FIELD CONSTRAINED BARGAINING DIFFUSIONNFT

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion alignment increasingly requires optimizing several rewards at once, such as visual quality, prompt fidelity, text rendering, and compositional correctness. Existing approaches commonly aggregate these rewards with fixed reward weights or balance their stochastic parameter gradients. Both views are incomplete for Diffusion Negative-aware FineTuning (DiffusionNFT), whose primitive learning signal is a scalar optimality that reweights the current clean-image distribution. A reward vector therefore has to be converted into a scalar target, and a target that appears attractive in reward space may still require a large modification of the forward velocity field. We formulate multi-reward DiffusionNFT as a constrained target-selection problem. For the linear optimality family $o_{\boldsymbol{a}} = r_0 + \boldsymbol{a}^\top \tilde{\boldsymbol{R}}$, we show that the exact distribution-level reward gain is $\boldsymbol{G}(\boldsymbol{a}) = \boldsymbol{C}\boldsymbol{a}/r_0$, where $\boldsymbol{C}$ is the covariance of the centered rewards. We further derive an NFT-native field Gram matrix $\boldsymbol{M}$ for which the population-optimal velocity displacement is exactly $4\boldsymbol{a}^\top \boldsymbol{M}\boldsymbol{a}/\beta_{\mathrm{NFT}}^2$. These two geometries lead to Field-Constrained BargainingNFT (FCB-NFT), a small convex program that maximizes Nash reward surplus under a velocity-field displacement budget and valid-optimality constraints. We also provide a finite-sample conservative estimator of $\boldsymbol{M}$ that avoids learning a conditional field probe. On two four-reward suites, FCB-NFT improves every reward and achieves the best worst-channel normalized gains, reaching 0.89 on OCR and 0.43 on compressibility.

1 INTRODUCTION

Diffusion models are increasingly aligned against several criteria, including visual quality, prompt fidelity, text rendering, and compositional correctness. These criteria have different scales and can conflict: improving one reward may degrade another. Many deployments nevertheless require a single model that performs well across criteria, motivating balanced multi-reward alignment (Golan et al., 2026; Li et al., 2025; Xue et al., 2025). We study this setting with black-box rewards, seeking joint improvement within one model without preference conditioning or multiple specialist models at inference time.

For Diffusion Negative-aware FineTuning (DiffusionNFT) (Zheng et al., 2025), this goal requires choosing what the model should learn. DiffusionNFT assigns each generated image a scalar optimality, which reweights the current clean-image distribution into positive and negative targets. Multiple rewards **must** therefore be converted into one scalar signal, and different combinations specify different distributional targets. The central question is how to select a target that improves the rewards together. Doing so requires understanding how the scalar target redistributes probability across generated images and how this redistribution affects the rewards together.

Existing methods typically approach this target-selection problem through reward weighting or parameter-gradient balancing. Fixed reward weights specify a trade-off without accounting for how rewards vary together on the current rollout distribution. In particular, increasing one reward's coefficient can also affect other rewards through their correlations on the same generated images. Gradient-based methods instead combine reward-specific parameter gradients through projection, minimum-norm optimization, or bargaining (Sener & Koltun, 2018; Yu et al., 2020; Navon et al., 2022). MARBLE (Zhao et al., 2026) applies this perspective to DiffusionNFT. Such gradients describe local optimization directions and depend on the network parameterization and sampled training signals. Their balance does not guarantee balanced reward gains at the distributional target.

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.

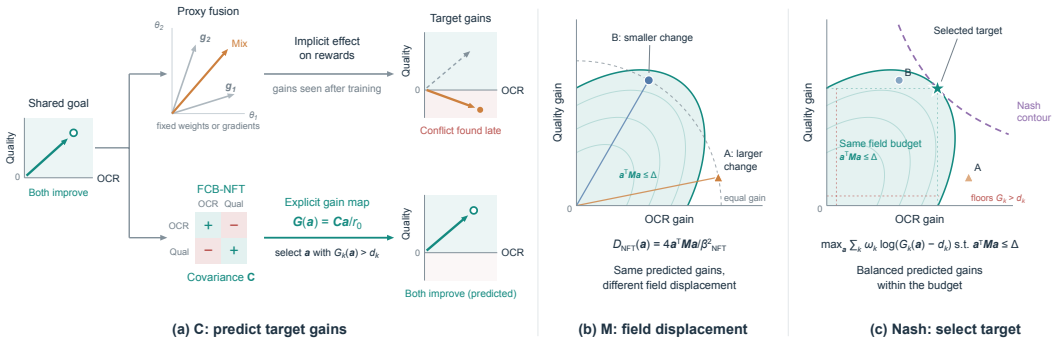

Figure 1: **Overview of FCB-NFT.** (a) Predict target gains through reward covariance. (b) Quantify NFT field displacement. (c) Select a balanced target through field-constrained Nash bargaining.

These strategies therefore leave implicit how a candidate scalar target changes the rewards together under the current rollout distribution.

We make the reward effects of candidate targets explicit through the mechanism of DiffusionNFT. Evaluating a target first requires translating its reward coefficients into reward gains. For the centered linear family $o_{\boldsymbol{a}} = r_0 + \boldsymbol{a}^\top \widetilde{\boldsymbol{R}}$, with nonnegative coefficients $\boldsymbol{a}$, we derive the exact gain of the positive target, $\boldsymbol{G}(\boldsymbol{a}) = \boldsymbol{C}\boldsymbol{a}/r_0$. Here, $\boldsymbol{C}$ is the reward covariance on current rollouts. Its diagonal entries measure available reward variation, while its off-diagonal entries capture cooperation or conflict. This relationship lets us assess each candidate combination by its target gains, accounting for the reward interactions that coefficients alone do not reveal.

The gain map is exact for reweighted targets constructed from the current rollout distribution. As training changes the generator, however, it also changes the reward statistics underlying this map. This motivates conservative target selection: we limit the field change requested in each round before refreshing the rollout statistics. To define this budget through the DiffusionNFT objective, we derive an NFT-native Gram matrix $\boldsymbol{M}$ whose quadratic form gives the exact target-induced displacement of the population-optimal field, $\mathcal{D}_{\mathrm{NFT}}(\boldsymbol{a}) = 4\boldsymbol{a}^\top \boldsymbol{M}\boldsymbol{a}/\beta_{\mathrm{NFT}}^2$. This parameterization-invariant measure captures how reward-induced field responses reinforce or oppose each other, distinguishing targets with similar gains but different field displacements.

Within this budget, we propose Field-Constrained BargainingNFT (FCB-NFT), which maximizes Nash welfare over reward gains above disagreement floors under valid-optimality constraints. Its logarithmic utility encourages balance through diminishing returns as each reward's surplus grows. For fixed statistics, the resulting convex program has one variable per reward and supplies scalar optimalities to the unchanged DiffusionNFT loss. To make the field geometry usable in training, we develop a lightweight estimator using current rollout rewards and a small number of forward-noise probes. It avoids fitting auxiliary conditional models while preserving the matrix structure needed for convex target selection.

Our contributions are threefold:

- We formulate multi-reward DiffusionNFT as target selection and derive an exact covariance-based map from scalarization coefficients to target reward gains.
- We derive an NFT-native field geometry that quantifies target-induced velocity displacement, together with a conservative estimator requiring no additional learned model.
- We develop FCB-NFT for balanced target selection. On two four-reward suites with SD3.5-Medium, it achieves the best worst-channel normalized gain, with ablations demonstrating the value of field geometry beyond reward covariance.

2 Related Work

Multi-reward diffusion alignment. Existing methods balance rewards through sample construction, normalization, or adaptive scalarization. Parrot (Lee et al., 2024) selects batch-wise non-

dominated samples, while CaPO (Lee et al., 2025) calibrates rewards into win rates and constructs preference pairs from Pareto frontiers. Multi-GRPO (Lyu et al., 2025) normalizes reward-specific advantages before aggregation. APEX (Chen et al., 2026) combines per-objective normalization with priorities based on gradient strength, conflict, and remaining improvement room. CMO (Wi et al., 2026) uses reward correlations to emphasize jointly difficult concepts, while PG-OT (Ba et al., 2026) transports dominated samples toward prompt-specific Pareto frontiers. These methods alter training samples or reward aggregation. FCB-NFT instead treats scalar-optimality coefficients as target-selection variables. It uses reward covariance to predict the gains induced by a candidate target under the current rollout distribution, then selects a target subject to an explicit budget on the resulting NFT field displacement.

Gradient-space multi-task optimization. Multi-task optimization typically resolves objective interference in parameter-gradient space. MGDA (Sener & Koltun, 2018) selects a minimum-norm convex gradient combination, PCGrad (Yu et al., 2020) projects away conflicting components, and CAGrad (Liu et al., 2021) seeks a conflict-averse direction. Nash-MTL (Navon et al., 2022) treats gradient combination as bargaining, while FairGrad (Ban & Ji, 2024) applies fair resource allocation to first-order task improvements. MARBLE (Zhao et al., 2026) adapts gradient balancing to DiffusionNFT, periodically computing normalized per-reward gradients, solving for a common direction, and amortizing the resulting coefficients between solver updates. FCB-NFT resolves reward interactions before parameter backpropagation. Its bargaining utility is defined over predicted reward gains above explicit disagreement floors, while its feasible geometry reflects target-induced velocity-field displacement rather than parameter-gradient norms.

Preference-conditioned and front-covering alignment. A complementary line of work learns a family of trade-offs rather than selecting a single model. RiC (Yang et al., 2024) and MIRO (Dufour et al., 2026) condition generation on desired reward levels, enabling preference adjustment without retraining. Diffusion Blend (Cheng et al., 2026) and MSDDA (Zhang et al., 2026) construct new trade-offs at denoising time by combining models aligned separately to individual rewards. ParetoSlider (Golan et al., 2026) trains a preference-conditioned diffusion model to approximate a continuous Pareto front, with preferences adjustable at inference. These methods support deployment-time trade-off selection. FCB-NFT instead targets one unconditioned model, automatically selecting a balanced operating point subject to reward floors and a field budget, without preference conditioning, specialist models, or inference-time composition.

3 Background and Problem Formulation

We study how to construct one scalar DiffusionNFT target from K black-box rewards. Fix a condition c and let $P_c = \pi_{\text{old}}(\cdot \mid c)$ be the current rollout distribution over clean images. All rewards are oriented so that larger is better. Define

$$\widetilde{R}_k = R_k - \mathbb{E}_{P_c}[R_k], \qquad \widetilde{\boldsymbol{R}} = (\widetilde{R}_1, \dots, \widetilde{R}_K)^\top. \tag{1}$$

Bold symbols denote vectors and matrices, while indexed scalar components remain unbolded. Centering removes additive offsets and fixes the mean of the linear optimality introduced below. We retain reward units when constructing the target geometry. Reference scales enter the bargaining surplus separately.

