Confined Vibe Optimization: Closed-Loop Text-to-Appearance Inverse Rendering with Vision–Language Feedback

Chi-Chang Lee
University of Maryland, College Park
changlee@umd.edu

Wenxuan Wu
University of Maryland, College Park
wenxuan6@terpmail.umd.edu

Abstract

We study an unpaired text-to-appearance inverse rendering problem: given only a text instruction, we optimize a differentiable renderer's continuous control parameters to synthesize an image that matches the requested look. Direct CLIP-driven optimization is often unstable and under-constrained, frequently yielding attribute entanglement or degenerate solutions. We propose a closed-loop method that uses a vision-language model to (i) parse the current rendering into foreground and background objects with attributes, producing swap-safe prompt augmentations, and (ii) generate a parameter-wise optimization plan that selects a small set of parameters to update and restricts their feasible ranges. We then perform differentiable optimization under this plan using a region-aware CLIP objective and stability safeguards. Experiments show improved overall prompt adherence and object visibility compared to a naive CLIP-only baseline, and ablations indicate that our method supports controllable refinement under prompt extension while remaining robust when instructions are semantically underspecified. GitHub: `https://github.com/ChangLee0903/Vibe-Rendering`

1. Introduction

Inverse rendering aims to recover physically meaningful scene parameters, commonly geometry, material reflectance, and illumination, from one or more images, enabling relighting, retexturing, and physically grounded scene understanding. Despite decades of progress, inverse rendering remains fundamentally ill posed due to strong ambiguities among appearance factors, for example albedo and illumination trade offs, and the presence of global illumination effects. Classical analyses provide theoretical foundations for estimating lighting and reflectance and for characterizing these ambiguities [6, 8]. More recently, learning based approaches improve robustness by combining differentiable rendering with priors learned from data, often trained on large scale physically based synthetic datasets where ground truth intrinsics are available [4, 9]. However, such progress typically depends on paired supervision, either explicit ground truth decompositions and scene parameters or tightly controlled capture setups, because real world imagery rarely comes with paired target renderings or accurate per scene labels for geometry, lighting, and SVBRDF, motivating methods that operate with weaker, unpaired forms of supervision [4, 9].

A growing body of work therefore explores unpaired, target free inverse rendering by replacing pixel aligned targets with high level constraints provided by multimodal foundation models. One line of research uses pretrained image text encoders as differentiable objectives, optimizing a renderable 3D representation so that differentiably rendered images match a language description in a joint embedding space, for example CLIP guided optimization of textured meshes [3]. Another line leverages pretrained text to image diffusion models as powerful natural image priors, distilling their guidance through differentiable rendering to optimize 3D parameters without paired targets, popularized by score distillation objectives and extended to higher quality mesh based pipelines [5, 7]. Subsequent work improves stability and diversity through refined distillation formulations such as variational score distillation [10]. Complementarily, instruction guided pipelines demonstrate that diffusion based image editing can be coupled with 3D optimization loops to impose textual edits consistently across views [1, 2], highlighting the promise of language as a flexible supervisory signal when paired targets are unavailable.

Existing unpaired formulations, however, are still far from delivering reliable parameter estimation in realistic inverse rendering settings. Because language or generative priors supervise images at semantic or perceptual levels, optimization can exploit shortcut solutions that satisfy the prompt while drifting away from plausible parameters, and these failure modes are exacerbated by the intrinsic ambiguities of inverse rendering [6, 8]. This challenge becomes particularly acute when the optimization variables are high dimensional or over parameterized, leading to unstable convergence, poor identifiability, and inconsistent results across views, which has been widely observed in diffusion guided optimization settings [7, 10]. Moreover, many prior unpaired methods optimize implicit 3D representations or free form textures, whereas practical inverse rendering pipelines often require structured, physically interpretable parameter sets with explicit bounds, sharing constraints, and trainable or frozen decisions [4]. In this work, we propose a language constrained inverse rendering framework that turns unpaired supervision into actionable and stable parameter updates by explicitly confining the optimization space. Our key idea is to introduce a constraint designer, implemented with a vision language model, that takes rendered images, a structured parameter specification, and textual feedback as input and outputs a parameter wise optimization plan, including which parameters are trainable, which are shared or locked to prevent scale ambiguities, and what tightened confidence ranges and priors should be applied. The differentiable renderer then optimizes only within this confined space under language based guidance, and the updated renderings are fed back for iterative refinement. This closed loop interaction stabilizes optimization in the absence of paired targets while preserving physically interpretable controls and preventing common degeneracies.

