DELIGHTING 3D ASSETS VIA REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review



Figure 1: We propose an RL framework for 3D delighting. Compared with variants without exploration (w/o Exr) or consistency constraints (w/o Cons), Ours effectively removes undesired lighting-caused traces while faithfully preserving the original content.

ABSTRACT

Recently, 3D generation has made progress. However, unwanted highlights in generated textures reduce realism while limiting the editability of 3D assets. Despite this progress, 3D delighting remains an underdeveloped research direction, with existing methods only mitigating lighting as a generation by-product without dedicated evaluation systems or optimization paradigms. In this paper, we highlight the lighting issue in 3D and redefine the delighting paradigm in a comprehensive manner. We first construct several lighting datasets and develop three structurally distinct evaluation models for lighting assessment. Together, they establish a unified evaluation benchmark for 3D delighting optimization. Then, we propose an adaptive decoupled multi-objective Supervised Fine-Tuning (SFT) and Reinforcement Learning (RL) joint architecture: SFT injects basic delighting capability, while RL further enhances performance. In experiments, our method improves delighting performance by 18.7% while reducing FID by 55.0%, demonstrating substantially better content consistency. To the best of our knowledge, this paper is the first to introduce an RL paradigm and reward models to the 3D delighting tasks. Code and model weights will be made public after publication.

1 INTRODUCTION

Generating high-quality 3D content is essential for movies (Engelhardt et al., 2025), games (Aldemir et al., 2025), and AR/VR (Cools et al., 2025; Franke et al., 2025) applications. Recently, rapid advances in generative models have fueled the flourishing of 3D content creation. However, 3D delighting remains underexplored. Lighting issues pose significant challenges for 3D generation, as most applications require relighting across different scenes. When lighting is baked into assets, it can conflict with new illumination during relighting, causing incorrect shadow distributions. Therefore, removing unwanted lighting from generated assets is critical for reliable 3D rendering.

To understand why generated 3D assets contain illumination effects, it is necessary to analyze the training data distribution and learning mechanism of generative models. Existing 3D generative models typically learn the overall appearance of objects from large-scale 3D asset datasets, which often contain scene-related lighting information. Therefore, during progressive generation of 3D textures, models simultaneously learn intrinsic object textures and additional illumination features.

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.

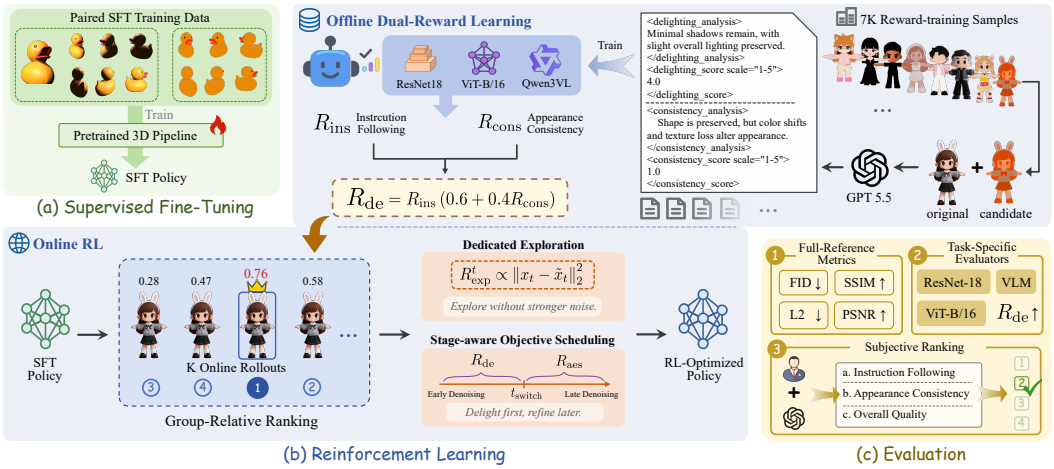


Figure 2: Overall framework of the proposed RL-based 3D delighting. (a) The warm-up multi-view sft. (b) Delighting-reward-based Reinforcement Learning. (c) Delighting Performance Evaluation.

Since the generation process does not constrain illumination, illumination factors are jointly encoded with texture details, causing generated assets to contain uncontrollable illumination effects.

To remove unwanted illumination effects, the industry relies on experienced artists to manually correct lighting artifacts in generated textures. However, this process often requires designing specialized procedural graphics and UV mapping, which is time-consuming and inefficient. Some works (Wei et al., 2018; Zeng et al., 2024) have attempted to replace manual editing with automated methods, but most only implicitly alleviate lighting issues within the processing pipeline. These methods rely on simple feature extraction and processing without developing dedicated delighting solutions, resulting in limited robustness (Zhang et al., 2025; Huang et al., 2025). Recent papers involving illumination removal, such as Hunyuan3D (Zhao et al., 2025b), treat delighting as an independent image translation process before texture generation, using a pretrained module to pre-remove illumination factors from reference images rather than dynamically guiding illumination removal during diffusion denoising. Since delighting is decoupled from texture generation, this method cannot jointly optimize illumination suppression and appearance preservation, leading to color shifts and image consistency degradation in generated textures (Shao et al., 2025). Moreover, Hunyuan3D only treats delighting as an auxiliary component, lacking systematic investigation and optimization for this problem. More detailed ***related work*** is provided in the appendix.

In summary, research on delighting tasks remains inadequate, and existing methods suffer from the following issues (Cools et al., 2025; Rohden et al., 2025; Zhao et al., 2025b): (1) The field lacks mature quantitative evaluation systems for delighting quality, limiting model optimization and method comparison. (2) Existing delighting methods usually require specialized designs for specific 3D generation pipelines, making them difficult to directly integrate into 3D generation models and limiting their generality. (3) Although some lighting can be removed, these methods decouple delighting from 3D texture generation and lack adaptive joint optimization, inevitably over-altering the color distribution and texture characteristics, causing significant discrepancies between optimized images and reference images while excessively compromising texture aesthetics.

To address the above issues, we establish a comprehensive technological foundation for 3D delighting. First, we construct a delighting dataset for quantitative evaluation and develop three complementary models for residual lighting assessment. Among them, the VLM-based model provides a holistic score jointly accounting for lighting removal and appearance consistency. To achieve collaborative optimization of lighting suppression and original texture characteristics preservation, unlike existing methods introducing independent delighting modules, we directly fine-tune the pretrained generative model with Supervised Fine-Tuning (SFT) and Reinforcement Learning (RL) joint optimization, integrating delighting into the original generation process and improving method compatibility and generality. This paper makes three main contributions: (1) We highlight the delighting issues and reformulate the 3D delighting paradigm with a complete formal investigation.

(2) We propose an RL-based joint optimization architecture with exploration supplement and multi-objective decoupling: SFT injects basic delighting capability, while RL further pushes performance. (3) We establish a comprehensive 3D delighting benchmark comprising multi-level score annotations and heterogeneous lighting reward models, lighting-affected samples curated specifically for RL training, and a unified evaluation protocol for assessing delighting performance.

2 METHOD

Since existing methods treat delighting as an independent module, causing lighting removal and texture generation to be decoupled, they struggle to simultaneously preserve lighting suppression performance and original texture consistency. Therefore, we directly fine-tune the pretrained generative model with Reinforcement Learning (RL) assisted by Supervised Fine-Tuning (SFT), integrating the delighting capability into the generation process, and achieve collaborative optimization of lighting removal and texture aesthetics preservation through adaptive decoupled multi-objective optimization. Specifically, considering the multi-view complexity of 3D content and the strong coupling between lighting and asset appearance, we construct an RL-based delighting optimization pipeline. First, the SFT stage endows the model with basic multi-view delighting capability. Then, the RL guided by lighting rewards further optimizes the model's understanding of the delighting task, achieving high-quality delighting results with multi-view consistency.

2.1 MULTI-VIEW DELIGHTING SFT

Multi-view Representation. Texture delighting removes baked-in illumination while preserving object identity and cross-view geometric consistency. Given a reference image $\mathbf{I}_{\text{ref}}$ and a 3D object, we render six normal maps $\{\mathbf{N}^{(v_p)}\}_{v_p=1}^{6}$ from predefined viewpoints and arrange them into a fixed 2×3 canvas $\mathbf{N} = \mathcal{T}_{2\times3}(\mathbf{N}^{(1)}, \ldots, \mathbf{N}^{(6)})$, where $\mathcal{T}_{2\times3}$ is a deterministic tiling operator. We use the same layout for the corresponding delighted RGB targets $\{\mathbf{Y}_0^{(v_p)}\}_{v_p=1}^{6}$, forming $\mathbf{Y}_0 = \mathcal{T}_{2\times3}(\mathbf{Y}_0^{(1)}, \ldots, \mathbf{Y}_0^{(6)})$. The shared canvas lets all viewpoints exchange information within a single denoising trajectory instead of being synthesized independently.

Geometry and Appearance Conditioning. The normal canvas specifies the surface orientation at each output location, while the reference image provides the appearance to be preserved. We encode these complementary inputs with the SAM 3D (SAM 3D Team et al., 2025) image encoder (E_{S3D}), obtaining $\mathbf{f}_N = E_{\text{S3D}}(\mathbf{N})$, $\mathbf{f}_I = E_{\text{S3D}}(\mathbf{I}_{\text{ref}})$, and form the condition stream as $\mathbf{H}_{\text{cond}}^{(0)} = [\mathbf{f}_N; \mathbf{f}_I]$, where $[\cdot;\cdot]$ denotes token concatenation. Specifically, $\mathbf{f}_N$ is the visual token sequence extracted from the normal canvas and encodes surface orientations across the six viewpoints, while $\mathbf{f}_I$ is the visual token sequence extracted from the reference image and encodes its spatial appearance cues. Such spatially resolved conditioning is important for delighting because the model must distinguish albedo from illumination using local surface orientation and the corresponding reference appearance. The noisy texture latent $\mathbf{x}_t$ initializes the texture stream $\mathbf{H}_{\text{tex}}^{(0)}$. Following the MMDiT dual-stream design (Esser et al., 2024), the texture and condition streams use separate projections and interact through shared self-attention, allowing texture tokens to attend jointly to geometry and reference-image tokens. Visual tokens preserve local correspondences but lack global appearance summarization. We extract a CLIP embedding $\mathbf{g}_I$ from $\mathbf{I}_{\text{ref}}$ (Radford et al., 2021), fuse it with the timestep embedding, and use it to modulate MMDiT blocks via AdaLN. Thus, SAM 3D provides aligned geometry and appearance cues, while CLIP supplies image-level appearance guidance.

Training Objective. We train the flow-matching backbone (Lipman et al., 2023) with supervised fine-tuning. Let $\mathbf{X}_0$ be the clean target latent encoded from $\mathbf{Y}_0$, with $\mathbf{x}_0 \sim \mathbf{X}_0$, and sample noise $\mathbf{x}_1 \sim \mathbf{X}_1$, where $\mathbf{X}_1 = \mathcal{N}(\mathbf{0}, \mathbf{I})$. For timestep $t \sim \mathcal{U}[0,1]$, the linear probability path shared by the SFT and subsequent RL stages is $\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\mathbf{x}_1$, where $\mathbf{v} = \mathbf{x}_1 - \mathbf{x}_0$ is the target velocity field. The adapted model predicts the target velocity from the noisy texture latent and the geometry-appearance conditions:

$$\mathcal{L}_{\text{SFT}}(\theta) = \mathbb{E}_{t,\mathbf{x}_0\sim\mathbf{X}_0,\mathbf{x}_1\sim\mathbf{X}_1}\left[\left\|\mathbf{v} - v_\theta\left(\mathbf{x}_t, t; \mathbf{f}_N, \mathbf{f}_I, \mathbf{g}_I\right)\right\|_2^2\right], \tag{1}$$

where θ denotes the trainable parameters. During inference, we initialize $\mathbf{x}_1$ from Gaussian noise and integrate the learned field from $t = 1$ to $t = 0$.