DiffusionNFT receives a scalar optimality $o(\boldsymbol{x}_0, c) \in [0, 1]$, with $m_o = \mathbb{E}_{P_c}[o] \in (0, 1)$, and defines

$$Q_o^+ = \frac{o}{m_o} P_c, \qquad Q_o^- = \frac{1 - o}{1 - m_o} P_c. \tag{2}$$

The positive target increases the mass of samples with large o, while the negative target uses the complementary weights. Combining multiple rewards therefore amounts to selecting a scalar optimality and, through it, a distributional target. To describe how this target changes the model, let $z = (\boldsymbol{x}_t, c, t)$ and consider the forward process

$$\boldsymbol{x}_t = \alpha_t \boldsymbol{x}_0 + \sigma_t \boldsymbol{\epsilon}, \qquad \boldsymbol{v} = \dot{\alpha}_t \boldsymbol{x}_0 + \dot{\sigma}_t \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}). \tag{3}$$

For the population analysis, we take the old field to be the conditional mean,

$$\boldsymbol{v}_{\text{old}}(z) = \mathbb{E}[\boldsymbol{v} \mid z]. \tag{4}$$

DiffusionNFT forms two implicit branches and optimizes

$$\boldsymbol{v}_\theta^{\pm} = \boldsymbol{v}_{\text{old}} \pm \beta_{\text{NFT}}(\boldsymbol{v}_\theta - \boldsymbol{v}_{\text{old}}), \tag{5}$$

$$\mathcal{L}_{\text{NFT}} = \mathbb{E}\big[o\|\boldsymbol{v}_\theta^{+} - \boldsymbol{v}\|_2^2 + (1-o)\|\boldsymbol{v}_\theta^{-} - \boldsymbol{v}\|_2^2\big]\,. \tag{6}$$

Our decision variable is the scalar target o. We select it using its reward gains and requested field change, then apply the original DiffusionNFT loss and rollout update.

4 Field-Constrained BargainingNFT

We construct the scalar NFT target through its reward gains and requested field change. We first derive these two geometries, then show how to obtain the required statistics from rollouts. Finally, we combine them through field-constrained Nash bargaining and use the selected target in the original NFT training loop.

4.1 Linear scalar targets and reward-gain geometry

A reward coefficient does not directly specify its target gain, because reweighting samples for one reward also affects the others through their correlations. To expose these interactions, we use the centered linear family and its reward covariance,

$$o_{\boldsymbol{a}}(\boldsymbol{x}_0, c) = r_0 + \boldsymbol{a}(c)^\top \widetilde{\boldsymbol{R}}(\boldsymbol{x}_0, c), \qquad \boldsymbol{a}(c) \in \mathbb{R}_+^K, \tag{7}$$

$$\boldsymbol{C}(c) = \mathbb{E}_{P_c}[\widetilde{\boldsymbol{R}}\widetilde{\boldsymbol{R}}^\top]. \tag{8}$$

Here $r_0 \in (0,1)$ is a neutral baseline, and nonnegative coefficients preserve each reward's orientation. Centering gives $\mathbb{E}_{P_c}[o_{\boldsymbol{a}}] = r_0$ for every $\boldsymbol{a}$. The same coefficients apply across all forward timesteps of a rollout group, so each noisy state inherits one clean-image target. Let $Q_{\boldsymbol{a}}^{+}$ denote Eq. (2) with $o = o_{\boldsymbol{a}}$.

Proposition 1 (exact reward gains). For every valid $o_{\boldsymbol{a}}$, the positive target changes reward k by

$$G_k(\boldsymbol{a}; c) := \mathbb{E}_{Q_{\boldsymbol{a}}^{+}}[R_k] - \mathbb{E}_{P_c}[R_k] = \frac{1}{r_0}[\boldsymbol{C}(c)\boldsymbol{a}]_k. \tag{9}$$

Consequently, $\boldsymbol{G}(\boldsymbol{a}; c) = \boldsymbol{C}(c)\boldsymbol{a}/r_0$.

Substituting $o_{\boldsymbol{a}}$ into the positive tilt yields this identity directly from centering. The diagonal entries C_{kk} measure available reward variation, while off-diagonal entries C_{kj} determine how emphasizing reward j affects reward k. Thus, nonnegative coefficients can still induce a negative gain through adverse cross-effects. The covariance map makes these effects explicit, allowing us to select targets by their reward gains.

4.2 NFT-native field geometry

The gain map in Eq. (9) exactly evaluates targets built from the current rollout distribution. Since model updates change the reward statistics underlying this map, we select targets that request controlled changes before refreshing those statistics. We quantify the requested change through the population-optimal response of the DiffusionNFT objective, providing a budget on target-induced field displacement.

Proposition 2 (optimal NFT field response). Under Eq. (4), the population optimum of Eq. (6) satisfies

$$\boldsymbol{v}_o^*(z) - \boldsymbol{v}_{\text{old}}(z) = \frac{2}{\beta_{\text{NFT}}}\text{Cov}(o(\boldsymbol{x}_0, c), \boldsymbol{v} \mid z). \tag{10}$$

The response depends on the association between optimality and the velocity uncertainty remaining after z is observed. For the linear target, this response separates into reward-specific fields,

$$\boldsymbol{B}_k(z) = \text{Cov}(\widetilde{R}_k, \boldsymbol{v} \mid z), \qquad \boldsymbol{B}(z) = [\boldsymbol{B}_1(z), \ldots, \boldsymbol{B}_K(z)], \tag{11}$$

$$\text{Cov}(o_{\boldsymbol{a}}, \boldsymbol{v} \mid z) = \boldsymbol{B}(z)\boldsymbol{a}, \tag{12}$$

$$\boldsymbol{M}(c) = D_v^{-1}\mathbb{E}_{z\sim\zeta_c}[\boldsymbol{B}(z)^\top \boldsymbol{B}(z)]. \tag{13}$$

Here D_v is the velocity dimension and ζ_c includes the timestep distribution and fixed loss weighting. The columns $\boldsymbol{B}_k(z) \in \mathbb{R}^{D_v}$ form a matrix $\boldsymbol{B}(z) \in \mathbb{R}^{D_v \times K}$, so $\boldsymbol{M} \in \mathbb{R}^{K \times K}$ is positive semidefinite. Its diagonal entries measure individual field energies. Its off-diagonal entries indicate whether reward-induced field responses reinforce, oppose, or remain nearly orthogonal, adding information beyond clean-reward covariance.

Theorem 1 (exact target-induced field displacement). Under Eq. (4), the target in Eq. (7) induces

$$\begin{aligned}\mathcal{D}_{\mathrm{NFT}}(\boldsymbol{a}; c) &:= \mathbb{E}_{z\sim\zeta_c}\left[D_v^{-1}\|\boldsymbol{v}^*_{\boldsymbol{a}}(z) - \boldsymbol{v}_{\mathrm{old}}(z)\|_2^2\right] \\ &= \frac{4}{\beta^2_{\mathrm{NFT}}}\boldsymbol{a}^\top \boldsymbol{M}(c)\boldsymbol{a}. \end{aligned} \tag{14}$$

Together, $\boldsymbol{C}$ and $\boldsymbol{M}$ describe the gains offered by a target and the field movement it requests. The displacement is defined between velocity functions and is invariant to network reparameterization. It measures the requested population-optimal response, independently of whether finite optimization reaches that field. To use this geometry for target selection, we next make it computable from rollout data.

4.3 Estimating field geometry from rollouts

Computing $\boldsymbol{M}$ directly requires the conditional fields $\boldsymbol{B}_k(z)$. To avoid fitting additional models for these fields, we construct a sample-based field metric from reward vectors and old-model velocity residuals, using a small number of forward-noise probes per rollout.

Let $\boldsymbol{e} = \boldsymbol{v} - \boldsymbol{v}_{\mathrm{old}}(z)$ and $q = D_v^{-1}\|\boldsymbol{e}\|_2^2$. Under Eq. (4), $\mathbb{E}[\widetilde{R}_k\boldsymbol{e} \mid z] = \boldsymbol{B}_k(z)$. Conditional Jensen's inequality relates these observable reward-residual products to the exact field metric:

$$\boldsymbol{M} \preceq \mathbb{E}\Big[q\,\widetilde{\boldsymbol{R}}\widetilde{\boldsymbol{R}}^\top\Big], \tag{15}$$

$$\boldsymbol{a}^\top\boldsymbol{M}\boldsymbol{a} \le \mathbb{E}\Big[q\,(\boldsymbol{a}^\top\widetilde{\boldsymbol{R}})^2\Big]. \tag{16}$$

The matrix moment in Eq. (15) uses the same rollout and forward-process measure as $\boldsymbol{M}$ and bounds its quadratic form in every coefficient direction. We estimate this moment from samples, writing $\widehat{\boldsymbol{M}}$ for the resulting empirical field metric. The upper-bound guarantee holds at the population level.

For each prompt, draw B clean images and center their rewards by the group mean. Let $\boldsymbol{S} \in \mathbb{R}^{B\times K}$ collect the centered reward vectors $\widetilde{\boldsymbol{R}}_i^\top$ as rows. For each of L probes per image, sample a timestep and forward noise, construct the velocity target $\boldsymbol{v}_{i\ell}$, and evaluate the old field at $z_{i\ell}$. We compute

$$\widehat{\boldsymbol{C}} = \frac{1}{B-1}\boldsymbol{S}^\top\boldsymbol{S}, \tag{17}$$

$$q_{i\ell} = D_v^{-1}\|\boldsymbol{v}_{i\ell} - \boldsymbol{v}_{\mathrm{old}}(z_{i\ell})\|_2^2, \tag{18}$$

$$\widehat{\boldsymbol{M}} = \frac{1}{BL}\sum_{i=1}^{B}\sum_{\ell=1}^{L} q_{i\ell}\widetilde{\boldsymbol{R}}_i\widetilde{\boldsymbol{R}}_i^\top. \tag{19}$$

Weighting reward outer products by velocity-residual energy links reward co-variation to the old model's field residuals. When residual energy varies systematically with rewards, this changes the relative cost of combining reward directions. Nonnegative weights preserve positive semidefiniteness and hence convex target selection. The additional computation consists of L old-model probes per sample, with no auxiliary learned model. We stabilize the estimates with an EMA and symmetrization, and add a trace-normalized ridge to $\widehat{\boldsymbol{M}}$. The resulting statistics provide the gain and field information needed to select a target.

