergy in native unified multimodal models: From representation, task to system. *arXiv preprint arXiv:2609.01607*, 2026.

Size Wu, Wenwei Zhang, Lumin Xu, Sheng Jin, Zhonghua Wu, Qingyi Tao, Wentao Liu, Wei Li, and Chen Change Loy. Harmonizing visual representations for unified multimodal understanding and generation. *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17739–17750, 2025b.

Ji Xie, Trevor Darrell, Luke Zettlemoyer, and XuDong Wang. Reconstruction alignment improves unified multimodal models. *arXiv preprint arXiv:2509.07295*, 2025.

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024.

Shuchen Xue, Chongjian Ge, Shilong Zhang, Yichen Li, and Zhi-Ming Ma. Advantage weighted matching: Aligning rl with pretraining in diffusion models. *arXiv preprint arXiv:2509.25050*, 2025.

Sen Ye, Mengde Xu, Shuyang Gu, Di He, Liwei Wang, and Han Hu. Understanding vs. generation: Navigating optimization dilemma in multimodal models. *arXiv preprint arXiv:2602.15772*, 2026.

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. *arXiv preprint arXiv:2509.16117*, 2025.

Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. *arXiv preprint arXiv:2408.11039*, 2024.

# A TRAINING DETAILS

Table 4 lists the general hyperparameters shared across all (model, method, reward) combinations, and Table 5 details the method-specific hyperparameters. Naive Joint Training and our staged strategy use identical hyperparameters. For the trainable modules, see Table 6.

Table 4: **General hyperparameters** shared across all (model, method, reward) combinations.

| Hyperparameter | Value |
|---|---|
| LoRA rank $r$ | 64 |
| LoRA $\alpha$ | 128 |
| Learning rate | $1\times10^{-4}$ |
| Optimizer | AdamW |
| Weight decay | $1\times10^{-4}$ |
| Betas | $(0.9, 0.999)$ |
| $\epsilon$ | $1\times10^{-8}$ |
| Max grad norm | 1.0 |
| Rollout resolution | 512 |
| Rollout sampling steps | 10 |
| Group size | 16 |
| Evaluation resolution | 1024 |
| Evaluation sampling steps | 50 |

Table 5: **Method-specific hyperparameters.**

| Method | Adv. Agg. | $\#t_{\text{train}}$ | Time Shift | Adv. Clip | Other |
|---|---|---|---|---|---|
| NFT | sum | 2 | 3.0 | 5.0 | $\beta = 1$; EMA decay 0.5 |
| Flow-GRPO | gdpo | 1 | 3.0 | 5.0 | KL v-based $\beta = 0$; EMA decay 0.9 |
| AWM | sum | 2 | 3.0 | 5.0 | ghuber power 0.25; KL $\beta = 10^{-4}$; EMA KL 0.1 |

Table 6: **LoRA target modules** for the understanding (UND) and generation (GEN) experts of BAGEL and SenseNova-U1.

| Expert | LoRA target modules |
|---|---|
| UND (BAGEL) | `self_attn.q_proj`, `self_attn.k_proj`, `self_attn.v_proj`, `self_attn.o_proj`, `mlp.gate_proj`, `mlp.up_proj`, `mlp.down_proj` |
| GEN (BAGEL) | `self_attn.q_proj_moe_gen`, `self_attn.k_proj_moe_gen`, `self_attn.v_proj_moe_gen`, `self_attn.o_proj_moe_gen`, `mlp_moe_gen.gate_proj`, `mlp_moe_gen.up_proj`, `mlp_moe_gen.down_proj` |
| UND (SenseNova-U1) | Same as UND (BAGEL) |
| GEN (SenseNova-U1) | `self_attn.q_proj_mot_gen`, `self_attn.k_proj_mot_gen`, `self_attn.v_proj_mot_gen`, `self_attn.o_proj_mot_gen`, `mlp_mot_gen.gate_proj`, `mlp_mot_gen.up_proj`, `mlp_mot_gen.down_proj` |

# B COMPUTING THE VELOCITY-SPACE PROBES

This appendix details how the compensability $\gamma$ and transport $\tau$ probes of Section 4.1 are computed for a checkpoint transition $k \to k{+}1$. The construction is model-agnostic since it only requires access to the velocity field $v_\theta(z)$ and the ability to perturb $\theta_U$ and $\theta_G$ separately.

