CREDITEX: VIEW-LEVEL CREDIT ASSIGNMENT FOR HUMAN-ALIGNED 3D TEXTURING

Anonymous authors
Paper under double-blind review



Figure 1: 3D assets with high quality textures generated by our method.

ABSTRACT

Despite advances in joint multi-view texture generation, individual views can still contain artifacts, residual illumination, and implausible details. A promising approach is to use post-training with human preference feedback, but existing asset-level rewards obscure quality differences across views and provide the same feedback to views of varying quality. To address this issue, we introduce CrediTex, a post-training framework for 3D texture generation with context-aware per-view rewards and fine-grained credit assignment. First, we construct a dataset of 48K human preference pairs between corresponding views to train the reward model. Given a target-view query, the reward model predicts a continuous scalar reward for the specified view using the reference image and complete view set as context. Subsequently, we compute relative advantages from these rewards across candidates at each fixed viewpoint to guide view-level reinforcement learning. These advantages provide distinct feedback to individual views of the same 3D asset, preserving view-level credit assignment throughout policy optimization. Experiments across multiple models and evaluation metrics demonstrate state-of-the-art performance in both reward modeling and 3D texture generation.

1 INTRODUCTION

High-quality textures are essential for creating realistic and usable 3D assets. Whereas geometry defines an object's shape, texture determines its surface appearance through colors, material characteristics, and fine details. Given a reference image and an untextured geometry, image-guided

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.

Reward formulation	Adapted to 3D tasks	Per-view output	Main limitation
Reward3D / MVReward / HyperScore	✓	×	Lacks view attribution
HPS / PickScore / ImageReward	×	✓	Not tailored to 3D albedo
CrediTex (Ours)	✓	✓	–

Table 1: Comparison of reward formulations by adaptation to 3D tasks and output granularity. Existing 3D evaluators lack view attribution and image reward models are not tailored to 3D albedo evaluation, whereas CrediTex provides per-view rewards tailored to this task.

texture generation aims to synthesize a complete texture consistent with the reference (Huang et al., 2025; Shao et al., 2025; Liu et al., 2026; He et al., 2025; Liang et al., 2026; He et al., 2026). Since the reference image reveals only part of the object, texture generation must preserve the observed appearance while completing unobserved surfaces to produce coherent albedo textures without baked illumination. Achieving plausible completion of occluded regions while preserving fine texture details remains a central challenge.

Post-training with human preference feedback provides a direct way to address local defects in pretrained texture generators. However, existing 3D and multi-view evaluators, including MVReward, Reward3D, and HyperScore (Wang et al., 2025; Ye et al., 2024; Zhang et al., 2025), summarize each candidate through one or more asset-level scores but cannot provide distinct quality feedback for individual views. Using this score for post-training assigns identical scalar feedback to views of different quality, potentially misdirecting local optimization. In addition, HPS, PickScore, ImageReward, and the vision language model (VLM)-based RewardDance score individual images for visual appeal and text-image alignment (Wu et al., 2023; Kirstain et al., 2023; Xu et al., 2023; Ma et al., 2025; Wu et al., 2025b). Their training objectives do not specifically address implausible content in occluded regions of albedo images. These limitations, summarized in Tab. 1, motivate a central question: *How can we assess individual views within the full multi-view context and preserve their relative quality in feedback to a joint generation policy?*

To address this challenge, we introduce *CrediTex*, a post-training framework for 3D texture generation with context-aware per-view rewards and fine-grained credit assignment. We first construct a large scale preference dataset by asking annotators to compare corresponding views while inspecting the reference image and both complete candidate view sets. The comparisons assess reference fidelity, material plausibility, and albedo cleanliness, yielding $47{,}714$ preference pairs with clear quality differences. We then use these preference pairs to finetune a VLM with a Bradley–Terry ranking loss, yielding our reward model. Given the reference image, a complete view set, and a target view query, the model returns a continuous scalar reward for the queried view. This *global context with local evaluation* design provides viewpoint context to help the model identify the target viewpoint and assess local texture quality more accurately.

Subsequently, we freeze the trained reward model and score each view of multiple candidates generated under the same reference image and geometric conditions to provide feedback for online post-training. These scores, however, vary in scale with reference coverage and occlusion across viewpoints, making direct cross-view comparisons inappropriate. Thus, we standardize rewards across candidates independently at each fixed viewpoint, obtaining an advantage that measures each view's relative performance against the corresponding views of other candidates. These advantages determine the positive and negative branch weights of DiffusionNFT (Zheng et al., 2026) separately for each view. Finally, the corresponding per-view losses jointly update the shared generation policy, preserving the view-level granularity of human judgments through policy optimization.

We evaluate CrediTex through post-training experiments on MV-Adapter (Huang et al., 2025), Qwen-Image-Edit (Wu et al., 2025a), and FLUX.2-klein (Black Forest Labs, 2026). As shown in Table 2, CrediTex improves all three backbones across multiple texture quality metrics. On MV-Adapter, it reduces $\mathrm{FID}_{\mathrm{CLIP}}$ from 18.13 to 17.03 and LPIPS from 0.421 to 0.402. On Qwen-Image-Edit, it reduces CMMD from 0.324 to 0.296 and increases PSNR by 0.58. On FLUX.2-klein, it reduces LPIPS from 0.302 to 0.292 and increases SSIM from 0.604 to 0.612. These results demonstrate that CrediTex improves texture quality across different base models.

Our contributions are summarized as follows:

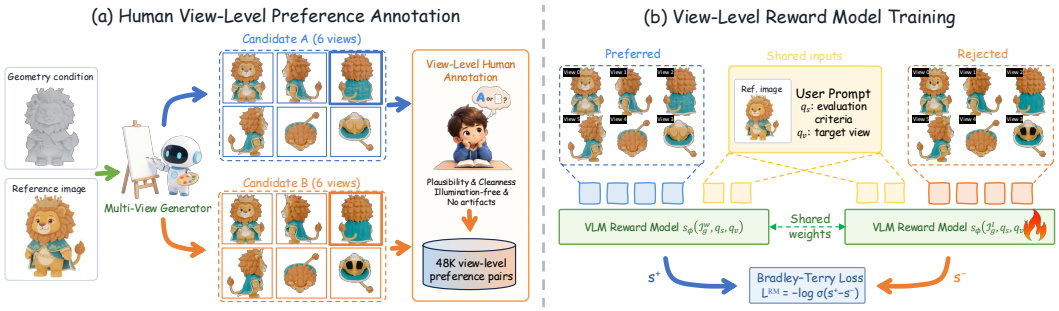


Figure 2: **Pipeline of Context-aware per-view reward modeling.** (a) Annotators compare corresponding views using the reference image and both complete view sets, yielding 48K view-level preference pairs. (b) A shared VLM learns view-specific scores from these pairs through a Bradley–Terry ranking objective.