4.4 Field-constrained bargaining and training

With target gains and field change characterized, the remaining choice is how to distribute improvement across rewards. We use Nash welfare, whose logarithmic utility gives diminishing marginal value to additional surplus on each channel. Let $d_k(c)$ be the disagreement gain and $s_k^{\mathrm{ref}} > 0$ a fixed or slowly updated reference scale. The surplus $G_k(\boldsymbol{a}; c) - d_k(c)$ measures improvement above

the floor. A zero floor requires positive gain, while a negative floor permits a bounded decrease. Reference scales make the utility dimensionless.

Writing the selection rule with population statistics, FCB-NFT solves

$$\max_{\boldsymbol{a}\in\mathbb{R}_+^K} \quad \sum_{k=1}^{K} \omega_k \log\left(\frac{[\boldsymbol{C}(c)\boldsymbol{a}]_k/r_0 - d_k(c)}{s_k^{\mathrm{ref}}}\right) \tag{20}$$

$$\text{s.t.} \quad \boldsymbol{a}^\top \boldsymbol{M}(c)\boldsymbol{a} \le \Delta(c), \tag{21}$$

$$\frac{1}{r_0}[\boldsymbol{C}(c)\boldsymbol{a}]_k > d_k(c), \quad k = 1,\dots,K, \tag{22}$$

$$\epsilon_o \le r_0 + \widetilde{\boldsymbol{R}}(\boldsymbol{x}_{0,i}, c)^\top \boldsymbol{a} \le 1 - \epsilon_o, \quad i = 1,\dots,B. \tag{23}$$

The constant $4/\beta_{\mathrm{NFT}}^2$ is absorbed into Δ. The constraints bound field displacement, enforce positive surplus, and keep sampled optimalities valid. The objective is concave and the feasible set is convex, with one variable per reward. The solution is Pareto efficient within the corresponding feasible target-gain set. Convexity, the KKT interpretation, single-reward consistency, and reward-unit invariance are detailed in the Appendix A.

In training, $\boldsymbol{C}$ and $\boldsymbol{M}$ are replaced with the estimates from Section 4.3. The Pareto-efficiency statement then applies to the estimated target gains and does not imply global Pareto efficiency after neural-network training. The positive priorities ω_k are initialized uniformly and calibrated during training.

Training procedure. At each outer iteration, the rollout model generates B images per prompt. We center their rewards without group standardization, estimate $\widehat{\boldsymbol{C}}$ and $\widehat{\boldsymbol{M}}$, and solve Eq. (20) with these statistics. The solution defines cached optimalities $o_i = r_0 + \widetilde{\boldsymbol{R}}_i^\top \boldsymbol{a}^*$. Statistics and solver outputs are stop-gradient quantities and remain fixed during group reuse. We then optimize the original DiffusionNFT loss in Eq. (6) and update the rollout model through its standard EMA. Thus, target selection enters training solely through the scalar optimalities. Pseudocode is provided in algorithm 1 (Appendix B.2).

5 Experiments

Our experiments test three claims. First, selecting the scalar target by field-constrained bargaining improves four competing rewards more evenly than fixed, gradient-space, and adaptive scalarizations, and avoids the degenerate solutions of a strongly conflicting suite (Section 5.2). Second, the empirical field metric $\widehat{\boldsymbol{M}}$ improves target selection beyond reward covariance, with additional structure concentrated in cross-reward interactions (Section 5.3). Third, the solver is numerically reward-unit invariant and adds about 2% wall-clock overhead (Section 5.4).

5.1 Experimental setup

Models and reward suites. We fine-tune SD3.5-Medium with LoRA using a matched reimplementation of DiffusionNFT on the OCR prompts of Liu et al. (2025). All methods use the same checkpoint, prompts, optimizer, reward queries, and update budget. The *OCR suite* combines OCR, PickScore (Kirstain et al., 2023), HPSv2 (Wu et al., 2023), and CLIPScore (Hessel et al., 2021). The harder *compressibility suite* replaces CLIPScore by the JPEG-compressibility reward of Black et al. (2024), $-\mathrm{kB}/500$. This reward opposes the preference rewards and admits both blank-image and text-on-blank solutions.

Baselines. We compare against the pretrained model, one single-reward DiffusionNFT specialist per reward, and three fixed scalarizations: equal-weight raw summation, summation after global normalization, and per-prompt group z-scoring followed by summation. We also include a validation-tuned fixed sum selected from a coefficient grid, MARBLE (Zhao et al., 2026) as a gradient-space method, and APEX (Chen et al., 2026) as an adaptive multi-reward baseline. Reward-only bargaining retains the Nash objective and $\widehat{\boldsymbol{C}}$ but replaces the field metric by the identity in the solver's normalized coefficient coordinates. FCB-NFT uses the full empirical field metric $\widehat{\boldsymbol{M}}$. Applicable baselines share centered reward observations.

Table 1: Multi-reward alignment on the OCR suite. Entries show raw scores (normalized gains), and "Worst" is $\min_k g_k^{\text{norm}}$. Specialist entries come from separate single-reward models. Bold denotes the best value among multi-reward methods.

Method	OCR	PickScore	HPSv2	CLIPScore	Worst ↑
Base model	0.125 (0.00)	0.772 (0.00)	0.192 (0.00)	0.270 (0.00)	0.00
Specialist	0.968 (1.00)	0.929 (1.00)	0.348 (1.00)	0.325 (1.00)	n/a
Multi-reward methods					
Equal weighted sum	**0.962** (0.99)	0.762 (−0.06)	0.139 (−0.34)	0.264 (−0.11)	−0.34
Global-normalized sum	0.929 (0.95)	0.855 (0.53)	0.289 (0.62)	0.312 (0.76)	0.53
Group-z-score sum	0.858 (0.87)	0.891 (0.76)	0.325 (0.85)	0.323 (0.96)	0.76
Tuned fixed sum	0.953 (0.98)	0.862 (0.57)	0.286 (0.60)	0.311 (0.75)	0.57
Reward-only bargaining	0.860 (0.87)	0.880 (0.69)	0.282 (0.58)	0.322 (0.95)	0.58
MARBLE	0.944 (0.97)	0.841 (0.44)	0.270 (0.50)	0.315 (0.82)	0.44
APEX	0.928 (0.95)	0.865 (0.59)	0.271 (0.51)	0.302 (0.58)	0.51
FCB-NFT	0.878 (0.89)	**0.911** (0.89)	**0.336** (0.92)	**0.326** (1.02)	**0.89**

Table 2: Multi-reward alignment on the compressibility suite. Entries show raw scores (normalized gains), and "Worst" is $\min_k g_k^{\text{norm}}$. Specialist entries come from separate single-reward models. Bold denotes the best value among multi-reward methods.

Method	OCR	Compress.	PickScore	HPSv2	Worst ↑
Base model	0.125 (0.00)	−0.206 (0.00)	0.772 (0.00)	0.192 (0.00)	0.00
Specialist	0.968 (1.00)	−0.009 (1.00)	0.929 (1.00)	0.348 (1.00)	n/a
Multi-reward methods					
Equal weighted sum	0.000 (−0.15)	**−0.010** (0.99)	0.704 (−0.43)	0.113 (−0.51)	−0.51
Group-z-score sum	0.000 (−0.15)	**−0.010** (0.99)	0.704 (−0.43)	0.113 (−0.51)	−0.51
Global-normalized sum	0.962 (0.99)	−0.141 (0.33)	0.752 (−0.13)	0.119 (−0.47)	−0.47
Tuned fixed sum	0.961 (0.99)	−0.090 (0.59)	0.818 (0.29)	0.205 (0.08)	0.08
Reward-only bargaining	0.000 (−0.15)	**−0.010** (0.99)	0.703 (−0.44)	0.113 (−0.51)	−0.51
MARBLE	0.000 (−0.15)	**−0.010** (0.99)	0.704 (−0.43)	0.113 (−0.51)	−0.51
APEX	0.961 (0.99)	−0.060 (0.74)	0.835 (0.40)	0.230 (0.24)	0.24
FCB-NFT	**0.964** (1.00)	−0.122 (0.43)	**0.870** (0.62)	**0.294** (0.65)	**0.43**

Metrics and configuration. The main result tables report each raw reward with its normalized gain in parentheses. We normalize improvement over the base model by the corresponding single-reward specialist,

$$g_k^{\text{norm}}(\pi) = \frac{R_k(\pi) - R_k(\pi_{\text{base}})}{R_k(\pi_k^{\text{single}}) - R_k(\pi_{\text{base}})}. \tag{24}$$

Here $R_k(\pi)$ denotes the evaluated mean reward. A normalized gain of 0 matches the base model and 1 matches the corresponding specialist. Our primary balance metric is $\min_k g_k^{\text{norm}}$, reported as "Worst". Unless stated otherwise, $r_0 = 0.5$, $B = 12$, $L = 1$, and $\epsilon_o = 0.03$. We use the same single-reward calibration rule for Δ across suites. More details are in the Appendix B.

5.2 Multi-reward alignment

OCR suite. FCB-NFT achieves the most balanced improvement on this suite: the best worst-channel gain, 0.89, together with the strongest PickScore, HPSv2, and CLIPScore among multi-reward methods (Table 1). Raw summation attains the highest OCR score but reduces the other three rewards, yielding a worst normalized gain of -0.34. Global normalization, tuned fixed weights, reward-only bargaining, MARBLE, and APEX remain imbalanced, with worst gains of 0.53, 0.57, 0.58, 0.44, and 0.51, respectively, and group-z summation reaches 0.76. The advantage of FCB-NFT therefore lies in the balance across reward channels rather than in any single reward.

