arXiv:2601.17259v1 [cs.CV] 24 Jan 2026

†共同贡献

扩散采样中由推理时损失引导的颜色保留

Aarush Ram Anandh[1], † 和 Angad Singh Ahuja[1], †[1]
受约束图像合成实验室

摘要。 在文本到图像扩散系统中，精确的颜色控制仍是一个持续存在的失效问题，尤其是在设计导向的工作流程中，生成结果必须满足用户明确指定的颜色目标值。我们提出一种推理时的区域约束颜色保留方法，无需额外训练即可引导预训练扩散模型。该方法结合了：(i) 基于感兴趣区域的图像修复，以实现空间选择性；(ii) 重新施加背景潜变量，以防止感兴趣区域之外的颜色漂移；以及 (iii) 通过梯度引导对潜变量进行微调，所用复合损失定义于 CIE Lab 和线性 RGB 空间。该损失不仅用于控制感兴趣区域的均值颜色，也通过类 CVaR 惩罚和软最大值惩罚来控制逐像素误差分布的尾部，并采用延迟启动门控和随时间变化的调度方案，以稳定各去噪步骤中的引导。我们表明，仅关注均值的基线方法虽能满足平均颜色约束，却可能造成显著影响感知的局部失误，这也说明了采用分布感知目标的必要性。由此得到的方法提供了一种实用且无需训练的机制，可实现有针对性的颜色遵循，并可集成到标准 Stable Diffusion 图像修复管线中。

关键词： 扩散模型；Stable Diffusion；推理时引导；潜变量微调；可控生成；颜色保持；CIELab；感兴趣区域 (ROI)；图像修复；条件风险价值；分布感知损失；调色板与颜色约束。

引言与动机

扩散模型已成为高质量图像合成与编辑的标准基础模型，但其可控性仍主要依赖语义条件（文本）和结构条件（边缘、姿态、分割、图像修复掩码）。这些机制虽能有效控制构图与几何结构，却难以可靠地施加低层级数值属性约束，例如在指定感兴趣区域 (ROI) 内实现精确的颜色目标值。然而，在实际设计工作流中，颜色往往被指定为明确的约束（例如品牌调色板中的某种颜色、UI 主题色或包装色样），要求并非仅仅是让输出“看起来大致是蓝色”，而是在可接受的容差范围内匹配给定的 sRGB 或感知颜色编码，同时避免图像其他区域发生意外偏移。

本文将推理时的颜色保持视为一个受约束采样问题。我们将扩散采样轨迹视为控制面，并在去噪过程中直接向潜变量状态施加轻量级引导信号。我们的损失设计结合了两种互补表示：一项随时间衰减的线性 RGB 项，使去噪早期偏向目标色彩方向；以及一项逐渐增强、对尾部敏感的 Lab 空间项，在结构逐渐成形时惩罚持续存在的离群值。由此，我们得到一种模块化的推理时控制机制，无需重新训练，也不会引入新参数，并且可以直接接入现有的 Stable Diffusion 图像修复管线。

核心动机是在计算资源和集成约束下优先考虑可控性。许多旨在提升约束条件下视觉保真度的方法（例如增强纹理真实感、保留高频细节，或学习区域特定外观先验）都需要额外训练、辅助网络，或反复进行内循环优化，从而显著增加推理成本和系统复杂度。相比之下，我们的目标更为聚焦：我们希望

在尽量不扰动基础模型生成先验的前提下，实现可预测且局限于感兴趣区域的颜色一致性。因此，所提出的方法不采用显式的纹理或图像块级外观匹配损失。该方法旨在为颜色关键编辑任务提供一种轻量、经过优化的基线，并作为基础，以便在额外复杂度确有必要时，进一步叠加更丰富的外观约束（包括纹理感知或印刷感知目标）。

文献综述

扩散模型从根本上将图像生成确立为一个逐步加噪与去噪的过程。Ho 等人 [8] 提出了去噪扩散概率模型，通过学习逆转数据逐渐经历的高斯加噪过程，生成高保真图像。他们采用的公式侧重于优化一个变分上界，该上界与分数匹配相关，形式如下 -

$$q(\mathbf{x}_t \mid \mathbf{x}_0) \;=\; \mathcal{N}\Big(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0,\, (1-\bar{\alpha}_t)\mathbf{I}\Big) \quad\Longleftrightarrow\quad \mathbf{x}_t \;=\; \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon},\;\; \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{1}$$

$$\mathcal{L}_{\text{simple}}(\theta) \;=\; \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\Big[\big\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\big\|_2^2\Big]. \tag{2}$$

Nichol 和 Dhariwal [14] 通过学习噪声方差并调整扩散调度，显著改进了这一范式，使采样速度提升了多个数量级，而质量损失几乎可以忽略不计。与此同时，Song 等人 [20] 提出了 DDIM，它沿用了 DDPM 的训练方法，但采用了非马尔可夫采样器，因此能够以确定性方式生成样本，或通过减少采样步数来生成样本，而性能仅略有下降。Song 等人 [21] 最终通过将扩散过程构建为随机微分方程，统一了这一理论基础：数据会持续加噪至先验分布，而学习得到的、随时间变化的分数函数则定义了反向随机微分方程，用于按以下公式采样新数据。

$$d\mathbf{x} \;=\; \mathbf{f}(\mathbf{x}, t)\,dt \;+\; g(t)\,d\mathbf{w}_t, \tag{3}$$

$$d\mathbf{x} \;=\; \Big[\mathbf{f}(\mathbf{x}, t) \;-\; g(t)^2\,\nabla_{\mathbf{x}} \log p_t(\mathbf{x})\Big]\,dt \;+\; g(t)\,d\bar{\mathbf{w}}_t. \tag{4}$$

这使得新的采样过程和测试时约束成为可能。就我们的研究而言，它展示了如何向生成随机微分方程中注入额外的能量项或引导项，以引导输出。这些进展表明，扩散采样过程本身提供了调节生成的接口，而这正是我们打算利用的。