2.2 RL FOR 3D DELIGHTING

Markov Decision Process (MDP). Flow matching denoising can be viewed as an MDP $(\mathcal{S}, \mathcal{A}, \rho_0, P, R)$. The state $\in \mathcal{S}$, action $\in \mathcal{A}$, and policy are defined as: $\mathbf{s}_t \triangleq (\mathbf{c}, t, \mathbf{x}_t)$, $\mathbf{a}_t \triangleq \mathbf{x}_{t-1}$, and $\pi(\mathbf{a}_t \mid \mathbf{s}_t) \triangleq p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{c})$, respectively. Where $\mathbf{c}$ denotes the prompt embedding. The deterministic transition is $P(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \boldsymbol{a}_t) \triangleq (\delta_{\mathbf{c}}, \delta_{t-1}, \delta_{\mathbf{x}_{t-1}})$, and the initial distribution is $\rho_0(\mathbf{s}_0) \triangleq (p(\mathbf{c}), \delta_1, \mathcal{N}(\mathbf{0}, \mathbf{I}))$, where δ_y denotes the Dirac delta distribution centered at y. The reward follows $R(\mathbf{s}_t, \mathbf{a}_t) \triangleq r(\mathbf{x}_0, \mathbf{c})$ for $t = 0$, and 0 otherwise.

Flow-GRPO. The reinforcement learning (RL) objective maximizes the expected cumulative reward of policy π_θ with KL divergence (Liu et al., 2026): $\max_\theta \mathbb{E}\left[\sum_{t=0}^{T} \left(R(\mathbf{s}_t, \mathbf{a}_t) - \beta D_{\mathrm{KL}}(\pi_\theta(\cdot|\mathbf{s}_t)||\pi_{\mathrm{ref}}(\cdot|\mathbf{s}_t))\right)\right]$, where π_{ref} is the reference policy.

Dedicated Exploration for 3D Delighting. RL requires stochastic exploration, while deterministic ODE-based Flow Matching lacks exploration capability. Although Flow-GRPO introduces SDE noise to enable exploration, increasing noise improves exploration at the cost of generation stability. This trade-off is unsuitable for 3D delighting, where simple, fixed texture structures cause noise to propagate directly into pixel space, resulting in artifacts and interfering with reward evaluation. Therefore, we propose a latent variation-based exploration reward instead of noise amplification, enhancing exploration while keeping SDE noise low. The exploration reward is defined as:

$$R^t_{\mathrm{exp}}(\mathbf{s}_t, \mathbf{a}_t) = \|\Delta \mathbf{x}_t\|_2^2 = \|\mathbf{x}_t - \mathbf{x}_{t-1}\|_2^2 . \tag{2}$$

Finally, the overall reward is defined as follows; λ is a hyperparameter.

$$R_{\mathrm{total}}(\mathbf{s}_t, \mathbf{a}_t) = \underbrace{R\left(\mathbf{s}_t, \mathbf{a}_t\right)}_{\text{Task Reward}} + \lambda \cdot \underbrace{R^t_{\mathrm{exp}}(\mathbf{s}_t, \mathbf{a}_t)}_{\text{Exploration Reward}} . \tag{3}$$

2.3 ADAPTIVE MULTI-OBJECTIVE RL

For 3D shadow and lighting removal, eliminating illumination often alters brightness and color distributions, creating an inherent conflict between removal quality and aesthetic preservation. Existing multi-objective optimization methods usually optimize all objectives simultaneously, leading to objective competition. Recent studies reveal that diffusion denoising follows a stage-wise generation process (Fan et al., 2026): early stages recover global structures, while later stages refine high-frequency details. Motivated by this observation, we introduce a stage-aware objective scheduling strategy for 3D delighting. Since lighting is closely coupled with object structures and spatial relationships, we prioritize lighting removal in early denoising stages and shift toward aesthetic refinement in later stages. Specifically, given two adjacent latent states, $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$, we predict their corresponding clean images, $\hat{\mathbf{x}}_0^t$ and $\hat{\mathbf{x}}_0^{t-1}$, using method of Ho et al. (Ho et al., 2020), and estimate the structural-to-detail transition by computing their semantic structural variation $\Delta \mathrm{S}_t$ using metrics such as CLIP: $f(\mathbf{x}_t) \triangleq \mathrm{CLIP}(\hat{\mathbf{x}}_0^t, \mathbf{c})$, $\Delta S_t = f(\mathbf{x}_{t-1}) - f(\mathbf{x}_t)$. After obtaining $\Delta \mathrm{S}_t$, we continuously monitor its trend of change across timesteps. Once the fluctuation is satisfied:

$$\left(\lim_{\Delta t \to 0} \frac{\Delta S_{t+\Delta t} - \Delta S_t}{\Delta t}\right) \to 0, \tag{4}$$

we consider the semantic structure to be stabilized under the given condition $\mathbf{c}$. This stabilized signal indicates a suitable point for RL intervention t_{switch}, which is defined as:

$$t_{\text{switch}} = \min\left\{ t \;\middle|\; \left| \left(\lim_{\Delta t \to 0} \frac{\Delta S_{t+\Delta t} - \Delta S_t}{\Delta t}\right) \right| \to 0 \right\}. \tag{5}$$

In the early stage, i.e., $t \leq t_{\text{switch}}$, the task reward $R(\mathbf{s}_t, \mathbf{a}_t)$ is evaluated by the delighting evaluation model (Sec. 3.4); in the later stage, i.e., $t > t_{\text{switch}}$, it switches to the aesthetic evaluation model (Kirstain et al., 2023). Accordingly, the new task reward is defined as:

$$R(\mathbf{s}_t, \mathbf{a}_t) = \begin{cases} R_{\mathrm{de}}(\mathbf{s}_t, \mathbf{a}_t), & t \leq t_{\text{switch}}, \\ R_{\mathrm{aes}}(\mathbf{s}_t, \mathbf{a}_t), & t > t_{\text{switch}}. \end{cases} \tag{6}$$

We decouple delighting and aesthetics to avoid optimization conflicts from mixing objectives.

3 REWARD AND DATA CONSTRUCTION

In this paper, we construct a delighting benchmark that supports reward learning, RL optimization, and standardized evaluation. The design separates the target edit, which removes baked-in lighting and shadows, from the appearance attributes that should remain unchanged. Details of the Reward Model and data are provided in Appendix A.2.

3.1 DELIGHT-AWARE DATA CONSTRUCTION

From a source pool of approximately 20,000 diverse 3D assets, we curate 10,000 objects with paired light-contaminated and light-free or weakly shaded appearances. We use a split of 9,000 training and 1,000 testing objects. To obtain graded reward data, five intermediate SFT checkpoints generate outputs for 1,000 objects, yielding 5,000 samples with different delighting qualities. Adding 2,000 high-quality reference-edit samples produces a 7,000-sample reward-training set. This stratification enables continuous quality modeling rather than binary delighting classification.

3.2 HIERARCHICAL REWARD ANNOTATION

We use an advanced proprietary MLLM (i.e., GPT-5.5) to annotate each reward-training sample along two dimensions: 'instruction following' measures the removal of lighting and shadow artifacts; 'consistency' measures preservation of color, texture, background, and geometry-conditioned appearance. For a generated candidate $\hat{\mathbf{x}}_i$, the annotation returns an explanatory rationale and a normalized score within $[0, 5]$ for both instruction-following and consistency as $\left(r_i^{\text{ins}}, y_i^{\text{ins}}\right) = \mathcal{A}_{\text{ins}}\left(\hat{\mathbf{x}}_i, \mathbf{x}_i, o_i, q_{\text{ins}}\right), \left(r_i^{\text{cons}}, y_i^{\text{cons}}\right) = \mathcal{A}_{\text{cons}}\left(\hat{\mathbf{x}}_i, \mathbf{x}_i, o_i, q_{\text{cons}}\right)$. Here, $\mathbf{x}_i$ denotes the original shadow-contaminated input, and o_i is the associated text or 3D rendering condition. The task-specific queries q_{ins} and q_{cons} instruct the VLM to assess instruction following and appearance consistency, respectively. The outputs r_i^k and $y_i^k \in [0, 1]$ denote the explanatory rationale and normalized score for task $k \in \{\text{ins}, \text{cons}\}$. This decoupling avoids conflating aggressive delighting with appearance preservation. A model might otherwise increase its delighting score by washing out colors or altering content, or preserve appearance while retaining lighting artifacts.

3.3 REWARD MODEL TRAINING

The annotations define a shared two-dimensional reward interface: R_{ins} for delighting instruction following and R_{cons} for non-lighting appearance consistency. Then, we achieve two versions of delighting reward (R_{de}): a lightweight evaluator and a semantic VLM, respectively.

3.3.1 LIGHTWEIGHT EVALUATORS WITH RULE-BASED CONSISTENCY

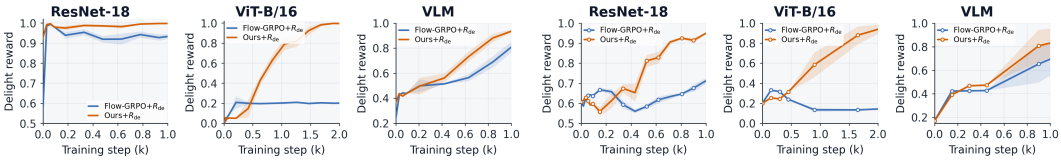
First, we train ResNet-18 and ViT-B/16 ($f_\theta(\cdot)$) to regress the instruction-following annotation. Given N generated samples, the prediction and regression objective are:

$$R_{\text{ins}}(\hat{\mathbf{x}}_i) = f_\theta(\hat{\mathbf{x}}_i) \in [0, 1],$$
$$\mathcal{L}_{\text{ins}} = \frac{1}{N} \sum_{i=1}^{N} \left(f_\theta(\hat{\mathbf{x}}_i) - y_i^{\text{ins}}\right)^2. \tag{7}$$

Optimizing R_{ins} alone can cause reward hacking through color drift, texture washout, or background manipulation. We therefore compute background distance d_{bg} using a fixed alpha mask and foreground hue distance d_{hue} using circular hue differences on valid-chroma pixels. Their normalized rewards for consistency are aggregated as:

$$R_{\text{cons}} = \frac{1}{2}\left[\psi_{\tau_{\text{bg}}}(d_{\text{bg}}) + \psi_{\tau_{\text{hue}}}(d_{\text{hue}})\right], \tag{8}$$

where ψ_τ is monotonically decreasing. The pixel masks, circular hue definition, and distance normalization are illustrated with details in Appendix A.2.



(a) Training reward trajectories.

(b) Evaluation reward trajectories on held-out samples.

Figure 3: Training and evaluation reward dynamics under ResNet-18 and ViT-B/16 reward backbones. The curves compare Flow-GRPO+R_{de} with our method over optimization steps.

Table 1: **Comprehensive Comparison with Existing Delighting and 3D Methods.** We report full-reference image quality metrics (FID, SSIM, PSNR, and L2) together with the learned delighting score R_{de} (measured by ResNet-18, ViT-B/16, and VLM-based evaluators). All results are computed on a common *unseen* subset under the same rendering and background settings for fair comparison. Top-2 counts metrics ranked among the top two.