Contributions. This paper makes three contributions. First, we formulate unpaired inverse rendering as a closed loop optimization problem in which language and vision foundation models provide supervision in lieu of paired target images, while a differentiable renderer supplies gradients to structured scene parameters. Second, we introduce a constraint designer that produces parameter wise optimization plans, including trainable versus frozen decisions, parameter sharing to mitigate scale ambiguities, and tightened confidence ranges with optional priors, enabling stable optimization for complex parameterizations. Third, we propose an interactive refinement protocol that alternates between constrained differentiable optimization and constraint redesign, and we show that this interaction improves convergence, controllability, and robustness under purely unpaired, instruction driven supervision.

2. Related Work

Inverse rendering has been studied extensively, with early work analyzing the ambiguity between reflectance and illumination and establishing principled formulations for recovering intrinsic components [6, 8]. Learning based inverse rendering methods commonly combine differentiable rendering with learned priors and rely on synthetic supervision, particularly for indoor scenes with spatially varying lighting and SVBRDF [4, 9]. While these approaches can be accurate under paired supervision, they face a supervision gap in realistic settings where ground truth parameters or paired target renderings are unavailable.

Recent unpaired alternatives replace pixel aligned targets with language and generative priors. CLIP guided optimization maximizes similarity between rendered images and textual descriptions in a joint embedding space, enabling text driven mesh and texture optimization without paired targets [3]. Diffusion guided approaches distill gradients from pretrained text to image diffusion models through differentiable rendering, enabling target free optimization of 3D representations, as demonstrated by score distillation objectives and mesh refinement pipelines [5, 7]. Follow up work improves stability and diversity by modifying the distillation objective, for example via variational formulations [10]. Instruction guided pipelines further demonstrate that textual edits can be applied consistently across views by coupling diffusion based editing with 3D optimization loops [1, 2]. Despite these advances, reliable parameter estimation remains challenging when the optimization space is large and structured, motivating methods that explicitly control trainable sets, parameter tying, and feasible ranges.

3. Method

3.1. Problem Setup

We optimize a differentiable renderer $\mathcal{R}$ that maps a structured scene parameter vector $\boldsymbol{\theta}$ to an RGB image:

$$\mathbf{I} = \mathcal{R}(\boldsymbol{\theta}). \tag{1}$$

The parameter vector $\boldsymbol{\theta}$ spans camera configuration, illumination, material like shading controls, participating media effects (fog), and post processing. Supervision is unpaired: we are given only a text instruction $\mathtt{t}$ describing a target appearance, without any reference image. The goal is to estimate $\boldsymbol{\theta}$ such that the rendered image is semantically aligned with $\mathtt{t}$ under a vision language similarity metric, while maintaining numerical stability during gradient based optimization.

3.2. Differentiable Parameterization and Rendering

We design $\boldsymbol{\theta}$ as a collection of continuous, differentiable variables so that gradients can flow from an image level objective back to each controllable factor. The parameterization includes camera viewpoint, light intensity and color with an ambient term, material like controls that affect diffuse and specular appearance, fog terms that modulate depth and height dependent attenuation, and post processing controls such as exposure, contrast, hue, saturation, gamma, and vignetting. To ensure stable optimization under weak supervision, each parameter is constrained to a feasible domain via projection after every update. We additionally sanitize non finite values and apply gradient norm clipping to prevent exploding updates.