早期的可控扩散研究引入了引导策略，在采样过程中加入显式梯度。Dhariwal 和 Nichol [7] 首先证明，扩散模型在图像质量和多样性方面能够超越生成对抗网络，并提出了分类器引导，以此在多样性和保真度之间进行权衡。分类器引导展示了测试时引导这一原理：无需重新训练生成模型，就能将任意可微目标注入生成动态中。Ho 和 Salimans [9] 随后将这一方法推广为无分类器引导（CFG）。CFG 因为简单有效，成为现代文本到图像扩散中的事实标准（如 Stable Diffusion、GLIDE [15], 等所采用的方式）：与无引导样本相比，它能更忠实地呈现提示词中的细节。然而，CFG 的控制是隐式且全局的，这意味着它无法保证遵循特定属性（例如精确的颜色值），而这些属性

往往仍是模型“涌现”出来的行为。因此，虽然我们大致可以要求生成“一辆蓝色汽车”，并得到一辆带蓝色调的汽车，却无法精确限定其色调，也无法阻止模型引入调色板之外的色调。这一局限促使人们开发更具结构性的控制机制，而这类机制也是工业和设计应用场景所必需的。

研究人员引入了更强的条件信息流和架构改进，以控制扩散模型的输出。一个例子是潜在扩散模型（LDM） [19]，由 Rombach 等人提出；该模型将扩散过程转移到由自编码器学习得到的低维潜在空间中。

$$\mathbf{z} = E(\mathbf{x}), \quad \mathbf{z}_t \;=\; \sqrt{\bar{\alpha}_t}\,\mathbf{z} \;+\; \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}, \;\; \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \mathcal{L}_{\mathrm{LDM}}(\theta) \;=\; \mathbb{E}_{t,\mathbf{z},\boldsymbol{\epsilon}}\Big[\big\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t, \mathbf{c})\big\|_2^2\Big]. \tag{5}$$

Rombach 等人 [19] 为扩散模型配备了交叉注意力机制，以便基于文本嵌入进行条件控制；这些文本嵌入通常来自 CLIP [16] 或 BERT[6]等预训练语言—图像模型。然而，由于 Radford 等人的 CLIP 模型侧重于整体语义对齐 [16]，某些低层属性（例如精确的色调或准确的几何形状）无法仅通过文本得到可靠或直接的指定 [3]。实际上，当前的文生图模型往往难以处理指定罕见或非常具体颜色的提示词（例如“一个带有潘通 448 C 色肩带的黄绿色手提包”） [3]。提示词中的颜色词可以引导总体色相，但模型可能会偏向更常见的颜色，或引入非预期的颜色混合，这是因为语言条件信息经过学习，旨在满足人类可描述的语义，而不是施加像素级约束。因此，纯文本颜色控制充其量只能是间接的。近期评估证实了这一点：在一项综合基准测试 GenColorBench（Butt 等人 2025年） [3] 中，包含覆盖 400 多种命名颜色或数值颜色的 4.4万条颜色相关提示，最先进的扩散模型在颜色准确性和一致性方面都表现出显著误差。

因此，研究人员针对其他图像属性，特别是结构或空间控制，开发了专门的条件机制，这些机制可以为颜色控制提供启发。例如，ControlNet（Zhang 等人） [24]通过添加一个可训练分支来扩展冻结的扩散模型，该分支接收额外的条件图像（如边缘图、人体姿态骨架、分割图等）。该分支通过零卷积层连接到主模型，并经过训练，以根据条件图像调节去噪过程。值得注意的是，基础模型的权重保持不变（从而避免灾难性遗忘），并且可以组合多个条件。关于结构和空间控制的研究凸显出一种模式：添加有针对性的条件路径可以显著提升可控性。颜色控制也可以从这一角度来看待：与其依赖文本编码器的弥散影响，不如设想一个专用于颜色属性的条件通道（例如调色板或目标直方图），以更可靠地引导生成。事实上，Pu 等人的一篇最新综述沿着这一思路对可控生成方法进行了分类 [4]，指出当前系统对空间和结构条件的处理能力远胜于对精确颜色或数值条件的处理能力。

扩散模型中的显式颜色控制直到最近才受到重点关注，目前仍有诸多局限。一类方法通过给定的调色板或颜色分布来对生成过程进行条件控制。Vavilala 等人（2023年） [22] 率先提出了基础的调色板条件控制方法，证明只需将调色板信息拼接到模型输入中，标准扩散模型便能大致遵循所提供的调色板，但控制效果较为粗略。Aharoni 等人则提出了更为系统的改进方案：采用调色板适配器模块 [2]实现调色板对齐扩散。在他们的方法中，用户提供一组参考颜色，模型则受到引导，在生成结果中使用

且仅使用这些颜色。该方法将调色板视为目标颜色直方图，并引入两个标量参数来控制模型遵循该直方图的严格程度。通过在包含已知颜色分布的大型图像数据集上训练，适配器学会将扩散生成与调色板“对齐”，其颜色保真度显著优于简单的基于提示词的控制。这有力证明，只要采用适当的条件机制，扩散模型就能内化严格的颜色约束。Agarwal 等人提出的另一项相关工作探索了一种免训练的颜色与风格解耦技术 [1]。他们没有对模型进行微调，而是在推理时调整模型的潜特征，使特征协方差与参考图像的颜色统计相匹配，并操控亮度通道与色度通道上的自注意力，以分离风格与颜色。值得注意的是，这一方法无需额外训练，因为它是针对每张图像进行的优化。这凸显了测试时优化在颜色控制方面的潜力：只要能计算某种可微的颜色相似度指标，就可以直接将其融入扩散过程。总体而言，这些工作表明，直接控制颜色是可行的，但要么需要额外训练（调色板适配器），要么只能较弱地约束颜色（事后调整潜变量）。关键在于，目前还没有方法将颜色约束与空间选择性结合起来（确保正确的区域呈现正确的颜色）。通常，这些方法会影响整幅图像的调色板。我们的方法旨在填补这一空缺，在测试时实现基于区域的颜色约束。

