REINFORCEMENT CONTINUAL LEARNING FOR ONE-STEP IMAGE GENERATORS

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce **Reinforcement Continual Learning (RCL)**, a post-training framework that converts evaluation feedback into image-level supervision through a generate–construct–select–update loop. For each current-model output, RCL constructs candidate alternatives, determines preference eligibility, selects a suitable target, and learns from it at the original noise input and conditioning through an independently specified matching loss. Distinguishing preference eligibility, source–target compatibility, and matching allows black-box evaluation to guide learning while preserving the inference path. We instantiate RCL on a pMF-H one-step ImageNet generator with retrieved targets and ImageReward or Qwen preferences. Under their respective preference criteria, ImageReward and Qwen scores continue to improve throughout post-training. In comparison with random target-selection policies, the ImageReward-guided variant forms the empirical Pareto frontier for ImageReward and the multi-representation Fréchet-distance ratio (FDr^6). In a single-run comparison with Direct ImageReward, RCL also yields a more favorable ImageReward–FDr^6 trajectory. These results demonstrate the framework's feasibility and characterize the preference–distribution trade-offs of its retrieval-based instantiation.

1 INTRODUCTION

One-step image generation has advanced through consistency learning, distribution-matching distillation, and average-velocity modeling (Song et al., 2023; Yin et al., 2024; Geng et al., 2025), with Pixel Mean Flow extending this direction to latent-free synthesis (Lu et al., 2026). Improving these generators after training requires turning output evaluations into learning signals. Differentiable rewards provide image gradients (Clark et al., 2024), whereas policy-gradient and preference-learning methods can use feedback without differentiating the evaluator (Black et al., 2024; Wallace et al., 2024). Recent forward-process methods further connect reward optimization to regression (Zheng et al., 2026; Bergmeister et al., 2026). The remaining design question is how to construct useful supervision for a particular current output while preserving its one-step generation interface.

Our starting point is to make evaluation actionable through a preferred image target. DrPO already constructs feature targets from reward-ranked samples, and DiffusionOPSD converts reward gradients into refreshed prediction targets (Jiang et al., 2026; Zhou et al., 2026). We investigate a complementary route: acquire candidate images relative to each current output, use preference and compatibility to choose one, and learn it at the original generation input. A preferred image specifies what to learn, and a separately chosen matching loss supplies the gradient.

We introduce **Reinforcement Continual Learning (RCL)**, a framework that couples evaluation-guided target construction with generator learning (Figure 1). Starting from a current output, RCL acquires candidate images and uses feedback to select a preferred target compatible with that output. It then trains the generator to match the target at the original noise and condition. The updated generator supplies the next round of source images, closing the loop between target construction and learning.

The framework separates the choice of supervision from the gradient used to learn it. Evaluation acts through the target; the matching loss acts through the generated image. This division lets

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.

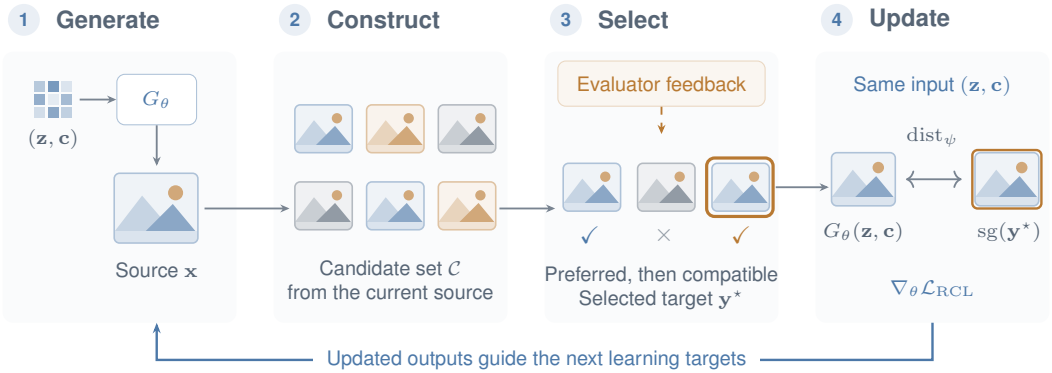


Figure 1. The RCL learning loop. A current output queries candidate targets; evaluation and compatibility determine which image to learn (amber outline). Matching at the original input updates the generator, whose new outputs guide subsequent target construction. Image cards are schematic.

differentiable rewards and black-box judgments guide the same update procedure. Target acquisition can use retrieval or other candidate providers, while inference remains a single generator call.

We study a retrieval-based implementation on ImageNet-1K, starting from a pretrained pMF-H generator. ImageReward and black-box Qwen scores both guide learning through the RCL loop. Comparisons with random target policies and direct reward maximization examine its practical behavior. We track preference and distribution agreement together: both initially improve, while later preference gains accompany a partial reversal in distribution agreement.

Contributions. We contribute (i) RCL, an online post-training framework that links evaluation, image-target construction, and generator updates in a learning loop; (ii) a retrieval-based implementation that accommodates differentiable and black-box rewards through the same matching procedure; and (iii) an empirical study of its learning behavior, target-selection policies, and preference–distribution trade-offs.

2 Related Work

From fast sampling to one-step generation. Fast solvers reduce sampling evaluations (Lu et al., 2022); progressive distillation, consistency models, and distribution-matching distillation compress or directly learn short generation paths (Salimans & Ho, 2022; Song et al., 2023; Yin et al., 2024). MeanFlow models average rather than instantaneous velocity (Geng et al., 2025), and pMF separates prediction and loss spaces for latent-free one-step generation (Lu et al., 2026). RCL addresses the subsequent post-training stage.

Reward gradients and policy optimization. ImageReward, DRaFT, and AlignProp backpropagate differentiable rewards (Xu et al., 2023; Clark et al., 2024; Prabhudesai et al., 2023); LeapAlign shortens training trajectories to improve this route (Liang et al., 2026). DDPO and Flow-GRPO instead use policy gradients (Black et al., 2024; Liu et al., 2025). DiffusionNFT, Advantage Weighted Matching (AWM), and Reinforce Adjoint Matching (RAM) connect reward-based updates to forward noising and score/flow regression (Zheng et al., 2026; Xue et al., 2026; Bergmeister et al., 2026). MeanFlowNFT extends forward-process RL to average-velocity generators (Huang et al., 2026). These developments motivate distinguishing RCL's selected-image matching from both reward differentiation and reward-conditioned denoising objectives.

Preference optimization and filtered self-training. Diffusion-DPO adapts pairwise preference optimization to diffusion (Wallace et al., 2024), DGPO uses online group preferences (Luo et al., 2026), and FlowCPO learns from fixed preferred and dispreferred examples (Han et al., 2026). For fast generators, Diff-Instruct++ and Diff-Instruct* combine rewards with distribution regularization, while PSO adapts timestep-distilled models using pairwise objectives (Luo, 2025; Luo et al., 2025;

Miao et al., 2025). RAFT and ReST establish iterative reward-filtered learning from generated data (Dong et al., 2023; Gulcehre et al., 2023). RCL instead constructs an image target associated with each source input.