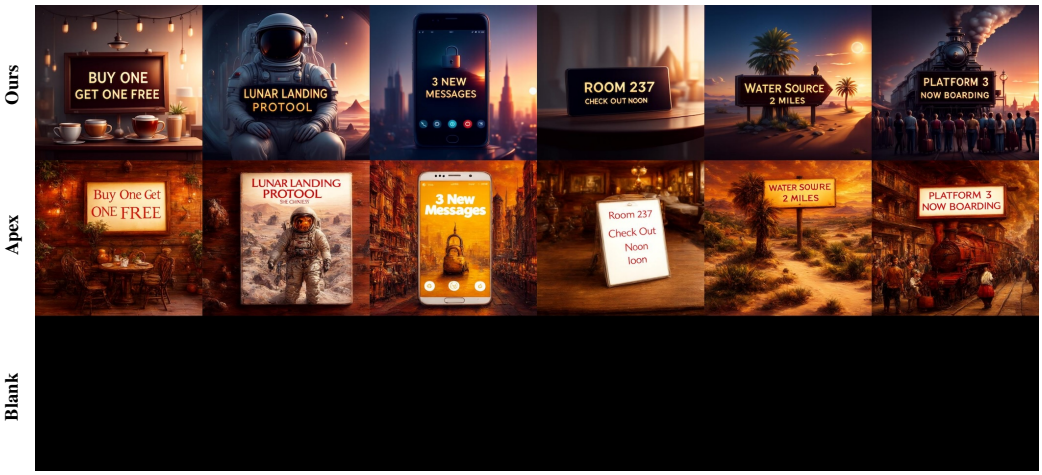

Figure 2: Qualitative comparison on the compressibility suite with SD3.5-Medium. FCB-NFT preserves text and visual content, while competing scalarizations can favor blank outputs.

Compressibility suite. FCB-NFT attains the best worst-channel gain under this conflict, 0.43, while most baselines fall into one of two degenerate solutions (Table 2 and Fig. 2). Blank images abandon OCR, and text-on-blank images retain OCR but lose semantic preference. Raw summation, group-z summation, reward-only bargaining, and MARBLE collapse to blank images (more precisely, these methods all converge to pure black images) within 100 iterations: their OCR scores fall to zero, compressibility approaches the specialist value, and both preference rewards regress. Global normalization retains OCR but settles on text-on-blank outputs with PickScore and HPSv2 below the base model.

Tuned fixed weights and APEX improve all four rewards, with worst normalized gains of 0.08 and 0.24. FCB-NFT reaches 0.43, retaining near-ceiling OCR (0.964), increasing PickScore from 0.772 to 0.870 and HPSv2 from 0.192 to 0.294, and improving compressibility from -0.206 to -0.122. It allocates less improvement to compressibility than APEX while achieving stronger OCR and preference rewards, yielding the best worst-channel gain under this conflict.

5.3 Validating the NFT field geometry

The main results evaluate the complete method. We next isolate whether the empirical field metric $\widehat{\boldsymbol{M}}$ adds useful information beyond reward covariance. As defined in Eq. (19), it estimates the population bound using residual-weighted reward outer products. For the raw group estimates, write $q_{i\ell} = \bar{q}_g + \widetilde{q}_{i\ell}$, where $\bar{q}_g = (BL)^{-1} \sum_{i,\ell} q_{i\ell}$. Then

$$\widehat{\boldsymbol{M}} = \frac{B-1}{B} \bar{q}_g \widehat{\boldsymbol{C}} + \frac{1}{BL} \sum_{i=1}^{B} \sum_{\ell=1}^{L} \widetilde{q}_{i\ell} \widetilde{\boldsymbol{R}}_i \widetilde{\boldsymbol{R}}_i^{\top}. \tag{25}$$

The first term is a scaled covariance budget, which controls the empirical variance of the scalar tilt. The second records the association between residual energy and reward outer products. If residual energy were independent of rewards, this additional structure would vanish at the population level. We test whether this association changes the selected target.

Table 3: Field-geometry ablation on the OCR suite. All rows differ only in the field matrix. $\kappa_g \widehat{\boldsymbol{C}}$ matches the scale of $\widehat{\boldsymbol{M}}$, while shuffled-q removes reward dependence but preserves finite-sample noise. "Active" is the fraction of groups with a binding field budget.

Field matrix	Uses q	OCR	PickScore	HPSv2	CLIPScore	Worst ↑	Active (%)
$\boldsymbol{I}$ (reward-only bargaining)	no	0.860	0.880	0.282	0.322	0.58	100
$\kappa_g \widehat{\boldsymbol{C}}$ (scale-matched)	scale only	0.872	0.880	0.286	0.325	0.60	100
Shuffled-q $\widehat{\boldsymbol{M}}$ (noise-only shape)	permuted	0.869	0.878	0.287	0.319	0.61	100
$\widehat{\boldsymbol{M}}$ (FCB-NFT)	yes	0.878	0.911	0.336	0.326	0.89	100

Table 4: Structure of $\widehat{M}$ beyond $\widehat{C}$. Relative deviations ρ are split into diagonal and off-diagonal entries; shuffled-q gives the finite-sample noise floor. "Diagonal spread" compares the largest and smallest $\widehat{M}_{kk}/\widehat{C}_{kk}$.

Suite	Statistic	$\widehat{M}$	Shuffled-q	Ratio	Diagonal spread
OCR	ρ, all entries	0.205	0.112	$1.8\times$	$< 3\%$
	ρ_{diag} / ρ_{off}	0.060/0.195	0.064/0.087		
Compress.	ρ, all entries	0.53	0.35	$1.5\times$	$< 23\%$
	ρ_{diag} / ρ_{off}	0.188/0.495	0.195/0.275		

Effect on target selection. The full field metric improves reward balance over the covariance and shuffled controls. Table 3 fixes the Nash objective, $\widehat{C}$, and the remaining constraints. The scale-matched metric $\kappa_g \widehat{C}$ yields a worst normalized gain of 0.60, compared with 0.89 for the full $\widehat{M}$. Shuffling q within each group breaks its association with rewards while retaining finite-sample fluctuations and yields 0.61. Together with the identity metric's 0.58, these controls fall within 0.03 of one another and remain below the full metric. These comparisons identify the association between residual energy and rewards as useful information for target selection. Relative to the scale-matched metric, PickScore's normalized gain rises from 0.69 to 0.89 and HPSv2's from 0.60 to 0.92, while OCR and CLIPScore remain similar. The field budget binds in every group and the other constraints are almost never active.

Cross-reward structure. The additional structure in $\widehat{M}$ is concentrated in cross-reward interactions (Table 4). Off-diagonal deviations from $\kappa_g \widehat{C}$ are approximately $2.2\times$ and $1.8\times$ those of the shuffled-q control on the OCR and compressibility suites, respectively. Diagonal deviations remain comparable to the control on both suites: 0.060 versus 0.064 on OCR and 0.188 versus 0.195 on compressibility. This pattern indicates that residual weighting primarily adds structure to the interactions between reward directions, changing their estimated field cost when combined. Together with Table 3, these results support the role of reward-dependent residual structure in balanced target selection.

5.4 Invariance and efficiency

Reward-unit invariance. We rescale rewards as $R'_k = \gamma R_k + b$ with $\gamma \in \{0.01, 0.1, 10, 1000\}$ on single or all channels, transform d_k and s_k^{ref} accordingly, re-solve, and map back to the original units. Over about 1.4×10^5 paired solves, the selected optimalities change by at most 1.1×10^{-4}, and coefficients and predicted gains by under 0.4%. Raw summation shifts optimalities by up to 0.76. These results support numerical reward-unit invariance of the implemented solver, consistent with the property established in Appendix A.4.

Efficiency. The convex program takes 9 ms per group and the old-model probes add 1.0 s per training step. Over $1{,}000$ steps, FCB-NFT averages 74.9 s per step versus 73.2 s for raw summation, an overhead of approximately 2.3%, with unchanged peak memory. Querying four reward models contributes additional cost relative to single-reward DiffusionNFT, but this cost is shared by the multi-reward methods.

6 Conclusion

Multi-reward DiffusionNFT reduces to choosing one scalar target, and FCB-NFT makes that choice a small convex program. The covariance $\boldsymbol{C}$ predicts the reward gains a candidate target induces on the current rollouts, the Gram matrix $\boldsymbol{M}$ measures the velocity-field displacement it requests, and the DiffusionNFT loss is left unchanged. On two four-reward suites the method attains the best worst-channel gain, and ablations attribute this to the structure of M beyond reward covariance. The approach has limits: it selects one balanced operating point rather than a front, its conservative estimator assumes the old field approximates the conditional mean, and we have evaluated it on a single base model and prompt distribution. Extending the target family beyond linear scalarizations is a natural next step.

AI USE STATEMENT

We used generative AI tools (large language models) in the following ways. (i) Writing: to improve the clarity, grammar, and presentation of text written by the authors; AI was not used to draft any section of the paper from scratch. (ii) Retrieval and discovery: to help identify and summarize related work during the literature search; every cited work was subsequently read and verified by the authors. (iii) Mathematical derivations: to assist in deriving and checking intermediate steps of some proofs and equations; the formulation of all mathematical claims, the overall proof strategies, and the final proofs are the authors' own, and every AI-assisted step was independently re-derived and verified by the authors. (iv) Implementation: to assist in writing the code (e.g., evaluation scripts, baseline implementations, and plotting utilities). The method itself, including its design and algorithmic details, was developed by the authors; AI tools were used to translate the authors' specifications into code. All AI-assisted code was reviewed line by line, tested against hand-verified cases, and validated by reproducing the reported results before use.

Generative AI was not used to generate or clean datasets, to develop the theoretical framework or research hypotheses, to design the experiments, to perform qualitative or thematic analysis, to interpret results, or for translation. The research questions, methodology, experimental design, results, and conclusions are the authors' own. We take full responsibility for the content of this paper, including all text, claims, code, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human participants, user studies, personal or sensitive data, or newly collected datasets. Our experiments use publicly available academic datasets and pretrained models released under licenses that permit academic research. We follow the applicable terms of use and do not redistribute restricted assets. To the best of our knowledge, this work raises no additional ethical concerns.

REPRODUCIBILITY STATEMENT

The main paper specifies the proposed method, assumptions, algorithms, and evaluation protocol. The appendix provides the complete derivations and proofs, implementation and training details, hyperparameters, dataset and model information, and additional experimental results. An anonymized implementation, including configuration files and evaluation scripts, will be available at `anonymous.4open.science/r/ICLR2027-FCBNFT`. Together, these materials are intended to support reproduction of all main and supplementary results.

REFERENCES

Ying Ba, Tianyu Zhang, Mohan Zhou, Yalong Bai, Wenyi Mo, Guiwei Zhang, Bing Su, and Ji-Rong Wen. Pareto-guided optimal transport for multi-reward alignment. In *International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2605.13155.

Hao Ban and Kaiyi Ji. Fair resource allocation in multi-task learning. *ArXiv*, abs/2402.15638, 2024.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations (ICLR)*, volume 2024, pp. 4965–4987, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/14f75513f0f1ca01de1e826b52e6b840-Paper-Conference.pdf.