- We construct a large-scale dataset of human view-level preferences and use it to train a context-aware texture reward model. By scoring each queried view within the reference image and complete multi-view context, the model provides fine-grained quality feedback that preserves local differences obscured by asset-level scores.
- We introduce a view-level DiffusionNFT post-training scheme that derives advantages from comparisons among candidates at the same viewpoint and assigns corresponding feedback to each view, enabling fine-grained credit assignment within a shared multi-view generation policy to accelerate convergence.
- We evaluate our reward model and post-training framework across different base models and multiple evaluation metrics, achieving state-of-the-art performance in reward modeling and 3D texture generation.

2 Related Work

2.1 Multi-View 3D Texture Generation

Recent 3D texturing systems typically synthesize multi-view appearances from a reference image and geometric conditions and then map them onto the target mesh. MV-Adapter, Hunyuan3D-Paint, Step1X-3D, and MVPainter combine image generation priors with geometric conditioning to synthesize consistent multi-view textures (Huang et al., 2025; Zhao et al., 2025; Li et al., 2025; Shao et al., 2025). RomanTex and CaliTex further incorporate 3D structure into attention to improve geometric alignment and cross-view consistency (Feng et al., 2025; Liu et al., 2026). Beyond multi-view synthesis, UniTEX learns to reconstruct textures in 3D space, while MaterialMVP extends generation to illumination-invariant PBR materials (Liang et al., 2026; He et al., 2025). These methods improve texture generation through data, conditioning, and architecture, whereas we study view-specific preference post-training.

2.2 Reinforcement Post-Training for Diffusion Models

Preference post-training for diffusion models broadly includes offline direct preference optimization and online reward-based optimization. Diffusion-DPO and IAPO optimize diffusion models using offline preferences at the image and instance levels, respectively (Wallace et al., 2024; Sun et al., 2026). Online methods such as Flow-GRPO (Liu et al., 2025) and DanceGRPO (Xue et al., 2025) compute group-relative advantages from newly generated candidates, whereas DiffusionNFT (Zheng et al., 2026) performs reward-weighted flow-matching updates on the forward noising process. For multi-view generation, Carve3D (Xie et al., 2024) uses set-level reconstruction consistency, while MVC-ZigAL (Zhang et al., 2026) combines single-image and shared multi-view rewards. Our method instead computes relative advantages across candidates at each fixed viewpoint from per-view rewards predicted under the full multi-view context, enabling fine-grained credit assignment to individual views.

3 METHOD

3.1 PROBLEM FORMULATION

Multi-View Texture Generation. Given a reference image I_{ref} and multi-view geometric conditions $\mathcal{G} = \{G_v\}_{v=1}^{N}$ derived from an untextured 3D geometry, our goal is to generate a set of high-quality, illumination-free, and mutually consistent texture images:

$$X = \mathcal{F}_\theta(I_{\text{ref}}, \mathcal{G}) = \{x_1, x_2, \ldots, x_N\}, \qquad N = 6. \tag{1}$$

Here, x_v denotes the texture image at the v-th fixed camera viewpoint. The views jointly describe the appearance of the same 3D asset and should remain consistent in object identity, color, material, texture layout, and geometric correspondence. Each view should also contain plausible and clean local textures without baked-in illumination.

3.2 CONTEXT-AWARE PER-VIEW TEXTURE REWARD MODEL

Human View-Level Preference Data Collection. Training a reward model adapted to 3D texturing requires human supervision with both multi-view context and view-level attribution. However, reference coverage varies across camera views: front views can draw on directly observed appearance, whereas rear views often require substantial completion of occluded surfaces. These differences in visible area and completion difficulty make it difficult for annotators to assign consistent absolute quality scores across viewpoints. Therefore, we collect same-view pairwise preferences between candidates generated under identical conditions. For each comparison, annotators inspect the reference image and both complete view sets to provide a preference judgment for the target view rather than an absolute quality score.

We collect $M = 18{,}343$ multi-view generation cases, each consisting of two candidates independently sampled with different random noise under the same reference image and geometric conditions. For case i, the two candidates are denoted as:

$$X^{A,(i)} = \left\{x_v^{A,(i)}\right\}_{v=1}^{N}, \qquad X^{B,(i)} = \left\{x_v^{B,(i)}\right\}_{v=1}^{N}, \qquad N = 6. \tag{2}$$

Annotators compare these candidates according to two criteria. **(1) Reference fidelity and material plausibility:** The target view should preserve the reference appearance, while unobserved regions should contain plausible textures consistent with the other views. We penalize structural and material defects, including Janus artifacts. **(2) Albedo cleanliness:** The base color should reflect the object's intrinsic appearance without baked illumination or spurious artifacts, while preserving valid material details.

Based on these criteria, annotators assign an A, B, or tie label to each fixed camera view v. A preference is recorded when one candidate is clearly better than the other at the target view. Otherwise, the comparison is labeled as a tie. Specifically, ties may occur when both candidates perform well or when both exhibit substantial defects:

$$y_v^{(i)} \in \{A, B, \text{tie}\}, \qquad v \in \{1, \ldots, N\}. \tag{3}$$

For training, we discard view-level records labeled as ties and retain winner–loser pairs with a clear quality difference at the target view. For each retained pair, the model input preserves the reference image and complete six-view context:

$$\mathcal{D}_{\text{pref}} = \left\{ \left(I_{\text{ref}}^{(i)}, X^{A,(i)}, X^{B,(i)}, v, y_v^{(i)} \right) \,\middle|\, y_v^{(i)} \in \{A, B\} \right\}. \tag{4}$$

After excluding ties, which account for 57% of all view-level judgments, we retain $47{,}714$ winner–loser pairs with clear quality differences to train the reward model.

Context-Aware Per-View Scoring. We obtain the reward model by finetuning a pretrained vision language model (VLM) and denote its parameters by ϕ. For a candidate $X_g = \{x_{g,v}\}_{v=1}^{N}$, we

concatenate all views in a fixed order and place the corresponding view index in the top-left corner of each view, as shown in Fig. 2(b):

$$I_g^{\text{grid}} = \text{Grid}\left(x_{g,1:N}\right), \qquad \mathcal{I}_g = \left(I_{\text{ref}}, I_g^{\text{grid}}\right). \tag{5}$$

To preserve fine texture details for evaluation, we retain each generated view at a resolution of 768×768 and arrange all six views in a 3×2 layout, forming a 2304×1536 multi-view grid. Within this grid, the rendered view indices identify camera positions and provide global viewpoint context.