Constructed targets and generation-conditioned search. DrPO fits detached feature targets synthesized from reward-ranked interactions and reference drift (Jiang et al., 2026); DiffusionOPSD fits reward-gradient targets at sampled denoising queries (Zhou et al., 2026). IMLE matches each data example to a generated sample (Li & Malik, 2018), while Reverse XM searches in the opposite direction, from a generation to a data target (Gladstone et al., 2026). The latter shares RCL's source-conditioned search and target regression. RCL combines relative preference eligibility, compatibility selection, and explicit image matching at the source's original input. Black-box feedback, detached targets, and online renewal each have precedents; this combination defines the framework studied here. Appendix E and Table 3 detail these relationships.

3 Preliminaries

3.1 One-Step Generator Training

Let $p_{\text{data}}(\mathbf{x}|\mathbf{c})$ be the conditional data distribution and $p(\mathbf{z})$ be a prior distribution. A one-step generator maps $(\mathbf{z}, \mathbf{c})$ directly to an image or latent:

$$\mathbf{x} = G_\theta(\mathbf{z}, \mathbf{c}), \qquad \mathbf{z} \sim p(\mathbf{z}), \quad \mathbf{c} \sim p(\mathbf{c}). \tag{1}$$

The generator may already be trained by a base objective

$$\mathcal{L}_{\text{base}}(\theta) = \sum_{j \in \mathcal{J}} \lambda_j \mathcal{L}_j(\theta), \tag{2}$$

where the components in $\mathcal{J}$ may include teacher distillation, distribution matching, adversarial training, consistency learning, or alignment losses. RCL assumes such a generator is available and asks how to continue improving it without changing its inference-time interface.

3.2 Image Matching in Representation Spaces

RCL learns from image targets that may differ from their sources in pose or layout. Perceptual objectives such as LPIPS provide a way to compare such images through deep features (Zhang et al., 2018). For an image pair $(\mathbf{x}, \mathbf{y})$ and frozen feature extractors $\{\psi_m\}_{m=1}^M$, a composite matching distance can be written as

$$d_{\text{perc}}(\mathbf{x}, \mathbf{y}) = \sum_{m=1}^{M} w_m \left\| \widehat{\psi_m(\mathbf{x})} - \widehat{\psi_m(\mathbf{y})} \right\|_p, \qquad w_m \geq 0, \tag{3}$$

Here, $\widehat{(\cdot)}$ denotes feature normalization, $w_m \geq 0$ weights each representation, and $\|\cdot\|_p$ is the chosen norm. A pixel term may be added. The encoders are frozen but pass gradients from generated-image features to the generator; the target branch is detached. Encoders used for target retrieval and selection may differ from those used for matching.

3.3 Multi-Representation Distribution Evaluation

Distribution evaluation compares sets of images rather than individual source–target pairs. For image sets $\mathcal{A}$ and $\mathcal{B}$, Fréchet distance in representation ϕ is

$$\text{FD}_\phi(\mathcal{A}, \mathcal{B}) = \|\boldsymbol{\mu}_\mathcal{A} - \boldsymbol{\mu}_\mathcal{B}\|_2^2 + \text{Tr}\Big(\boldsymbol{\Sigma}_\mathcal{A} + \boldsymbol{\Sigma}_\mathcal{B} - 2(\boldsymbol{\Sigma}_\mathcal{A}\boldsymbol{\Sigma}_\mathcal{B})^{1/2}\Big), \tag{4}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the empirical mean and covariance of features extracted by ϕ. FID (Heusel et al., 2017) is the special case using Inception features. Let $\mathcal{T}$, $\mathcal{V}$, and $\mathcal{G}$ denote the training, validation, and generated sets. Following Yang et al. (2026), the normalized ratio under representation ϕ_i and its six-representation aggregate are

$$\text{FDr}_{\phi_i}(\mathcal{G}) = \frac{\text{FD}_{\phi_i}(\mathcal{G}, \mathcal{T})}{\text{FD}_{\phi_i}(\mathcal{V}, \mathcal{T})}, \qquad \text{FDr}^6(\mathcal{G}) = \frac{1}{6}\sum_{i=1}^{6} \text{FDr}_{\phi_i}(\mathcal{G}). \tag{5}$$

The six representations are listed in Appendix A. Normalization makes their scales comparable and assigns the real validation set a ratio of 1. Lower FDr^6 indicates closer agreement with the training distribution in the first two feature moments.

4 METHOD

RCL alternates between constructing supervision from the current generator and learning from that supervision. Each iteration has four stages: generate a source, acquire candidates, select a learning target, and update the generator at the source input. The next iteration repeats this process with the updated model. Figure 1 shows the information flow, and Algorithm 1 gives the complete loop.

In RCL, *reinforcement* refers to feedback-guided learning, and *continual* to the renewal of supervision as the generator evolves. We define the target provider, evaluation and selection procedure, and matching loss below, then describe how they interact across updates.

4.1 CONSTRUCTING SUPERVISION FROM CURRENT OUTPUTS

Generate sources and construct candidates. At iteration t, generate a batch of source images $\mathbf{x}_b = G_{\theta_t}(\mathbf{z}_b, \mathbf{c}_b)$, $b = 1, \dots, B$. A target provider $\mathcal{P}$ proposes K candidates for each source:

$$\mathcal{C}_b = \{\mathbf{y}_{b,1}, \dots, \mathbf{y}_{b,K}\} = \mathcal{P}(\mathrm{sg}(\mathbf{x}_b), \mathbf{c}_b). \tag{6}$$

Here, sg denotes stop-gradient. The provider conditions on the source and its generation condition. Our implementation retrieves real images; refinement and teacher/editor proposals are further possibilities discussed in Appendix D.

Filter for preference improvement. First, a binary filter $A_e(\mathbf{y}; \mathbf{x}, \mathbf{c}) \in \{0, 1\}$ keeps a candidate when evaluator e judges it better than the source under the same condition. For a scalar score R_e, this can be expressed as

$$A_e(\mathbf{y}; \mathbf{x}, \mathbf{c}) = \mathbb{I}[R_e(\mathbf{y}, \mathbf{c}) > R_e(\mathbf{x}, \mathbf{c}) + \delta]\,. \tag{7}$$

Here, δ is the required improvement margin. The filter can also use pairwise judgments, rules, human feedback, or black-box evaluations.

Select a compatible target. Let $\mathcal{C}_b^+$ contain the candidates that pass the filter. If this set is nonempty, a selector $\mathcal{S}_q$ chooses one target according to criterion q:

$$\mathcal{C}_b^+ = \{\mathbf{y} \in \mathcal{C}_b : A_e(\mathbf{y}; \mathrm{sg}(\mathbf{x}_b), \mathbf{c}_b) = 1\}, \qquad \mathbf{y}_b^\star = \mathcal{S}_q(\mathcal{C}_b^+ \mid \mathbf{x}_b, \mathbf{c}_b). \tag{8}$$

Using feature distance as the selection criterion chooses the eligible image closest to the source, making similarity a proxy for source–target compatibility. Preference establishes that a candidate is worth learning from, while compatibility relates that candidate to the particular output being improved. If $\mathcal{C}_b^+$ is empty, the source contributes no target loss.