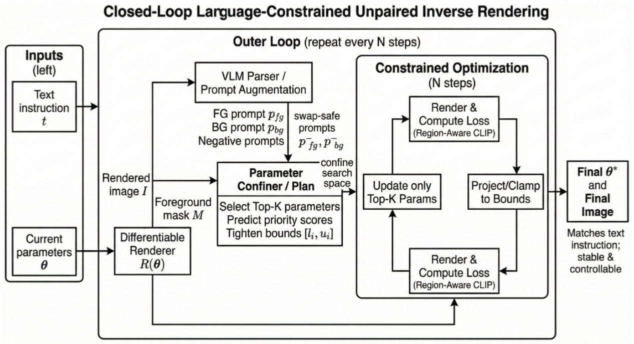
Closed-Loop Larguage-Constrained Unpaired Inverse Rendering
Outer Loop (repeat every N steps)
Inputs
(left)
Text instruction t
Current parameters $\boldsymbol{\theta}$
VLM Parser / Prompt Augmentation
FG prompt p_{fg}
BG prompt p_{bg}
Negative prompts
swap-safe prompts p^{-}_{fg}, p^{-}_{bg}
confine search space
Rendered image I
Foreground mask M
Parameter Confiner / Plan
Select Top-K parameters
Predict priority scores
Tighten bounds $[l_i, u_i]$
Differentiable Renderer $R(\boldsymbol{\theta})$
Constrained Optimization
(N steps)
Render & Compute Loss (Region-Aware CLIP)
Update only Top-K Params
Project/Clamp to Bounds
Render & Compute Loss (Region-Aware CLIP)
Final θ^* and Final Image
Matches text instruction; stable & controllable

Figure 1. System Diagram.

3.3. Overview

Our approach performs unpaired inverse rendering by alternating between constrained gradient descent and language mediated constraint redesign. At each outer round, we first render the current image and extract a soft foreground mask. We then parse the current rendering together with the user instruction into an objects list with region specific attributes, and convert these structured attributes into swap safe foreground and background prompts plus negative prompts. Next, a confiner produces a parameter wise plan consisting of refined feasible ranges and priority scores. We select a top K subset of parameters by priority for gradient updates while enforcing the refined ranges for all parameters through projection. Finally, we run a fixed number of inner optimization steps using region decomposed vision language losses, followed by numerical stabilization through projection and non finite repair. The full procedure is summarized in Algorithm 1 and Figure 1.

3.4. Text Guidance via Vision Language Similarity

We supervise optimization using a pretrained vision language model that embeds images and texts into a shared space. Let $\mathcal{E}_I(\cdot)$ and $\mathcal{E}_T(\cdot)$ denote the image and text encoders. For a rendered image $\mathbf{I}$ and a prompt $\mathbf{p}$, we define cosine similarity

$$s(\mathbf{I}, \mathbf{p}) = \left\langle \frac{\mathcal{E}_I(\mathbf{I})}{\|\mathcal{E}_I(\mathbf{I})\|}, \frac{\mathcal{E}_T(\mathbf{p})}{\|\mathcal{E}_T(\mathbf{p})\|} \right\rangle. \qquad (2)$$

We minimize a positive alignment loss

$$\mathcal{L}_{pos}(\mathbf{I}, \mathbf{p}) = 1 - s(\mathbf{I}, \mathbf{p}), \qquad (3)$$

and use a hinge style negative loss to discourage forbidden concepts:

$$\mathcal{L}_{neg}(\mathbf{I}, \mathbf{p}^{-}) = \max\big(0, s(\mathbf{I}, \mathbf{p}^{-}) - m\big), \qquad (4)$$

where m is a margin. The negative term is critical in our setting because unpaired text supervision can otherwise be satisfied through unintended attribute transfer, such as background attributes leaking into the foreground or vice versa.