We denote the shared evaluation instructions by q_s, which specify reference fidelity, material plausibility, and albedo cleanliness and ask whether the target view meets these quality criteria, with "yes" or "no" as the answer. To evaluate view v, we append a target view query q_v to q_s using the template `View to judge: view k:`, where `k` is set to $v-1$ to match the view indices displayed in the grid. Then we use the difference between the "yes" and "no" logits, $z_{g,v}^{\text{yes}}$ and $z_{g,v}^{\text{no}}$, as the reward to preserve a wider score range for subsequent group-relative advantage computation. The per-view reward is defined as

$$R_{g,v} = s_\phi\left(\mathcal{I}_g, q_s, q_v\right) = z_{g,v}^{\text{yes}} - z_{g,v}^{\text{no}}. \tag{6}$$

Using the same image input $\mathcal{I}_g$ and shared evaluation instructions q_s, we switch the target view by modifying only q_v. In this way, scoring each target view in turn yields the complete per-view reward vector:

$$\mathbf{R}_g = \left[R_{g,1}, R_{g,2}, \ldots, R_{g,N}\right]. \tag{7}$$

By relating the target view to the complete view set, the model can better identify its viewpoint and distinguish legitimate frontal details from Janus artifacts. We refer to this design as *global context with local evaluation*.

Pairwise Preference Learning. For each preference pair, we score the two candidates separately using the same target view query, with the winner and loser determined by the preference label at that view. Inspired by RewardDance (Wu et al., 2025b), we optimize a Bradley–Terry ranking objective for a preferred context $\mathcal{I}_g^w$, a rejected context $\mathcal{I}_g^l$, and target view v:

$$\mathcal{L}_{\text{RM}} = -\mathbb{E}_{(\mathcal{I}_g^w, \mathcal{I}_g^l, v)\sim\mathcal{D}_{\text{pref}}}\left[\log\sigma\left(s_\phi\left(\mathcal{I}_g^w, q_s, q_v\right) - s_\phi\left(\mathcal{I}_g^l, q_s, q_v\right)\right)\right]. \tag{8}$$

By scoring candidates separately during preference learning, the model can evaluate a single candidate at inference and return a scalar reward for the queried view. This avoids pairwise comparisons among rollout candidates and enables efficient reward computation for subsequent group-relative policy optimization.

Efficient Per-View Inference with a Shared KV Cache. When scoring different views of the same candidate, only the target view query q_v varies. We encode the shared multimodal prefix $(\mathcal{I}_g, q_s)$ once and cache its attention keys and values as H_g^{KV}. Each query q_v then reuses this cache in a separate branch to obtain the corresponding reward $R_{g,v}$:

$$H_g^{\text{KV}} = \text{Prefill}_\phi\left(\mathcal{I}_g, q_s\right), \qquad R_{g,v} = \text{Decode}_\phi\left(H_g^{\text{KV}}, q_v\right). \tag{9}$$

For each six-view candidate, this reduces multimodal prefix encoding from six forward passes to one, eliminating five redundant passes and accelerating reward inference.

3.3 View-Credited Online DiffusionNFT

Online Multi-Candidate Sampling. Following DiffusionNFT (Zheng et al., 2026), we maintain a trainable policy v_θ and a sampling policy v_{old} during online training. The sampling policy remains fixed within each online iteration and is softly updated from v_θ afterward. Given the same reference image and geometric conditions, v_{old} generates G multi-view candidates using different random noise:

$$X_g \sim p_{\theta_{\text{old}}}\left(X \mid I_{\text{ref}}, \mathcal{G}\right), \qquad g = 1, \ldots, G. \tag{10}$$

We freeze the reward model and score each candidate using q_s and q_v, obtaining a reward $R_{g,v}$ for each view. These scores retain the quality differences within each candidate, providing localized feedback for policy optimization.

View-Level Advantages. Fig. 6(b) illustrates that frontal views receive higher rewards than other views. This variation reflects differences in occluded area across viewpoints, leading to different reward scales and making raw scores unsuitable for direct comparison. Therefore, we standardize rewards independently at each fixed view across candidate rollouts generated under the same conditions. This puts view-level advantages on a common relative scale:

$$A_{g,v}^{\text{view}} = \frac{R_{g,v} - \mu_v}{\sigma_v + \epsilon}, \qquad \epsilon > 0. \tag{11}$$

The baseline μ_v and scale σ_v are the mean and standard deviation of the rewards from the G candidates at view v:

$$\begin{aligned} \mu_v &= \frac{1}{G}\sum_{g=1}^{G} R_{g,v}, \\ \sigma_v &= \sqrt{\frac{1}{G}\sum_{g=1}^{G}(R_{g,v} - \mu_v)^2}. \end{aligned} \tag{12}$$

We then convert these advantages into rewards $r_{g,v}$ for subsequent training (Zheng et al., 2026):

$$r_{g,v} = \frac{1}{2} + \frac{1}{2}\,\text{clip}\big(A_{g,v}^{\text{view}}, -1, 1\big)\,. \tag{13}$$

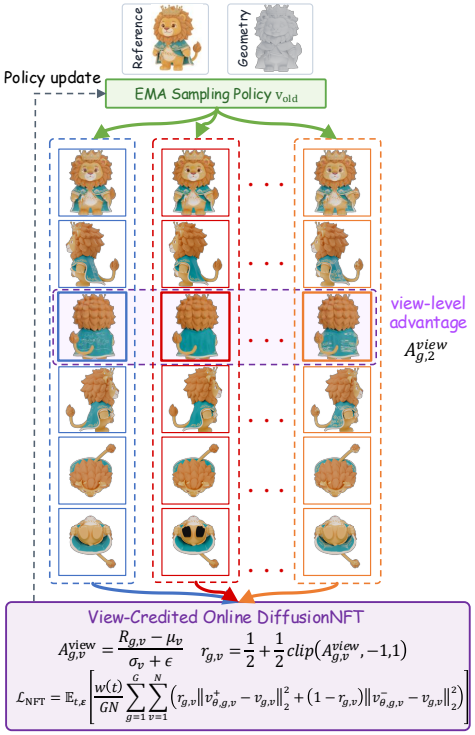


Figure 3: **View-Credited Online Diffusion-NFT.** Rewards are standardized across candidates at each viewpoint, and the resulting advantages guide the corresponding view losses to update the shared policy.