Dongliang Chen, Xinlin Zhuang, Junjie Xu, Luojian Xie, Zehui Wang, Jiaxi Zhuang, Haolin Yang, Liang Dou, Xiao He, Xingjiao Wu, and Ying Qian. APEX: Learning adaptive priorities for multi-objective alignment in vision-language generation. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 4922–4939, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.243. URL https://aclanthology.org/2026.findings-acl.243/.

Min Cheng, Fatemeh Doudi, Dileep Kalathil, Mohammad Ghavamzadeh, and Panganamala Kumar. Diffusion blend: Inference-time multi-preference alignment for diffusion models. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations (ICLR)*, volume 2026, pp. 51151–51181, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/54093759b45e92c1f68ccfc655820464-Paper-Conference.pdf.

Nicolas Dufour, Lucas Degeorge, Arijit Ghosh, Vicky Kalogeiton, and David Picard. MIRO: MultI-Reward cOnditioned pretraining improves T2I quality and efficiency. In *International Conference on Machine Learning (ICML)*, 2026.

Shelly Golan, Michael Finkelson, Ariel Bereslavsky, Yotam Nitzan, and Or Patashnik. Paretoslider: Diffusion models post-training for continuous reward control. *arXiv preprint arXiv:2604.20816*, 2026.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. In *Proceedings of EMNLP*, 2021.

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)*, volume 36, pp. 36652–36663, 2023.

Kyungmin Lee, Xiahong Li, Qifei Wang, Junfeng He, Junjie Ke, Ming-Hsuan Yang, Irfan Essa, Jinwoo Shin, Feng Yang, and Yinxiao Li. Calibrated multi-preference optimization for aligning diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, 2025.

Seung Hyun Lee, Yinxiao Li, Junjie Ke, Innfarn Yoo, Han Zhang, Jiahui Yu, Qifei Wang, Fei Deng, Glenn Entis, Junfeng He, Gang Li, Sangpil Kim, Irfan Essa, and Feng Yang. Parrot: Pareto-optimal multi-reward reinforcement learning framework for text-to-image generation. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part XXXVIII*, pp. 462–478, Berlin, Heidelberg, 2024. Springer-Verlag. ISBN 978-3-031-72919-5. doi: 10.1007/978-3-031-72920-1_26. URL https://doi.org/10.1007/978-3-031-72920-1_26.

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

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393.

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*, 2025.

Qiang Lyu, Zicong Chen, Chongxiao Wang, Haolin Shi, Shibo Gao, Ran Piao, Youwei Zeng, Jianlou Si, Fei Ding, Jing Li, et al. Multi-grpo: Multi-group advantage estimation for text-to-image generation with tree-based trajectories and multiple rewards. *arXiv preprint arXiv:2512.00743*, 2025.

Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. Multi-task learning as a bargaining game. *arXiv preprint arXiv:2202.01017*, 2022.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems (NeurIPS)*, 31, 2018.

Jungmyung Wi, Hyunsoo Kim, and Donghyun Kim. Correlation-weighted multi-reward optimization for compositional generation. *ArXiv*, abs/2603.18528, 2026.

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

Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, et al. Dancegrpo: Unleashing grpo on visual generation. *arXiv preprint arXiv:2505.07818*, 2025.

Rui Yang, Xiaoman Pan, Feng Luo, Shuang Qiu, Han Zhong, Dong Yu, and Jianshu Chen. Rewards-in-context: Multi-objective alignment of foundation models with dynamic preference adjustment. In *International Conference on Machine Learning*, pp. 56276–56297. PMLR, 2024.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Qi Zhang, Dawei Wang, and Shaofeng Zou. Step-level denoising-time diffusion alignment with multiple objectives. *arXiv preprint arXiv:2604.14379*, 2026.

Canyu Zhao, Hao Chen, Yunze Tong, Yu Qiao, Jiacheng Li, and Chunhua Shen. Marble: Multi-aspect reward balance for diffusion rl. *arXiv preprint arXiv:2605.06507*, 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.

A Proofs and Additional Properties

We fix a condition c throughout the proofs and suppress it from the notation. All expectations are population expectations under the current rollout distribution and the forward training measure defined in the main text.

A.1 Exact reward gains

Proof of Proposition 1. Let $\boldsymbol{\mu} = \mathbb{E}_P[\boldsymbol{R}]$, so that $\widetilde{\boldsymbol{R}} = \boldsymbol{R} - \boldsymbol{\mu}$ and $\mathbb{E}_P[\widetilde{\boldsymbol{R}}] = 0$. It follows that $\mathbb{E}_P[o_{\boldsymbol{a}}] = r_0$, and hence

$$Q^+_{\boldsymbol{a}}(\mathrm{d}x_0) = \frac{r_0 + \boldsymbol{a}^\top \widetilde{\boldsymbol{R}}(x_0)}{r_0} P(\mathrm{d}x_0). \tag{26}$$

The reward vector under this target is

$$\mathbb{E}_{Q^+_{\boldsymbol{a}}}[\boldsymbol{R}] = \frac{1}{r_0}\mathbb{E}_P\Big[(r_0 + \boldsymbol{a}^\top \widetilde{\boldsymbol{R}})\boldsymbol{R}\Big] \tag{27}$$

$$= \boldsymbol{\mu} + \frac{1}{r_0}\mathbb{E}_P[\boldsymbol{R}\widetilde{\boldsymbol{R}}^\top]\boldsymbol{a}. \tag{28}$$

Since $\boldsymbol{R} = \widetilde{\boldsymbol{R}} + \boldsymbol{\mu}$,

$$\mathbb{E}_P[\boldsymbol{R}\widetilde{\boldsymbol{R}}^\top] = \mathbb{E}_P[\widetilde{\boldsymbol{R}}\widetilde{\boldsymbol{R}}^\top] + \boldsymbol{\mu}\mathbb{E}_P[\widetilde{\boldsymbol{R}}]^\top = \boldsymbol{C}. \tag{29}$$

Subtracting $\mathbb{E}_P[\boldsymbol{R}] = \boldsymbol{\mu}$ gives

$$\boldsymbol{G}(\boldsymbol{a}) = \mathbb{E}_{Q^+_{\boldsymbol{a}}}[\boldsymbol{R}] - \mathbb{E}_P[\boldsymbol{R}] = \frac{1}{r_0}\boldsymbol{C}\boldsymbol{a}. \tag{30}$$

This is an exact identity whenever $o_{\boldsymbol{a}}$ defines a valid optimality. □

A.2 Population NFT field response

Proof of Proposition 2. Fix a forward state z and write

$$\boldsymbol{\delta} = v_\theta(z) - v_{\mathrm{old}}(z), \qquad e = v - v_{\mathrm{old}}(z). \tag{31}$$

Equation 4 gives $\mathbb{E}[e \mid z] = 0$. From the two branches in equation 5,

$$v^+_\theta - v = \beta_{\mathrm{NFT}}\boldsymbol{\delta} - e, \qquad v^-_\theta - v = -\beta_{\mathrm{NFT}}\boldsymbol{\delta} - e. \tag{32}$$

Conditioning the NFT loss on z and retaining only the terms that depend on $\boldsymbol{\delta}$ yields

$$\ell_z(\boldsymbol{\delta}) = \mathbb{E}\big[o\|\beta_{\mathrm{NFT}}\boldsymbol{\delta} - e\|_2^2 + (1-o)\|-\beta_{\mathrm{NFT}}\boldsymbol{\delta} - e\|_2^2 \mid z\big] \tag{33}$$

$$= \beta^2_{\mathrm{NFT}}\|\boldsymbol{\delta}\|_2^2 - 4\beta_{\mathrm{NFT}}\boldsymbol{\delta}^\top\mathbb{E}[oe \mid z] + \mathrm{const}(z). \tag{34}$$

The conditional objective is strictly convex in $\boldsymbol{\delta}$ for $\beta_{\mathrm{NFT}} > 0$. Its unique minimizer satisfies

$$\boldsymbol{\delta}^* = \frac{2}{\beta_{\mathrm{NFT}}}\mathbb{E}[oe \mid z]. \tag{35}$$

Finally,

$$\mathbb{E}[oe \mid z] = \mathbb{E}[o(v - \mathbb{E}[v \mid z]) \mid z] = \mathrm{Cov}(o, v \mid z), \tag{36}$$

which proves

$$v^*_o(z) - v_{\mathrm{old}}(z) = \frac{2}{\beta_{\mathrm{NFT}}}\mathrm{Cov}(o, v \mid z). \tag{37}$$

□

A.3 EXACT FIELD DISPLACEMENT

Proof of Theorem 1. For $o_{\boldsymbol{a}} = r_0 + \boldsymbol{a}^\top \widetilde{\boldsymbol{R}}$, linearity of conditional covariance gives

$$\mathrm{Cov}(o_{\boldsymbol{a}}, v \mid z) = \sum_{k=1}^{K} a_k \mathrm{Cov}(\widetilde{R}_k, v \mid z) = B(z)\boldsymbol{a}. \tag{38}$$

Proposition 2 therefore implies

$$v^*_{\boldsymbol{a}}(z) - v_{\mathrm{old}}(z) = \frac{2}{\beta_{\mathrm{NFT}}} B(z)\boldsymbol{a}. \tag{39}$$

Taking the squared norm and integrating over the forward training measure gives

$$\mathcal{D}_{\mathrm{NFT}}(\boldsymbol{a}) = \mathbb{E}_{z\sim\zeta}\left[D_v^{-1}\|v^*_{\boldsymbol{a}}(z) - v_{\mathrm{old}}(z)\|_2^2\right] \tag{40}$$

$$= \frac{4}{\beta^2_{\mathrm{NFT}}} \boldsymbol{a}^\top \mathbb{E}_{z\sim\zeta}\left[D_v^{-1} B(z)^\top B(z)\right] \boldsymbol{a} \tag{41}$$

$$= \frac{4}{\beta^2_{\mathrm{NFT}}} \boldsymbol{a}^\top \boldsymbol{M}\boldsymbol{a}. \tag{42}$$

The matrix $\boldsymbol{M}$ is positive semidefinite because it is an expectation of Gram matrices. □

A.4 PROPERTIES OF THE BARGAINING PROBLEM

Convexity. For fixed $\boldsymbol{C}$ and $\boldsymbol{M}$, each surplus $S_k(\boldsymbol{a}) = [\boldsymbol{C}\boldsymbol{a}]_k/r_0 - d_k$ is affine in $\boldsymbol{a}$. Since $\log S_k$ is concave on $S_k > 0$ and $\omega_k > 0$, the Nash objective is concave on its domain. The nonnegative orthant and the optimality-box constraints are convex, while $\boldsymbol{a}^\top \boldsymbol{M}\boldsymbol{a} \le \Delta$ is convex because $\boldsymbol{M} \succeq 0$. Maximizing the concave objective over this convex feasible set is therefore a convex optimization problem.