4.2 LEARNING THE TARGET AT THE SOURCE INPUT

The selected image becomes a training target for the source's original input $(\mathbf{z}_b, \mathbf{c}_b)$. Reusing this input preserves the association established during target construction: the target supervises the output for which it was selected. Let $\mathcal{I}_t = \{b : |\mathcal{C}_b^+| > 0\}$ index the sources with a selected target. The generator minimizes

$$\mathcal{L}_{\mathrm{RCL}}(\theta) = \frac{1}{B} \sum_{b \in \mathcal{I}_t} \mathrm{dist}_\psi(G_\theta(\mathbf{z}_b, \mathbf{c}_b), \mathrm{sg}(\mathbf{y}_b^\star))\,. \tag{9}$$

The loss is divided by the full batch size B; sources without a target contribute zero. Here, dist_ψ is a matching loss specified independently of the filter and selector, and differentiable with respect to the generated output. Since targets need not be pixel-aligned with their sources, matching may compare pixels, latents, or perceptual features (Zhang et al., 2018). An optional base objective or regularizer can be retained:

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{base}} + \lambda_{\mathrm{RCL}} \mathcal{L}_{\mathrm{RCL}}. \tag{10}$$

The weight λ_{RCL} controls the contribution of target matching. Targets and indices remain fixed during the update, so the gradient acts only through the generated branch.

4.3 Gradient Flow and Continual Target Construction

The evaluator affects learning through the selected target. Holding the targets and $\mathcal{I}_t$ fixed, the chain rule gives the gradient of Equation 9:

$$\nabla_\theta \mathcal{L}_{\mathrm{RCL}} = \frac{1}{B} \sum_{b \in \mathcal{I}_t} \left(\nabla_{\mathbf{x}_b} \operatorname{dist}_\psi(\mathbf{x}_b, \operatorname{sg}(\mathbf{y}_b^\star)) \right) \frac{\partial G_\theta(\mathbf{z}_b, \mathbf{c}_b)}{\partial \theta}. \tag{11}$$

The matching loss supplies the image gradient, which propagates through the generator. The evaluator influences this gradient through the selected target; neither A_e nor $\mathcal{S}_q$ is differentiated. Thus, the same update accepts a differentiable reward, a discrete comparison, or a score returned by a black-box interface.

After the update, new outputs query the target provider again. Even with a fixed database and evaluator, the supervision can change because it is constructed relative to the evolving generator. This dependence closes the RCL loop: each update changes the model that will construct the next training examples.

Matching a preferred target does not guarantee that every update raises the source reward. We therefore examine preference and reference-distribution agreement throughout training. Target construction and evaluation are training operations; inference remains a single call to $G_\theta(\mathbf{z}, \mathbf{c})$.

Algorithm 1 Reinforcement Continual Learning (RCL)

Require: Generator G_θ, target provider $\mathcal{P}$, preference filter A_e, target selector $\mathcal{S}_q$, matching loss dist_ψ, optimizer, weight λ_{RCL}, and optional base objective $\mathcal{L}_{\mathrm{base}}$.

for each continual post-training iteration **do**
  Sample priors and conditions $\{(\mathbf{z}_b, \mathbf{c}_b)\}_{b=1}^{B}$.
  Generate: obtain sources $\mathbf{x}_b = G_\theta(\mathbf{z}_b, \mathbf{c}_b)$.
  Initialize the selected-target index set $\mathcal{I}_t = \varnothing$.
  for each source b **do**
    Construct: propose $\mathcal{C}_b = \mathcal{P}(\operatorname{sg}(\mathbf{x}_b), \mathbf{c}_b)$.
    Evaluate: retain preferred candidates $\mathcal{C}_b^+$ using A_e.
    if at least one candidate remains **then**
      Select $\mathbf{y}_b^\star = \mathcal{S}_q(\mathcal{C}_b^+ \mid \mathbf{x}_b, \mathbf{c}_b)$ and add b to $\mathcal{I}_t$.
    else
      Set the RCL loss contribution of source b to zero.
    end if
  end for
  Compute $\mathcal{L}_{\mathrm{RCL}}$ on the selected targets, normalized by the full batch size B.
  Update: train G_θ with Equation 10 at the original $(\mathbf{z}_b, \mathbf{c}_b)$.
end for

5 Experiments

We instantiate RCL on ImageNet-1K (Deng et al., 2009) at 256×256, starting from a shared pMF-H one-step generator. Each source retrieves eight same-class training images in Inception space. After preference filtering, the nearest eligible image in DINOv2 space becomes the target for a composite perceptual and pixel matching loss. All runs use 600 post-training steps; Appendix A gives the full protocol.

5.1 Study Design

The study first examines whether the RCL loop can learn from different feedback interfaces. We then compare target-selection policies and direct reward maximization to understand the behavior of this implementation. Across these experiments, we follow the joint evolution of preference and distribution agreement.

ImageReward and Qwen measure in-loop preference; FDr^6 tracks reference-distribution agreement during post-training. We compare evaluated checkpoint trajectories: at comparable FDr^6, higher preference is favorable, and vice versa. Non-dominated sets use raw evaluated checkpoints without smoothing or interpolation.

5.2 Learning from Differentiable and Black-Box Feedback

ImageReward permits differentiation with respect to the input image; the Qwen interface generates text that is parsed into a score and provides no image gradient. RCL uses only the returned scores in both cases. We denote the resulting implementations by IRNearest and QwenNearest. They share the target provider, compatibility criterion, and matching objective, with evaluator-specific preference thresholds. This tests whether the same learning loop can accommodate both feedback interfaces. As a control, RetrievalRandom uniformly selects from the same retrieved neighborhood without preference filtering.

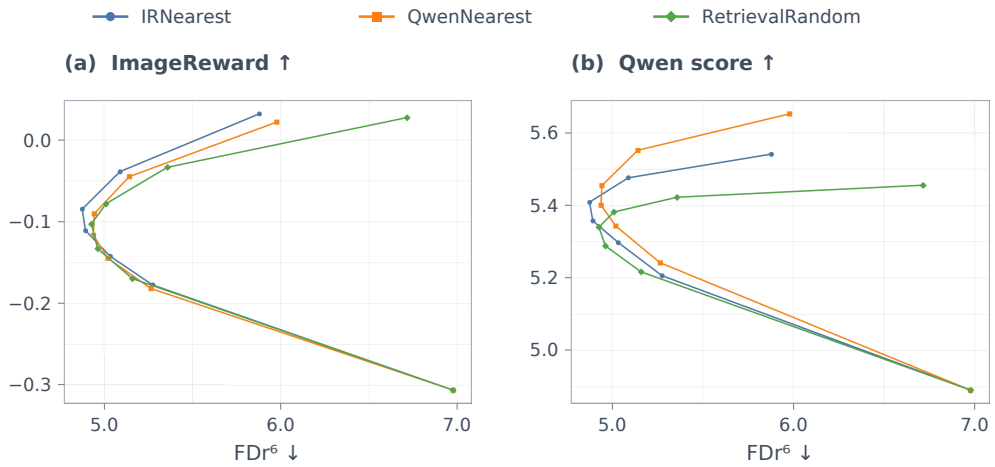


Figure 2. One learning loop, two feedback interfaces. Both RCL variants are evaluated by (a) ImageReward and (b) Qwen. Higher preference and lower FDr^6 are favorable. Each curve is one run from the same initialization.

Learning from feedback. Figure 2 shows evaluator-specific gains: IRNearest increases ImageReward from approximately -0.31 to 0.03, and QwenNearest increases Qwen score from 4.89 to 5.65. The Qwen result shows that feedback without image gradients can guide preference improvement through RCL's target construction and matching loop. At comparable FDr^6, IRNearest generally favors ImageReward, whereas QwenNearest extends toward higher Qwen scores.