Algorithm 1 Region aware language constrained unpaired inverse rendering

```
Require: Differentiable renderer $\mathcal{R}$, initial parameters $\boldsymbol{\theta}^{(0)}$, text instruction $\mathbf{t}$, outer rounds $R$, inner steps $T$, optimization budget $K$.
for $r = 0$ to $R - 1$ do
    Render image and mask: $(\mathbf{I}^{(r)}, M^{(r)}) \leftarrow \mathcal{R}(\boldsymbol{\theta}^{(r)})$.
    Parse objects and attributes: $\mathcal{O}^{(r)} \leftarrow \mathrm{Parse}(\mathbf{I}^{(r)}, \mathbf{t})$.
    Build prompts: $(\mathbf{p}_{fg}, \mathbf{p}_{bg}, \mathbf{p}^{-}_{fg}, \mathbf{p}^{-}_{bg}) \leftarrow \mathrm{Compose}(\mathcal{O}^{(r)}, \mathbf{t})$.
    Confiner plan: for each parameter $\theta_i$, predict bounds $[l_i, u_i]$ and priority $\pi_i$ using $(\mathbf{I}^{(r)}, \mathbf{t}, \mathbf{p}_{fg}, \mathbf{p}_{bg})$.
    Select update set: $\mathcal{S}_K^{(r)} \leftarrow \mathrm{TopK}(\{\theta_i\}, \pi_i, K)$.
    Apply bounds to all parameters by projection: $\boldsymbol{\theta}^{(r)} \leftarrow \Pi_{[l,u]}(\boldsymbol{\theta}^{(r)})$.
    for $t = 1$ to $T$ do
        Render: $(\mathbf{I}, M) \leftarrow \mathcal{R}(\boldsymbol{\theta})$.
        Masked images: $\mathbf{I}_{fg} \leftarrow \mathbf{I} \odot M, \mathbf{I}_{bg} \leftarrow \mathbf{I} \odot (1 - M)$.
        Compute loss $\mathcal{L}$ using region positives and negatives:
            $\mathcal{L} \leftarrow \lambda_{fg}\mathcal{L}_{pos}(\mathbf{I}_{fg}, \mathbf{p}_{fg}) + \lambda_{bg}\mathcal{L}_{pos}(\mathbf{I}_{bg}, \mathbf{p}_{bg}) + \lambda_{full}\mathcal{L}_{pos}(\mathbf{I}, \mathbf{t}) + \lambda_{neg}\big(\mathcal{L}_{neg}(\mathbf{I}_{fg}, \mathbf{p}^{-}_{fg}) + \mathcal{L}_{neg}(\mathbf{I}_{bg}, \mathbf{p}^{-}_{bg})\big)$.
        Gradient update only on $\mathcal{S}_K^{(r)}$: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}_{\mathcal{S}_K^{(r)}}} \mathcal{L}$.
        Stabilize: clip gradient norms, project $\boldsymbol{\theta} \leftarrow \Pi_{[l,u]}(\boldsymbol{\theta})$, repair non finite values.
    end for
    Set $\boldsymbol{\theta}^{(r+1)} \leftarrow \boldsymbol{\theta}$.
end for
return Estimated parameters $\boldsymbol{\theta}^{(R)}$.
```

3.5. Region Aware Prompt Decomposition with Objects and Attributes

A single text instruction often entangles foreground and background attributes, and abstract style words may not directly map to optimizable parameters. To reduce ambiguity, we introduce a scene parsing module that converts the input text and the current rendered image into an explicit set of objects and their attributes.

Given the current rendering $\mathbf{I}$ and instruction $\mathbf{t}$, the parser outputs an *objects list* $\mathcal{O} = \{o_j\}$, where each object o_j is assigned to a region $r_j \in \{\mathrm{fg}, \mathrm{bg}\}$ and accompanied by a structured attribute dictionary $\mathbf{a}_j$. Attributes include color, material cues, lighting descriptors, style descriptors, mood words, and coarse shape descriptors. When the instruction contains abstract terms such as cozy, cinematic, or dreamy, the parser grounds them into visually actionable attributes conditioned on the image context, for example warm soft lighting, low contrast, gentle vignetting, or slight desaturation. This grounding step converts vague language into constraints that can be expressed through the available parameter families.