在设计我们的解决方案时，我们从扩散模型受约束采样的成功经验中汲取灵感，这些经验已在扩散领域关于逆问题的文献中确立。一个核心思想是，在每个采样步骤中引入一个额外项，“引导”样本满足某些约束，从而实时执行拉格朗日优化。Chung 等人的扩散后验采样（DPS） [5] 是一种求解含噪逆问题的方法，他们在每次扩散迭代中加入如下所示的梯度——

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t \mid \mathbf{y}) \;\approx\; \mathbf{s}_\theta(\mathbf{x}_t, t) \;+\; \nabla_{\mathbf{x}_t} \log p\Big(\mathbf{y} \mid \hat{\mathbf{x}}_0(\mathbf{x}_t)\Big), \tag{6}$$

$$\hat{\mathbf{x}}_0(\mathbf{x}_t) \;=\; \frac{1}{\sqrt{\bar{\alpha}_t}}\Big(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\Big). \tag{7}$$

这会逐步将去噪轨迹引导至与观测结果一致的图像子集。重要的是，DPS 无需重新训练扩散模型；它作为测试时封装器运行。由此产生的采样器将原始生成分数与“流形约束”项相结合，相较于无引导扩散，能够得到更准确且更一致的解 [5]。后续工作如 DDRM（Kawar 等人 2022年） [12] 和 Wang 等人的（DDNM） [23] 则通过刻画测量的零空间，并确保扩散更新只影响不会产生矛盾的图像部分，进一步改进了这一方法。更广泛地说，在扩散采样过程中加入自定义能量或损失项，是施加约束的一种强大且具有理论依据的方式 [5]。近期综述也指出，当需要新的条件时，测试时条件控制如今已成为避免昂贵模型重新训练的标准方法 [4]。在我们的案例中，“测量”是颜色约束的集合（例如感兴趣区域中的目标直方图或目标 Lab 值），我们也以类似方式将其惩罚梯度纳入扩散更新。这样做使我们获得了一项关键优势：无需微调模型权重即可实现颜色约束——我们只调整采样路径，从而避免生成先验性能退化的风险，也无需收集大量特定颜色示例的训练数据集。这一策略与 Lobashev 等人的近期工作相似；他们提出使用切片 Wasserstein（SW）引导来实现颜色分布对齐 [13]。他们定义了生成图像的颜色分布与参考分布之间的可微损失（使用切片 Wasserstein

距离　基于 Lab 直方图），并在下文指定的每一步进行反向传播——

$$\mathcal{L}_{\mathrm{SW}}(\mathbf{x}) \;=\; \mathrm{SW}_p\Big(\mu(\phi(\mathbf{x})),\, \mu(\phi(\mathbf{x}^{\mathrm{ref}}))\Big), \qquad \mathrm{SW}_p(\mu,\nu) \;=\; \int_{\mathbb{S}^{d-1}} W_p\Big((\mathcal{P}_{\boldsymbol{\theta}})_{\#}\mu,\; (\mathcal{P}_{\boldsymbol{\theta}})_{\#}\nu\Big)\, d\boldsymbol{\theta}. \tag{8}$$

该方法应用于文本到图像扩散后，生成的图像在整体颜色使用上与参考调色板高度匹配，同时仍然遵循文本提示的语义。关键在于，无需进行模型训练。引导完全在采样时运行。我们的方法可以视为对这一思想的扩展，但具备更精细的控制能力（支持特定区域，并使用多个损失项来平衡保真度与质量）。

推理时损失的颜色转换

为计算本文所述的推理时损失，我们需要在多种颜色编码之间进行一致且可微的转换。我们的复合目标函数是若干子损失的加权和，这些子损失分别定义在 (i) CIE $L^*a^*b^*$ (Lab) 空间和 (ii) 线性光 RGB 空间中。用户提供单个目标颜色，形式为归一化的 sRGB 三元组 $[0,1]^3$, ，记作 $\mathrm{sRGB}_{\mathrm{user}}$。由于 sRGB 经过伽马压缩，因此与线性 RGB 并不等价；因此，当子损失定义在线性 RGB 中时，我们首先应用标准的 sRGB 逆传递函数，以获得 $\mathrm{LRGB}_{\mathrm{user}}$ （线性光 RGB）。此外，我们还使用与 CIE 1976 Lab 定义一致的可微 sRGB→Lab 转换，将用户提供的颜色转换为 Lab，并记作 $\mathrm{Lab}_{\mathrm{user}}$。

在推理时，我们的损失函数作用于扩散采样器生成的不断演变的样本。在每个时间步 t, 我们将当前潜变量解码为模型 RGB 输出空间中的图像，并将其限制到 $[0,1]$ ，以获得归一化的 sRGB 图像 sRGB_t。基于此，我们计算：(a) 通过 sRGB 线性化得到相应的线性光 RGB 图像 LRGB_t ；(b) 通过标准 Lab 转换得到相应的 Lab图像 Lab_t 。因此，我们维护损失项所用的四个主要量：(i) $\mathrm{LRGB}_{\mathrm{user}}$；(ii) $\mathrm{Lab}_{\mathrm{user}}$；(iii) LRGB_t；(iv) Lab_t。基于 Lab 的损失定义为 Lab_t 与 $\mathrm{Lab}_{\mathrm{user}}$ 之间的差异（通常在指定的感兴趣区域内计算），而基于线性 RGB 的损失则定义为 LRGB_t 与 $\mathrm{LRGB}_{\mathrm{user}}$之间的差异。所有转换均通过可微算子实现，以便在采样过程中将损失信号反向传播至潜变量 [10, 11, 17]。