As illustrated in Fig. 3, local defects occur at different viewpoints: the first candidate has stains on the back of the cape, while the second has blackened paw soles visible from below. Comparing candidates at each fixed viewpoint provides feedback specific to these local differences, and applying each signal to the corresponding view loss enables fine-grained credit assignment.

View-Level DiffusionNFT. The policy processes all N views jointly, and the regression losses are evaluated separately for each view. We denote the view v predictions of the trainable and sampling policies for candidate g by $v_{\theta,g,v}$ and $v_{\text{old},g,v}$, evaluated on the same noisy multi-view input at timestep t. Following DiffusionNFT, we define the positive and negative predictions as follows:

$$\begin{aligned} v_{\theta,g,v}^{+} &= (1-\beta)v_{\text{old},g,v} + \beta v_{\theta,g,v}, \\ v_{\theta,g,v}^{-} &= (1+\beta)v_{\text{old},g,v} - \beta v_{\theta,g,v}. \end{aligned} \tag{14}$$

Here, $\beta > 0$ is the branch coefficient and $v_{g,v}$ denotes the corresponding flow target. Each pair of branch losses is weighted by its corresponding $r_{g,v}$ and $1 - r_{g,v}$ before averaging over candidates and views. We further apply a timestep weight $w(t)$ using the same self-normalization scheme as DiffusionNFT. The resulting objective is

$$\mathcal{L}_{\text{NFT}} = \mathbb{E}_{t,\boldsymbol{\varepsilon}}\left[\frac{1}{GN}\sum_{g=1}^{G}\sum_{v=1}^{N} w(t)\left(r_{g,v}\left\|v_{\theta,g,v}^{+} - v_{g,v}\right\|_2^2 + (1 - r_{g,v})\left\|v_{\theta,g,v}^{-} - v_{g,v}\right\|_2^2\right)\right]. \tag{15}$$

This formulation preserves view-level supervision throughout policy optimization: each view is compared with the corresponding views of other candidates, and its relative advantage determines its own regression feedback. The resulting losses jointly update the shared multi-view generation policy.



Figure 4: **Qualitative comparison before and after CrediTex post-training.** Results on MV-Adapter (Huang et al., 2025), Qwen-Image-Edit (Wu et al., 2025a), and FLUX.2-klein (Black Forest Labs, 2026) show clearer texture details and fewer local artifacts.

4 Experiments

4.1 Experimental Setup

Reward Model Training. We fine-tune Qwen3.6-27B (Qwen Team, 2026) on 47,714 view-level preference pairs from 18,343 cases, each containing two candidates, using LoRA (Hu et al., 2021) with rank 16 and $\alpha = 32$. We resize each view to 768×768 and train with a learning rate of 1×10^{-4} and an effective batch size of 64.

View-Credited Online DiffusionNFT Training. We freeze the trained reward model and optimize the generation policy using our View-Credited Online DiffusionNFT scheme (Zheng et al., 2026). The policy jointly generates six albedo views per candidate. For each conditioning input, we sample $G = 8$ candidates using 20 sampling steps and a classifier-free guidance (CFG) scale of 1.0. We compute advantages independently at each viewpoint and set the NFT branch coefficient to $\beta = 1.0$. At inference, we use a CFG scale of 3.0.

Detailed training configurations are provided in App. A.

Baselines. We compare with four methods for texture generation: MV-Adapter (Huang et al., 2025), Hunyuan3D-2.1 (Team Hunyuan3D et al., 2025), MVPainter (Shao et al., 2025), and UniTEX (Liang et al., 2026). To evaluate CrediTex across different backbones, we apply it to MV-Adapter, Qwen-Image-Edit (Wu et al., 2025a), and FLUX.2-klein (Black Forest Labs, 2026) and compare each model before and after post-training. We adapt Qwen-Image-Edit and FLUX.2-klein using the same SFT procedure as Hunyuan3D-2.1. For MV-Adapter, which is built on SDXL (Podell et al., 2024), we compute the post-training loss in x_0 space. We benchmark all methods on the same 200 high-quality cases, each consisting of a reference image and its corresponding geometry. All images use albedo representations. We render each generated texture and its ground truth and evaluate the resulting view pairs under matched conditions.

Metrics. We report six metrics to evaluate texture quality. We use FID in CLIP feature space ($\mathrm{FID}_{\mathrm{CLIP}}$) and CLIP Maximum Mean Discrepancy (CMMD) to assess distributional similarity between generated and ground-truth renderings. CLIP-I (Radford et al., 2021) measures feature similarity between matched full images. On the foreground, we compute LPIPS, PSNR, and SSIM to assess perceptual, pixel-level, and structural fidelity, respectively.

Method	Distribution match		Reference fidelity			
	$\text{FID}_{\text{CLIP}}\downarrow$	CMMD↓	CLIP-I↑	LPIPS↓	PSNR↑	SSIM↑
Existing texture generators						
UniTEX (Liang et al., 2026)	25.64	1.179	0.877	0.438	13.87	0.409
Hunyuan3D-2.1 (Team Hunyuan3D et al., 2025)	21.03	0.899	0.896	0.426	15.49	0.417
MVPainter (Shao et al., 2025)	18.83	0.713	0.907	0.421	16.90	0.445
Backbones and CrediTex post-training						
MV-Adapter (Huang et al., 2025)	18.13	0.580	0.917	0.421	16.91	0.446
+ CrediTex	17.03	0.551	0.922	0.402	17.37	0.456
Qwen-Image-Edit (Wu et al., 2025a)	12.14	0.324	0.948	0.304	20.86	<u>0.626</u>
+ CrediTex	11.67	0.296	0.952	**0.290**	**21.44**	**0.642**
FLUX.2-klein (Black Forest Labs, 2026)	<u>11.54</u>	**0.241**	<u>0.953</u>	0.302	20.68	0.604
+ CrediTex	**11.01**	<u>0.247</u>	**0.955**	<u>0.292</u>	<u>21.11</u>	0.612

Table 2: Quantitative comparison on 200 cases using six-view unlit albedo renderings. Bold and underline indicate the best and second-best values, respectively.

4.2 Quantitative Comparison

Tab. 2 shows that CrediTex improves texture quality across MV-Adapter, Qwen-Image-Edit, and FLUX.2-klein, reducing FID_{CLIP} and LPIPS while increasing CLIP-I, PSNR, and SSIM on all three backbones. On MV-Adapter, it lowers FID_{CLIP} from 18.13 to 17.03 and LPIPS from 0.421 to 0.402. On Qwen-Image-Edit, it reduces CMMD from 0.324 to 0.296 and achieves the best LPIPS (0.290), PSNR (21.44), and SSIM (0.642) among the compared methods. On FLUX.2-klein, it achieves the best FID_{CLIP} (11.01) and CLIP-I (0.955), while SSIM improves from 0.604 to 0.612. These gains demonstrate that CrediTex improves both distributional similarity and texture fidelity across different generation backbones.