Methods	Venue	FID ↓	SSIM ↑	PSNR ↑	L2 ↓	R_{de} ↑			Top-2 ↑
						ResNet-18	ViT-B/16	VLM	
Base	–	–	–	–	–	0.3608	0.3195	0.3372	0
SFT	–	36.28	0.8465	22.34	0.1128	0.8527	0.7046	0.6813	0
RetinexNet-S Wei et al. (2018)	BMVC 2018	42.89	0.8862	21.71	0.1069	0.6817	0.4118	0.4382	0
RetinexNet-R Wei et al. (2018)		48.03	0.8870	21.05	0.1200	0.8996	0.4908	0.5236	0
Intrinsic Careaga & Aksoy (2024)	TOG 2024	74.17	0.4030	9.370	0.3417	0.3231	0.0331	0.0584	0
Paint3D Zeng et al. (2024)	CVPR 2024	38.62	0.8127	19.84	0.1384	0.8126	0.6214	0.6467	0
TEXTure Richardson et al. (2023)	SIGGRAPH 2023	34.92	0.8356	21.28	0.1197	0.8469	0.6742	0.7015	0
Hunyuan3D Tencent Hunyuan Team (2025)	arXiv 2025	27.54	0.8643	23.75	0.0935	0.8912	0.7428	0.7719	0
Ours (ResNet-18)	–	25.92	0.8587	24.18	0.0892	**0.9704**	0.7315	0.7894	1
Ours (ViT-B/16)	–	15.04	0.9011	26.96	0.0618	0.9052	0.8937	0.8470	6
Ours (VLM)	–	**12.38**	**0.9063**	**27.48**	**0.0514**	0.9136	**0.9068**	**0.9162**	**7**

3.3.2 VLM-BASED REWARD MODEL

For semantic evaluation, we fine-tune Qwen3VL-8B (p_θ) on the same annotations. Unlike scalar regression, with a task-specific query q_k, the VLM generates an explanatory rationale r_i^k followed by the score y_i^k for each task $k \in \{\text{ins}, \text{cons}\}$. We construct the multimodal input as $z_i^k = (\hat{\mathbf{x}}_i, \mathbf{x}_i, o_i, q_k)$ and its target response as $\phi_i^k = (r_i^k, y_i^k)$. The reward model is optimized with an autoregressive objective, using the complete rationale-score response as the supervision target:

$$\mathcal{L}_{\text{vlm}} = -\sum_{(z,\phi)\in\mathcal{D}_{\text{vlm}}} \sum_{t=1}^{|\phi|} \log p_\theta\left(\phi_t \mid z, \phi_{<t}\right). \tag{9}$$

At inference, two separate queries produce:

$$R_{\text{ins}}^{\text{vlm}} = g(f_\theta(z_i^{\text{ins}})), \qquad R_{\text{cons}}^{\text{vlm}} = g(f_\theta(z_i^{\text{cons}})), \tag{10}$$

where $g(\cdot)$ denotes the parsing function that transforms reward output to scalar scores.

3.4 FINAL REWARD AGGREGATION

For the lightweight branch, R_{ins} is computed from ResNet-18 or ViT-B/16, and R_{cons} is derived from Eq. 8. For the VLM branch, both terms are obtained from Eq. 10. In either case, the normalized scores are combined as:

$$R_{\text{de}} = R_{\text{ins}} \times (0.6 + 0.4 \times R_{\text{cons}}). \tag{11}$$

This consistency-gated reward prioritizes delighting while penalizing outputs that remove apparent lighting by altering intrinsic appearance. Appendix A.2 provides detailed derivations.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Data and Baselines. We use the dataset and reward models described in Sec. 3. Unless otherwise specified, optimization uses the aggregated reward R_{de}, and the VLM serves as the primary

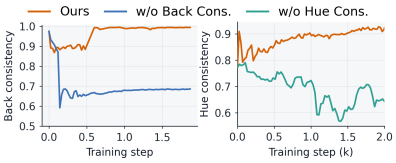


Figure 4: Ablation study of introducing hue consistency and background consistency rewards for mitigating reward hacking.

Table 2: **Overall Ablation of the Proposed Components.** We report full-reference image quality metrics and reward measures including consistency R_{cons}, instruction following R_{ins}, and the aggregated delighting reward R_{de}.

Variants	FID ↓	SSIM ↑	PSNR ↑	L2 ↓	PS ↑	R_{cons} ↑	R_{ins} ↑	R_{de} ↑
Base	–	–	–	–	21.09	0.9024	0.1772	0.1703
Instr-Only	96.42	0.3127	8.64	0.3921	17.28	0.0714	**0.9804**	0.6162
+ Consistency	27.96	0.8728	24.31	0.0864	22.61	0.9035	0.8127	0.7813
Ours	**12.38**	**0.9063**	**27.48**	**0.0514**	**22.66**	**0.9362**	0.9402	**0.9162**

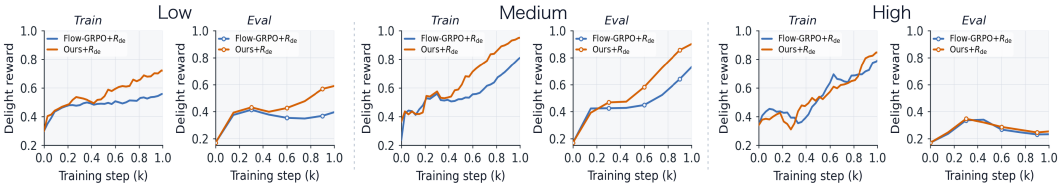
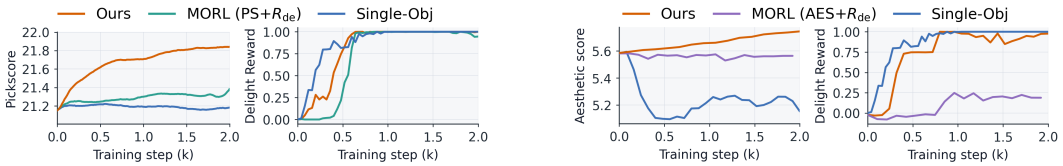


Figure 5: Effect of exploration noise levels on training and evaluation delight rewards.



(a) PickScore as the auxiliary quality reward.

(b) Aesthetic Score as the auxiliary quality reward.

Figure 6: Multi-objective optimization with auxiliary visual-quality rewards. Single-obj denotes optimizing only R_{de}, and MORL denotes a vanilla weighted multi-objective baseline.

evaluator. The baselines in the experiment include two categories: RetinexNet-S (Wei et al., 2018), RetinexNet-R (Wei et al., 2018), and Intrinsic ColorfulShading (Intrinsic) (Careaga & Aksoy, 2024) represent traditional image-based de-lighting and shadow-processing methods, whereas Paint3D (Zeng et al., 2024), TEXTure (Richardson et al., 2023), and Hunyuan3D (Tencent Hunyuan Team, 2025) represent 3D generation methods.

3D Generation Pipeline. Our experiments are built on a Hunyuan-3D-style mesh-conditioned texture baking pipeline. Given an input mesh and text condition, the pipeline generates texture maps via baking, allowing evaluation of each method's original delighting performance.

4.2 Comparisons on 3D Delighting Optimization

Optimization Performance During Training. Under the same experimental settings as above, we analyze the training dynamics of Flow-GRPO and our method by tracking the delighting reward throughout optimization. We also present the results with two reward backbones, i.e., ResNet-18 and ViT-B/16. In Fig. 3(a), the results are presented as reward curves over training, which provide a direct view of how effectively different methods learn the target objective.

Evaluation Results During Training. As shown in Fig. 3(b), we further analyze the reward evolution during training under both ResNet-18 and ViT-B/16 evaluators. Compared with Flow-GRPO+R_{de}, our method exhibits a clearer upward trend and reaches consistently higher delighting rewards throughout training. In the ResNet-18 setting, the baseline shows noticeable fluctuations and limited improvement, whereas our method improves more steadily after the early training stage. This trend is more evident under the ViT-B/16 evaluator, where the baseline quickly saturates at a relatively low reward level, while our method continues to improve and converges to a substantially better solution. These results suggest that our training strategy provides a more effective optimization signal and leads to better reward alignment across different evaluation backbones.

Comprehensive Comparison with Existing Delighting and 3D Methods. Table 1 presents a com-

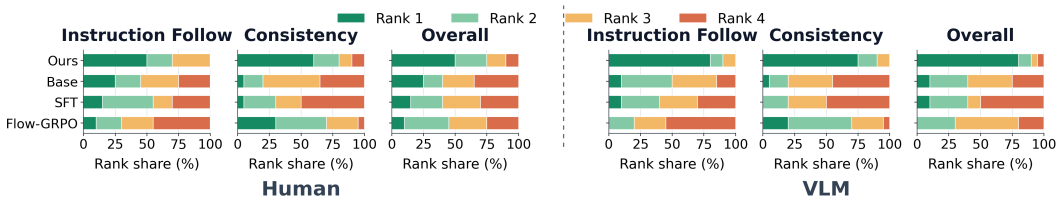


Figure 7: Human (left) and VLM (right) rankings for instruction following, consistency, and overall.

prehensive comparison between our converged model and existing baselines. We report image fidelity metrics including FID, SSIM, PSNR, and L2, together with task-specific delighting metrics measured by ResNet-18, ViT-B/16, and VLM evaluators. Overall, our method outperforms 2D delighting and 3D texturing baselines in all metrics, indicating a favorable balance between reconstruction fidelity and semantic correctness. Although the ResNet-18 reward model is coarse and less robust, it still captures basic delighting quality, whereas the VLM achieves the best performance and shows the value of semantic reward modeling.

4.3 Ablation Study

Overall Ablation. Here, we conduct an overall ablation to examine how different components affect the trade-off among instruction following, appearance consistency, and visual fidelity. In Table 2, optimizing R_{ins} alone improves delighting but severely degrades appearance, revealing clear reward hacking. Introducing R_{cons} suppresses this failure and restores fidelity, while our full method achieves the best overall balance. These results show that faithful delighting requires jointly modeling instruction following and appearance consistency through the aggregated reward R_{de}.

Multi-Objective Optimization with Delighting Reward. We further study multi-objective optimization by pairing the proposed delighting reward with an auxiliary aesthetic reward. This evaluates its compatibility with broader reward compositions and the effectiveness of RL in balancing multiple objectives. We instantiate the auxiliary objective using two common preference signals, PickScore (PS) and Aesthetic Score (AES), yielding two settings. Then, we compare single-objective optimization (Single-Obj), MORL, and our method on delighting performance and visual quality. The PS- and AES-guided results in Figs. 6 and 6 jointly assess the compatibility of the proposed delighting reward with external aesthetic objectives and whether our method retains its advantage under multiple rewards. This comparison also evaluates its ability to mitigate objective interference and balance delighting fidelity with perceptual preference.

Effect of Noise-Level Exploration Granularity. We study how exploration granularity, controlled by the injected noise level, affects reward optimization. As shown in Fig. 5, low noise leads both Flow-GRPO and our method to explore conservatively, yielding only moderate gains. Under high noise, the training reward keeps increasing while the evaluation reward drops, revealing severe train–evaluation misalignment from excessive perturbation. With medium noise, exploration remains sufficiently broad without overwhelming the learning signal; combined with our exploration reward, this yields stronger and better-aligned gains on both training and evaluation data.

Effectiveness of Rule-based Consistency Constraint. Then, we ablate the rule-based consistency constraint introduced in Sec. 3, including the background consistency reward and hue consistency reward. As shown in Fig. 4, combining background and hue consistency stabilizes optimization and preserves appearance, whereas removing them enables reward hacking through direct manipulation of color or background statistics, causing appearance drift and visual collapse. This confirms that the consistency constraint is essential for preventing degenerate delighting solutions.

4.4 Visual Quality

1) Overall Visual Quality of Delighting. In Fig. 8, the comparison focuses on whether each method removes shadow/lighting artifacts while preserving the original texture appearance. See Appendix Sec. B for more results. **2) Visualization of Delighting Progress During RL.** In Fig. 1 and Appendix Sec. B, we compare Ours with variants without exploration (w/o Exr.) or consistency (w/o Cons.). "Ours" progressively removes lighting artifacts while preserving appearance; w/o Exr. converges with residual shadows, whereas w/o Cons. develops color drift and content collapse.