转换流程

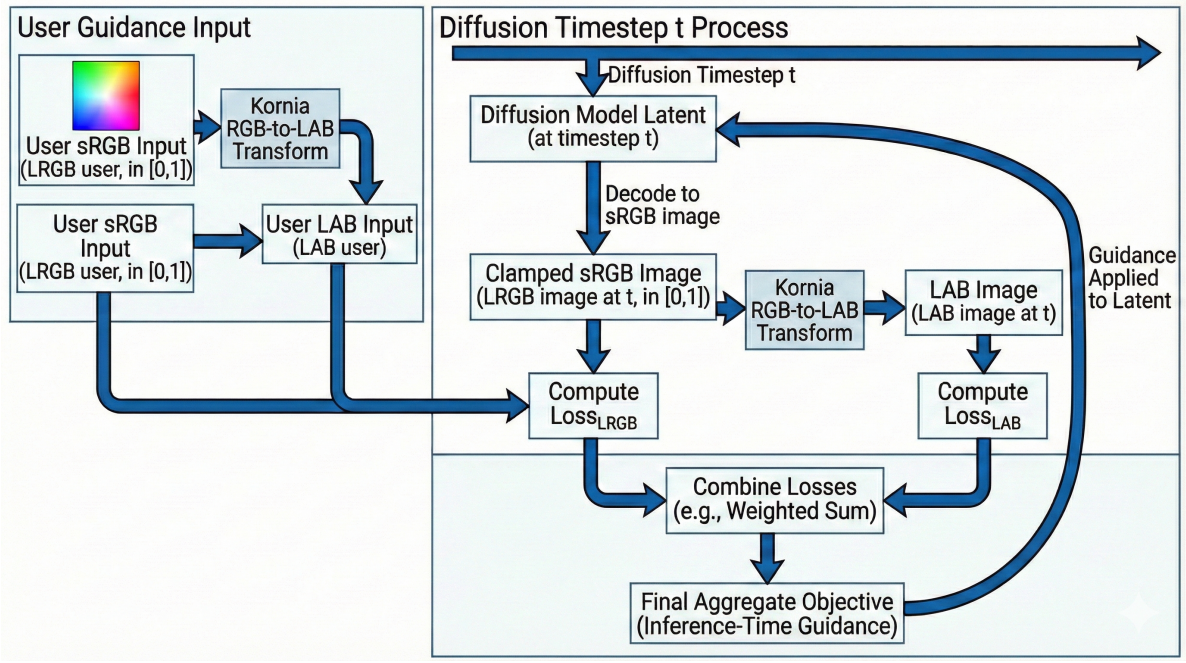


图 1：**转换流程**。 可微颜色转换与损失计算流程概览。

损失函数和图像修复流程

我们首先假设存在一个在推理时应用的单一主损失函数。该损失函数以用户指定目标值、时间步t的模型输出，以及少量权重和超参数作为输入。在扩散过程中，每一步都会生成一个潜变量表示。在每个时间步 t，我们解码潜变量并执行前文所述的颜色空间转换，随后计算时间步 t处潜变量的主损失函数 L_t 。

为施加并应用这些损失，我们采用一种在此称为潜变量微调的推理时引导策略。该方法可以理解为在每个时间步，通过根据损失对潜变量执行梯度步来施加校正性推动。由于损失被用作引导信号，以局部调整去噪轨迹，因此理想情况下，有效损失信号不应在连续时间步之间发生过于 abrupt 的变化。微调更新如下：

$$z_t \leftarrow z_t - \eta \nabla_{z_t} L_t, \tag{9}$$

其中， η 为固定步长。更新 z_t后，我们以更新后的潜变量 z_t 为参考，继续标准扩散过程，在 $t{+}1$ 生成潜变量（或者根据所选调度器使用下一个索引）。我们谨慎选择 η ，以避免其导致扩散过程不稳定。这种通过梯度注入实现的测试时引导，与既有的扩散引导和后验采样方法一致；这些方法表明，可微目标函数能够在采样过程中被纳入，而无需重新训练生成模型 [5, 7]。

对于本文所述的实现，我们使用 Stable Diffusion v1.5，因为它原生支持在潜在空间中进行图像修复 [19]。首先，我们指定一个感兴趣区域（ROI）

，并希望在该区域内强制使用目标颜色。图像修复需要一张初始图像，因此我们使用 matplotlib 生成一张简单的画布图像，并为其设置所需的背景颜色。随后，我们将目标掩码内的像素替换为零，从而生成一张带有“空洞”的图像。接着，将该图像输入扩散图像修复流水线，模型会尝试仅填充掩码区域。实际运行时，模型在每个步骤都会接收完整图像（包括保留的背景），这意味着完整图像都会经历加噪与去噪过程：模型在根据提示词对目标区域进行去噪的同时，也会将周围像素去噪，使其恢复为原始背景。实践中，这会导致掩码外的原始颜色发生不理想的漂移。我们还观察到，由于完整图像都会被去噪，背景颜色会影响目标区域内的生成结果。为抑制背景漂移，我们在每个步骤中都通过覆盖非掩码区域，将保留区域的潜变量重新设为对应于所需背景颜色的潜变量。