From $\mathcal{O}$ we derive region specific prompts:

$$\mathbf{p}_{fg} = \mathrm{Prompt}(\{o_j : r_j = \mathrm{fg}\}), \qquad (5)$$

$$\mathbf{p}_{bg} = \mathrm{Prompt}(\{o_j : r_j = \mathrm{bg}\}), \qquad (6)$$

together with swap prevention negative prompts $\mathbf{p}^-_{fg}$ and $\mathbf{p}^-_{bg}$ that explicitly forbid foreground attributes in the background and background attributes in the foreground. The resulting prompts are short, region specific, and designed to minimize attribute swapping.

3.6. Confiner: Priority Based Parameter Selection and Range Refinement

Even with improved prompts, many parameters remain coupled and can compensate for each other. Optimizing all parameters simultaneously often leads to degenerate solutions. We therefore introduce a confiner module that produces a parameter wise optimization plan conditioned on the current image and the decomposed prompts.

For each parameter θ_i, the confiner predicts (i) a feasible range $[l_i, u_i]$ (or component wise bounds for vector parameters), and (ii) an integer priority score π_i indicating how important it is to optimize this parameter in the current outer round. The confiner is informed by the instruction, region specific prompts $\mathbf{p}_{fg}$ and $\mathbf{p}_{bg}$, negative prompts, and the current rendering, enabling it to focus on parameters most likely to explain the remaining mismatch.

We enforce a top K optimization budget. Let $\mathcal{S}_K$ be the set of K parameters with the smallest priorities among those deemed relevant. Only parameters in $\mathcal{S}_K$ receive gradients in the subsequent inner loop, while all other parameters are held fixed:

$$\mathcal{S}_K = \mathrm{TopK}\big(\{\theta_i\}, \pi_i, K\big), \quad \nabla_{\theta_i}\mathcal{L} = 0 \ \text{ for } \ \theta_i \notin \mathcal{S}_K. \qquad (7)$$

Separately, the confiner provided bounds are applied to all parameters through projection, which prevents drift even for frozen parameters. This separation between *range refinement* (global) and *gradient updates* (top K) improves identifiability and stability.

3.7. Region Decomposed Objective and Closed Loop Optimization

We optimize an objective that explicitly separates foreground and background alignment. Let M denote a soft foreground mask obtained from rendering, and define foreground and background composites $\mathbf{I}_{fg}$ and $\mathbf{I}_{bg}$ by masking:

$$\mathbf{I}_{fg} = \mathbf{I} \odot M, \quad \mathbf{I}_{bg} = \mathbf{I} \odot (1 - M). \qquad (8)$$

We combine region specific positive losses, a global positive loss, and negative losses:

$$\begin{aligned}\mathcal{L} = \lambda_{fg}\mathcal{L}_{pos}(\mathbf{I}_{fg}, \mathbf{p}_{fg}) + \lambda_{bg}\mathcal{L}_{pos}(\mathbf{I}_{bg}, \mathbf{p}_{bg}) \qquad (9)\\ + \lambda_{full}\mathcal{L}_{pos}(\mathbf{I}, \mathbf{t}) + \lambda_{neg}\Big(\mathcal{L}_{neg}(\mathbf{I}_{fg}, \mathbf{p}^-_{fg})\\ + \mathcal{L}_{neg}(\mathbf{I}_{bg}, \mathbf{p}^-_{bg})\Big). \qquad (10)\end{aligned}$$

Optionally, when the instruction specifies a concrete background color, we add a weak color prior on the mean background color. We also add a boundary penalty that discourages the foreground mask from collapsing onto the image borders, which reduces trivial solutions caused by cropping.

Our overall procedure alternates between (i) parsing and prompt decomposition, (ii) confiner based plan generation with priorities and bounds, and (iii) constrained gradient descent on the selected top K parameters for a fixed number of inner steps. After each inner step, we project parameters back to feasible ranges and repair numerical issues. This outer loop repeats until convergence or a fixed budget is reached, yielding a parameter estimate that matches the instruction while remaining stable under optimization.