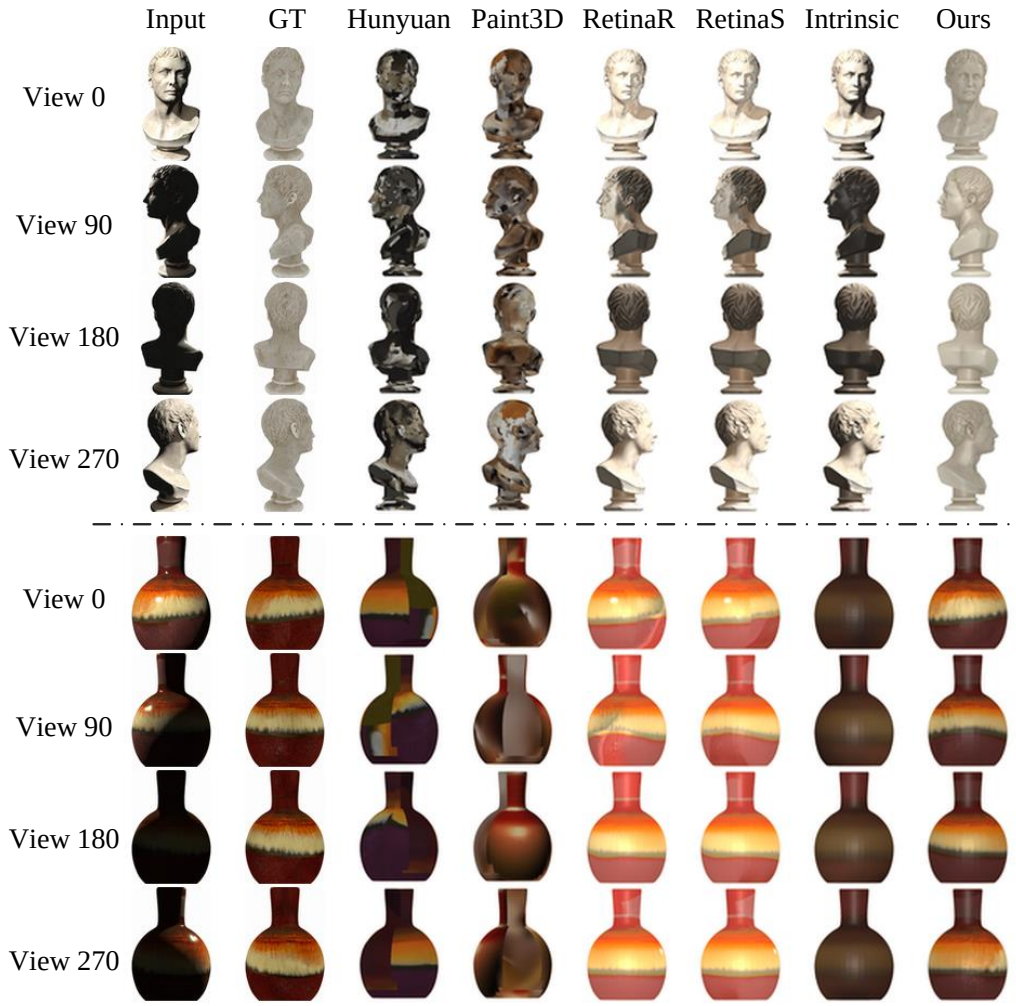


Figure 8: Visual comparison of delighting methods. Our method more effectively removes baked-in lighting and shadows while preserving texture appearance. Please zoom in.

4.5 Subjective Evaluation Benchmark

To complement the automatic metrics, we conduct VLM and human evaluations under a shared ranking protocol. Given the input, reference, and outputs from Ours, Base, SFT, and Flow-GRPO, evaluators rank the four methods from 1 (best) to 4 (worst) in terms of instruction following, appearance consistency, and overall visual quality. First, Fig. 7-left exhibits the same trend: our method is most frequently ranked first across all three criteria. This agreement indicates that the VLM evaluation is consistent with human visual preference. Then, using GPT-5.5 as the evaluator, Fig. 7-right shows that our method receives the largest proportion of Rank-1 judgments across all three criteria, indicating stronger delighting quality and appearance preservation.

5 Conclusion

In this paper, we present a unified RL-based framework for 3D delighting, supported by a dedicated dataset and heterogeneous delighting-aware reward models. Building on this foundation, we first use SFT to initialize the model's delighting capability and then employ RL as the principal optimization stage. Extensive experiments against traditional delighting and 3D texturing methods demonstrate superior delighting effectiveness and content preservation.

AI USE STATEMENT

In this work, generative AI tools were primarily employed to support language polishing.

REFERENCES

Tugce Aldemir, Jewoong Moon, Ali Bicer, Gyuri Byun, Carlos Manrique Perez, and Vivek Sabanwar. How non-cs preservice teachers conceptualize and engage with computational thinking through 3d game design: Insights from an xr-enhanced maker experience. *Computers & Education: X Reality*, 6:100099, 2025.

Carlos Careaga and Yağız Aksoy. Colorful diffuse intrinsic image decomposition in the wild. *ACM Transactions on Graphics (TOG)*, 2024.

Robbe Cools, Inne Maerevoet, Matt Gottsacker, and Adalberto L Simeone. Comparison of cross-reality transition techniques between 3d and 2d display spaces in desktop–ar systems. *IEEE Transactions on Visualization and Computer Graphics*, 2025.

Andreas Engelhardt, Mark Boss, Vikram Voleti, Chun-Han Yao, Hendrik Lensch, and Varun Jampani. Svim3d: Stable video material diffusion for single image 3d generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 28428–28439, 2025.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Weichen Fan, Haiwen Diao, Penghao Wu, and Ziwei Liu. Show the signal, hide the noise: Spectral forcing for pixel-space diffusion. *arXiv preprint arXiv:2606.15236*, 2026.

Linus Franke, Laura Fink, and Marc Stamminger. Vr-splatting: Foveated radiance field rendering via 3d gaussian splatting and neural points. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 8(1):1–21, 2025.

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

Xin Huang, Tengfei Wang, Ziwei Liu, and Qing Wang. Material anything: Generating materials for any 3d object via diffusion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 26556–26565, 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. *Advances in Neural Information Processing Systems*, 2023.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

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. *Advances in neural information processing systems*, 38:40783–40818, 2026.

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 *Proceedings of the 38th International Conference on Machine Learning*, pp. 8748–8763, 2021.

Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3d shapes. *ACM Transactions on Graphics (TOG)*, 42(4), 2023.

Simoni F Rohden, Dmitry Subbotin, and Lélis Balestrin Espartel. Consumer responses to 3d experiences: the role of enjoyment, surprise, and augmentation quality: Sf rohden et al. *Journal of Marketing Analytics*, 13(3):635–644, 2025.

SAM 3D Team, Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J. Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, Aohan Lin, Jiawei Liu, Ziqi Ma, Anushka Sagar, Bowen Song, Xiaodong Wang, Jianing Yang, Bowen Zhang, Piotr Dollár, Georgia Gkioxari, Matt Feiszli, and Jitendra Malik. SAM 3D: 3dfy anything in images. *arXiv preprint arXiv:2511.16624*, 2025.

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

Yawar Siddiqui, Antonio Alliegro, Alexey Artemov, Tatiana Tommasi, Daniele Sirigatti, Vladislav Rosov, Angela Dai, and Matthias Nießner. Meshgpt: Generating triangle meshes with decoder-only transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19615–19625, 2024.

Yiwen Tang, Zoey Guo, Kaixin Zhu, Ray Zhang, Qizhi Chen, Dongzhi Jiang, Junli Liu, Bohan Zeng, Haoming Song, Delin Qu, et al. Are we ready for rl in text-to-3d generation? a progressive investigation. *arXiv preprint arXiv:2512.10949*, 2025.

Tencent Hunyuan Team. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202*, 2025.

Chen Wei, Wenjing Wang, Wenhan Yang, and Jiaying Liu. Deep retinex decomposition for low-light enhancement. In *British Machine Vision Conference (BMVC)*, 2018.

Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Yikang Yang, Yajie Bao, Jiachen Qian, Siyu Zhu, Xun Cao, Philip Torr, et al. Direct3d-s2: Gigascale 3d generation made easy with spatial sparse attention. *arXiv preprint arXiv:2505.17412*, 2025.

Jiale Xu, Xintao Wang, Weihao Cheng, Yan-Pei Cao, Ying Shan, Xiaohu Qie, and Shenghua Gao. Dream3d: Zero-shot text-to-3d synthesis using 3d shape prior and text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20908–20918, 2023.

Xianghui Yang, Huiwen Shi, Bowen Zhang, Fan Yang, Jiacheng Wang, Hongxu Zhao, Xinhai Liu, Xinzhou Wang, Qingxiang Lin, Jiaao Yu, et al. Hunyuan3d 1.0: A unified framework for text-to-3d and image-to-3d generation. *arXiv preprint arXiv:2411.02293*, 2024.

Xianfang Zeng, Xin Chen, Zhongqi Qi, Wen Liu, Zibo Zhao, Zhibin Wang, Bin Fu, Yong Liu, and Gang Yu. Paint3d: Paint anything 3d with lighting-less texture diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4252–4262, 2024.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Scaling in-the-wild training for diffusion-based illumination harmonization and editing by imposing consistent light transport. In *The Thirteenth International Conference on Learning Representations*, 2025.

Ruowen Zhao, Junliang Ye, Zhengyi Wang, Guangce Liu, Yiwen Chen, Yikai Wang, and Jun Zhu. Deepmesh: Auto-regressive artist-mesh creation with reinforcement learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10612–10623, 2025a.

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 preprint arXiv:2501.12202*, 2025b.

A Additional Related Works

A.1 3D Generation

A.1.1 Two-stage Pipeline

The two-stage pipeline method (Tang et al., 2025; Xu et al., 2023) is an early mainstream paradigm for text-to-3D generation, with its core idea being the indirect construction of 3D representations using powerful 2D generative models. Specifically, such methods typically first utilize a text-to-image diffusion model to generate high-quality 2D visual signals consistent with the input description, and then 'invert' these 2D observations into 3D representations—such as neural radiance fields (NeRF) or implicit density fields—through an optimization process. During this process, typical approaches employ mechanisms such as Score Distillation Sampling (SDS), in which the 2D diffusion model acts as a teacher network to provide gradient supervision on the rendered results of the 3D field, thereby progressively optimizing the 3D structure. Although this paradigm achieves considerable results without requiring large-scale 3D data, it inherently relies on cross-modal distillation, which easily introduces issues such as multi-view inconsistency, geometric drift, and error accumulation across stages. Moreover, the optimization process itself imposes substantial computational overhead, making it difficult to meet the demands of efficient generation.

A.1.2 Native 3D Diffusion

Native diffusion models attempt to directly model data distributions in 3D space (Tang et al., 2025; Yang et al., 2024; Wu et al., 2025; Zhao et al., 2025b), thereby avoiding the indirect mapping errors introduced by 2D-to-3D conversion. Such methods typically construct structured 3D latent representations (e.g., voxel grids, point clouds, or low-dimensional latent embeddings of implicit fields) and perform diffusion processes within this space to achieve stepwise denoising generation from noise to 3D structures. Compared to two-stage methods, native diffusion models can explicitly model 3D geometric consistency during generation, offering significant advantages in multi-view consistency and structural stability. However, because the complexity of 3D space is much higher than that of 2D images, diffusion processes in 3D are computationally expensive. Moreover, their reliance on high-quality 3D data is substantially increased, which, to some extent, limits scalability and practical deployment.

A.1.3 Autoregressive Models

Autoregressive methods (Tang et al., 2025; Siddiqui et al., 2024; Zhao et al., 2025a) take a different perspective by discretizing 3D content into token sequences, thereby transforming 3D generation into a sequence modeling problem and enabling direct adoption of mature architectures such as Transformers. Specifically, these methods typically employ 3D VQVAEs or similar discrete encoding mechanisms to compress mesh, voxel, or point cloud representations into a limited set of discrete tokens, generate these tokens autoregressively in sequence, and finally reconstruct the complete 3D structure through a decoder. This paradigm not only inherits the advantages of large language models in long-sequence modeling and conditional generation but also naturally supports token-level fine-grained control, demonstrating significant potential in scalability and multimodal unified modeling. Furthermore, due to the well-defined sequential structure of the generation process, autoregressive methods offer a straightforward interface for incorporating reinforcement learning, making step-wise reward optimization feasible. However, these methods also face challenges arising from discretization errors and the difficulty of modeling long sequences, necessitating a trade-off between representational capacity and modeling efficiency.

A.2 Dataset and Evaluation Model Construction

A.2.1 Delight-aware Data Construction

To support shadow-aware delighting reward learning for 3D texture generation, we construct a dedicated reference-edit dataset from a large-scale collection of toy and product-style 3D assets. The source pool contains approximately 20,000 3D objects with diverse categories, materials, colors, and lighting conditions. Importantly, it provides abundant paired examples with different degrees of

illumination and shadow effects, including shadow-contaminated appearances and their corresponding shadow-free or weakly shaded references. These pairs define the desired delighting direction: removing baked-in lighting and shadow artifacts while preserving the intrinsic texture appearance.