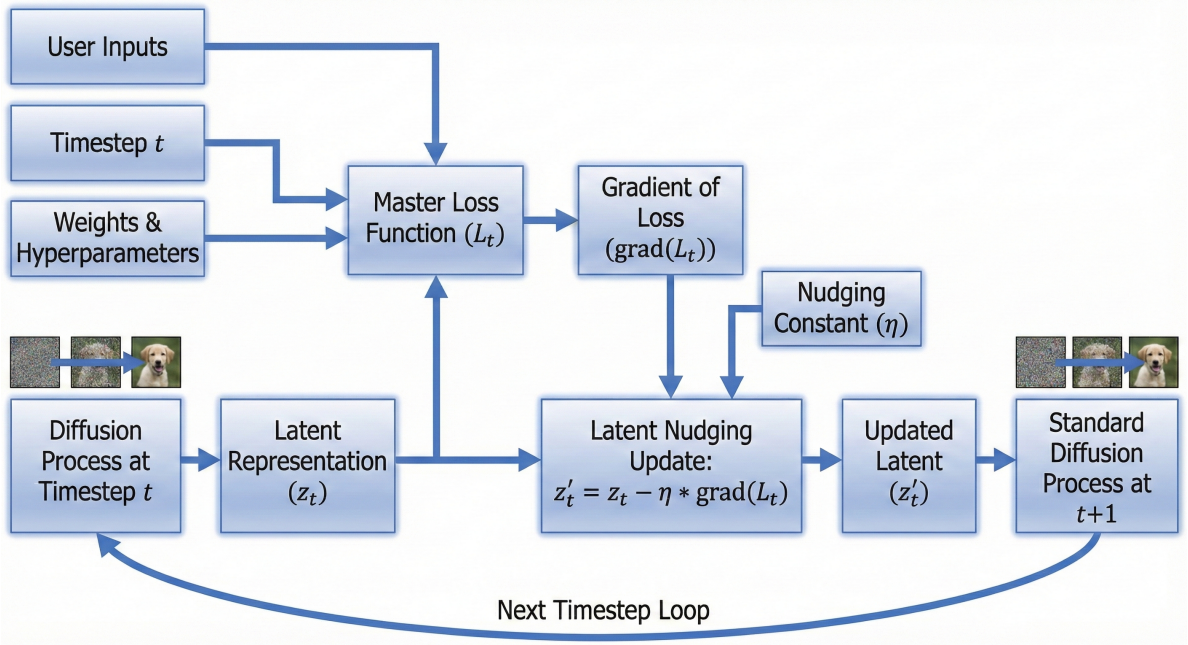
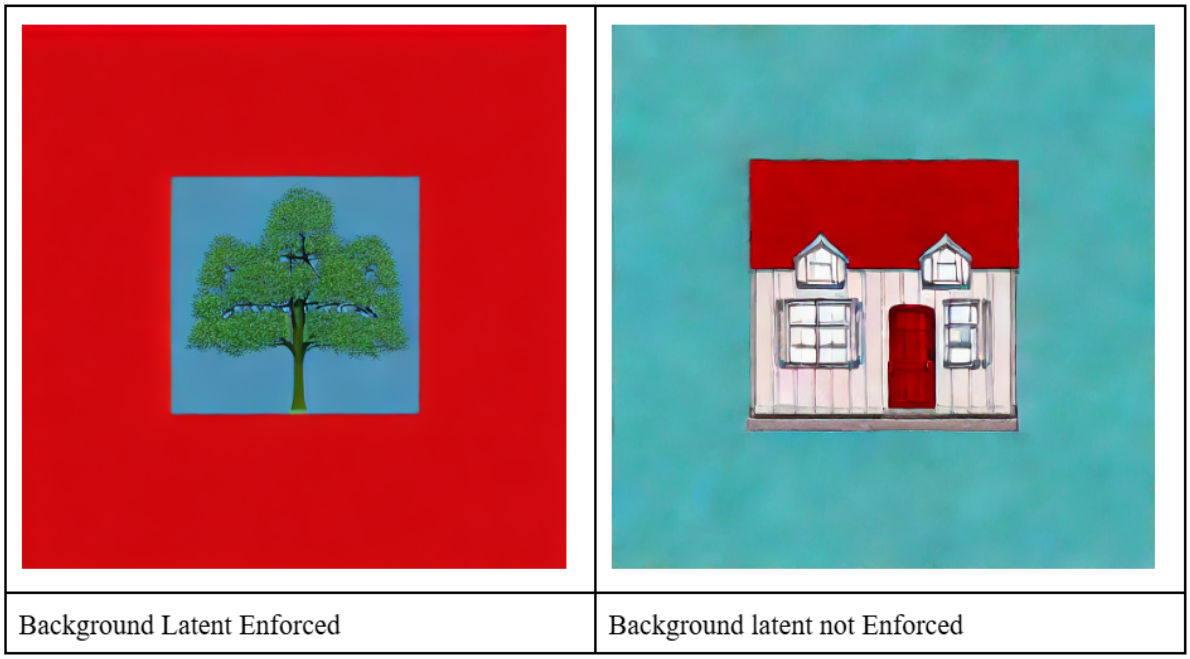


图 2：**损失函数流程**。展示推理时优化过程和数据流的示意图。



Background Latent Enforced | Background latent not Enforced

图3：**背景一致性**。请注意，潜变量得到强化时（左侧）背景颜色均匀一致；未得到强化时（右侧），背景则呈现纹理且不一致。

在介绍我们的最终目标之前，我们评估了两种简单的平均颜色损失，作为基线。这些损失刻意设计得十分精简：它们为测试 (i) 推理时潜变量微调和 (ii) 图像修复调度器提供了一个易于理解的起点，但并不试图控制掩码区域内颜色的完整分布。

算法1计算掩码区域内所有有效像素的线性 RGB 颜色均值，并惩罚其与目标颜色之间的欧氏距离平方偏差。从功能上看，该损失约束了平均颜色，因此能够有效纠正明显的色相偏差。然而，我们没有将其用作主要目标，因为它的约束不足：许多视觉上截然不同的颜色场具有相同的均值。实际上，这会导致一些失效模式，例如生成结果趋于模糊色调，或少量像素变得高度饱和（“尖峰”），而均值仍接近目标。此外，作为纯粹基于均值的约束，它无法保留细粒度细节（例如，确保叶片与树皮呈现正确的色调），因为它不会促使区域内实现像素级一致性。