Constrained Pareto efficiency. Let $\boldsymbol{a}^*$ solve equation 20–equation 23. Suppose another feasible $\boldsymbol{a}'$ satisfies $G_k(\boldsymbol{a}') \ge G_k(\boldsymbol{a}^*)$ for every k and a strict inequality for at least one k. The same relations hold for the surpluses because the disagreement point is fixed. As every term $\omega_k \log(S_k/s_k^{\mathrm{ref}})$ is strictly increasing in S_k, the objective at $\boldsymbol{a}'$ is strictly larger than that at $\boldsymbol{a}^*$, which is a contradiction. Thus, $\boldsymbol{a}^*$ is Pareto efficient within the gain set induced by the current rollout distribution, linear target family, field budget, disagreement point, and validity constraints. This argument does not imply global Pareto optimality after finite neural-network training. □

Interior KKT interpretation. Assume that $\boldsymbol{M}$ is positive definite, the field constraint is active, and the nonnegativity, surplus, and optimality-box constraints are inactive at the solution. Define

$$\lambda_k^* = \frac{\omega_k}{G_k(\boldsymbol{a}^*) - d_k}. \tag{43}$$

With $\gamma > 0$ denoting the multiplier of the field constraint, stationarity of

$$\sum_{k=1}^{K} \omega_k \log \frac{G_k(\boldsymbol{a}) - d_k}{s_k^{\mathrm{ref}}} - \gamma(\boldsymbol{a}^\top \boldsymbol{M}\boldsymbol{a} - \Delta) \tag{44}$$

gives

$$\frac{1}{r_0}\boldsymbol{C}^\top \boldsymbol{\lambda}^* - 2\gamma \boldsymbol{M}\boldsymbol{a}^* = 0. \tag{45}$$

Solving for $\boldsymbol{a}^*$ yields

$$\boldsymbol{a}^* = \frac{1}{2\gamma r_0} \boldsymbol{M}^{-1}\boldsymbol{C}^\top \boldsymbol{\lambda}^*, \tag{46}$$

Boundary solutions are governed by the additional KKT multipliers and are handled directly by the convex program. □

Single-reward consistency. When $K = 1$, every nonzero admissible target has the form $o_a = r_0 + a\widetilde{R}_1$ with $a > 0$. The only target direction is therefore the standard centered single-reward direction. If $C_{11} > 0$, then $G_1(a) = C_{11}a/r_0$ and the Nash objective is strictly increasing in a until a field or validity constraint becomes active. FCB-NFT consequently differs from standard single-reward DiffusionNFT only in how it selects the scalar target magnitude. If $C_{11} = 0$, the current rollout support contains no reward variation and no positive gain is available within this target family. □

Reward-unit invariance. Consider $R'_k = q_k R_k + b_k$ with $q_k > 0$, and define $\boldsymbol{D}_q = \mathrm{diag}(q_1, \ldots, q_K)$. Centering removes the offsets, so

$$\widetilde{\boldsymbol{R}}' = \boldsymbol{D}_q \widetilde{\boldsymbol{R}}, \qquad \boldsymbol{C}' = \boldsymbol{D}_q \boldsymbol{C} \boldsymbol{D}_q. \tag{47}$$

The conditional fields and their Gram matrix transform as

$$B'(z) = B(z)\boldsymbol{D}_q, \qquad \boldsymbol{M}' = \boldsymbol{D}_q \boldsymbol{M} \boldsymbol{D}_q. \tag{48}$$

Set $\boldsymbol{a}' = \boldsymbol{D}_q^{-1}\boldsymbol{a}$, $\boldsymbol{d}' = \boldsymbol{D}_q \boldsymbol{d}$, and $\boldsymbol{s}^{\mathrm{ref}\prime} = \boldsymbol{D}_q \boldsymbol{s}^{\mathrm{ref}}$. Then

$$\boldsymbol{a}'^{\top} \widetilde{\boldsymbol{R}}' = \boldsymbol{a}^{\top} \widetilde{\boldsymbol{R}}, \qquad \boldsymbol{a}'^{\top} \boldsymbol{M}' \boldsymbol{a}' = \boldsymbol{a}^{\top} \boldsymbol{M} \boldsymbol{a}, \tag{49}$$

so the scalar optimality, validity constraints, and field cost are unchanged. Proposition 1 gives $\boldsymbol{G}'(\boldsymbol{a}') = \boldsymbol{D}_q \boldsymbol{G}(\boldsymbol{a})$, and therefore

$$\frac{G'_k(\boldsymbol{a}') - d'_k}{s_k^{\mathrm{ref}\prime}} = \frac{G_k(\boldsymbol{a}) - d_k}{s_k^{\mathrm{ref}}}. \tag{50}$$

The two programs have identical normalized objectives under a one-to-one mapping of their feasible sets and hence return the same scalar target. The same transformation holds for $\overline{\boldsymbol{M}}$ and its empirical estimator before ridge regularization. □

Strict-improvement infeasibility. For $\boldsymbol{d} = 0$, the surplus constraints require $G_k(\boldsymbol{a}) > 0$ for every reward. Proposition 1 makes these inequalities exact statements about the positively reweighted clean distribution. If the program is infeasible, no coefficient vector satisfies all strict gain, field-budget, nonnegativity, and validity constraints simultaneously. The certificate is restricted to this feasible linear target family. It does not exclude a weak Pareto improvement, a nonlinear target, or an improvement after the rollout distribution changes. □

A.5 Conservative field matrix

Proof of $\boldsymbol{M} \preceq \overline{\boldsymbol{M}}$. Let $e = v - v_{\mathrm{old}}(z)$ and collect the random vector-valued reward residuals in

$$Y = e\widetilde{\boldsymbol{R}}^{\top} \in \mathbb{R}^{D_v \times K}. \tag{51}$$

Because $v_{\mathrm{old}}(z) = \mathbb{E}[v \mid z]$,

$$\mathbb{E}[Y \mid z] = \Big[\mathrm{Cov}(\widetilde{R}_1, v \mid z), \ldots, \mathrm{Cov}(\widetilde{R}_K, v \mid z)\Big] = B(z). \tag{52}$$

Moreover,

$$\overline{\boldsymbol{M}} = \mathbb{E}[D_v^{-1} Y^{\top} Y], \qquad \boldsymbol{M} = \mathbb{E}[D_v^{-1} \mathbb{E}[Y \mid z]^{\top} \mathbb{E}[Y \mid z]]. \tag{53}$$

For any $\boldsymbol{u} \in \mathbb{R}^K$,

$$\boldsymbol{u}^{\top}(\overline{\boldsymbol{M}} - \boldsymbol{M})\boldsymbol{u} = \mathbb{E}\big[D_v^{-1}\left(\mathbb{E}[\|Y\boldsymbol{u}\|_2^2 \mid z] - \|\mathbb{E}[Y \mid z]\boldsymbol{u}\|_2^2\right)\big] \tag{54}$$

$$= \mathbb{E}\big[D_v^{-1}\mathbb{E}[\|(Y - \mathbb{E}[Y \mid z])\boldsymbol{u}\|_2^2 \mid z]\big] \geq 0. \tag{55}$$

Hence $\overline{\boldsymbol{M}} - \boldsymbol{M} \succeq 0$. Combining this relation with Theorem 1 shows that replacing $\boldsymbol{M}$ by $\overline{\boldsymbol{M}}$ cannot underestimate the population target-induced field displacement under the old-population-field assumption. □

Positive semidefiniteness of the empirical estimator. Each summand of equation 19 has the form $q_{i\ell}\widetilde{\boldsymbol{R}}_i\widetilde{\boldsymbol{R}}_i^\top$ with $q_{i\ell} \geq 0$ and is therefore positive semidefinite. Their average, its symmetrized version, and the version with a nonnegative ridge are also positive semidefinite. With population-centered rewards and independent forward probes, the expectation of the unregularized estimator is $\overline{\boldsymbol{M}}$. When rewards are centered using the same finite rollout group, the estimator instead targets the corresponding empirical analogue; the EMA and ridge used in practice trade exact unbiasedness for stability.

B Training Configuration

This appendix records the settings that determine the behaviour of FCB-NFT and of the compared scalarizations on the two main suites. Both suites are run with the *same* configuration; the only difference is the reward set. Generic optimizer and sampler settings follow the public DiffusionNFT SD3.5 recipe and are listed only where they interact with the method.

B.1 Shared setup

Model and optimizer. We fine-tune SD3.5-Medium at 512×512 with LoRA (rank 32, $\alpha = 64$) on the query, key, value, and output projections of both streams of every MM-DiT attention block, in fp16 mixed precision. AdamW uses learning rate 4×10^{-4}, $(\beta_1, \beta_2) = (0.9, 0.999)$, weight decay 10^{-4}, and gradient-norm clipping at 1.0. All methods train for $1{,}000$ outer iterations with one optimizer step per iteration.

Rollout and update budget. Every outer iteration samples 48 prompt groups of $B = 12$ images each (576 rollouts) from the OCR training prompts of Liu et al. (2025), with 25 deterministic sampling steps (DPM-Solver-2, no classifier-free guidance). Each rollout contributes 7 of its 25 forward timesteps to the loss (timestep fraction 0.28). The NFT implicit-branch coefficient is $\beta_{\mathrm{NFT}} = 0.1$. The same 576-sample iteration is executed on $8\times$H200 (micro-batch 9, 8 accumulation steps) or $16\times$RTX PRO 6000 (micro-batch 6, 6 steps); the two topologies are semantically identical and are used interchangeably.

Rollout model. The rollout ("old") model is an exponential moving average of the trainable LoRA weights with the conservative anchor schedule used for OCR in DiffusionNFT: the EMA decay is 0 for the first 75 iterations (rollout model equal to the live model), then increases linearly by 0.0075 per iteration and is held at 0.999 afterwards. This schedule is shared by FCB-NFT and every baseline; with the faster default schedule, OCR training was unstable for the method and the baselines alike.

Rewards. The OCR suite uses OCR accuracy (PaddleOCR, as in Liu et al. (2025)), PickScore v1, HPSv2.1, and CLIPScore (CLIP ViT-L/14). The compressibility suite replaces CLIPScore by the JPEG-compressibility reward of Black et al. (2024), $-\mathrm{kB}/500$. Rewards are computed once per rollout and shared by all methods; centering is per prompt group.