Distribution agreement during learning. Appendix Figure 6 shows that FDr^6 initially falls from 6.97 to approximately 4.9, then rebounds as preference improves. Terminal values are 5.88 for IRNearest, 5.97 for QwenNearest, and 6.71 for RetrievalRandom. Both RCL variants retain better reference-distribution agreement than initialization at the terminal checkpoint. The later rebound shows why preference gains and distribution agreement should be assessed together.

5.3 Target-Selection Policies

To examine the target-selection policy used in this implementation, we compare IRNearest with RetrievalRandom and SameClassRandom. The latter samples uniformly from the source class without source-conditioned retrieval. These controls replace the target-selection policy as a whole while retaining the matching update. The random policies reach their lowest observed FDr^6 near step 250, versus step 300 for IRNearest; later preference gains accompany increasing FDr^6. All globally non-dominated displayed checkpoints in this comparison belong to IRNearest (Figure 3).

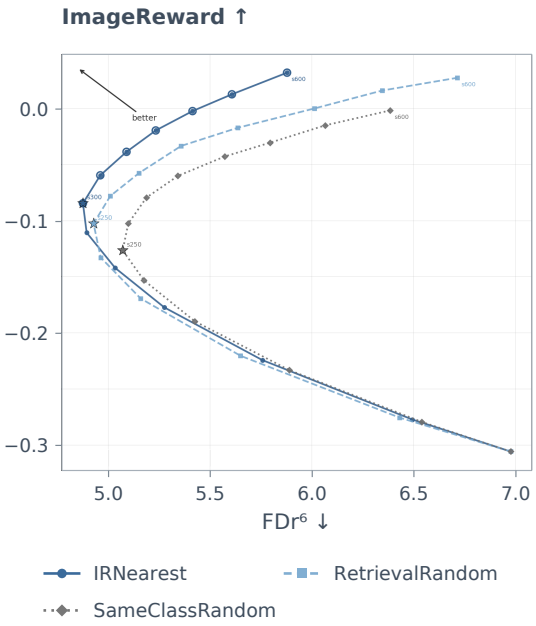


Figure 3. Target-selection policies. Stars mark each method's minimum FDr^6; rings mark globally non-dominated checkpoints. Lines connect evaluated checkpoints. One run per configuration.

5.4 Comparison with Direct Reward Maximization

ImageReward allows us to compare RCL with a second route from evaluation to learning: direct reward maximization. This baseline backpropagates through frozen ImageReward:

$$\mathcal{L}_{\mathrm{direct}}(\theta) = -w\,\mathbb{E}_{\mathbf{z},\mathbf{c}}[R_{\mathrm{IR}}(G_\theta(\mathbf{z},\mathbf{c}),\mathbf{c})]\,. \tag{12}$$

Direct IR and IRNearest share initialization and optimization conditions. RCL additionally uses real-image targets and composite matching, whereas Direct IR uses only the reward objective. Appendix B records preprocessing and gradient computation.

By step 300, RCL reaches approximately 4.88 FDr^6 and -0.085 ImageReward, versus 6.20 and -0.20 for Direct IR (Appendix Figure 7). Direct IR's best observed FDr^6 is about 6.03 near step 550. RCL later attains positive ImageReward as FDr^6 rebounds; its joint trajectory remains more favorable over the evaluated checkpoints (Figure 4).

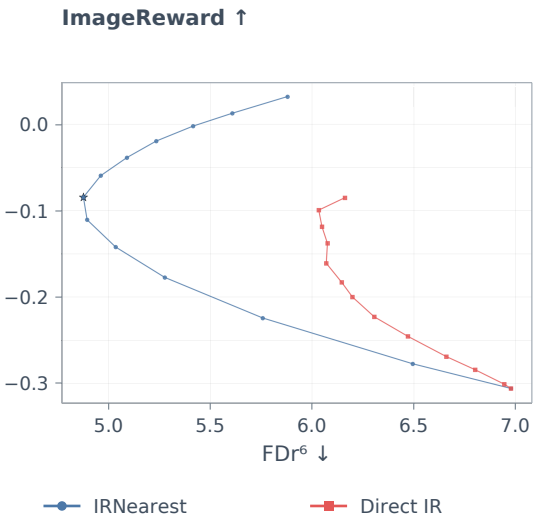


Figure 4. RCL versus Direct IR in the preference–distribution plane. The star marks RCL's minimum-FDr^6 checkpoint. One run per configuration, with shared initialization and optimization conditions.

5.5 Qualitative Results

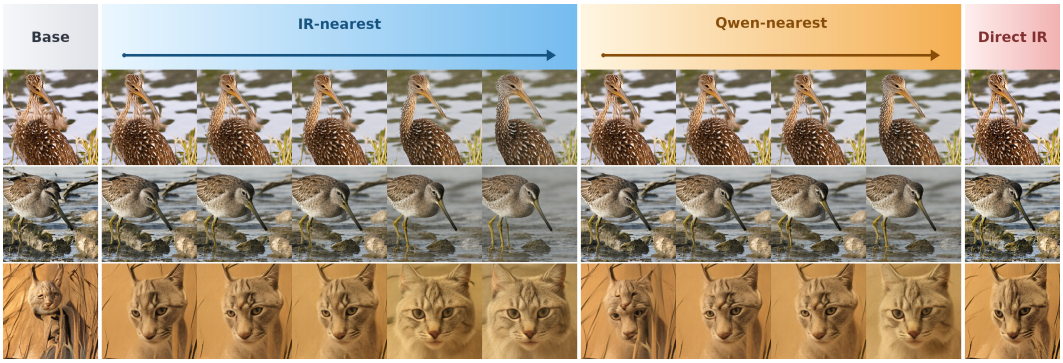


Figure 5. Fixed-source checkpoint comparison. Columns show the base model, successive IRNearest and QwenNearest checkpoints, and the final Direct IR checkpoint. Foregrounds improve under RCL, while backgrounds become increasingly similar.

Figure 5 follows fixed inputs across checkpoints. Both RCL variants produce cleaner subject structure and local detail in these examples, while Direct IR changes them more modestly. The accompanying changes in background appearance illustrate that target matching can alter more than the foreground, complementing the distribution trajectories.

6 Discussion and Conclusion

RCL proposes a modular post-training framework that connects evaluation to learning through image-target supervision. Its target provider, preference filter, compatibility selector, and matching objective have distinct roles within the generate–construct–select–update loop. The retrieval-based configuration is one instantiation of these interfaces. Each component can be replaced or improved while preserving the association between a selected target and the source's original generation input. This separation provides a basis for studying both individual design choices and their interactions.

Online dynamics and complementary objectives. Even with a fixed database and preference criterion, updated outputs can retrieve different candidates and change which ones pass the filter. The observed preference gains and late FDr^6 rebound motivate tracking how target availability, compatibility, and coverage evolve during learning. RCL can also use teachers to propose candidates and retain distillation, distribution matching, or adversarial objectives in $\mathcal{L}_{\text{base}}$. These objectives can complement selected-target matching; the reported experiments use only the matching objective ($\mathcal{L}_{\text{base}} = 0$).

Current evidence and evaluation scope. The study demonstrates learning from ImageReward and black-box Qwen feedback on one generator and dataset, with one run per configuration. Both variants improve their in-loop preference and retain better distribution agreement than initialization. Candidate-level diagnostics, independent human or VLM judgments, repeated runs, and diversity and near-duplicate metrics would help assess target quality and its effects beyond these measurements. Database coverage limits the candidates available to each source, while evaluator behavior and source–target differences in pose or layout affect which targets are useful supervision.