4. Experiments

We evaluate our approach on one benchmark comparison and two ablation studies. The benchmark compares our full method against a naive CLIP-only optimization baseline to test whether structured parameter control and our guidance strategy improve text adherence and object visibility under the same differentiable renderer. The first ablation studies prompt specificity by contrasting concrete, renderer-supported attribute language with an abstract description for the same object, probing robustness under semantically underspecified instructions. The second ablation studies prompt length via incremental extension, examining whether adding constraints yields predictable and controllable changes in appearance rather than destabilizing optimization. Together, these configurations are designed to isolate three practical questions in unpaired text-to-appearance inverse rendering: whether our method outperforms a direct CLIP objective, whether it remains stable under ambiguous semantics, and whether it supports controllable refinement as more information is provided.

4.1. Experimental Setup

We optimize differentiable renderer parameters from a fixed initialization for each object. The parameter set includes camera pose and field-of-view, point and ambient illumination, material-like controls (diffuse tint, specular strength and color, roughness, shininess), fog parameters, and post-processing parameters (exposure, contrast, saturation, hue shift, gamma, vignette). All parameters are updated via gradient-based optimization through the renderer at 512 × 512 resolution for a fixed iteration budget.

- **Baseline (CLIP-only).** The baseline performs end-to-end optimization using a single global CLIP alignment loss between the full rendered image and the raw user prompt. It does not use region decomposition, negative constraints, or any external model guidance.
- **Ours.** Our method augments CLIP guidance with structured text supervision that separates foreground and background intent and introduces constraints that discourage attribute leakage across regions. In addition, we use a VLM-guided step to restrict the search to a small subset of parameters at each stage, while projecting parameters to valid ranges to improve numerical stability.

Our setting is an unpaired text-to-appearance optimization problem: for each mesh we are given only a text instruction and no reference image. As a result, pixel-based metrics (e.g., PSNR, SSIM) are not applicable. We also do not report other numerical metrics in this work, because they can be misleading for small-scale, open-ended appearance targets.

Instead, we adopt a tentative evaluation protocol based on a vision-language model (VLM) as a blind judge. For each trial, we render a single 2×3 comparison figure where the top row is *Ours* and the bottom row is *CLIP-only*. Each column corresponds to one object (Dolphin, Teapot, Tree), and the target prompt is shown under each column. We then ask the VLM to decide the overall winner across the three objects, focusing only on visible prompt adherence and object visibility.

We use the following evaluation prompt verbatim:

Unpaired Text-to-Appearance Evaluation Instructions

Role: You are an evaluator for an unpaired text-to-appearance task.

You will see ONE 2×3 **comparison figure:**
- **Top row** = Ours
- **Bottom row** = CLIP-only
- Columns are three objects: **Dolphin**, **Teapot**, **Tree**.
- Each column has its **target prompt** shown under the images.

Task: Decide which method is better overall across the 3 objects.

How to judge (only what is visible):
- **Prompt adherence:** color, material (glossy/matte), lighting, fog/no-fog, vignette, background
- **Object visibility:** object should be clearly visible (not too dark or washed out)
- **Attribute correctness** matters more than aesthetics

Output format (strict):

```
WINNER = OURS or CLIP-ONLY or TIE
REASON = one sentence summary
```

4.2. Benchmark Comparison: Ours vs CLIP-only

Figure 2 shows the 2×3 comparison across Dolphin, Teapot, and Tree, with each target prompt displayed under its corresponding column. Using the evaluation protocol above, the VLM returned:

WINNER OURS

REASON Ours matches the key prompt attributes much better overall (teal glossy dolphin on a light studio background and a blue-fog vignetted tree silhouette), while CLIP-only is mostly too dark/off-color despite slightly better teapot visibility.

Qualitative analysis. For **Dolphin**, our output achieves a clearly visible teal subject against a bright studio-like background, whereas the CLIP-only baseline collapses to a dark, low-contrast rendering that obscures the object. For **Tree**, our method produces a recognizable silhouette and better conveys the requested cool fog and vignette, while the CLIP-only baseline again yields an under-exposed result with weak subject–background separation. For **Teapot**, the judge preferred the CLIP-only baseline mainly due to higher visibility. This indicates that visibility and exposure can be a failure mode in similarity-driven optimization, and that additional constraints may occasionally over-restrict the search and lead to under-exposed outputs for dark materials.