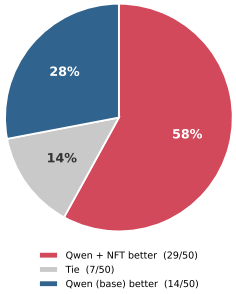


Figure 5: User Study.

User study. We further evaluate human preference by comparing Qwen-Image-Edit before and after CrediTex on 50 cases. Fig. 5 summarizes the outcomes: participants prefer the CrediTex results in 58% of cases and the base model in 28%, with 14% ties. This preference for CrediTex complements the improvements in the quantitative metrics.

4.3 Qualitative Comparison

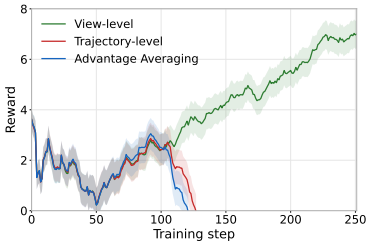
Fig. 4 compares textures before and after CrediTex post-training on three backbones. CrediTex improves local details while reducing artifacts in regions not visible in the reference image. On MV-Adapter, it produces more coherent fur patterns and reduces misplaced colors on the back and underside of the dress. On Qwen-Image-Edit, it makes the package text and garment patterns clearer and removes the spurious colors beneath the figure. On FLUX.2-klein, it sharpens the jar label and reduces unwanted markings on the reverse side. These examples show that CrediTex improves reference fidelity while producing cleaner textures on unobserved surfaces.

4.4 Reward Model Analysis

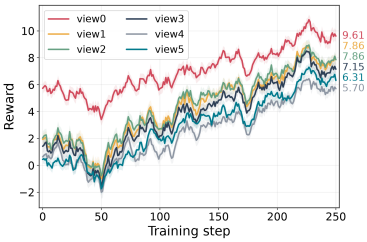
On view-level pairs with clear human preferences from 1,024 unseen cases, we measure agreement with human preferences at the target view. We compare our reward model with a single-view variant that receives the reference image and target view under identical training. Full context raises pairwise accuracy from 71.27% to 78.73% (Tab. 3), reducing ranking errors by approximately 26%. The largest accuracy gain occurs at the frontal view, where accuracy rises from 61.2% to

Table 3: **Reward model analysis.** KV caching reuses the shared input images across target-view queries in the six-view setting.

Context	Cache	Acc.	s/candidate
Single view	–	0.7127	11.62
Six views	×	0.7873	34.20
Six views	✓	**0.7873**	**7.34**



(a) View vs. Trajectory-Level Credit



(b) Rewards across Viewpoints

Reward model	FID_{CLIP}	LPIPS
MV-Adapter	18.13	0.421
PickScore	17.59	0.413
MVReward	18.01	0.421
Ours	**17.03**	**0.402**

(c) Reward Model Comparison

Figure 6: Reward curves during MV-Adapter training and reward model comparison.

75.5% (App. C). Without the complete view set, a scorer may confuse legitimate frontal details with Janus artifacts. Full context helps resolve this ambiguity by showing whether these details appear at the appropriate viewpoint. Futhermore, KV caching reduces scoring time per candidate from 34.20 s to 7.34 s by reusing the shared multimodal prefix, compared with 11.62 s for single-view scoring.

4.5 View-Level Credit Analysis

Ablation on advantage construction. We compare three advantage formulations on MV-Adapter: view-level credit assignment, trajectory-level credit assignment, and advantage averaging. Our view-level method normalizes rewards across candidates independently at each viewpoint and retains separate advantages for individual views. The trajectory-level baseline averages raw rewards over views before normalization, whereas advantage averaging first computes view-level advantages, averages them within each candidate, and normalizes these averages across candidates. Both baselines assign the resulting shared advantage to all views. As shown in Fig. 6(a), both baselines exhibit a sharp reward decline late in training, whereas our method continues to improve. Sharing advantages can reinforce defective views in otherwise strong candidates and suppress clean views in otherwise weak ones. This misassigned feedback can steer the corresponding view gradients toward updates that degrade local texture quality. These results demonstrate the importance of preserving view-level credit assignment during policy optimization.

Ablation on reward models. We replace our reward model with PickScore (Kirstain et al., 2023) or MVReward (Wang et al., 2025) during MV-Adapter post-training. Fig. 6(c) shows that PickScore yields modest improvements, reaching FID_{CLIP} and LPIPS of 17.59 and 0.413, while MVReward slightly improves FID_{CLIP} to 18.01. Our reward model achieves larger improvements in both metrics, reaching 17.03 and 0.402. We attribute PickScore's limited gains to its focus on 2D image preferences rather than defects specific to 3D texturing. MVReward jointly evaluates RGB appearance and normals, whereas our task varies only albedo with fixed normals, a mismatch that may weaken rollout discrimination and make rankings sensitive to noise. Consequently, broadcasting a single scalar advantage to all six views can amplify this noise through shared preference feedback. Together, these results support combining a reward model adapted to 3D albedo textures with fine-grained view-level credit assignment.

5 Conclusion

We presented CrediTex, a post-training framework that aligns multi-view texture generation with human preferences. We construct a large-scale dataset of view-level preferences to train a context-aware reward model and use its scores to guide View-Credited Online DiffusionNFT. Our analysis shows that full multi-view context helps the reward model interpret local texture quality, while view-specific feedback directs policy updates toward the views that need improvement. Together, these designs improve agreement with human judgments and support more stable training than trajectory-level aggregation, which can reinforce defective views within otherwise strong candidates and suppress clean views within otherwise weak ones. Experiments on MV-Adapter, Qwen-Image, and FLUX demonstrate consistent improvements in texture quality across different generation backbones. These findings highlight the importance of matching reward granularity to the local structure of the generation task.

6 AI USE STATEMENT

We used generative AI tools to polish the language of the manuscript. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Black Forest Labs. FLUX.2 [klein]: Towards Interactive Visual Intelligence. Official blog post, 2026. URL https://bfl.ai/blog/flux2-klein-towards-interactive-visual-intelligence.