6.1 Replaceable Components and Future Directions

The framework exposes several opportunities for further development; the following alternatives remain to be evaluated.

Candidate acquisition and search budgets. Stronger retrieval features, multi-stage reranking, perturb-and-refine proposals through renoising and denoising, and stronger teachers or fine-tuned editors can provide alternative candidates. Our implementation uses $K = 8$ candidates per source. Larger sets may offer better targets but increase retrieval and evaluation cost and may introduce

mismatches. Adaptive budgets and coarse-to-fine pruning could manage this trade-off. Comparisons should control source streams, acceptance rules, and compute budgets.

Preference filtering and compatibility selection. The evaluator and acceptance margin determine which candidates are eligible; the compatibility selector determines which eligible image supervises a particular source. Alternative scalar evaluators, pairwise or human judgments, and uncertainty-aware thresholds could change eligibility, while task-specific representations or selection rules could improve compatibility. Holding the provider and matching loss fixed would help isolate these choices and their effects on target availability and learning.

Matching objectives and optimization. Our composite loss and LoRA/optimizer configuration are fixed. Alternative feature losses, adaptive weighting, confidence-aware or curriculum updates, distribution regularization, and other optimizers or parameter-efficient schemes could improve transfer from selected targets while limiting overfitting to incidental appearance differences. The separately specified matching objective allows these choices to be studied with the same target-construction procedure.

Target-data sources and computational cost. The ImageNet retrieval pool reuses original training data. Curated external corpora and domain-specific collections could supply new visual knowledge, subject to condition compatibility, domain shift, duplication, and provenance. Target construction adds post-training overhead: a precomputed index amortizes retrieval, whereas refinement-based proposals and VLM evaluation may cost more. Evaluating gains against this cost is essential when replacing components. These operations leave the one-step inference path unchanged.

6.2 Conclusion

RCL links feedback, source-conditioned target construction, and generator updates through a shared learning loop. The retrieval-based study establishes its feasibility with differentiable and black-box evaluators and characterizes the resulting preference–distribution trade-offs. Its modular interfaces support further study of which preferred images provide useful supervision for which current outputs, and how improvements to target acquisition, selection, and matching translate into better generators.

AI use statement

Generative AI tools were used solely for surveying related work, language polishing, manuscript formatting, and figure preparation. The authors are responsible for the paper's claims, citations, results, and final submission.

References

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, et al. Qwen3-VL Technical Report. *arXiv preprint arXiv:2511.21631*, 2025. URL https://arxiv.org/abs/2511.21631.

Andreas Bergmeister, Stefanie Jegelka, Nikolas Nüsken, Carles Domingo-Enrich, and Jakiw Pidstrigach. Reinforce Adjoint Matching: Scaling RL Post-Training of Diffusion and Flow-Matching Models. *arXiv preprint arXiv:2605.10759*, 2026. URL https://arxiv.org/abs/2605.10759.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training Diffusion Models with Reinforcement Learning. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2305.13301.

Kevin Clark, Paul Vicol, Kevin Swersky, and David J Fleet. Directly Fine-Tuning Diffusion Models on Differentiable Rewards. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2309.17400.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *The IEEE Conference on Computer Vision and Pattern Recognition*

(CVPR), 2009. URL https://image-net.org/static_files/papers/imagenet_cvpr09.pdf.

Mingyang Deng, He Li, Tianhong Li, Yilun Du, and Kaiming He. Generative modeling via drifting. *arXiv preprint arXiv:2602.04770*, 2026. URL https://arxiv.org/abs/2602.04770.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. URL https://arxiv.org/abs/2105.05233.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. RAFT: Reward ranked finetuning for generative foundation model alignment. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=m7p5O7zblY.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean Flows for One-step Generative Modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2505.13447.

Alexi Gladstone, Heng Ji, and Yilun Du. Explorative modeling: Unlocking a third pretraining axis and end-to-end generation. *arXiv preprint arXiv:2607.27372*, 2026. URL https://arxiv.org/abs/2607.27372.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud Doucet, Orhan Firat, and Nando de Freitas. Reinforced self-training (ReST) for language modeling. *arXiv preprint arXiv:2308.08998*, 2023. URL https://arxiv.org/abs/2308.08998.

Yansen Han, Shengyi Liao, Peng Sun, Deyuan Liu, Yuanxing Zhang, Pengfei Wan, and Tao Lin. FlowCPO: A Unified Divergence View of Preference Alignment for Flow Models. *arXiv preprint arXiv:2609.09905*, 2026. URL https://arxiv.org/abs/2609.09905.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked Autoencoders Are Scalable Vision Learners. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. URL https://arxiv.org/abs/2111.06377.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. URL https://arxiv.org/abs/1706.08500.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. URL https://arxiv.org/abs/2006.11239.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://arxiv.org/abs/2106.09685.

Yushi Huang, Xiangxin Zhou, Jun Zhang, Liefeng Bo, and Tianyu Pang. MeanFlowNFT: Bringing Forward-Process RL to Average-Velocity Generators. *arXiv preprint arXiv:2607.15273*, 2026. URL https://arxiv.org/abs/2607.15273.

Zhou Jiang, Yandong Wen, and Zhen Liu. Drifting preference optimization for one-step generative models. *arXiv preprint arXiv:2606.02521*, 2026. URL https://arxiv.org/abs/2606.02521.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://arxiv.org/abs/2206.00364.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-Pic: An Open Dataset of User Preferences for Text-to-Image Generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://arxiv.org/abs/2305.01569.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved Precision and Recall Metric for Assessing Generative Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. URL https://arxiv.org/abs/1904.06991.

Ke Li and Jitendra Malik. Implicit Maximum Likelihood Estimation. *arXiv preprint arXiv:1809.09087*, 2018. URL https://arxiv.org/abs/1809.09087.

Zhanhao Liang, Tao Yang, Jie Wu, Chengjian Feng, and Liang Zheng. LeapAlign: Post-Training Flow Matching Models at Any Generation Step by Building Two-Step Trajectories. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. URL https://arxiv.org/abs/2604.15311.

Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating Text-to-Visual Generation with Image-to-Text Generation. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2024. URL https://arxiv.org/abs/2404.01291.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for Generative Modeling. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2210.02747.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2505.05470.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. URL https://arxiv.org/abs/2201.03545.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://arxiv.org/abs/1711.05101.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://arxiv.org/abs/2206.00927.

Yiyang Lu, Susie Lu, Qiao Sun, Hanhong Zhao, Zhicheng Jiang, Xianbang Wang, Tianhong Li, Zhengyang Geng, and Kaiming He. One-step latent-free image generation with pixel mean flows. *arXiv preprint arXiv:2601.22158*, 2026. URL https://arxiv.org/abs/2601.22158.

Weijian Luo. Diff-instruct++: Training one-step text-to-image generator model to align with human preferences. *Transactions on Machine Learning Research*, 2025. URL https://arxiv.org/abs/2410.18881.

Weijian Luo, Colin Zhang, Debing Zhang, and Zhengyang Geng. David and goliath: Small one-step model beats large diffusion with score post-training. In *Proc. of the International Conference on Machine Learning (ICML)*, 2025. URL https://proceedings.mlr.press/v267/luo25s.html.

Yihong Luo, Tianyang Hu, and Jing Tang. Reinforcing Diffusion Models by Direct Group Preference Optimization. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2510.08425.

Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. HPSv3: Towards Wide-Spectrum Human Preference Score. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, pp. 15086–15095, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Ma_HPSv3_Towards_Wide-Spectrum_Human_Preference_Score_ICCV_2025_paper.html.