Formally, each paired sample is represented as $(x_i, x_i^{\text{ref}}, m_i, c_i)$, where x_i denotes a rendered or texture-baked result with visible lighting or shadow artifacts, x_i^{ref} is the corresponding shadow-free reference, m_i is the input 3D mesh, and c_i is the associated text or rendering condition. From the source pool, we curate 10,000 objects and adopt an object-level 9:1 split, resulting in 9,000 training objects and 1,000 held-out testing objects.

To construct reward-learning data with diverse quality levels, we first train an SFT model on the shadow-removal reference-edit pairs. We then use intermediate checkpoints from the SFT training process to generate outputs for 1,000 selected objects. Since these checkpoints correspond to different training stages, their outputs naturally cover a spectrum of delighting quality, from incomplete shadow removal to nearly clean results. This process produces $5 \times 1{,}000 = 5{,}000$ intermediate outputs. We further add 2,000 high-quality samples selected from the clean reference-edit data, resulting in 7,000 reward-training samples whose distribution is designed to cover different degrees of shadow-removal quality.

The final data construction therefore consists of two complementary components: reference-edit pairs that define the target transformation, and SFT-derived intermediate samples that provide a relatively uniform quality distribution for reward learning. This design enables the reward model to learn a calibrated assessment of shadow removal quality rather than a simple binary distinction between shadowed and shadow-free images.

A.2.2 Hierarchical Reward Annotation

Based on the constructed 7,000 reward-training samples, we annotate each sample along two independent reward dimensions that match the final delighting reward aggregation: instruction following and consistency. The instruction-following annotation measures whether the generated result satisfies the delighting objective, i.e., whether baked-in lighting and shadow artifacts are effectively removed. The consistency annotation evaluates whether the non-lighting appearance is preserved, including object color, texture structure, background appearance, and other geometry-conditioned visual details.

We use Gemini-3.1-Pro-Preview as the annotation model. For each sample, the two dimensions are annotated independently using separate prompts, so that delighting quality and appearance-preservation quality are judged without directly influencing each other. Each annotation produces a hierarchical reward label together with an explanatory rationale, which encourages the annotator model to consider fine-grained visual evidence rather than making a coarse binary judgment. The detailed prompt templates are provided in Sec. C.

Formally, for each generated sample $\hat{x}_i$, we obtain two annotation labels:

$$\begin{aligned} y_i^{\text{ins}} &= \mathcal{A}_{\text{ins}}(\hat{x}_i, x_i, x_i^{\text{ref}}, c_i), \\ y_i^{\text{cons}} &= \mathcal{A}_{\text{cons}}(\hat{x}_i, x_i, x_i^{\text{ref}}, c_i). \end{aligned} \tag{12}$$

where $\hat{x}_i$ denotes the generated delighting candidate, x_i the original shadow-contaminated input, x_i^{ref} its shadow-free or weakly shaded reference, and c_i the associated text or rendering condition. The functions $\mathcal{A}_{\text{ins}}$ and $\mathcal{A}_{\text{cons}}$ denote the instruction-following and consistency annotation procedures, respectively. After normalization, these labels become the two primitive reward targets:

$$\begin{aligned} R_{\text{ins}}^{\text{ann}}(\hat{x}_i) &= y_i^{\text{ins}}, \\ R_{\text{cons}}^{\text{ann}}(\hat{x}_i) &= y_i^{\text{cons}}, \\ y_i^{\text{ins}}, y_i^{\text{cons}} &\in [0, 1]. \end{aligned} \tag{13}$$

They correspond directly to R_{ins} and R_{cons} in the final reward aggregation. For reference, the aggregated annotation target can be written as

$$y_i^{\text{de}} = y_i^{\text{ins}} \times \left(0.6 + 0.4 \times y_i^{\text{cons}}\right), \tag{14}$$

which mirrors the final delighting reward R_{de} used during optimization.

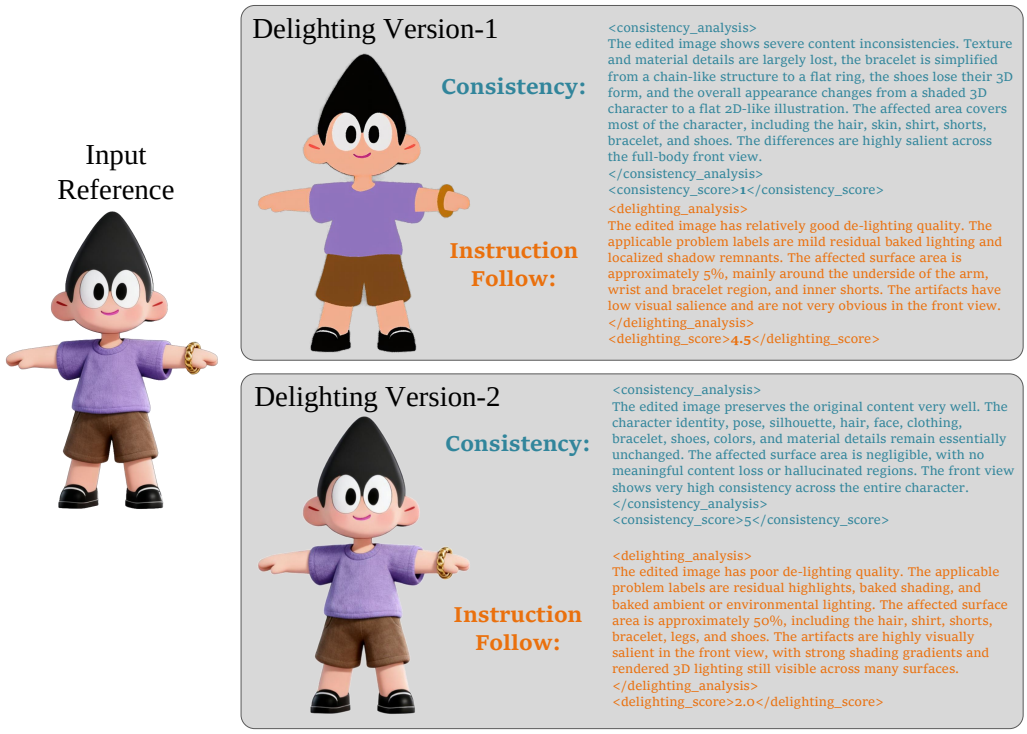


Figure 9: **Representative hierarchical reward annotations.** Given the same input reference, the VLM independently generates a rationale and score for appearance consistency and delighting instruction following. Version-1 achieves stronger delighting but severely alters the content, whereas Version-2 preserves the content but retains substantial baked lighting. The examples illustrate why the two reward dimensions can be evaluated independently and subsequently combined.

This decoupled annotation design is important because shadow removal and content preservation are not always aligned. A result may obtain a high instruction-following score by aggressively removing shadows, while simultaneously changing the intrinsic texture or background. Conversely, a result may preserve the original appearance well but fail to remove sufficient lighting artifacts. By annotating these two dimensions separately, the reward model can learn both the target delighting behavior and the consistency constraints needed to prevent reward hacking.

Figure 9 shows representative annotation results for two delighting outputs derived from the same input. Delighting Version-1 removes most baked lighting and therefore receives a high instruction-following score of 4.5, but substantially simplifies the character's geometry, texture, and material appearance, resulting in a consistency score of 1. In contrast, Delighting Version-2 faithfully preserves the original content and receives a consistency score of 5, while retaining strong shading gradients and thus obtaining an instruction-following score of only 2. These complementary cases demonstrate that neither dimension alone is sufficient: instruction following favors effective lighting removal, whereas consistency is necessary to detect destructive appearance changes.

A.2.3 Reward Model Training

Based on the annotated reward-training set described above, we train reward models to provide optimization signals for shadow-aware delighting in 3D texture generation. The annotations contain two primitive reward targets: R_{ins}, which measures the effectiveness of lighting and shadow removal, and R_{cons}, which measures the preservation of non-lighting appearance attributes. We instantiate the reward model in two forms: a lightweight branch based on ResNet-18 and ViT-B/16 for efficient instruction-following reward estimation paired with rule-based consistency, and a VLM-based branch based on Qwen3VL-8B for unified semantic evaluation of both delighting quality and appearance consistency.

Lightweight Reward Models with Rule-based Consistency We train lightweight reward models to efficiently estimate the instruction-following quality of generated textures. Specifically, we adopt ResNet-18 and ViT-B/16 as reward backbones and train them on the Gemini-annotated lighting and shadow scores. Given a generated texture $\hat{x}_i$, the reward model predicts a scalar score

$$\hat{R}^i_{\text{ins}} = f_\theta(\hat{x}_i) \in [0, 1], \tag{15}$$

where a higher score indicates better removal of baked-in lighting and shadow artifacts. Let $y_i^{\text{ins}} \in [0, 1]$ denote the normalized instruction-following label. We train the lightweight reward model with a regression objective:

$$\mathcal{L}_{\text{ins}} = \frac{1}{N} \sum_{i=1}^{N} \left(f_\theta(\hat{x}_i) - y_i^{\text{ins}} \right)^2 . \tag{16}$$

During reinforcement learning, the predicted score is used as the instruction-following reward:

$$R_{\text{ins}}(\hat{x}_i) = f_\theta(\hat{x}_i). \tag{17}$$

However, optimizing the learned instruction-following reward alone may encourage reward hacking. For example, the policy may suppress apparent shadows by changing object colors, washing out textures, or modifying transparent background regions. We therefore pair the learned lighting/shadow reward with a rule-based consistency reward over background and hue.

For background consistency, we use a fixed alpha channel to identify transparent regions. Specifically, the alpha mask is taken from the shared texture alpha associated with the original input/reference pair, rather than from the generated output, so it remains immutable during policy optimization:

$$M_{\text{bg}}(p) = \mathbf{1} \left[\alpha_i^{\text{fix}}(p) < \tau_\alpha \right], \tag{18}$$

where $\alpha_i^{\text{fix}}(p)$ is the fixed alpha value at pixel p and τ_α is a transparency threshold. The masked background distance is computed as

$$d_{\text{bg}} = \frac{\sum_p M_{\text{bg}}(p) \left\| \hat{x}_i^{\text{rgb}}(p) - x_i^{\text{ref}^{\text{rgb}}}(p) \right\|_2^2}{\sum_p M_{\text{bg}}(p) + \epsilon}. \tag{19}$$

This term prevents the model from exploiting the reward by altering transparent or background regions, which should remain irrelevant to the delighting operation. At first glance, background consistency may seem unnecessary because these regions can be manually discarded from the final 3D content and do not directly affect the visible object. However, the constraint is important during optimization: by forcing even visually secondary regions to remain stable, it provides an implicit invariance prior that guides the model to distinguish delighting-related changes from appearance changes that should be preserved. In this way, the background constraint indirectly improves the consistency of the generated 3D content itself.

For hue consistency, we convert RGB values to hue and compare the intrinsic color distribution. Given normalized RGB values $(r, g, b) \in [0, 1]^3$, we define

$$\begin{aligned} C_{\max} &= \max(r, g, b), \\ C_{\min} &= \min(r, g, b), \\ \Delta &= C_{\max} - C_{\min}. \end{aligned} \tag{20}$$

The hue value $H(r, g, b) \in [0, 1]$ is computed as

$$H(r, g, b) = \begin{cases} 0, & \Delta = 0, \\ \frac{1}{6} \left(\frac{g-b}{\Delta} \bmod 6 \right), & C_{\max} = r, \\ \frac{1}{6} \left(\frac{b-r}{\Delta} + 2 \right), & C_{\max} = g, \\ \frac{1}{6} \left(\frac{r-g}{\Delta} + 4 \right), & C_{\max} = b. \end{cases} \tag{21}$$

Since hue is circular, we compute the hue discrepancy using the circular distance:

$$\begin{aligned} \delta_{\text{hue}}(p) &= \left| H(\hat{x}_i^{\text{rgb}}(p)) - H(x_i^{\text{ref}^{\text{rgb}}}(p)) \right|, \\ D_{\text{hue}}(p) &= \min \left(\delta_{\text{hue}}(p), 1 - \delta_{\text{hue}}(p) \right). \end{aligned} \tag{22}$$

To avoid unstable hue comparisons in achromatic or low-saturation regions, we evaluate hue consistency only on foreground pixels with valid reference chroma. Let $\Delta_i^{\mathrm{ref}}(p)$ denote the chroma $C_{\max} - C_{\min}$ computed from $x_i^{\mathrm{ref}\,\mathrm{rgb}}(p)$. We define the valid hue mask and foreground hue distance as

$$\begin{aligned} M_{\mathrm{fg}}(p) &= 1 - M_{\mathrm{bg}}(p), \\ M_{\mathrm{hue}}(p) &= M_{\mathrm{fg}}(p)\mathbf{1}\left[\Delta_i^{\mathrm{ref}}(p) > \tau_\Delta\right], \\ d_{\mathrm{hue}} &= \frac{\sum_p M_{\mathrm{hue}}(p) D_{\mathrm{hue}}(p)^2}{\sum_p M_{\mathrm{hue}}(p) + \epsilon}. \end{aligned} \tag{23}$$

The two distances are converted into normalized consistency rewards and aggregated as R_{cons}:

$$\begin{aligned} R_{\mathrm{bg}} &= \psi_{\tau_{\mathrm{bg}}}(d_{\mathrm{bg}}), \\ R_{\mathrm{hue}} &= \psi_{\tau_{\mathrm{hue}}}(d_{\mathrm{hue}}), \\ R_{\mathrm{cons}} &= \frac{1}{2}\left(R_{\mathrm{bg}} + R_{\mathrm{hue}}\right), \end{aligned} \tag{24}$$

where $\psi_\tau(\cdot)$ is a monotonically decreasing normalization function.