Yifei Feng, Mingxin Yang, Shuhui Yang, Sheng Zhang, Jiaao Yu, Zibo Zhao, Yuhong Liu, Jie Jiang, and Chunchao Guo. Romantex: Decoupling 3d-aware rotary positional embedded multi-attention network for texture synthesis. In *ICCV*, 2025.

Huiang He, Shengchu Zhao, Jianwen Huang, Jie Li, Jiaqi Wu, Hu Zhang, Pei Tang, Heliang Zheng, Yukun Li, and Rongfei Jia. Hitem3d 2.0: Multi-view guided native 3d texture generation. *arXiv:2604.09231*, 2026.

Zebin He, Mingxin Yang, Shuhui Yang, Yixuan Tang, Tao Wang, Kaihao Zhang, Guanying Chen, Yuhong Liu, Jie Jiang, Chunchao Guo, et al. Materialmvp: Illumination-invariant material generation via multi-view pbr diffusion. In *ICCV*, 2025.

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. *arXiv preprint arXiv:2106.09685*, 2021.

Zehuan Huang, Yuan-Chen Guo, Haoran Wang, Ran Yi, Lizhuang Ma, Yan-Pei Cao, and Lu Sheng. Mv-adapter: Multi-view consistent image generation made easy. In *ICCV*, 2025.

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. *NeurIPS*, 2023.

Weiyu Li, Xuanyang Zhang, Zheng Sun, Di Qi, Hao Li, Wei Cheng, Weiwei Cai, Shihao Wu, Jiarui Liu, Zihao Wang, et al. Step1x-3d: Towards high-fidelity and controllable generation of textured 3d assets. *arXiv:2505.07747*, 2025.

Yixun Liang, Kunming Luo, Xiao Chen, Rui Chen, Hongyu Yan, Weiyu Li, Jiarui Liu, Fei-Peng Tian, and Ping Tan. Unitex: Universal high fidelity generative texturing for 3d shapes. In *CVPR*, 2026.

Chenyu Liu, Hongze Chen, Jingzhi Bao, Lingting Zhu, Runze Zhang, Weikai Chen, Zeyu Hu, Yingda Yin, Keyang Luo, and Xin Wang. Calitex: Geometry-calibrated attention for view-coherent 3d texture generation. In *CVPR*, 2026.

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. *NeurIPS*, 2025.

Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. Hpsv3: Towards wide-spectrum human preference score. In *ICCV*, 2025.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *ICLR*, 2024.

Qwen Team. Qwen3.6-27B: Flagship-level coding in a 27B dense model, April 2026. URL https://qwen.ai/blog?id=qwen3.6-27b.

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 *ICML*, 2021.

Mingqi Shao, Feng Xiong, Zhaoxu Sun, and Mu Xu. Mvpainter: Accurate and detailed 3d texture generation via multi-view diffusion with geometric control. *arXiv:2505.12635*, 2025.

Jiayang Sun, Pin Wang, Hongbo Wang, Xinyue Liu, Huaibo Huang, and Ran He. Towards fine-grained attribution: Instance-aware preference optimization for aligning diffusion models. In *CVPR*, 2026.

Team Hunyuan3D, Shuhui Yang, Mingxin Yang, Yifei Feng, Xin Huang, Sheng Zhang, Zebin He, Di Luo, Haolin Liu, Yunfei Zhao, Qingxiang Lin, Zeqiang Lai, Xianghui Yang, Huiwen Shi, Zibo Zhao, Bowen Zhang, Hongyu Yan, Lifu Wang, Sicong Liu, Jihong Zhang, Meng Chen, Liang Dong, Yiwen Jia, Yulin Cai, Jiaao Yu, Yixuan Tang, Dongyuan Guo, Junlin Yu, Hao Zhang, Zheng Ye, Peng He, Runzhou Wu, Shida Wei, Chao Zhang, Yonghao Tan, Yifu Sun, Lin Niu, Shirui Huang, Bojian Zheng, Shu Liu, Shilin Chen, Xiang Yuan, Xiaofeng Yang, Kai Liu, Jianchen Zhu, Peng Chen, Tian Liu, Di Wang, Yuhong Liu, Linus, Jie Jiang, Jingwei Huang, and Chunchao Guo. Hunyuan3D 2.1: From Images to High-Fidelity 3D Assets with Production-Ready PBR Material. *arXiv:2506.15442*, 2025.

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 *CVPR*, 2024.

Weitao Wang, Haoran Xu, Yuxiao Yang, Zhifang Liu, Jun Meng, and Haoqian Wang. Mvreward: Better aligning and evaluating multi-view diffusion models with human preferences. In *AAAI*, 2025.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-Image Technical Report. *arXiv:2508.02324*, 2025a.

Jie Wu, Yu Gao, Zilyu Ye, Ming Li, Liang Li, Hanzhong Guo, Jie Liu, Zeyue Xue, Xiaoxia Hou, Wei Liu, et al. Rewarddance: Reward scaling in visual generation. *arXiv:2509.08826*, 2025b.

Xiaoshi Wu, Keqiang Sun, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score: Better aligning text-to-image models with human preference. In *ICCV*, 2023.

Desai Xie, Jiahao Li, Hao Tan, Xin Sun, Zhixin Shu, Yi Zhou, Sai Bi, Sören Pirk, and Arie E Kaufman. Carve3d: Improving multi-view reconstruction consistency for diffusion models with rl finetuning. In *CVPR*, 2024.

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. *NeurIPS*, 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:2505.07818*, 2025.

Junliang Ye, Fangfu Liu, Qixiu Li, Zhengyi Wang, Yikai Wang, Xinzhou Wang, Yueqi Duan, and Jun Zhu. Dreamreward: Text-to-3d generation with human preference. In *ECCV*, 2024.

Yujie Zhang, Bingyang Cui, Qi Yang, Zhu Li, and Yiling Xu. Benchmarking and learning multi-dimensional quality evaluator for text-to-3d generation. In *ICCV*, 2025.

Ziyi Zhang, Li Shen, Deheng Ye, Yong Luo, Huangxuan Zhao, Meng Liu, Wei Yu, and Lefei Zhang. Refining few-step text-to-multiview diffusion via reinforcement learning. In *CVPR*, 2026.

Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, et al. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv:2501.12202*, 2025.

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

A TRAINING CONFIGURATIONS

Tables 4 and 5 summarize the configurations for reward model training and MV-Adapter post-training, respectively. We train the reward model with LoRA for three epochs and then keep it frozen during policy optimization. The NFT configuration distinguishes the CFG scale used for rollout sampling from that used in the training forward pass. Here, NFT training steps refer to online iterations, while its effective batch size counts sample–timestep pairs per optimizer update. Reward model training takes 50.6 hours on 16 H20 GPUs, and MV-Adapter post-training takes 19.7 hours on eight H20 GPUs for 250 online iterations. Both training times report wall-clock duration.