算法1 线性 RGB 约束损失$(I, M, c^\star)$

输入： 线性 RGB 图像 $I \in \mathbb{R}^{3\times H\times W}$、掩码 $M \in \{0,1\}^{H\times W}$、目标 $c^\star \in \mathbb{R}^3$
定义有效像素：

$$\Omega = \{(h,w) \mid M_{h,w} = 1,\ I_{:,h,w} \text{ finite}\}$$

如果 $|\Omega| = 0$ **则返回** 0
结束如果
计算均值颜色：

$$\mu = \frac{1}{|\Omega|} \sum_{(h,w)\in\Omega} I_{:,h,w}$$

返回 $\|\mu - c^\star\|_2^2$

算法 2 延续了上述方法，但在 CIE Lab 色彩空间中运行，并测量平均Lab颜色与目标之间的逐通道欧几里得偏差。出于感知方面的原因（Lab 近似均匀），这种方法优于 RGB；根据经验，与仅使用 RGB 的惩罚相比，它还能减少一些色相偏移伪影。然而，出于相同的结构性原因，我们没有将该损失选为主要驱动因素：它仍然是一种均值约束，因此无法防止异质解，即部分像素远离目标，而均值却是正确的。换言之，它无法充分控制掩码内颜色误差分布的尾部，而视觉显著性失效恰恰集中在这一部分。

算法 2 Lab 欧几里得损失($I_{\text{lab}}, M, c^{\star}_{\text{lab}}$)

输入： $I_{\text{lab}} \in \mathbb{R}^{3\times H\times W}$, $M \in \{0,1\}^{1\times H\times W}$, $c^{\star}_{\text{lab}} \in \mathbb{R}^3$

将 $I_{\text{lab}}, M, c^{\star}_{\text{lab}}$ 转换为单精度浮点数将 $c^{\star}_{\text{lab}} \to$ 重塑为 (1,3, 1, 1)计算像素数量：

$$n \leftarrow \sum_{i,j} M_{:,i,j}$$

如果 $n = 0$ **，则返回** 0

结束 if掩码 LAB 值：

$$I_{\Omega} \leftarrow I_{\text{lab}} \odot M$$

计算平均 LAB 颜色：

$$\mu \leftarrow \frac{\sum_{i,j} I_{\Omega,:,i,j}}{n+\epsilon}$$

与目标的差异：

$$d \leftarrow \mu - c^{\star}_{\text{lab}}$$

逐通道欧氏距离：

$$\text{dist} \leftarrow \sqrt{\sum_{c=1}^{3} d_c^2 + \delta}$$

返回均值(dist)

上述均值颜色基线可用作诊断，但无法解决面向设计的生成中的核心失效模式：模型可能满足均值约束，却仍生成局部颜色区域不正确的结果（例如，几片叶子偏成了错误的绿色，或出现少量视觉上十分醒目的高饱和色块）。为防止模型收敛到模糊色调并保留语义伪影，我们在目标区域采用逐像素方法。其指导原则是：如果某个特定物体区域需要特定色调，那么损失就应当 (i) 在像素级作用于掩码，并且 (ii) 显式惩罚掩码内的平均偏差和最坏情况偏差。

我们的最终损失由多个惩罚项组合而成，这些惩罚项源自 Lab 空间中的逐像素距离场。该构造具有模块化特点：算法 3 定义了一个逐像素距离 u_t，用于衡量其与目标颜色之间的距离，该距离通过在“安全”的替代距离和更直接的感知平方距离之间插值获得。算法 4 随后生成一组互补的惩罚项（均值约束、逐像素铰链损失、尾部风险、软最大值和方差）。算法 5 将这些项组合成单个标量 L_t ，供推理时潜变量微调用。

我们引入了一种简单的门控调度 $g(t)$，其随去噪进度增加。直观而言，早期时间步主要受噪声影响，解码图像在语义上尚不稳定；此时过于激进的感知惩罚可能会引入高方差梯度。因此，我们的门控采用延迟启动：在达到指定的进度阈值之前，引导较弱或不存在；此后则平滑增强。在实际操作中，这一选择反映了经验观察：当样本进入具有语义意义的阶段后，施加严格的感知约束最为有效。

最后，我们强调，复合损失经过刻意设计，能够感知分布。除了控制均值偏差外，它还会惩罚：(i) 典型的像素偏差，通过 L_{pix}实现；(ii) 最差情况下的偏差，通过尾部聚合项（条件风险价值，CVaR）和平滑最大值（对数求和指数）实现；(iii) 掩码内离散度，通过方差约束实现。选择这种组合，是为了直接应对仅均值目标中观察到的失效模式：少数像素与目标值偏差很大，但均值仍在可接受范围内。加入 CVaR 组件，正是为了将优化压力集中在表现最差的一部分像素（尾部风险）上，而不是对所有像素一视同仁 [18]。

算法 3 定义了两种距离：通道加权代理量 q_{safe} 和直接平方距离 q_{00}。该代理量对亮度和色度赋予不同权重（通过 w_L 和 w_{ab}），以减少推理早期引导的不稳定性；而直接距离则会在采样后期提高感知上的严格程度。门控 g 在这两种距离之间进行插值，得到 q_t；开平方后的场 u_t 则作为稳健且可微的逐像素误差幅度。算法 4 根据 u_t 和 Lab 值计算五种惩罚项：均值约束确保平均颜色处于容差范围内；逐像素铰链损失抑制广泛存在的逐像素误差；CVaR 尾部项针对最大误差（前 $(1-\alpha)$ 比例）；基于对数求和指数的平滑最大值在避免不可微性的同时近似最大值惩罚；方差项则抑制掩码内异质的“混合”解。算法 5 将这些组件结合起来，根据去噪进度应用延迟启动门控，并返回加权和 L_t，作为推理时引导信号。

算法 3 距离场(ℓ, μ, g)

输入： Lab图像 $\ell \in \mathbb{R}^{3\times H\times W}$，目标 $\mu \in \mathbb{R}^3$，门控 $g \in [0,1]$