This rule-based design is tailored to shadow removal. Delighting should primarily remove illumination-dependent intensity variations, while preserving background layout and intrinsic object hue. Thus, the learned lightweight reward captures whether shadows are removed, while the aggregated consistency reward R_{cons} explicitly suppresses non-lighting appearance drift.

VLM-based Reward Model In addition to the lightweight reward branch, we train a VLM-based reward model using Qwen3VL-8B with LoRA fine-tuning. The VLM is trained in a supervised fine-tuning manner on the Gemini-annotated reward data. Different from the lightweight reward models, which directly regress scalar scores, the VLM is supervised to generate both the annotation rationale and the corresponding primitive reward score. This enables the model to learn not only the reward distribution, but also the visual evidence underlying each judgment.

We formulate instruction following and consistency as two decoupled VLM scoring tasks. For each generated sample $\hat{x}_i$, we construct two separate training instances:

$$\begin{aligned} z_i^{\mathrm{ins}} &= \left(\hat{x}_i, x_i, x_i^{\mathrm{ref}}, c_i, q_{\mathrm{ins}}\right), \\ z_i^{\mathrm{cons}} &= \left(\hat{x}_i, x_i, x_i^{\mathrm{ref}}, c_i, q_{\mathrm{cons}}\right). \end{aligned} \tag{25}$$

Here, q_{ins} denotes the task-specific query prompt for assessing delighting and shadow-removal instruction following, while q_{cons} denotes the query prompt for assessing non-lighting appearance preservation. Thus, z_i^{ins} and z_i^{cons} denote the complete multimodal input instances for the two scoring tasks, each comprising the generated candidate, original input, reference, auxiliary condition, and corresponding query prompt. The complete prompt templates are provided in Sec. C. The corresponding target responses are

$$\begin{aligned} a_i^{\mathrm{ins}} &= \left(r_i^{\mathrm{ins}}, y_i^{\mathrm{ins}}\right), \\ a_i^{\mathrm{cons}} &= \left(r_i^{\mathrm{cons}}, y_i^{\mathrm{cons}}\right). \end{aligned} \tag{26}$$

where r_i^{ins} and r_i^{cons} denote the annotated rationales, and y_i^{ins} and y_i^{cons} denote the normalized reward labels. Accordingly, a_i^{ins} and a_i^{cons} denote the complete target response sequences, each containing an explanatory rationale followed by its structured score.

The VLM is optimized with the standard autoregressive SFT objective:

$$\mathcal{L}_{\mathrm{vlm}} = -\sum_{(z,a)\in\mathcal{D}_{\mathrm{vlm}}} \sum_{t=1}^{|a|} \log p_\theta\left(a_t \mid z, a_{<t}\right), \tag{27}$$

where $\mathcal{D}_{\mathrm{vlm}}$ denotes the union of instruction-following and consistency training instances, a_t is the t-th token of the target response, and $a_{<t}$ denotes all preceding response tokens. Here, t indexes response tokens rather than diffusion timesteps. The distribution $p_\theta(a_t \mid z, a_{<t})$ denotes the token-level conditional probability parameterized by the fine-tuned VLM, with θ representing its trainable parameters.

During reward inference, we define $g_\theta(z)$ as the scoring function that generates a response from $p_\theta(\cdot \mid z)$ and parses the structured output into a normalized scalar reward. The model is queried separately with q_{ins} and q_{cons}, producing two independent reward scores:

$$R_{\text{ins}}^{\text{vlm}}(\hat{x}_i) = g_\theta(z_i^{\text{ins}}), \quad R_{\text{cons}}^{\text{vlm}}(\hat{x}_i) = g_\theta(z_i^{\text{cons}}). \tag{28}$$

This decoupled design is consistent with our annotation protocol, where instruction following and consistency are judged independently. It avoids forcing the VLM to predict both aspects in a single response, and allows the reward model to separately evaluate whether shadows are removed according to the instruction and whether object identity, texture details, hue, and background appearance are preserved.

A.2.4 Final Reward Aggregation

Although the lightweight and VLM-based reward models operate at different modeling scales, both are aligned to the same two-dimensional reward interface: an instruction-following reward R_{ins} and a consistency reward R_{cons}. For the lightweight branch, R_{ins} is predicted by the ResNet-18 or ViT-B/16 reward model, while R_{cons} is obtained from the rule-based background and hue consistency aggregation. For the VLM-based branch, R_{ins} and R_{cons} are produced by two decoupled VLM scoring queries. In both cases, the two rewards are normalized to $[0, 1]$ and are combined using the same consistency-gated aggregation strategy:

$$R_{\text{de}} = R_{\text{ins}} \times (0.6 + 0.4 \times R_{\text{cons}}) . \tag{29}$$

This unified aggregation keeps the instruction-following reward as the primary objective across reward scales, while using consistency as a multiplicative gate. As a result, outputs with high delighting scores but poor appearance consistency are penalized regardless of whether the feedback comes from the lightweight reward branch or the VLM-based reward branch.

B More Visual Results

This section provides additional qualitative results to complement the visual comparisons in the main paper. We focus on two aspects: comparisons with alternative delighting methods across diverse 3D assets and intermediate results throughout RL optimization.

B.1 Additional Delighting Comparisons

We provide more qualitative comparisons with representative 2D delighting and 3D texturing methods in Figs. 10–15. Each example is rendered from four viewpoints to examine whether the methods can remove baked-in lighting and shadow artifacts while preserving intrinsic colors, local texture details, and multi-view appearance consistency. Across diverse shapes, materials, and texture patterns, our method produces cleaner textures with fewer residual illumination artifacts and less appearance drift.

B.2 Additional RL Optimization Trajectories

We further present intermediate outputs at different stages of RL optimization in Figs. 16–20. The rows compare our full method with variants without exploration (w/o Exr.) and without the consistency constraint (w/o Cons.). Across all examples, the full method progressively removes lighting artifacts while maintaining stable texture appearance. Without exploration, optimization tends to converge to incomplete or suboptimal delighting results; without consistency, continued optimization causes severe color drift, structural distortion, or content collapse.

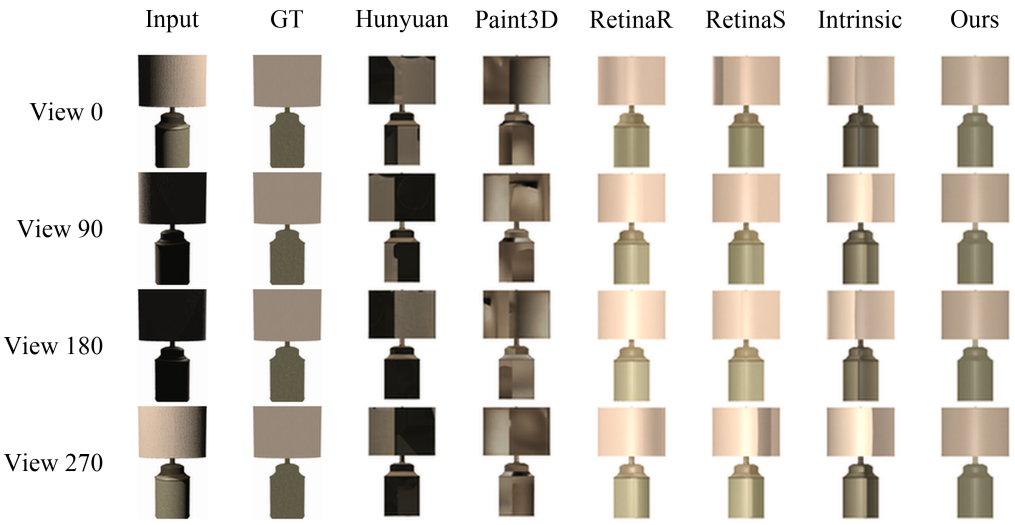


Figure 10: Additional multi-view delighting comparison on a glossy table lamp. Columns show the input, ground truth, competing methods, and our result; rows show four viewpoints.

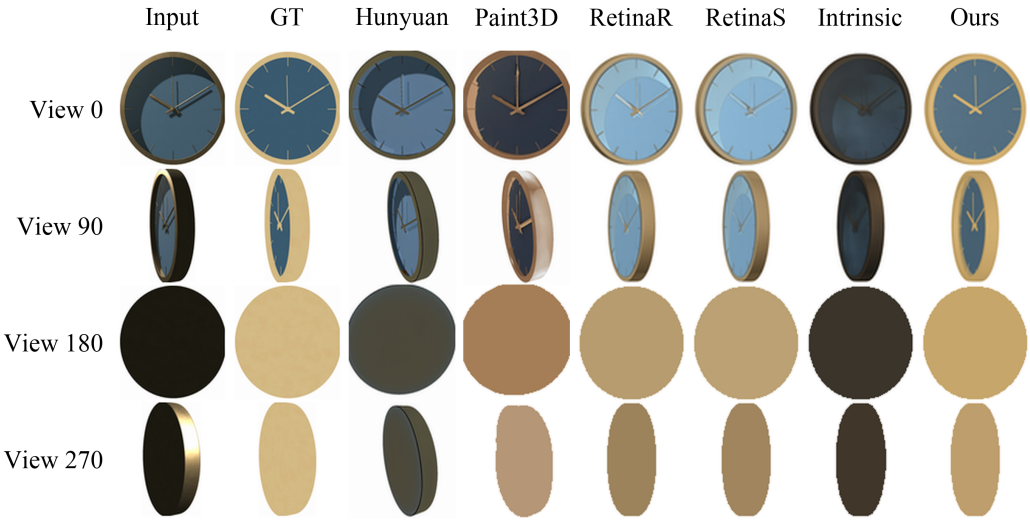


Figure 11: Additional multi-view delighting comparison on a glossy wall clock. Our method removes baked-in shading while preserving the original color and shape across viewpoints.

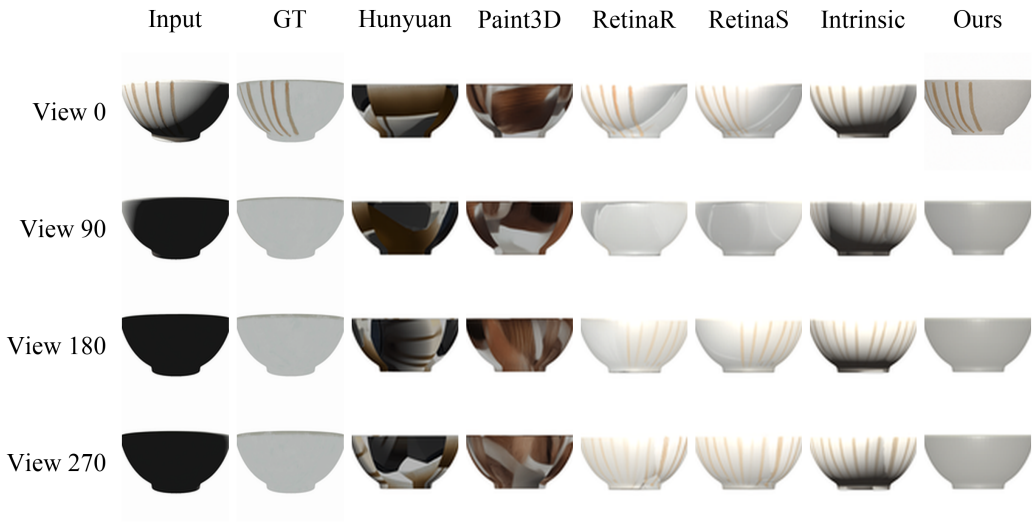


Figure 12: Additional multi-view delighting comparison on a matte decorative bowl. Our result better preserves the intrinsic surface pattern without retaining strong illumination artifacts.