Held-out evaluation. All numbers are computed from the EMA weights at iteration 1000 on the full held-out OCR test set ($1{,}018$ prompts, one image per prompt). The initial noise of each prompt is fixed across models, so per-prompt differences are paired. Evaluation uses 40 deterministic steps without guidance at 512×512, and every reward is scored for every model, including the single-reward specialists that define the normalized-gain denominators in Eq. (24).

B.2 FCB-NFT

Algorithm 1 summarizes one outer iteration of target selection; the DiffusionNFT loss and rollout update are unchanged. Table 5 lists the design hyperparameters and Table 6 the fixed numerical constants. None of them is tuned per suite.

Reference scales. The per-reward reference scale s_k^{ref} is a slow EMA of the standard deviation of reward k over all 576 rollouts of the current iteration. It is used only to make the Nash objective dimensionless, to express $\boldsymbol{d}$ and all tolerances, and to calibrate Δ; it never rescales the rewards that enter $\widehat{\boldsymbol{C}}$ or $\widehat{\boldsymbol{M}}$.

Algorithm 1 FCB target selection for one outer iteration. Lines 1–2 are the feedback controller of Section C; the remainder is the convex program of Section 4.4.

Require: rollout groups $\{(x_i, \boldsymbol{R}_i)\}_{i=1}^{B}$ per prompt, rollout field v_{old}, controller state $(\boldsymbol{s}^{\text{ref}}, \widehat{\boldsymbol{R}}^{\text{run}}, \boldsymbol{\eta})$ and last $\boldsymbol{G}^{\text{pred}}$

1: update EMAs of s_k^{ref}, $\widehat{R}_k^{\text{run}}$, $\widehat{R}_k^{\max}$, η_k from realized rewards and $\boldsymbol{G}^{\text{pred}}$ ▷ decay ρ
2: $\omega_k \leftarrow \text{clip}(1/\eta_k, [1/\omega_{\max}, \omega_{\max}])$; $\omega_k \leftarrow \min(\omega_k, 1)$ if at ceiling ▷ Table 7
3: **for** each prompt group **do**
4: $\widetilde{\boldsymbol{R}}_i \leftarrow \boldsymbol{R}_i - \frac{1}{B}\sum_j \boldsymbol{R}_j$; $\widehat{\boldsymbol{C}} \leftarrow \frac{1}{B-1}\sum_i \widetilde{\boldsymbol{R}}_i \widetilde{\boldsymbol{R}}_i^\top$ ▷ Eq. (17)
5: draw (t_i, ϵ_i); $q_i \leftarrow D_v^{-1}\|v_i - v_{\text{old}}(z_i)\|_2^2$; $\widehat{\boldsymbol{M}} \leftarrow \frac{1}{B}\sum_i q_i \widetilde{\boldsymbol{R}}_i \widetilde{\boldsymbol{R}}_i^\top + \text{ridge}$ ▷ Eq. (19), EMA, symmetrize
6: $\mathcal{U} \leftarrow \{k : \text{std}_i(\widetilde{\boldsymbol{R}}_{ik}) \geq \epsilon_{\text{sat}} s_k^{\text{ref}}\}$; $\Delta \leftarrow \text{median}_{k\in\mathcal{U}}\, \widehat{\boldsymbol{M}}_{kk}/(2As_k^{\text{ref}})^2$ ▷ Eq. (56)
7: **for** $\tau_d \in \{0,\ 0.01,\ 0.02,\ 0.04,\ 0.08\}$ **do** ▷ relaxation ladder
8: $d_k \leftarrow -\tau_d\, s_k^{\text{ref}}$ for $k \in \mathcal{U}$
9: $\boldsymbol{a}^\star \leftarrow \arg\max_{\boldsymbol{a}\geq 0} \sum_{k\in\mathcal{U}} \omega_k \log\big(([\widehat{\boldsymbol{C}}\boldsymbol{a}]_k/r_0 - d_k)/s_k^{\text{ref}}\big)$ s.t. $\boldsymbol{a}^\top \widehat{\boldsymbol{M}}\boldsymbol{a} \leq \Delta$; Eqs. (21) and (22) on $\mathcal{U}$; $a_k = 0$ and $[\widehat{\boldsymbol{C}}\boldsymbol{a}]_k/r_0 \geq -\epsilon_{\text{floor}} s_k^{\text{ref}}$ for $k \notin \mathcal{U}$
10: **break** if feasible
11: **end for**
12: $\boldsymbol{a}^\star \leftarrow \boldsymbol{0}$ if no ladder level was feasible
13: $o_i \leftarrow r_0 + \widetilde{\boldsymbol{R}}_i^\top \boldsymbol{a}^\star$; $\boldsymbol{G}^{\text{pred}} \leftarrow \widehat{\boldsymbol{C}}\boldsymbol{a}^\star/r_0$ ▷ stop-gradient; cached for group reuse
14: **end for**
15: **return** $\{o_i\}$ to the DiffusionNFT loss, Eq. (6)

Matrices and probes. $\widehat{\boldsymbol{C}}$ follows Eq. (17). $\widehat{\boldsymbol{M}}$ follows Eq. (19) with $L = 1$ probe per rollout: one (t, ϵ) pair is drawn from the training forward distribution, and q is the mean-squared residual between the flow-matching target and the rollout model's velocity, computed with a single gradient-free forward pass. A trace-normalized ridge is added and both matrices are symmetrized. The program is solved in per-channel z-units (each coefficient rescaled by $\widehat{C}_{kk}^{-1/2}$) for conditioning.

Field budget. Δ is calibrated to the field cost of an ordinary single-reward DiffusionNFT update. The standard practical mapping produces an optimality $o = \frac{1}{2} + \text{adv}/(2A)$ from a clipped normalized advantage with $A = 5$, which corresponds to a single-reward coefficient $a_k = 1/(2As_k^{\text{ref}})$. We therefore set, per group,

$$\Delta = \text{median}_{k\in\mathcal{U}}\ \frac{\widehat{\boldsymbol{M}}_{kk}}{(2As_k^{\text{ref}})^2}, \tag{56}$$

where $\mathcal{U}$ is the active channel set. The multi-reward tilt is thus permitted to move the field by no more than a typical single-reward tilt would; the rule is identical on both suites.

Disagreement policy. The strict program with $\boldsymbol{d} = 0$ is solved first. If it is infeasible, $d_k = -\tau_d s_k^{\text{ref}}$ is applied with τ_d doubled from 0.01 to a cap of 0.08; if the capped problem remains infeasible the group receives $\boldsymbol{a} = 0$. A strict surplus floor of $10^{-3} s_k^{\text{ref}}$ prevents solver-tolerance solutions with vanishing gain from being accepted as feasible.

Controller. The priorities $\boldsymbol{\omega}$ and active set $\mathcal{U}$ are written by the feedback controller of Section C (algorithm 1, lines 1–2, and Table 7). The main-table FCB-NFT rows use exactly these rules.

B.3 Baselines

All baselines share the rollouts, rewards, optimizer, anchor schedule, and update budget above; they differ only in how the scalar optimality is formed. Each first builds a scalar reward $\bar{R}$ from the K centered observations and then applies the standard DiffusionNFT mapping: per-group centering, division by the standard deviation of $\bar{R}$ over all 576 rollouts, clipping to $[-A, A]$ with $A = 5$, and $o = \frac{1}{2} + \text{adv}/(2A)$.

- *Single-reward specialists*: $K = 1$; these are ordinary DiffusionNFT runs on one reward and define the normalized-gain denominators.

Table 5: FCB-NFT design hyperparameters. All values are shared by the OCR and compressibility suites; controller thresholds are in units of s_k^{ref} and were set once by hand.

Group	Setting	Value
Estimation	group size B / field probes L	12 / 1
Budget	Δ	single-reward calibration, Eq. (56), $A = 5$
Controller	smoothing decay ρ (η)	0.95
	weight clip $\omega_{\max}$	2
	saturation threshold ϵ_{sat}	0.15

Table 6: Fixed numerical constants. None was tuned.

Constant	Value
baseline optimality r_0 / box margin ϵ_o	0.5 / 0.03
ridge on $\widehat{\boldsymbol{M}}$	$10^{-3}\,\mathrm{tr}/K$
s^{ref} EMA decay / η window	0.9 / 25 iterations
running-maximum decay	0.997 per iteration
ceiling margin ϵ_{ceil} / saturation floor $\epsilon_{\mathrm{floor}}$	0.1 / 0.02
relaxation ladder τ_d / surplus floor	$0.01 \to 0.08$ (doubling) / 10^{-3}
solver	CLARABEL (SCS fallback)

- *Raw summation*: $\bar{R} = \sum_k R_k$ in native units.
- *Global normalization*: $\bar{R} = \sum_k R_k/\hat{\sigma}_k$, where $\hat{\sigma}_k$ is the all-rollout standard deviation of reward k measured at the first iteration and then frozen.
- *Group-z summation*: $\bar{R} = \sum_k (R_k - \mu_{k,g})/(\sigma_{k,g} + 10^{-4})$ with per-group statistics.
- *Reward-only bargaining*: identical to FCB-NFT (objective, $\widehat{\boldsymbol{C}}$, disagreement policy, controller) except that the field constraint Eq. (21) is replaced by $\|\boldsymbol{a}\|_2^2 \le \Delta$ in z-units with $\Delta = 1/(2A)^2$, the same single-reward calibration as Eq. (56) with $\widehat{\boldsymbol{M}}$ replaced by the identity.

The field-geometry ablations of Section 5.3 (diagonal $\widehat{\boldsymbol{M}}$, scale-matched $\kappa_g \widehat{\boldsymbol{C}}$, shuffled q, frozen budget) change only the matrix or the budget inside the FCB-NFT program and keep every other setting in Tables 5 and 6.

C Realized-gain feedback

The program of Section 4.4 certifies gains in the currency of the estimated linear model, $[\widehat{\boldsymbol{C}}\boldsymbol{a}]_k/r_0 > d_k$, not in the currency of realized rewards after the induced parameter update. The two can diverge systematically: rewards that respond to low-frequency image changes realize more than predicted, rewards that require fine structure realize less. Writing $G_k^{\mathrm{real}}(\tau)$ for the realized change of the rollout mean of reward k at outer iteration τ and $G_k^{\mathrm{pred}}(\tau)$ for the group-averaged certified surplus, we track the *realization efficiency*

$$\eta_k(\tau) = \mathrm{EMA}_\rho\Big[G_k^{\mathrm{real}}(\tau)/G_k^{\mathrm{pred}}(\tau)\Big]. \tag{57}$$

Left uncorrected, every per-group solution can certify a balanced predicted improvement while the executed updates drift along the over-realizing channel without violating any constraint. On the compressibility suite this drift ends in blank images.