定义通道加权代理量：

$$q_{\text{safe}}(\ell,\mu) = w_L(\ell_L - \mu_L)^2 + w_{ab}\|\ell_{ab} - \mu_{ab}\|_2^2$$

定义感知平方距离：

$$q_{00}(\ell,\mu) = \|\ell - \mu\|_2^2$$

门控插值：

$$q_t = (1-g)\, q_{\text{safe}} + g\, q_{00}$$

返回距离场

$$u_t = \sqrt{q_t + \epsilon}$$

算法 4 ROI损失项$(u,\ \ell,\ \mu,\ M)$

输入： 距离场 u，Lab图像 ℓ，目标 μ，掩码 M 感兴趣区域集合 $\Omega = \{i \mid M_i = 1\}$，$\phi_p(r) = \max(0,\ r)^p$ 平均Lab颜色：

$$\hat{\mu} = \frac{1}{|\Omega|} \sum_{i \in \Omega} \ell_i$$

均值约束：

$$L_{\text{mean}} = \phi_p\big(\|\hat{\mu} - \mu\|_2 - \tau_{\text{mean}}\big)$$

逐像素铰链惩罚：

$$L_{\text{pix}} = \frac{1}{|\Omega|} \sum_{i \in \Omega} \phi_p(u_i - \tau_{\text{pix}})$$

尾部风险（条件风险价值）：

$$\text{CVaR}_\alpha(u) = \frac{1}{k} \sum_{u_i \in \text{Top}(1-\alpha)} u_i$$

$$L_{\text{tail}} = \phi_p\big(\text{CVaR}_\alpha(u) - \tau_{\text{tail}}\big)$$

软最大值（对数和指数）：

$$\text{LSE}_\beta(u) = \frac{1}{\beta} \log\left(\tfrac{1}{|\Omega|} \sum_{i \in \Omega} e^{\beta u_i}\right)$$

$$L_{\max} = \phi_p\big(\text{LSE}_\beta(u) - \tau_{\max}\big)$$

方差约束：

$$\text{Var}(u) = \frac{1}{|\Omega|} \sum_{i \in \Omega} (u_i - \bar{u})^2$$

$$L_{\text{var}} = \phi_p\big(\text{Var}(u) - \tau_{\text{var}}\big)$$

返回 $\{L_{\text{mean}}, L_{\text{pix}}, L_{\text{tail}}, L_{\max}, L_{\text{var}}\}$

算法 5 总损失(x_t, t)

1: 解码并转换为 Lab： $\ell \leftarrow$Lab(VAE(x_t))2: 计算
去噪进度：

$$\mathrm{prog}(t) = \frac{T - t}{T - t_{\min}}$$

3: 延迟启动门控：

$$g(t) = \begin{cases} 0, & \mathrm{prog}(t) \le s \\ \dfrac{\mathrm{prog}(t) - s}{1 - s}, & \mathrm{prog}(t) > s \end{cases}$$

4: 距离场：

$$u_t \leftarrow \texttt{DistanceField}(\ell, \mu, g(t))$$

5: 感兴趣区域惩罚项：

$$(L_{\mathrm{mean}}, L_{\mathrm{pix}}, L_{\mathrm{tail}}, L_{\max}, L_{\mathrm{var}}) \leftarrow \texttt{ROILossTerms}(u_t, \ell, \mu, M)$$

6: 总损失：

$$L_t = \lambda_{\mathrm{mean}} L_{\mathrm{mean}} + \lambda_{\mathrm{pix}} L_{\mathrm{pix}} + \lambda_{\mathrm{tail}} L_{\mathrm{tail}} + \lambda_{\max} L_{\max} + \lambda_{\mathrm{var}} L_{\mathrm{var}}$$

7: **返回** L_t

我们采用较晚的起始时机，因为初始潜变量噪声很大，可能导致误差非常大，并对潜变量进行不必要的调整。我们发现，虽然这一损失确实能将目标颜色约束为均值，但无法实现我们期望的输出。从视觉上看，这一结果很明显：图像中的小块区域会变成饱和的深蓝色。我们还观察到，若在提示词中提及目标颜色，同时只使用这一损失函数，采样过程可能会变得不稳定，导致损失值激增的失败案例。

从损失值和输出的特征来看，我们推测这可能是因为采样后期的损失变得过强。为验证这一点，我们将损失项乘以 $1/t$来衰减损失，其中 t 是时间步索引。

我们观察到，输出中的颜色约束质量显著提升。与衰减相一致，记录的损失轨迹呈现出更平滑的下降趋势。尽管这些记录引发了一个疑问：优化是否集中在采样的早期阶段，但我们强调，提示词完全没有提及期望的颜色。这一约束完全由推理时损失引入。

局限性

由于逐步解码、可微颜色空间转换以及通过损失进行反向传播以获取引导梯度，我们的方法会增加额外的推理时计算；这种开销会随时间步数量和感兴趣区域尺寸而增加，在高分辨率下可能使交互式使用难以实现。性能对掩码质量十分敏感：不准确的感兴趣区域边界、柔和的边缘或分割泄漏，可能导致颜色约束扩散到非预期区域，或在真实边界处约束不足；当物体包含精细结构时，这一问题尤为突出。该方法依赖引导过程中所用 RGB 到 Lab 转换的保真度和可微性；数值近似、色域裁剪或实现差异可能导致梯度偏移，并产生系统性的颜色误差。尽管复合目标函数

提高了逐像素遵循度，但更强的约束可能会牺牲纹理真实感和局部细节，导致色度变平、出现色带或产生不自然的过渡，尤其是在目标颜色与提示词驱动的语义不一致时。该方法无法正式保证达到全局最优解或满足约束：引导强度、延迟启动阈值和调度方案选择仍取决于超参数，在复杂提示词、高纹理物体、极端目标颜色或提示词与颜色强烈冲突的情况下可能失效。