4.3. Ablation: Concrete vs Abstract Instructions

Figure 3 compares two prompts that describe the same scene at different levels of specificity. The **concrete** prompt specifies directly controllable attributes (matte black, warm soft light, no fog), while the **abstract** prompt describes a high-level mood (quiet, cozy) without explicit appearance variables.

Observation. Although the abstract prompt is semantically underspecified and does not explicitly name renderer-controllable attributes, our inverse rendering procedure still produces an output that remains contextually consistent with the concrete instruction. As shown in Fig. 3, both prompts lead to a similar overall "studio teapot" interpretation under a dark background, with the main discrepancy primarily reflected in secondary appearance choices such as highlight emphasis, rim-like responses, and the relative scale of the teapot. This suggests that our method can preserve scene-level context and produce semantically aligned renderings even when the text instruction is vague, while concrete prompts mainly act as stronger constraints that reduce ambiguity in the final appearance.

4.4. Ablation: Prompt Length Extension

Figure 4 evaluates sensitivity to prompt length by progressively extending a base teapot instruction: (1) object and background description only, (2) plus explicit key light and rim light, (3) plus global tone controls (high contrast, slight vignette) and an explicit no fog constraint.

Observation. Fig. 4 shows that as we progressively extend the prompt with additional, renderer-controllable constraints, our method can reliably steer the same underlying scene toward more specific target conditions. Even the shortest instruction already yields a plausible solution (glowing green teapot on a magenta studio background), while the longer prompts refine the result in predictable directions by tightening lighting and post-processing requirements. In particular, adding explicit key and rim lighting encourages stronger, more structured specular responses, and further appending high-contrast and vignette constraints produces a more stylized global tone without breaking object-background separation. Overall, this ablation highlights the controllability of our renderer: prompt augmentation provides extra constraints that reduce ambiguity, and the optimization can incorporate these additional conditions to produce renderings that remain consistent with the intended appearance specification.

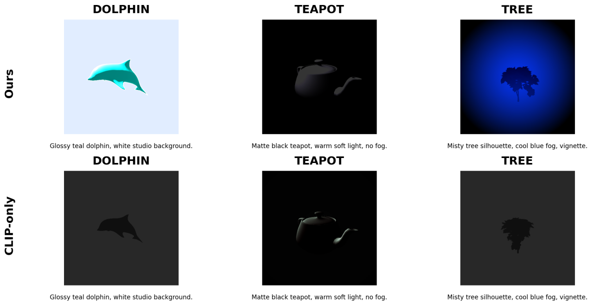
DOLPHIN
TEAPOT
TREE
Ours
Glossy teal dolphin, white studio background.
Matte black teapot, warm soft light, no fog.
Misty tree silhouette, cool blue fog, vignette.
DOLPHIN
TEAPOT
TREE
CLIP-only
Glossy teal dolphin, white studio background.
Matte black teapot, warm soft light, no fog.
Misty tree silhouette, cool blue fog, vignette.

Figure 2. Benchmark comparison. Top row: *Ours*. Bottom row: *CLIP-only*. Columns correspond to Dolphin, Teapot, and Tree, with the target prompt shown under each column.

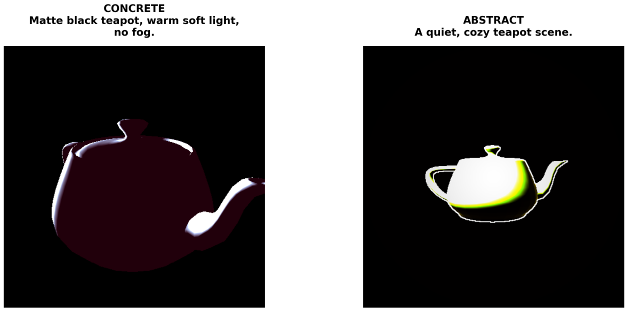
CONCRETE
Matte black teapot, warm soft light, no fog.
ABSTRACT
A quiet, cozy teapot scene.