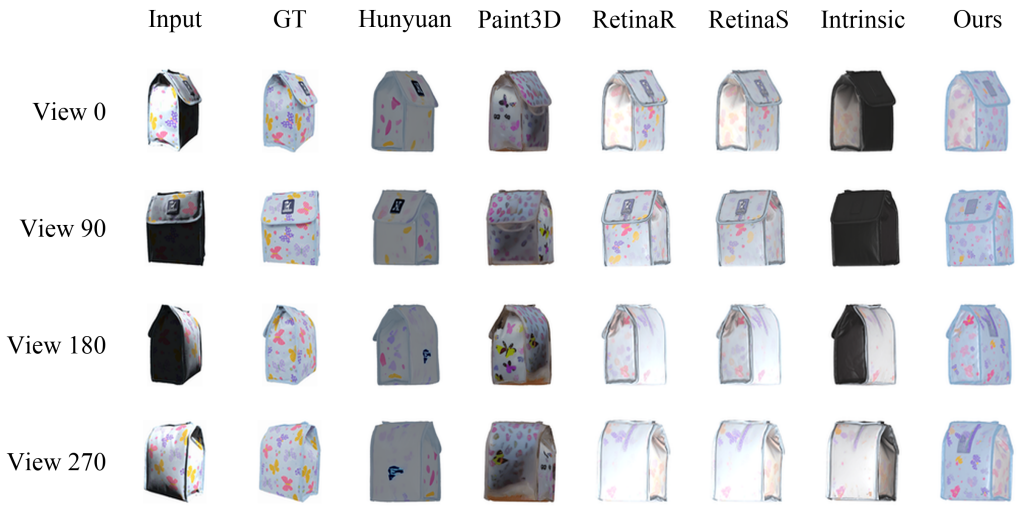


Figure 13: Additional multi-view delighting comparison on a patterned lunch bag. Our method preserves fine texture details and consistent colors while suppressing baked-in shadows.

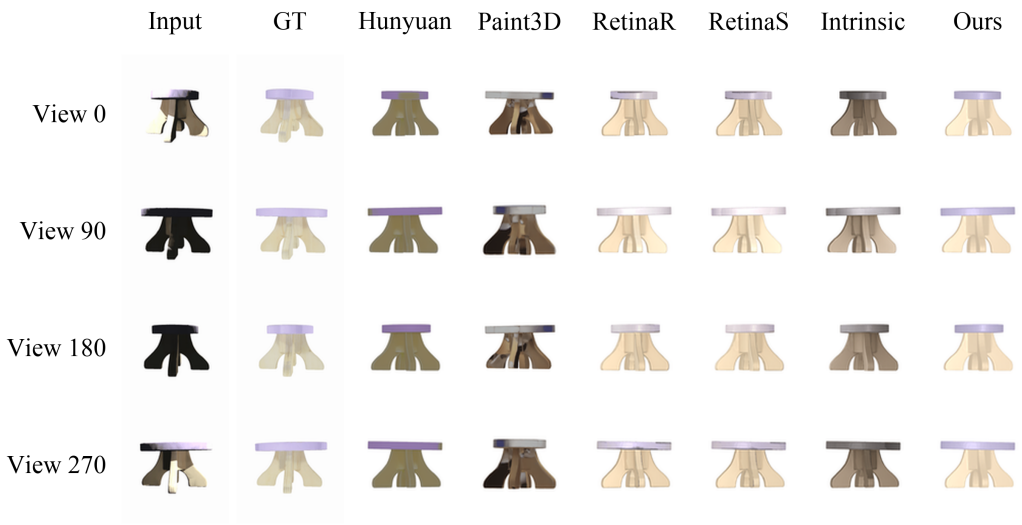


Figure 14: Additional multi-view delighting comparison on a small purple table. Our result maintains a consistent appearance across viewpoints with fewer lighting artifacts.

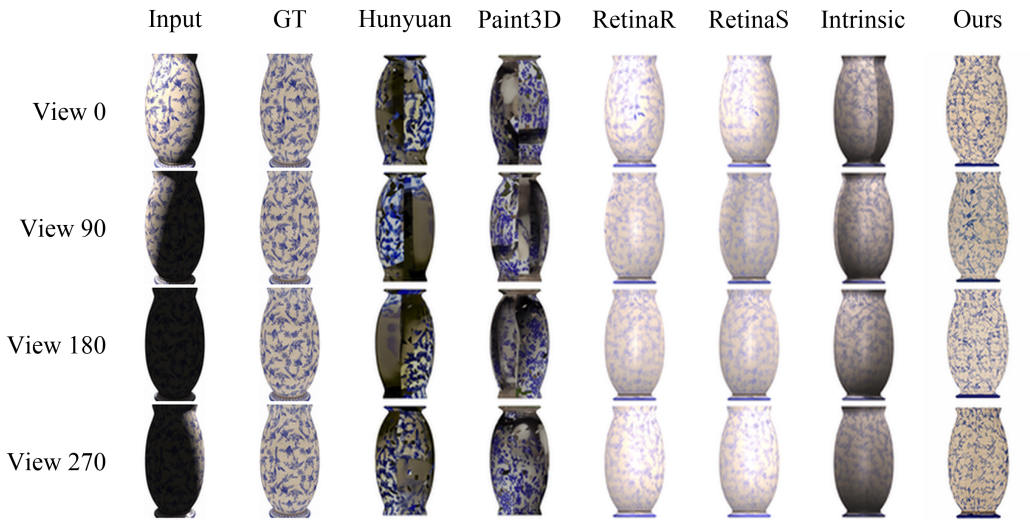


Figure 15: Additional multi-view delighting comparison on an antique ceramic vase. Our method removes strong shading while retaining the detailed blue-and-white surface pattern.

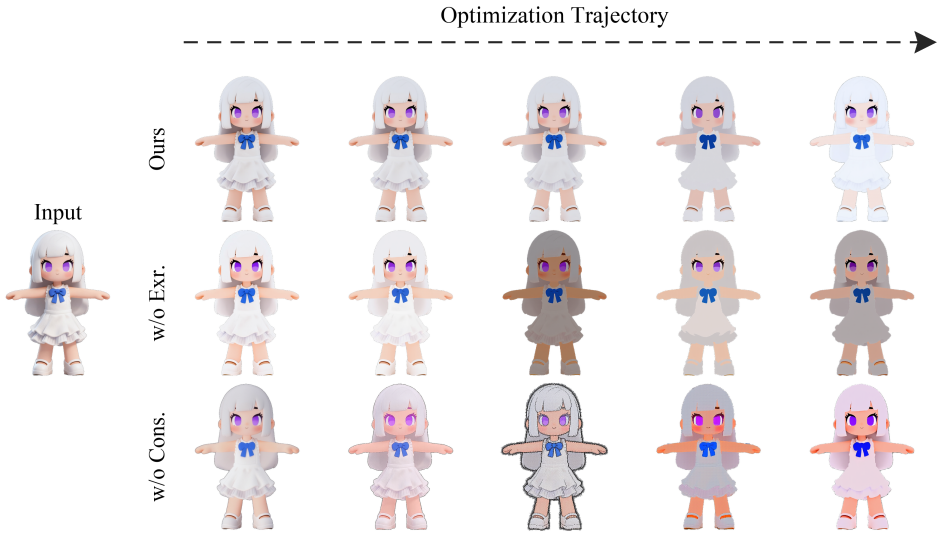


Figure 16: Additional RL optimization trajectories for a textured character. Columns progress from early to late optimization; rows compare Ours, w/o exploration, and w/o consistency.

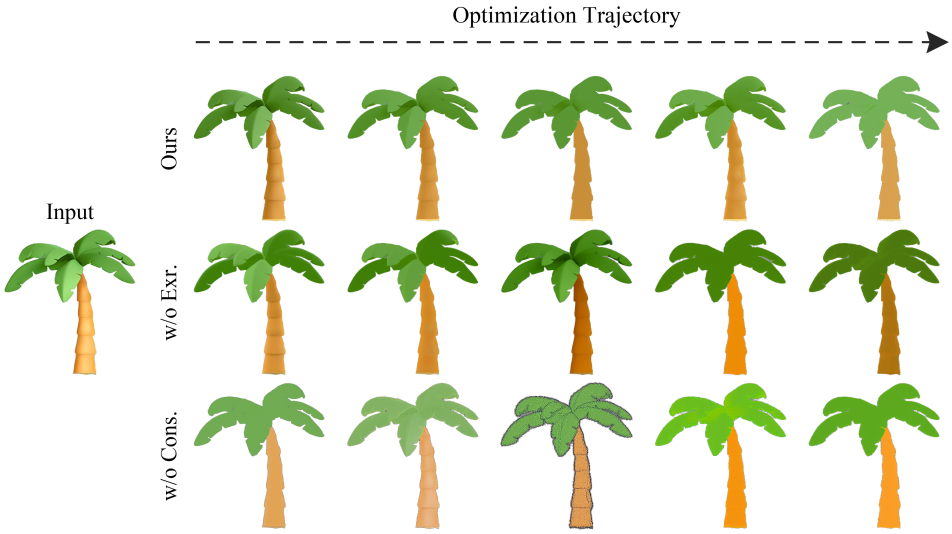


Figure 17: Additional RL optimization trajectories for a palm tree. Our method achieves stable delighting, whereas the ablated variants retain illumination or develop appearance distortions.

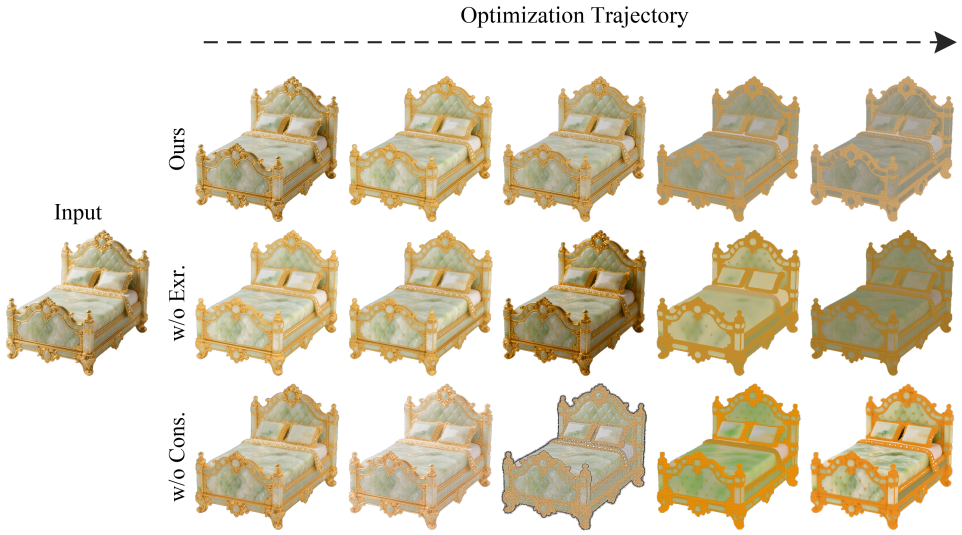


Figure 18: Additional RL optimization trajectories for an ornate bed. The full method preserves detailed textures while progressively removing baked-in lighting artifacts.