Zichen Miao, Zhengyuan Yang, Kevin Lin, Ze Wang, Zicheng Liu, Lijuan Wang, and Qiang Qiu. Tuning timestep-distilled diffusion model using pairwise sample optimization. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=fXnE4gB64o.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, et al. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=a68SUt6zFt.

Mihir Prabhudesai, Anirudh Goyal, Deepak Pathak, and Katerina Fragkiadaki. Aligning Text-to-Image Diffusion Models with Reward Backpropagation. *arXiv preprint arXiv:2310.03739*, 2023. URL https://arxiv.org/abs/2310.03739.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *Proc. of the International Conference on Machine Learning (ICML)*, 2021. URL https://arxiv.org/abs/2103.00020.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://arxiv.org/abs/2305.18290.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. URL https://arxiv.org/abs/2112.10752.

Tim Salimans and Jonathan Ho. Progressive Distillation for Fast Sampling of Diffusion Models. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://arxiv.org/abs/2202.00512.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://arxiv.org/abs/2010.02502.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency Models. In *Proc. of the International Conference on Machine Learning (ICML)*, 2023. URL https://arxiv.org/abs/2303.01469.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. URL https://arxiv.org/abs/1512.00567.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, et al. SigLIP 2: Multilingual Vision-Language Encoders with Improved Semantic Understanding, Localization, and Dense Features. *arXiv preprint arXiv:2502.14786*, 2025. URL https://arxiv.org/abs/2502.14786.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion Model Alignment Using Direct Preference Optimization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. URL https://arxiv.org/abs/2311.12908.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://arxiv.org/abs/2304.05977.

Shuchen Xue, Chongjian Ge, Shilong Zhang, Yichen Li, and Zhi-Ming Ma. Advantage Weighted Matching: Aligning RL with Pretraining in Diffusion Models. In *Proc. of the International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2509.25050.

Jiawei Yang, Zhengyang Geng, Xuan Ju, Yonglong Tian, and Yue Wang. Representation Fréchet loss for visual generation. *arXiv preprint arXiv:2604.28190*, 2026. URL https://arxiv.org/abs/2604.28190.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T. Freeman, and Taesung Park. One-step Diffusion with Distribution Matching Distillation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. URL https://arxiv.org/abs/2311.18828.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2410.06940.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. URL https://arxiv.org/abs/1801.03924.

Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. DiffusionNFT: Online Diffusion Reinforcement with Forward Process. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2509.16117.

Wei Zhou, Xiongwei Zhu, Lingdong Kong, Bo Chen, Lei Zhang, Yongyuan Liang, et al. On-Policy Self-Distillation in Diffusion Models. *arXiv preprint arXiv:2608.24646*, 2026. URL https://arxiv.org/abs/2608.24646.

A RETRIEVAL-BASED EXPERIMENTAL PROTOCOL

Table 1. Shared experimental configuration. Candidate-set construction and target selection are specified separately for each study.

Setting	Configuration
Task and data	ImageNet-1K class-conditional generation at 256×256; the data source used to construct candidate targets is specified per experiment.
Base generator	A shared pMF-H one-step generator checkpoint (Lu et al., 2026).
Post-training	LoRA (Hu et al., 2022) with rank 64 and alpha 512; approximately 74.7M trainable parameters (about 7.3%).
Optimization	All experiments use AdamW (Loshchilov & Hutter, 2019) with learning rate 10^{-6}, $\beta = (0.9, 0.95)$, zero weight decay, a constant learning-rate schedule, and no warmup. Training uses 600 steps with global batch size 128 (76,800 source samples).
Training objectives	Every reported comparison sets $\mathcal{L}_{\text{base}} = 0$. RCL runs optimize the selected-target matching loss; Direct IR minimizes $-\operatorname{mean}(R_{\text{IR}})$ with reward weight $w = 1$.
RCL matching loss	MAE, DINOv2, Inception, LPIPS, and pixel ℓ_1 terms weighted by 1.0, 0.7, 0.5, 0.3, and 0.05.
Evaluation	Every 50 steps; fixed sets of 50,000 images for FDr^6 and ImageReward and 10,000 images for Qwen, with shared class order and generation noise.

Table 2. Key protocol for the retrieval-based RCL instantiation. Shared settings appear in Table 1.

Field	Retrieval-based configuration
Task and retrieval pool	ImageNet-1K class-conditional generation at 256×256. Retrieval uses 1,281,167 training images across all 1,000 classes, with 732–1,300 images per class.
Training budget	Starting from the shared pMF-H checkpoint, all runs use rank-64, alpha-512 LoRA for 600 optimizer steps with global batch size 128, processing 76,800 sources. They use the shared optimizer configuration in Table 1, and the original base objective is disabled ($\mathcal{L}_{\text{base}} = 0$).
Target provider	Every source undergoes candidate search. We retrieve the top eight same-class training images by cosine similarity in the Inception feature space.
Candidate acceptance	IRNearest and QwenNearest apply the evaluator-specific margin rules in Equation 13. If none pass, the source contributes zero matching loss without fallback.
Target selection	Among accepted candidates, both methods choose the target nearest to the source in DINOv2 feature space. Thus, the preference evaluator controls eligibility, while the final selector is held fixed.
Random controls	RETRIEVALRANDOM uniformly selects one of the eight Inception-retrieved candidates without preference filtering. SAMECLASSRANDOM removes source-conditioned retrieval and uniformly samples one target from all training images of the source class.
Matching objective	The loss combines MAE, DINOv2, Inception, LPIPS, and pixel ℓ_1 terms with weights 1.0, 0.7, 0.5, 0.3, and 0.05. It is normalized by the full training batch, with zero contributions from rejected sources.
Preference evaluators	ImageReward (Xu et al., 2023) uses ImageReward-v1.0 with package version 1.5. Qwen uses `Qwen/Qwen3-VL-32B-Instruct` (Bai et al., 2025) with deterministic decoding; its exact prompt and parsing rule are given below.
Evaluation	Checkpoints are saved and evaluated every 50 steps. FDr^6 and ImageReward use a fixed set of 50,000 generated images, while Qwen uses a fixed set of 10,000; methods share class order and generation noise.

Distribution evaluation. The six representations in FDr^6 are Inception (Szegedy et al., 2016), ConvNeXt (Liu et al., 2022), DINOv2 (Oquab et al., 2024), MAE (He et al., 2022), SigLIP2 (Tschannen et al., 2025), and CLIP (Radford et al., 2021). Non-dominated sets are computed from raw evaluated checkpoints without smoothing or interpolation. We report one run per configuration.

Preference thresholds. For source s and candidate target t, the evaluator-specific acceptance rules are

$$\begin{aligned} \text{IRNEAREST}: &\quad R_{\text{IR}}(t) - R_{\text{IR}}(s) > 0.5, \\ \text{QWENNEAREST}: &\quad R_{\text{Qwen}}(t) - R_{\text{Qwen}}(s) > 1.8. \end{aligned} \tag{13}$$

Qwen prompting and parsing. For class c, the target description and scoring instruction are:

```
a high-quality photo of a {class_name}

Rate this image for the following target description: {prompt}.
Consider semantic match, realism, image quality, and visible artifacts.
Return only JSON in this exact form: {"score": 0.0}.
The score must be a number from 0 to 10.
```

The response is first parsed as JSON; if that fails, the first numeric value is extracted and clipped to $[0, 10]$. If a scoring attempt fails to produce a valid score, we retry until one is obtained. This retry policy applies to source images, candidate targets, and evaluation images. The Qwen evaluation mean is computed over all 10,000 images in the fixed evaluation set at each checkpoint.