Table 4: Reward model training configuration.

Setting	Value
Backbone	Qwen3.6-27B
Preference cases	18,343
Candidates per case	2
Dataset preference pairs	47,714
Resolution per view	768×768
Grid layout	3×2
Grid resolution	2304×1536
Training method	LoRA
LoRA rank	16
LoRA alpha	32
Optimizer	AdamW
Learning rate	1×10^{-4}
Learning rate schedule	Cosine
Effective batch size	64
Training epochs	3
Warmup steps	50
Training precision	BF16
Hardware	$16\times$ H20
Training time	50.6 h

Table 5: MV-Adapter NFT configuration.

Setting	Value
Backbone	MV-Adapter + SDXL-base-1.0
Candidates per condition G	8
Number of views N	6
Sampling steps	20
Rollout CFG scale	1.0
Inference CFG	3.0
LoRA rank	32
LoRA alpha	64
NFT branch coefficient β	1.0
Optimizer	AdamW
Learning rate	3×10^{-5}
Learning rate schedule	Constant
Effective batch size	32
Gradient clipping	1.0
Training steps	250
Training precision	BF16
Hardware	$8\times$ H20
Training time	19.7 h

B TRAINING ALGORITHMS

CrediTex consists of two training stages. We first train a reward model on view-level preference pairs, using the complete multi-view context to score each target view. We then freeze this model and use its rewards to optimize the generation policy with View-Credited Online DiffusionNFT. Algorithms 1 and 2 summarize these stages.

B.1 REWARD MODEL TRAINING

Each preference label compares two candidates at the same target view. We score the candidates separately under their respective multi-view contexts and optimize a Bradley–Terry objective using the score difference. This lets the model learn from pairwise supervision while returning a scalar reward for a single candidate and target view at inference.

B.2 VIEW-CREDITED ONLINE DIFFUSIONNFT

At each online iteration, the sampling policy generates multiple candidates under the same condition. We reuse the reward model's shared prefix cache to score all views, then normalize rewards across candidates separately at each viewpoint. The resulting advantages determine the positive and negative branch weights for each view. Both policies process the full noisy multi-view input, while the loss retains separate reward weights for individual views. After policy updates, we refresh the sampling policy through an exponential moving average.

Algorithm 1 Context-Aware Per-View Reward Model Training

Require: Pretrained VLM parameters ϕ_0, non-tie preference pairs $\mathcal{D}_{\text{pref}}$, evaluation instructions q_s, view queries $\{q_v\}_{v=1}^{N}$, learning rate η_{RM}.
Initialize: Reward model parameters $\phi \leftarrow \phi_0$.
for each training iteration **do**
  Sample a minibatch $\mathcal{B} \subset \mathcal{D}_{\text{pref}}$.
  for each record j: $(I_{\text{ref}}^{(i)}, X^{A,(i)}, X^{B,(i)}, v, y_v^{(i)}) \in \mathcal{B}$ **do**
    Order the two candidates as (X^w, X^l) using $y_v^{(i)}$ at view v.
    for each $b \in \{w, l\}$ **do**
      Form an indexed grid $I_g^{b,\text{grid}} \leftarrow \text{Grid}(X^b)$.
      Construct $\mathcal{I}_g^b \leftarrow (I_{\text{ref}}^{(i)}, I_g^{b,\text{grid}})$.
      Score the candidate separately: $s^b \leftarrow s_\phi(\mathcal{I}_g^b, q_s, q_v) = z_{g,v}^{b,\text{yes}} - z_{g,v}^{b,\text{no}}$.
    end for
    Compute the pair loss $\ell_j \leftarrow -\log \sigma(s^w - s^l)$.
  end for
  $\mathcal{L}_{\text{RM}} \leftarrow |\mathcal{B}|^{-1} \sum_{j=1}^{|\mathcal{B}|} \ell_j$.
  Update trainable parameters: $\phi \leftarrow \phi - \eta_{\text{RM}} \nabla_\phi \mathcal{L}_{\text{RM}}$.
end for
Output: Reward model s_ϕ.

Here, $v_{g,v}$ is the view-v slice of the joint flow target $\mathbf{v}_g$. The sampling policy and reward model remain fixed during each policy update. The time-dependent coefficients α_t, σ_t specify the forward noising process, while σ_v denotes reward dispersion across candidates at one viewpoint. The hat indicates a sampled training objective rather than the expectation over all timesteps and noise.

C Per-View Reward Model Accuracy

We break down reward model accuracy by viewpoint on the held-out preference pairs used in the main evaluation. Figure 7 shows that full multi-view context improves pairwise accuracy at all six viewpoints. The largest gain occurs at the frontal view (View 0), where accuracy increases from 61.2% to 75.5%, followed by View 4, which improves from 66.2% to 77.4%. The remaining views gain 3.6 to 7.2 percentage points. Full context also narrows the accuracy gap between the best and worst viewpoints from 15.2 to 4.6 percentage points, indicating more consistent evaluation across views. The pronounced improvement at View 0 supports the role of multi-view context in interpreting viewpoint-dependent appearance, where a faithful frontal texture may otherwise be mistaken for a Janus artifact.

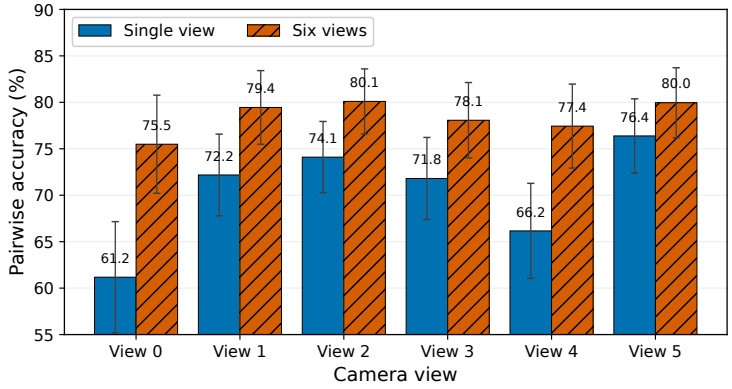


Figure 7: **Reward model accuracy at each viewpoint.** Full multi-view context improves pairwise accuracy over single-view scoring across all six views.