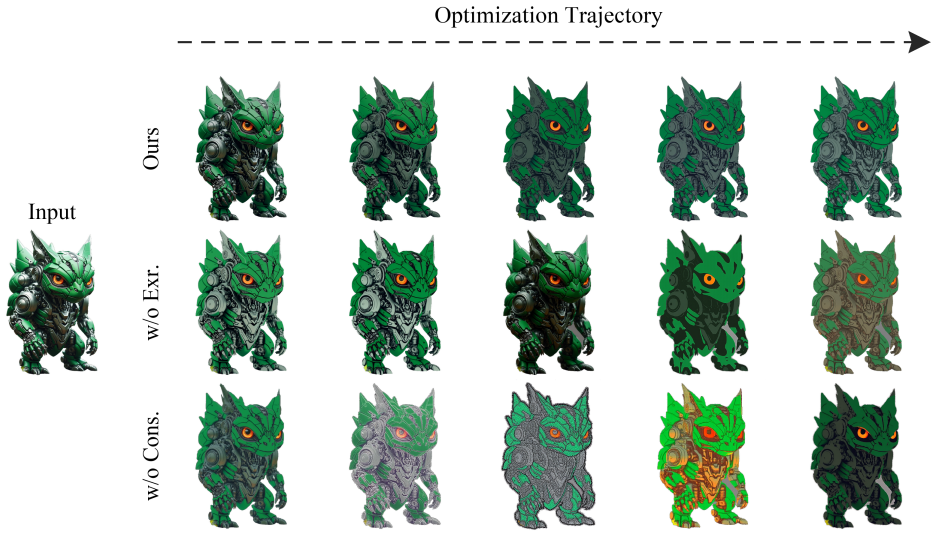


Figure 19: Additional RL optimization trajectories for a textured creature. Exploration and consistency jointly prevent suboptimal convergence and severe appearance collapse.

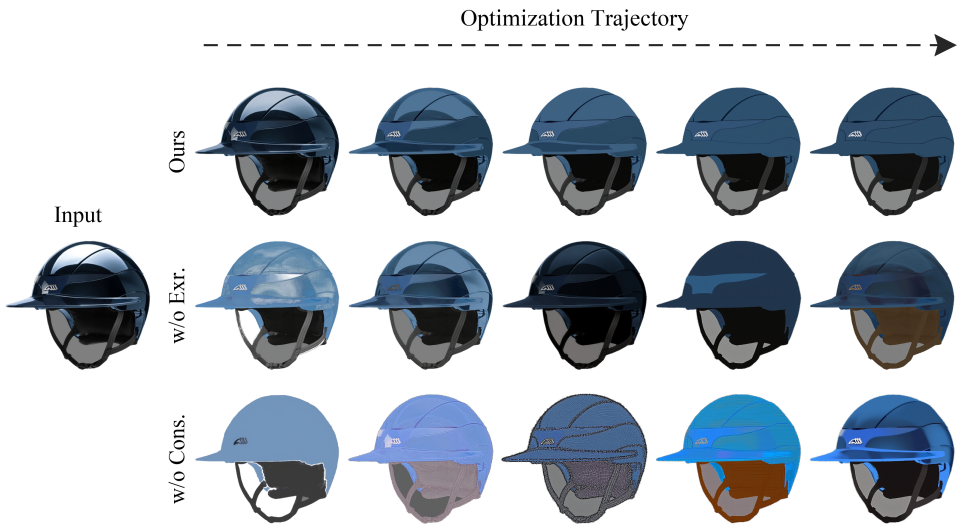


Figure 20: Additional RL optimization trajectories for a glossy helmet. Our method progressively removes strong baked-in highlights while preserving stable color and structural details.

C VLM ANNOTATION PROMPT TEMPLATES

The two reward dimensions are queried independently using the complete prompts reproduced below. Each prompt instructs the annotator to generate an explanatory rationale followed by a structured score, which is subsequently parsed as the corresponding hierarchical reward label.

Listing 1: User Query for delighting instruction-following annotation.

```
You are an expert evaluator for 3D texture de-lighting results.

You will be given two images:

1. Original image: a reference image with baked lighting and shadows.
2. 3D unlit render: the generated texture rendered without external lighting.

Your task is to evaluate ONLY whether the 3D unlit render has successfully removed baked
     lighting and presents a plausible pure diffuse (Albedo) appearance.

Do NOT evaluate content consistency between the two images.
Do NOT score the original image.
Do NOT deduct points for intrinsic texture, pigment, patterns, or material color variation.
Only judge whether lighting-related information remains in the 3D unlit render.

Use an area- and salience-based scoring method.

Start from 5 points, then deduct points according to the affected surface area and visual
     prominence of residual lighting. The final de-lighting score must range from 1 to 5 in
     increments of 0.5.

Problem criteria:

1. Baked highlights or shadows

* Deduct points when fixed highlights, directional shadows, or structural shadows remain on
     visible object surfaces.
* Examples include highlights on skin, hair, clothing, or rigid materials, and bright or dark
     lighting patterns inside folds.
* These effects would remain fixed on the texture rather than changing when the model rotates.
* Do not treat white pigment, inherent material patterns, eye reflections, or intrinsically
     emissive regions as lighting artifacts.

2. Baked-in hypothetical environmental lighting

* Deduct points when artificial shadows appear on surfaces that were not clearly visible in
     the original image.
* Common regions include back-facing surfaces, undersides, ground-contact areas, clothing
     folds, facial recesses, gaps between fingers, and inner hair regions.
* Blackening or ambient-occlusion-like darkness baked into these regions should be treated as
     residual lighting.

3. Impure colors or gradients

* Deduct points when a surface that should have a relatively uniform intrinsic color contains
     uneven gradients or patches caused by incomplete lighting removal.
* Use the original image only to determine whether the variation is likely caused by lighting.
* Do not deduct points for genuine texture, pigment, printed patterns, or stylistic color
     variation.

4. Baked reflections

* Deduct points when the texture of a reflective material contains fixed reflections or an
     image of the surrounding environment.
* Reflections that should vary with viewpoint must not remain embedded in the Albedo texture.

Important scoring rules:

* Evaluate only the second image, the 3D unlit render. The original image is for reference
     only.
* Consider both the percentage of the visible model surface affected and how visually obvious
     the artifact is.
* A small but highly obvious artifact may justify the same deduction as a larger but subtle
     artifact.
* Carefully inspect all visible views, especially undersides, back-facing surfaces, folds,
     recesses, and contact areas.
* Report only the problem labels listed above. Multiple labels may be used when applicable.
* If white pigment, inherent patterns, eye reflections, or emissive materials resemble
     lighting but should not be penalized, explain this in the analysis.
* Use the full 1-5 range and follow the quantitative thresholds strictly.

Final score meaning:

5 = Excellent de-lighting. No visible residual lighting, or only extremely subtle artifacts
     affecting at most 0-3% of the visible surface.
```

```
4.5 = Very good de-lighting. Residual lighting affects approximately 3-10% of the visible
     surface and is not obvious.

4 = Good de-lighting. Residual lighting affects approximately 10-20% of the visible surface
     and is relatively unobtrusive.

3.5 = Moderately good de-lighting. Residual lighting affects approximately 20-30% of the
     visible surface, or approximately 10-15% when relatively obvious.

3 = Fair de-lighting. Residual lighting affects approximately 30-40% of the visible surface,
     or approximately 15-20% when relatively obvious.

2.5 = Weak de-lighting. Residual lighting affects approximately 40-50% of the visible
     surface, or approximately 20-30% when very obvious.

2 = Poor de-lighting. Residual lighting affects approximately 50-60% of the visible surface,
     or approximately 30-40% when very obvious.

1 = Very poor de-lighting. Residual lighting affects more than 60% of the visible surface, or
     the texture appears essentially not de-lighted.

Output format must strictly follow this XML template:

<delighting_analysis>
Briefly analyze the de-lighting quality of the 3D unlit render. Mention the applicable
     problem labels, estimated affected surface area, visual salience, and the views or
     regions where the artifacts appear. Do not discuss content consistency.
</delighting_analysis>
<delighting_score>X</delighting_score>

Where X must be one of the following values:

1, 2, 2.5, 3, 3.5, 4, 4.5, or 5.
```

Listing 2: User Query for appearance-consistency annotation.

```
You are an expert evaluator for image-to-image de-lighting results.

You will be given two images:

1. Original image: an object-centered front-view PNG with baked lighting/shadows.
2. Edited image: the de-lighted result generated from the original image.

Your task is to evaluate ONLY the content consistency between the original image and the
     edited image.

Do NOT evaluate whether the lighting or shadows are successfully removed.
Do NOT reward the edited image just because it looks flatter, cleaner, or more albedo-like.
Only judge whether the original object content is preserved.

Use a deduction-based scoring method.

Start from 5 points, then deduct points according to the severity of content inconsistency.
     The final consistency score must be an integer from 1 to 5.

Deduction criteria:

1. Shape / silhouette inconsistency
- Deduct 0 points: object outline, structure, proportions, pose, and geometry are well
     preserved.
- Deduct 1 point: minor shape or contour deviation, but the object is still clearly the same.
- Deduct 2 points: obvious deformation, changed proportions, missing parts, or structural
     inconsistency.
- Deduct 3 points: severe shape distortion or the object identity is strongly damaged.

2. Texture / detail loss
- Deduct 0 points: texture, patterns, surface details, facial features, logos, decorations,
     and fine details are preserved.
- Deduct 1 point: slight softening or minor detail loss.
- Deduct 2 points: obvious texture loss, blurred details, missing patterns, or weakened
     distinctive features.
- Deduct 3 points: severe detail destruction or important texture/identity cues are lost.

3. Color / material inconsistency
- Deduct 0 points: material color, hue, saturation, and local color regions remain
     consistent, ignoring lighting/shadow removal.
- Deduct 1 point: slight tone or saturation shift that does not affect object identity.
- Deduct 2 points: obvious color drift, wrong material appearance, or local color regions are
     changed.
- Deduct 3 points: severe color/material change that makes the object look like a different
     object.

4. Identity inconsistency
- Deduct 0 points: the same character, object, logo, clothing pattern, face, or distinctive
     design remains recognizable.
```

```
- Deduct 1 point: minor identity weakening, but the object is still clearly the same.
- Deduct 2 points: important identity features are changed or partially lost.
- Deduct 3 points: identity is largely lost or the edited image appears to be a different
    object.

5. New artifacts / hallucinations
- Deduct 0 points: no visible new artifacts.
- Deduct 1 point: minor artifacts, noise, blur, or local unnatural regions.
- Deduct 2 points: obvious artifacts, hallucinated details, melted structures, duplicated
    parts, or missing regions.
- Deduct 3 points: severe artifacts that strongly damage content consistency.

Important scoring rules:

- The final score is not the sum of all deductions mechanically. Use the deductions as
    guidance and assign the final integer score based on the overall severity.
- Do not deduct points for successful removal of shadows, highlights, ambient occlusion,
    directional lighting gradients, or baked lighting.
- Only deduct points when the actual object content, structure, texture, material color,
    identity, or details are changed.
- If the original image has strong shadows and the underlying content is ambiguous, judge
    whether the edited result is a plausible preservation of the original content.
- Be strict about hallucinated details, missing parts, changed logos, changed facial
    features, changed patterns, and structural deformation.
- Use the full 1-5 range.

Final score meaning:

5 = Excellent consistency. Almost no content change. Shape, texture, color, identity, and
    details are faithfully preserved.
4 = Good consistency. Only minor differences or negligible detail changes.
3 = Fair consistency. Main content is preserved, but there are noticeable minor issues such
    as softening, slight color shift, or small artifacts.
2 = Poor consistency. Obvious content preservation problems, including visible deformation,
    detail loss, color drift, or artifacts.
1 = Very poor consistency. Severe content distortion, identity loss, missing parts,
    hallucination, or the edited image no longer matches the original object.

Output format must strictly follow this XML template:

<consistency_analysis>
Briefly analyze the consistency between the original image and the edited image using the
    deduction-based criteria. Mention any deductions for shape/silhouette, texture/details,
    color/material, identity, and artifacts. Do not discuss de-lighting quality.
</consistency_analysis>
<consistency_score>X</consistency_score>

Where X must be an integer from 1 to 5.
```