B DIFFERENTIABLE IMAGEREWARD BASELINE

We use ImageReward-v1.0 from `image-reward==1.5` and freeze all reward-model parameters while retaining gradients with respect to the input image. Before scoring, images are clamped to $[0, 1]$, cast to FP32, resized to 224×224 with antialiased bicubic interpolation, and normalized with the CLIP mean and standard deviation. For class c, the text prompt is `a high-quality photo of a {class_name}`, with a maximum token length of 35. We take the CLS representation from the BLIP cross-modal text encoder, pass it through the ImageReward MLP, and standardize the resulting score using the mean and standard deviation stored by the model.

The direct-reward run uses $\mathcal{L}_{\text{direct}} = -\text{mean}(R_{\text{IR}})$, i.e., $w = 1$ in Equation 12, with $\mathcal{L}_{\text{base}} = 0$. Each device evaluates rewards in micro-batches of eight until the complete local batch has been scored. The run shares initialization, trainable parameters, optimizer, update budget, and noise–condition inputs with IRNearest. All other shared optimization settings follow Table 1.

C Additional Trajectories

Figure 6 reports stepwise distribution agreement for IRNearest, QwenNearest, and RetrievalRandom; Figure 7 compares RCL and Direct IR at matched training steps. The corresponding preference–distribution trajectories appear in Figures 2 and 4 in the main text.

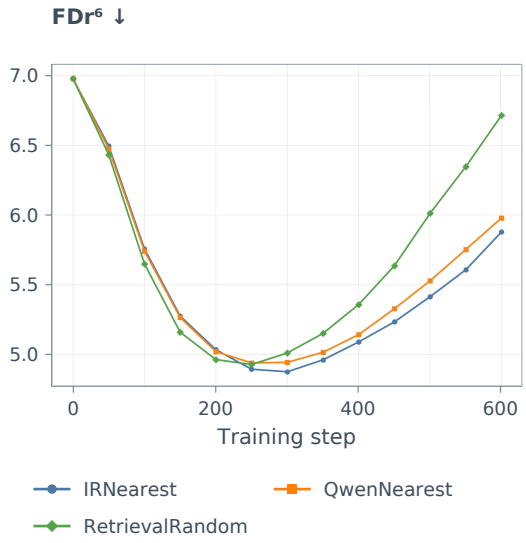


Figure 6. Stepwise reference-distribution agreement. All three configurations initially reduce FDr^6 and later partially rebound. IRNearest and QwenNearest remain below initialization at step 600. Each curve is one run.

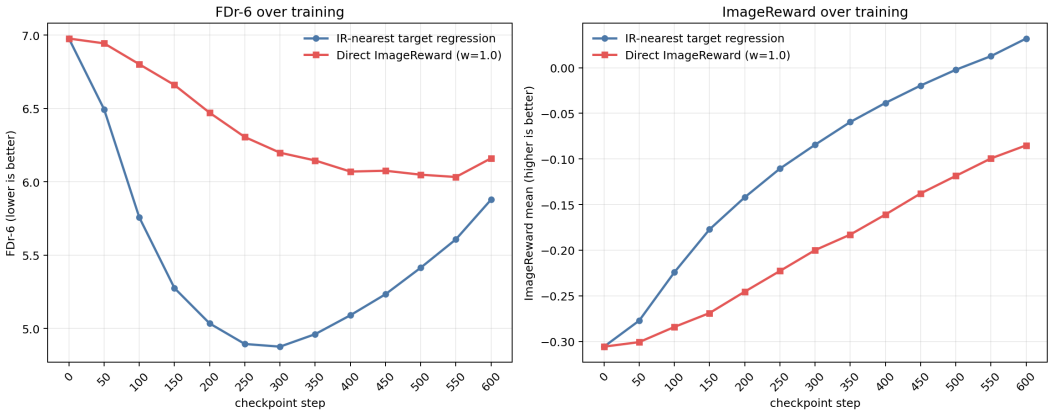


Figure 7. RCL and Direct IR at matched steps. One run per configuration, with shared initialization and optimization conditions. RCL uses additional image-target supervision; Direct IR uses only the reward objective.

D Update and Target-Provider Details

Algorithm 1 in the main text defines the learning loop. This appendix describes the matching and target-provider interfaces used to instantiate it.

Matching interface. RCL leaves the matching objective abstract:

$$\mathrm{dist}_\psi : (\mathbf{x}, \mathbf{y}) \mapsto \mathbb{R}. \tag{14}$$

It can use pixel, latent, perceptual, or task-specific losses that are differentiable with respect to the generated output, and can be combined with base losses or regularizers. The resulting generator gradient is given in Equation 11.

Optional selection rules. The filter and selector can share a criterion ($q = e$): a score may both filter candidates and rank those that remain. Setting $A_e \equiv 1$ bypasses preference filtering while leaving target selection in place.

D.1 Target Providers and Retrieval-Based Instantiation

Target-provider design space. RCL leaves $\mathcal{P}$ replaceable: it may retrieve condition-compatible references, perturb and refine the source through renoising and denoising, or use a stronger teacher or editing model to propose alternatives under the same condition. The provider can combine deterministic or stochastic mechanisms, with its outputs used as detached targets. The interface requires source-conditioned candidates that can serve as compatible supervision after selection.

Retrieval-based provider. Our experiments use same-class retrieval from a fixed real-image database. Let $\mathcal{D} = \{(\mathbf{r}_n, c_n)\}_{n=1}^{N}$ be a reference database, and let a frozen representation ϕ define the normalized feature distance

$$\bar{\phi}(\mathbf{x}) = \frac{\phi(\mathbf{x})}{\|\phi(\mathbf{x})\|_2}, \qquad d_\phi(\mathbf{x}, \mathbf{y}) = 1 - \bar{\phi}(\mathbf{x})^\top \bar{\phi}(\mathbf{y}). \tag{15}$$

For a source $\mathbf{x}$ with condition $\mathbf{c}$, the provider returns the K nearest compatible references

$$\mathrm{NN}_K^{\phi}(\mathbf{x}, \mathbf{c}; \mathcal{D}) = \operatorname*{arg\,min}_{\substack{\mathcal{A} \subseteq \mathcal{D}_{\mathbf{c}} \\ |\mathcal{A}| = K}} \sum_{(\mathbf{r}, c) \in \mathcal{A}} d_\phi(\mathbf{x}, \mathbf{r}), \tag{16}$$

where $\mathcal{D}_{\mathbf{c}}$ is the compatible subset (the same class in our class-conditional setting). The retrieval representation determines which external real images form the source-conditioned candidate neighborhood.

Candidate eligibility is determined by ImageReward or Qwen through the preference-margin rule in Equation 7, while $\mathcal{S}_q$ selects the eligible target closest to the source in DINOv2 space. The evaluator supplies eligibility decisions, DINOv2 supplies the selection criterion, and the matching loss supplies the gradient for the selected target. Because retrieved targets need not be pixel-aligned with their sources, we use the composite pointwise feature-matching distance in Equation 3, augmented with a pixel-space term. The retrieval encoder, preference thresholds, compatibility selector, and exact matching weights are specified in Appendix A.