Figure 3. Concrete vs abstract prompt ablation on the teapot. The concrete prompt specifies renderer-controllable attributes, while the abstract prompt describes a high-level mood.

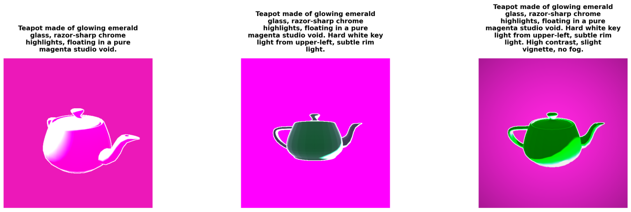
Teapot made of glowing emerald glass, razor-sharp chrome highlights, floating in a pure magenta studio void.
Teapot made of glowing emerald glass, razor-sharp chrome highlights, floating in a pure magenta studio void. Hard white key light from upper-left, subtle rim light.
Teapot made of glowing emerald glass, razor-sharp chrome highlights, floating in a pure magenta studio void. Hard white key light from upper-left, subtle rim light. High contrast, slight vignette, no fog.

Figure 4. Prompt length extension ablation on the teapot. We progressively extend the prompt with explicit lighting and global tone constraints.

5. Conclusion

We presented a closed-loop approach for unpaired text-to-appearance inverse rendering that optimizes a differentiable renderer's controllable parameters using vision-language guidance. The key idea is to reduce ambiguity and instability in CLIP-only optimization by introducing two forms of structured guidance: swap-safe prompt augmentation derived from foreground and background attribute parsing, and a parameter confiner that proposes an optimization plan with a top-K update set and tightened ranges. Across a benchmark comparison, our method achieves better overall prompt adherence than the naive CLIP baseline. In ablations, we find that the method produces contextually consistent results even when the text is abstract, indicating robustness to underspecified semantics. Moreover, extending the prompt with additional constraints yields predictable changes in appearance, demonstrating that the renderer remains controllable and can incorporate augmented information without collapsing optimization. Our current evaluation relies on pairwise visual judgments from a vision-language evaluator, which is suitable for an unpaired generative setting but remains imperfect. Future work includes broader object and material coverage, stronger multi-view consistency constraints, and more systematic human studies, as well as extending the framework to richer scenes and more physically grounded rendering models.

Author Contributions

- **Chi-Chang Lee -** co-proposed the LLM confiner module, co-developed the core pipeline that combines large language models with CLIP-based optimization, contributed advanced designs for integrating (visual) large language models with CLIP guidance as the confining strategy, and led the paper writing and overall presentation of the work.
- **Wenxuan Wu -** co-proposed the Vibe rendering framework, implemented the initial CLIP-only baseline, and co-developed the core pipeline that combines large language models with CLIP-based optimization.

References

[1] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 2

[2] Ayaan Haque, Matthew Tancik, Alexei A. Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. *arXiv preprint arXiv:2303.12789*, 2023. 1, 2

[3] Nasir Mohammad Khalid, Tianhao Xie, Eugene Belilovsky, and Tiberiu Popa. Clip-mesh: Generating textured meshes from text using pretrained image-text models. In *SIGGRAPH Asia 2022 Conference Papers (SA '22)*. ACM, 2022. 1, 2

[4] Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2

[5] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. *arXiv preprint arXiv:2211.10440*, 2022. 1, 2

[6] Gustavo Patow and Xavier Pueyo. A survey of inverse rendering problems. *Computer Graphics Forum*, 22(4):663–687, 2003. 1, 2

[7] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 1, 2

[8] Ravi Ramamoorthi and Pat Hanrahan. A signal-processing framework for inverse rendering. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*, pages 117–128. ACM, 2001. 1, 2

[9] Soumyadip Sengupta, Jinwei Gu, Kihwan Kim, Guilin Liu, David W. Jacobs, and Jan Kautz. Neural inverse rendering of an indoor scene from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 1, 2

[10] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *arXiv preprint arXiv:2305.16213*, 2023. 1, 2