## B.1 PROBE SETUP

For each transition we evaluate the velocity field on a fixed probe input $z = (x_t, t, c)$, where $x_t$ is a random latent patch sequence sampled once per (transition, prompt) pair, $t$ is a fixed mid-level timestep, and $c$ is the conditioning prompt. Let

$$v_{\text{base}} = v_{[\theta_U^{(k)};\, \theta_G^{(k)}]}(z) \in \mathbb{R}^D \tag{6}$$

be the velocity at step $k$, with $D$ the flattened dimension of the velocity output. The velocity change induced by updating only the understanding expert is

$$\Delta v_U = v_{[\theta_U^{(k+1)};\, \theta_G^{(k)}]}(z) - v_{\text{base}}. \tag{7}$$

## B.2 COMPENSABILITY VIA SAMPLED JACOBIAN-VECTOR PRODUCTS

Compensability measures the fraction of $\Delta v_U$ that lies in the range of the generation expert's Jacobian $J_G = \partial v / \partial \theta_G$. We approximate this range by sampling $m$ random directions $d_i$ in $\theta_G$ space, each a unit vector over the concatenated parameter blocks, and computing forward finite-difference Jacobian-vector products:

$$y_i = \frac{v_{[\theta_U^{(k)};\, \theta_G^{(k)} + \varepsilon d_i]}(z) - v_{\text{base}}}{\varepsilon}, \qquad i = 1, \ldots, m, \tag{8}$$

with step size $\varepsilon = 10^{-2}$. Stacking the $y_i$ and applying reduced QR decomposition yields an orthonormal basis $Q \in \mathbb{R}^{D \times m}$ for the sampled subspace of $\mathrm{range}(J_G)$. Compensability is the squared projection ratio

$$\gamma(\Delta v_U) = \frac{\|QQ^\top \Delta v_U\|_2^2}{\|\Delta v_U\|_2^2} \in [0, 1]. \tag{9}$$

Since $\mathrm{span}(Q)$ has rank at most $m$, this is a lower bound on the true explained fraction when $m < \mathrm{rank}(J_G)$. We use $m = 64$ throughout.

## B.3 TRANSPORT VIA REVERSE-MODE AUTOGRAD

Transport measures how strongly $\Delta v_U$ couples to the generation expert's parameters through the Jacobian. We form the scalar $\Delta v_U^\top v_{[\theta_U^{(k)};\, \theta_G^{(k)}]}(z)$ with $\Delta v_U$ held fixed as a constant, and differentiate with respect to $\theta_G$:

$$J_G^\top \Delta v_U = \nabla_{\theta_G} \left( \Delta v_U^\top v_{[\theta_U^{(k)};\, \theta_G^{(k)}]}(z) \right). \tag{10}$$

Transport is the norm of this gradient, normalized by the velocity change:

$$\tau(\Delta v_U) = \frac{\|J_G^\top \Delta v_U\|_2}{\|\Delta v_U\|_2}. \tag{11}$$

## B.4 RANDOM CONTROL

To calibrate whether the measured values reflect genuine geometric alignment rather than an artifact of magnitude, we repeat both computations for a random understanding-expert direction of the same norm: sample a unit direction $d_{\text{rand}}$ in $\theta_U$ space, scale it to $\|\Delta\theta_U\|_F$, and compute

$$\Delta v_{\text{rand}} = v_{[\theta_U^{(k)} + \|\Delta\theta_U\|_F \cdot d_{\text{rand}};\, \theta_G^{(k)}]}(z) - v_{\text{base}}, \tag{12}$$

then evaluate $\gamma(\Delta v_{\text{rand}})$ and $\tau(\Delta v_{\text{rand}})$ with the same basis $Q$ and the same autograd procedure. We report the excess $\gamma_{\text{excess}} = \gamma(\Delta v_U) - \gamma(\Delta v_{\text{rand}})$ and the ratio $\tau(\Delta v_U)/\tau(\Delta v_{\text{rand}})$ in the main text.