E Extended Related Work

From iterative generation to direct generators. Diffusion models learn iterative denoising (Ho et al., 2020; Song et al., 2021; Dhariwal & Nichol, 2021; Karras et al., 2022), while latent diffusion reduces its spatial cost (Rombach et al., 2022) and Flow Matching learns continuous transport through velocity regression (Lipman et al., 2023). Fast solvers reduce sampling evaluations (Lu et al., 2022); progressive distillation compresses a teacher's sampling path (Salimans & Ho, 2022); consistency models learn mappings that agree along a trajectory (Song et al., 2023); and DMD matches a one-step student's distribution to a diffusion teacher (Yin et al., 2024). MeanFlow directly models average velocity over an interval (Geng et al., 2025), and pMF extends this route to pixel prediction with a velocity-space loss (Lu et al., 2026). These works supply fast generators; RCL studies their subsequent adaptation through image targets.

Reward differentiation and trajectory policy gradients. ImageReward provides a learned preference model and reward-feedback tuning (Xu et al., 2023). DRaFT and AlignProp propagate reward gradients through generation (Clark et al., 2024; Prabhudesai et al., 2023). LeapAlign uses two-leap training trajectories to make reward backpropagation to early flow steps more tractable (Liang et al., 2026). Diff-Instruct++ combines reward maximization with integral KL regularization; Diff-Instruct*, published as *David and Goliath*, uses score-based divergence regularization (Luo, 2025; Luo et al., 2025). DDPO instead treats denoising as sequential decision-making, and Flow-GRPO enables online policy optimization of flow models through stochastic sampling (Black et al., 2024; Liu et al., 2025). RCL uses evaluator outputs only to construct supervision. Our Direct IR experiment is a matched direct-reward control, not an implementation of every method in this family.

Forward-process reward learning. DiffusionNFT contrasts positively and negatively rewarded generations through a forward-process flow-matching objective, using clean endpoints without requiring reverse sampling trajectories for the policy update (Zheng et al., 2026). AWM is a policy-gradient method that weights score/flow matching by advantages (Xue et al., 2026), whereas RAM

derives reward-corrected regression targets using adjoint-matching optimality and a REINFORCE identity (Bergmeister et al., 2026). MeanFlowNFT constructs an instantaneous-velocity predictor from a MeanFlow parameterization and applies DiffusionNFT to it while retaining average-velocity sampling (Huang et al., 2026). Consequently, neither regression-based reward learning nor post-training fast MeanFlow generators is unique to RCL. Our update instead matches a selected image at the source's original noise and condition, without introducing an auxiliary noising-time training objective.

Pairwise, groupwise, and offline preferences. DPO formulates preference learning through a classification objective (Rafailov et al., 2023); Diffusion-DPO adapts it using a diffusion likelihood bound (Wallace et al., 2024). PSO adapts timestep-distilled generators using training-image/model-sample pairs and also supports offline and online preference pairs (Miao et al., 2025); DGPO learns from online group preferences while allowing deterministic ODE sampling (Luo et al., 2026). FlowCPO derives a contrastive flow-matching surrogate for an offline forward-KL objective using preferred and dispreferred samples (Han et al., 2026). These objectives encode preferences in model fitting. RCL instead uses preference to determine whether an image can supervise a specific current output, then applies a separately specified matching loss.

Renewed supervision and constructed targets. RAFT reward-ranks model outputs for fine-tuning, while ReST alternates data generation with reward-filtered offline learning (Dong et al., 2023; Gulcehre et al., 2023). DrPO constructs detached feature targets from reward-ranked sample interactions and frozen-generator reference drift, supporting black-box rewards and one-step inference (Jiang et al., 2026). DiffusionOPSD uses reward gradients to form bounded positive and negative clean-output targets at sampled denoising queries, fits these targets, and refreshes its behavior policy (Zhou et al., 2026). RCL shares the principle of constructing and renewing supervision. Its retrieval-based instance obtains a concrete real image through source-conditioned search and relative preference filtering; reward gradients are unnecessary, and the target need not be synthesized as a feature displacement.

Matching direction and target search. IMLE assigns each data example a nearby model sample (Li & Malik, 2018). Forward XM likewise fixes data and searches generated candidates, whereas Reverse XM fixes a generation and searches data targets under a reconstruction criterion (Gladstone et al., 2026). Reverse XM also investigates vector-database retrieval in language modeling; this is evidence of a shared search mechanism, not a matched image-generation comparison. RCL's retrieval instance follows the generation-to-data direction, adding source-relative preference eligibility and a separate compatibility selector before matching. This direction does not ensure coverage of every data mode, motivating target-coverage and diversity diagnostics rather than a mode-coverage guarantee.

Representations, evaluators, and distribution agreement. LPIPS compares images using deep features (Zhang et al., 2018); REPA aligns network hidden states with clean-image representations (Yu et al., 2025); FD-loss matches feature moments (Yang et al., 2026); and drifting models build regression targets through attraction to data and repulsion from generated samples (Deng et al., 2026). These are different uses of representations. RCL's reported loss is pointwise matching to selected images, whereas FDr^6 is a set-level evaluation metric. PickScore and HPSv3 model human preferences (Kirstain et al., 2023; Ma et al., 2025), while VQAScore assesses image–text alignment through question answering (Lin et al., 2024). Our Qwen score is a task-specific prompting protocol, not an implementation of these metrics. Independent preference judgments and precision/recall diagnostics (Kynkäänniemi et al., 2019) would complement the reported in-loop scores and feature-moment distances; no such additional results are claimed here.

Table 3. Mechanisms most relevant to RCL. This compares how feedback becomes supervision, not empirical performance. The final row describes our retrieval-based implementation.

Method	Feedback / supervision	Update and relation to RCL
DDPO / Flow-GRPO (Black et al., 2024; Liu et al., 2025)	Rewards on generated endpoints.	Policy gradients on stochastic denoising transitions.
DiffusionNFT / AWM / RAM (Zheng et al., 2026; Xue et al., 2026; Bergmeister et al., 2026)	Rewards determine contrast, weights, or corrected targets.	Score/flow regression after forward noising; no selected external image at the original source input.
MeanFlowNFT (Huang et al., 2026)	Rewarded endpoints and an induced instantaneous-velocity predictor.	Forward-process RL for average-velocity generators; a close fast-generator post-training direction.
FlowCPO (Han et al., 2026)	Fixed preferred and dispreferred examples.	Offline contrastive flow matching, without online target renewal.
DrPO (Jiang et al., 2026)	Reward-ranked interactions plus reference drift.	Detached feature-target regression; already supports black-box rewards and one-step inference.
DiffusionOPSD (Zhou et al., 2026)	Reward gradients construct positive/negative prediction targets.	Detached target fitting at denoising queries, with behavior-policy refresh.
Reverse XM (Gladstone et al., 2026)	Data targets chosen by reconstruction compatibility.	Generation-conditioned search and selected-target regression; no required relative-preference filter.
RCL (retrieval instance)	Source-relative preference filter, then compatible real-image selection.	Match the selected image at the original noise and condition; renew targets from updated outputs.

Scope of the comparison. Table 3 locates RCL among existing mechanisms. The experiments establish feasibility and compare target-selection policies and direct reward maximization on one ImageNet generator. They do not establish superiority over DrPO, MeanFlowNFT, DiffusionOPSD, Reverse XM, or the other methods reviewed here. Matched implementations and compute-aware comparisons are needed to assess those empirical relationships.