Controller. We therefore close the loop with a slow, model-free controller that observes only realized rewards and acts on the program exclusively through inputs it already exposes: the priorities $\boldsymbol{\omega}$ and the active channel set $\mathcal{U}$. Per channel it keeps an EMA of the realized rollout mean $\widehat{R}_k^{\mathrm{run}}$ and a decayed running maximum $\widehat{R}_k^{\max}$. Each iteration it evaluates the conditions in Table 7 and writes the corresponding inputs; everything else in the program is untouched, so every solve remains the convex problem of Eq. (20).

Table 7: Controller conditions and their effect on the program. All thresholds are in units of s_k^{ref}. The first two rows are per channel and per iteration; the last is per channel and per group.

Condition	Test	Effect on the program
Active (default)	—	$\omega_k = \mathrm{clip}(1/\eta_k, [1/\omega_{\max}, \omega_{\max}])$
At ceiling	$\widehat{R}_k^{\mathrm{run}} \geq \widehat{R}_k^{\max} - \epsilon_{\mathrm{ceil}}\, s_k^{\mathrm{ref}}$	$\omega_k \leftarrow \min(\omega_k, 1)$
Degenerate (per group)	$\mathrm{std}_i(\widetilde{\boldsymbol{R}}_{ik}) < \epsilon_{\mathrm{sat}}\, s_k^{\mathrm{ref}}$	$k \notin \mathcal{U}$: $a_k = 0$, floor $G_k \geq -\epsilon_{\mathrm{floor}}\, s_k^{\mathrm{ref}}$

The rows address two failure modes. *Active* re-weights through $1/\eta_k$, demanding more predicted surplus from channels that historically under-realize; the Nash objective is invariant to rescaling of the surplus, so this is the only place η can enter. *At ceiling* prevents that re-weighting from penalizing a channel whose low efficiency reflects absent headroom rather than optimization failure. *Degenerate* removes channels whose group statistics carry no signal, so solver noise on a saturated channel can neither drive the tilt nor force the group into relaxation.

Properties. The controller is model-free by design, so it cannot be misled by the estimation bias it corrects. It introduces three design quantities: the smoothing decay ρ, the weight clip $\omega_{\max}$, and the saturation threshold ϵ_{sat}. The remaining constants (ϵ_{ceil}, $\epsilon_{\mathrm{floor}}$, the running-maximum decay) were set once by hand in s^{ref} units, are shared by both suites, and were never tuned (Table 6). The result is a two-timescale method: a model-based allocator that is sample-efficient within each rollout group, governed by a feedback loop anchored to measured outcomes.

D Separating the Field Metric from the Feedback Controller

Two components act on the scalar target. The empirical field metric $\widehat{\boldsymbol{M}}$ shapes the feasible set of the convex program in Section 4.4, and the controller of Section C adjusts the priorities $\boldsymbol{\omega}$ and the active set $\mathcal{U}$ from realized rewards. Because the controller writes only inputs the program already exposes, the two can be switched independently. This appendix reports the resulting 2×2 ablation on the compressibility suite, where the degenerate solutions of Section 5.2 make the distinction visible. Compute constraints limited the study to this suite; the OCR-suite comparisons of Table 3 were all run with the controller on.

Protocol. All runs share Section B. We cross the field matrix, $\widehat{\boldsymbol{M}}$ or the scale-matched $\kappa_g \widehat{\boldsymbol{C}}$ of Table 3, with the controller on or off. With the controller off, $\omega_k = 1$ throughout and every channel with nonzero group variance stays in $\mathcal{U}$. The relaxation ladder and the budget rule of Eq. (56) are identical in all four cells, so the cells differ only in the matrix defining the budget ellipsoid and in whether $\boldsymbol{\omega}$ and $\mathcal{U}$ are updated. The reward-only bargaining row of Table 2 (identity matrix, controller on) is repeated for reference. Each cell is a single run.

Table 8: Field metric and controller ablation on the compressibility suite. Worst is $\min_k g_k^{\mathrm{norm}}$; the binding channel is given in the last column. The $\widehat{\boldsymbol{M}}$/on row is the FCB-NFT entry of Table 2; the identity row is the reward-only bargaining entry of the same table.

Field matrix	Controller	OCR	Compress.	PickScore	HPSv2	Worst ↑	Binding
Base model	n/a	0.125	−0.206	0.772	0.192	0.00	–
Specialist	n/a	0.968	−0.009	0.929	0.348	n/a	–
$\widehat{\boldsymbol{M}}$	on	0.964	−0.122	0.870	0.294	**0.43**	Compress.
$\widehat{\boldsymbol{M}}$	off	0.900	−0.060	0.839	0.237	0.29	HPSv2
$\kappa_g \widehat{\boldsymbol{C}}$	on	0.832	−0.050	0.802	0.202	0.06	HPSv2
$\kappa_g \widehat{\boldsymbol{C}}$	off	0.000	−0.010	0.704	0.113	−0.51	OCR
I (reference)	on	0.000	−0.010	0.703	0.113	−0.51	OCR

Either component alone avoids collapse; the field metric does so at a far better operating point. With both switched off ($\kappa_g \widehat{\boldsymbol{C}}$, controller off) training enters the blank-image solution within the first

hundred iterations. Switching on either component prevents this, but not equally. The field metric alone ($\widehat{M}$, controller off) reaches a worst normalized gain of 0.29 with OCR at 0.900 and all four rewards above the base model. The controller alone ($\kappa_g \widehat{C}$, controller on) reaches only 0.06: it holds OCR at 0.832 but leaves HPSv2 within 0.01 raw of the base model, close to the text-on-blank regime. The two matrices have matched trace by construction, so the gap between the two controller-off rows cannot be attributed to the size of the budget; it reflects the shape of the ellipsoid. Under $\widehat{M}$ the coefficient direction that pushes images toward blank outputs is expensive, because the velocity residuals of low-frequency image changes co-vary strongly with the compressibility reward. Under $\kappa_g \widehat{C}$ that direction costs the same as any other direction of equal reward variance, and without feedback the program spends the budget on it whenever the predicted compressibility surplus is large. The reference row shows that the covariance shape itself matters as well: with the identity matrix, the controller alone does not prevent collapse.

With the field metric in place, the controller re-allocates rather than stabilizes. Turning the controller on under $\widehat{M}$ raises the worst gain from 0.29 to 0.43, and the pattern of the change is the one Section C is designed to produce. Compressibility, the over-realizing channel, gives up part of its gain (normalized $0.74 \to 0.43$), while the two preference channels, which under-realize, receive it (PickScore $0.43 \to 0.62$, HPSv2 $0.29 \to 0.65$); OCR moves from 0.900 to 0.964. The binding channel switches from HPSv2 to compressibility. Under $\kappa_g \widehat{C}$ the controller is instead doing the stabilizing work by itself, which is why its marginal effect there ($+0.57$) is four times its marginal effect under $\widehat{M}$ ($+0.14$). The two effects are therefore not additive; the components are partial substitutes for stability and complementary for allocation.

Reading of the four cells. The field metric determines the operating regime: whether training stays on a non-degenerate solution and, if so, how far from the text-on-blank boundary. The controller determines how the surplus within that regime is split between over- and under-realizing channels, and can substitute, poorly, for the metric when the metric is absent. The ordering of the two matrices is the same with the controller on and off, consistent with the scale and shuffled-q controls of Table 3 on the OCR suite. All cells are single runs, so differences of a few hundredths in Worst should not be over-read; the collapse/non-collapse distinction and the 0.29 versus 0.06 gap are well outside that range.

E More Ablations

Priority on the OCR channel. The Nash priorities $\boldsymbol{\omega}$ are the only place where a user preference enters FCB-NFT. To see how sensitive the selected target is to this input, we fix the base priority of the OCR channel to $\omega_{\mathrm{ocr}} \in \{1/16, 1/8, 1/4, 1/2, 1, 2\}$, keep the other three at 1, and leave every other setting of Tables 5 and 6 unchanged; the controller still applies its $1/\eta$ correction and clip on top of the base value. Table 9 reports the OCR suite.

The sweep behaves as the log utility predicts. Reducing ω_{ocr} trades OCR for the three preference rewards, which reach or exceed their specialist values at $\omega_{\mathrm{ocr}} \leq 1/8$, while OCR remains the binding channel; the gain along OCR is not monotone in this range because the field budget is spent mostly on the preference channels and OCR moves only through its covariance with them. Increasing ω_{ocr} to 2 pushes OCR to 0.950 but drops HPSv2 and CLIPScore to 0.60 and 0.42 in normalized units, so the binding channel flips to CLIPScore. The default $\omega_{\mathrm{ocr}} = 1$ is the maximizer of the worst-channel gain, which is consistent with FCB-NFT being tuned to balance rather than to any one channel. Two notes on reading the table: CLIPScore has the smallest specialist range (0.055 raw), so a raw difference of 0.01 moves its normalized gain by roughly 0.18, which explains the non-monotone step from $1/4$ to $1/2$; and every $\omega_{\mathrm{ocr}} \leq 1$ still improves all four rewards over the base model, so the priority mainly moves the operating point along the frontier rather than breaking balance.

Table 9: Sweep of the base Nash priority on the OCR channel, $\omega_{\mathrm{ocr}} \in \{1/16, \ldots, 2\}$, with the other priorities fixed at 1 (OCR suite). $\omega_{\mathrm{ocr}} = 1$ is the main-table FCB-NFT row. Worst is $\min_k g_k^{\mathrm{norm}}$; the binding channel is given in the last column.

Method	OCR	PickScore	HPSv2	CLIPScore	Worst ↑	Binding
Base model	0.125	0.772	0.192	0.270	0.00	–
Specialist	0.968	0.929	0.348	0.325	–	–
FCB-NFT with fixed base priority						
$\omega_{\mathrm{ocr}} = 1/16$	0.692	0.923	0.349	0.323	0.67	OCR
$\omega_{\mathrm{ocr}} = 1/8$	0.661	0.919	0.348	0.324	0.64	OCR
$\omega_{\mathrm{ocr}} = 1/4$	0.786	0.912	0.336	0.323	0.78	OCR
$\omega_{\mathrm{ocr}} = 1/2$	0.829	0.915	0.339	0.312	0.76	CLIPScore
$\omega_{\mathrm{ocr}} = 1$	0.878	0.911	0.336	0.326	**0.89**	PickScore
$\omega_{\mathrm{ocr}} = 2$	0.950	0.889	0.285	0.293	0.42	CLIPScore