# C Additional Experimental Results

## C.1 A fixed learning-rate schedule for Und

We test a fixed learning-rate schedule for Und, where the ratio of its learning rate to Gen's is annealed linearly from 1 to 0.1 over 200 steps and held constant afterwards, which removes the probe computation from the protocol. As shown in Figure 11, this variant stays close to naive joint training rather than matching our gains; this indicates that what matters is when Und's updates stop, not that they are small.

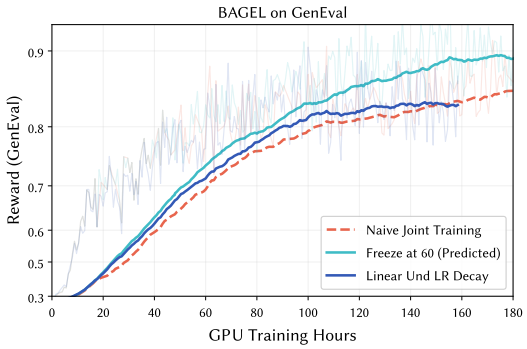



Figure 11: **A fixed learning-rate schedule is not a substitute for freezing Und.** Training reward on BAGEL with GenEval for naive joint training, our staged protocol, and a variant that decays Und's learning rate on a fixed schedule. The scheduled variant stays close to naive joint training, while our freeze rule reaches a higher reward within the same training budget.

## C.2 Overhead of the velocity-space probes

The dominant cost of a probe is the finite-difference JVP loop, which requires $m$ extra forward passes per task on top of a few base forwards. We probe a sequence of five transitions, $(0{\to}20), (20{\to}40), (40{\to}60), (60{\to}100), (100{\to}200)$, with 40 prompts each, and stop as soon as the freeze rule fires. In practice the freeze point is already determined by the second transition in the faster settings and only requires the fifth in the slowest ones, so a run uses between 2 and 5 transitions. Figure 12 reports the resulting overhead relative to a ∼800 GPU-hour training run.

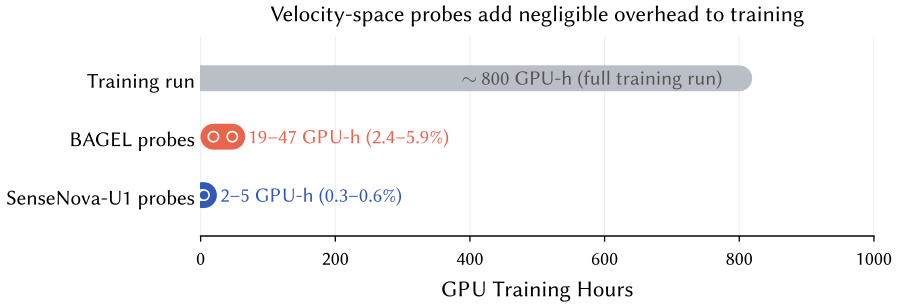



Figure 12: **GPU cost of the velocity-space probes.** A run uses 2–5 transitions with 40 prompts each (80–200 tasks), which adds only a small fraction of GPU-hours on top of a ∼800 GPU-hour training run.

Both models run the same number of forwards per task, and all probes are computed at $512 \times 512$ resolution. The difference comes down to how the two models compress the image: BAGEL ends up with a 1024-token sequence, whereas SenseNova uses 256 tokens, so each forward pass simply takes more time. If the overhead matters, one can also probe BAGEL at a lower resolution to achieve faster evaluation.

### C.3 Understanding capability of the RL-trained models

We report the image-understanding results of the RL-trained models on both backbones, where the RL algorithm is DiffusionNFT, as shown in Table 7. These results indicate that, when the UND expert is not frozen, applying RL to the UMM backbones on generation tasks does not significantly improve or hurt the model's understanding capability.