Algorithm 2 View-Credited Online DiffusionNFT

Require: Initial policy parameters θ_0, frozen reward model s_ϕ, conditioning data $\mathcal{C} = \{(I_{\text{ref}}, \mathcal{G})\}$, group size G, N views, branch coefficient $\beta > 0$, stabilizer $\epsilon > 0$, learning rate η, sampling-policy decay $\rho_k \in [0, 1)$.

Initialize: Trainable policy $\theta \leftarrow \theta_0$, sampling policy $\theta_{\text{old}} \leftarrow \theta_0$.

for each online iteration k **do**
  Initialize the rollout buffer $\mathcal{B} \leftarrow \varnothing$.
  for each sampled condition $c = (I_{\text{ref}}, \mathcal{G}) \in \mathcal{C}$ **do**
    Sample G candidates $X_g \sim p_{\theta_{\text{old}}}(X \mid c)$ with independent noise.
    for each candidate $g = 1, \ldots, G$ **do**
      $\mathcal{I}_g \leftarrow (I_{\text{ref}}, \text{Grid}(X_g))$; $H_g^{\text{KV}} \leftarrow \text{Prefill}_\phi(\mathcal{I}_g, q_s)$.
      Score all views: $R_{g,v} \leftarrow \text{Decode}_\phi(H_g^{\text{KV}}, q_v)$ for $v = 1, \ldots, N$.
    end for
    for each view $v = 1, \ldots, N$ **do**
      $\mu_v \leftarrow \frac{1}{G}\sum_{g=1}^{G} R_{g,v}$; $\sigma_v \leftarrow \sqrt{\frac{1}{G}\sum_{g=1}^{G}(R_{g,v} - \mu_v)^2}$.
      $A_{g,v}^{\text{view}} \leftarrow (R_{g,v} - \mu_v)/(\sigma_v + \epsilon)$ for all g.
      $r_{g,v} \leftarrow \frac{1}{2} + \frac{1}{2}\,\text{clip}(A_{g,v}^{\text{view}}, -1, 1)$ for all g.
    end for
    Store $(c, X_{1:G}, r_{1:G,1:N})$ in $\mathcal{B}$.
  end for
  for each sampled training group $(c, X_{1:G}, r_{1:G,1:N}) \in \mathcal{B}$ **do**
    Obtain clean joint latents $\mathbf{z}_{g,0}$ from X_g for all g.
    Sample t and Gaussian noise $\boldsymbol{\varepsilon}_g$; $\mathbf{z}_{g,t} \leftarrow \alpha_t \mathbf{z}_{g,0} + \sigma_t \boldsymbol{\varepsilon}_g$.
    Set the joint flow target $\mathbf{v}_g \leftarrow \dot{\alpha}_t \mathbf{z}_{g,0} + \dot{\sigma}_t \boldsymbol{\varepsilon}_g$.
    Evaluate both policies on $(\mathbf{z}_{g,t}, c, t)$ and extract $v_{\theta,g,v}$ and $v_{\text{old},g,v}$ for every view.
    $v^+_{\theta,g,v} \leftarrow (1-\beta) v_{\text{old},g,v} + \beta v_{\theta,g,v}$.
    $v^-_{\theta,g,v} \leftarrow (1+\beta) v_{\text{old},g,v} - \beta v_{\theta,g,v}$.
    $$\widehat{\mathcal{L}}_{\text{NFT}} \leftarrow \frac{1}{GN}\sum_{g=1}^{G}\sum_{v=1}^{N} w(t)\left[r_{g,v}\|v^+_{\theta,g,v} - v_{g,v}\|_2^2 + (1 - r_{g,v})\|v^-_{\theta,g,v} - v_{g,v}\|_2^2\right].$$
    Update the policy: $\theta \leftarrow \theta - \eta \nabla_\theta \widehat{\mathcal{L}}_{\text{NFT}}$.
  end for
  Update the sampling policy: $\theta_{\text{old}} \leftarrow \rho_k \theta_{\text{old}} + (1 - \rho_k)\theta$.
end for

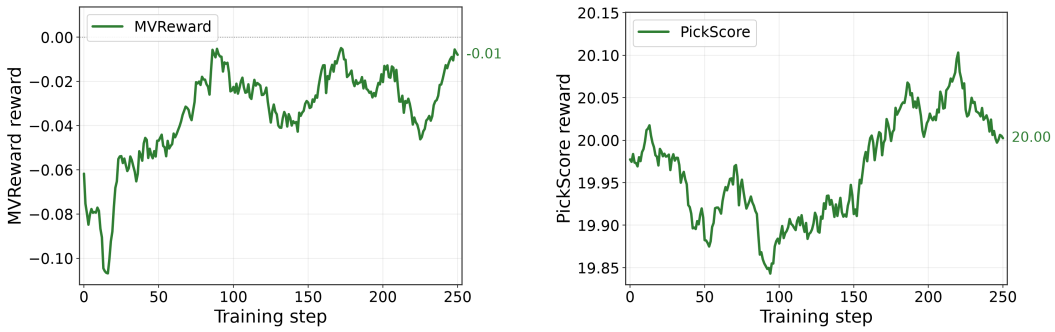
Output: Generation policy v_θ.

D Training Dynamics with Alternative Reward Models

We report the training reward curves for MV-Adapter optimized with MVReward and PickScore in Fig. 8. MVReward recovers after an initial drop but fluctuates throughout training, while PickScore ends close to its initial level after a temporary increase. Both reward models provide some gains, but they are not well suited to albedo generation, resulting in limited improvements and textures that still contain noticeable defects.

E Limitations and Future Works

Limitations. CrediTex relies on a learned reward model whose judgments may be less reliable for materials and appearance patterns underrepresented in the preference data. View-level credit assignment directs feedback to individual views but cannot correct errors in the reward itself. Our preference training also excludes ties, including comparisons in which the candidates have strengths and defects in different regions, leaving potentially useful local supervision unused. Moreover, optimization over six fixed views does not directly constrain poorly observed surface regions or artifacts introduced during texture baking.



(a) MVReward (b) PickScore

Figure 8: **Training dynamics with alternative reward models.** MV-Adapter is optimized using (a) MVReward or (b) PickScore. Each panel reports the corresponding model's training reward, whose numerical scale is not directly comparable across panels.

Future Work. A promising direction is to extend both reward modeling and policy optimization from views to regions. For example, one candidate may have a defective hat but clean clothing, while another exhibits the opposite pattern. Although we currently label this comparison as a tie, the individual regions still provide meaningful preference supervision. Region-level texture reward models could recover these signals, while reinforcement learning algorithms with region-level credit assignment could use them to guide more targeted updates. Together, these advances could improve texture evaluation and make better use of the collected preference data.