Table 7: **Understanding evaluation** of the RL-trained models.

| | BAGEL | | | | SenseNova-U1 (w/o think) | | |
|---|---|---|---|---|---|---|---|
| Benchmark | Baseline | RL on GenEval | RL on GenEval2 | RL on HPSv3 | Baseline | RL on GenEval2 | RL on HPSv3 |
| MMMU | 51.33 | **52.11** | 51.30 | 51.78 | **56.56** | 55.11 | 55.44 |
| MMVP | 70.67 | **72.67** | 70.67 | 70.67 | 57.33 | 58.67 | **62** |
| MME-P | **1654** | 1649 | 1650 | 1645 | 1628 | 1614 | **1646** |
| MME-C | **695** | 694 | **695** | 686 | **621** | 594 | 609 |

### C.4 Qualitative examples

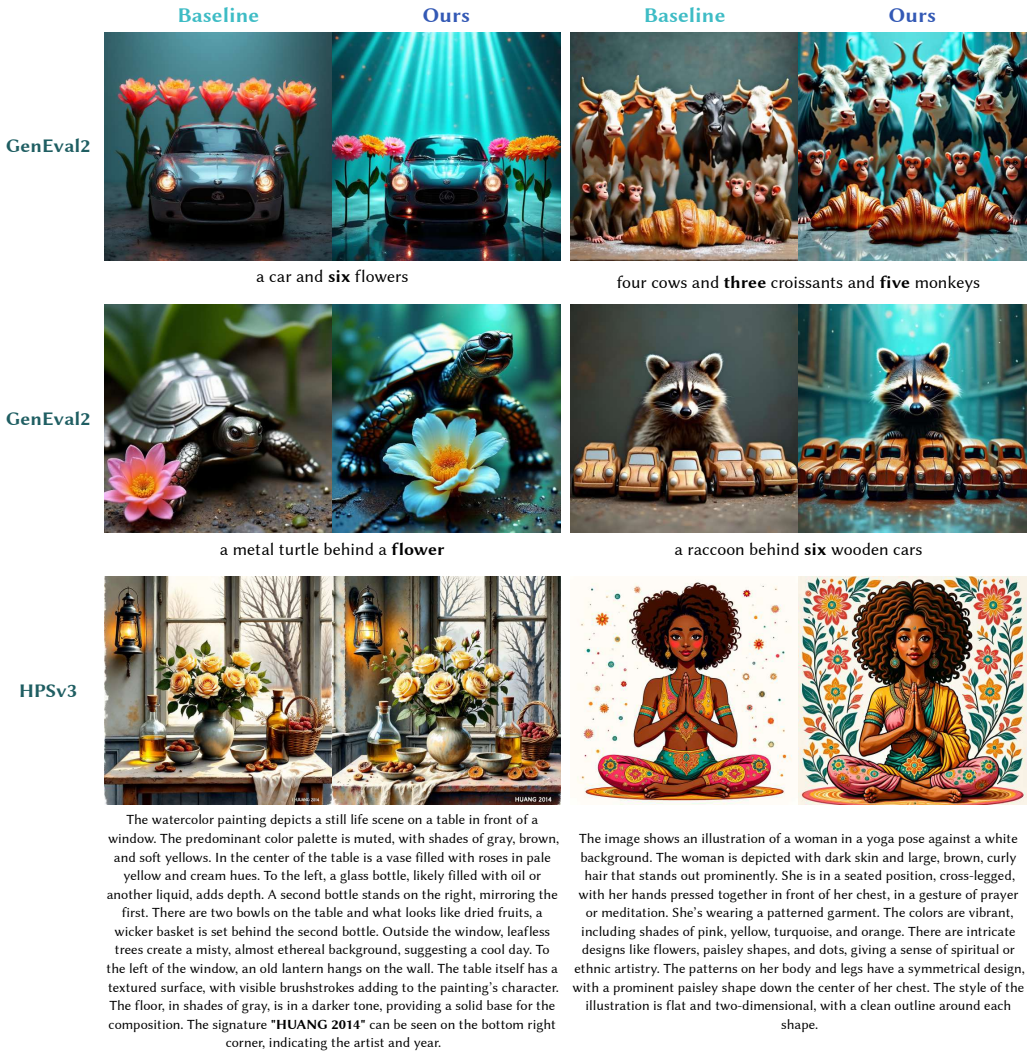



Figure 13: Some inference results of the RL-trained models w/ and w/o our staged training protocol.