未来工作

一个自然的延伸方向是通过边界感知正则化、阿尔法遮罩或显式边缘一致性项来改进ROI边界处理，在保留精细结构的同时减少颜色渗漏。我们还计划用能够响应去噪进度和损失曲率的自适应调度方案取代固定调度方案（例如，仅在ROI误差趋于平稳时增强引导，并在梯度导致采样不稳定时减弱引导）。另一个方向是从小型校准集中学习损失权重或阈值，从而自动调整 λ值和容差，以便针对不同的提示词类别和目标颜色，在贴合度与真实感之间取得平衡。更广泛地说，采用更丰富的颜色外观模型（超越Lab）、稳健处理色域约束，以及使用光照感知目标函数，有望减少因明暗变化和高光导致感知颜色偏移的失败案例。对于包装和工业设计的应用场景，我们计划整合印刷感知约束，例如墨量限制逻辑、基于 ΔE的参考色样容差，以及工艺先验（如基材和分色约束），使引导生成结果不仅在屏幕上视觉一致，也能满足下游再现流程的实际要求。

参考文献

[1] Aishwarya Agarwal、Srikrishna Karanam 和 Balaji Vasan Srinivasan。面向受限文生图合成的免训练颜色风格解耦，2024年。[2] Ido Aharoni、Omer Levi、Shai Hassid 和 Tomer Michaeli。调色板对齐的图像扩散，2025年。[3] Muhammad Atif Butt、Alexandra Gomez-Villa、Tao Wu、Javier Vazquez-Corral、Joost Van De Weijer 和 Kai Wang。GenColorBench：提升文生图生成中的颜色保真度，2025年。[4] Pu Cao、Feng Zhou、Qing Song 和 Lu Yang。基于文生图扩散模型的可控生成：综述，2024年。[5] Hyungjin Chung、Jeongsol Kim、Michael T. McCann、Marc L. Klasky 和 Jong Chul Ye。面向一般含噪逆问题的扩散后验采样，2022年。[6]Jacob Devlin、Ming-Wei Chang、Kenton Lee 和 Kristina Toutanova。BERT：用于语言理解的深度双向变换器预训练。载于 *NAACL-HLT* 论文集，2019年。[7] Prafulla Dhariwal 和 Alexander Quinn Nichol。扩散模型在图像合成方面胜过生成对抗网络，2021年。[8] Jonathan Ho、Ajay Jain 和 Pieter Abbeel。去噪扩散概率模型，2020年。

[9] Jonathan Ho 和 Tim Salimans ．无分类器扩散引导 2022 ，.[10] 国际色彩联盟（ICC）。sRGB色彩空间配置文件（sRGB2014及相关配置文件/传递规范材料），2014年。通过ICC配置文件注册库访问。[11] 国际标准化组织和国际照明委员会（CIE）。色度学——第4部分：CIE 1976 L*a*b*色彩空间（ISO/CIE 11664-4:2019），2019年。ISO/CIE联合标准。[12] Bahjat Kawar、Michael Elad、Stefano Ermon和Jiaming Song。去噪扩散复原模型，2022年。[13] Alexander Lobashev、Maria Larchenko和Dmitry Guskov。基于切片Wasserstein引导的颜色条件生成。载于神经信息处理系统进展（*NeurIPS*），2025年。海报。[14] Alexander Quinn Nichol和Prafulla Dhariwal。改进的去噪扩散概率模型，2021年。[15] Alexander Quinn Nichol、Prafulla Dhariwal、Aditya Ramesh、Pranav Shyam、Pamela Mishkin、Bob McGrew、Ilya Sutskever和Mark Chen。GLIDE：利用文本引导扩散模型实现照片级真实感图像生成与编辑，2021年。[16] Alec Radford、Jong Wook Kim、Chris Hallacy、Aditya Ramesh、Gabriel Goh、Sandhini Agarwal、Girish Sastry、Amanda Askell、Pamela Mishkin、Jack Clark、Gretchen Krueger和Ilya Sutskever。通过自然语言监督学习可迁移视觉模型，2021年。[17] Edgar Riba、Dmytro Mishkin、Daniel Ponsa、Ethan Rublee和Gary Bradski。Kornia：面向PyTorch的开源可微计算机视觉库。载于*IEEE/CVF*冬季计算机视觉应用会议（*WACV*），第3674–3683页，2020年。[18] R. Tyrrell Rockafellar和Stanislav Uryasev。条件风险价值的优化。《风险杂志》，2(3)：21–41，2000年。[19] Robin Rombach、Andreas Blattmann、Dominik Lorenz、Patrick Esser和Bj¨orn Ommer。利用潜在扩散模型进行高分辨率图像合成，2021年。[20] Jiaming Song、Chenlin Meng和Stefano Ermon。去噪扩散隐式模型，2020年。[21] Yang Song、Jascha Sohl-Dickstein、Diederik P. Kingma、Abhishek Kumar和Stefano Ermon。通过随机微分方程进行基于分数的生成建模，2020年。[22] Vaibhav Vavilala和David Forsyth。使用扩散模型在局部控制下应用色彩调色板，2023年。[23] Yinhuai Wang、Jiwen Yu和Jian Zhang。用于逆问题的去噪扩散零空间模型，2022年。[24] Lvmin Zhang、Anyi Rao和Maneesh Agrawala。为文本到图像扩散模型添加条件控制，2023年。

附录

附录 1： ΔE_{00} 的二阶泰勒近似

算法 6 ΔE_{00}的二阶泰勒近似

1: **输入：** Lab 颜色 $\ell_1 = (L_1,\ a_1,\ b_1)$, $\ell_2 = (L_2,\ a_2,\ b_2)$

2: 定义差值：

$$\Delta\ell = \ell_1 - \ell_2, \quad \Delta L = L_1 - L_2, \quad \Delta a = a_1 - a_2, \quad \Delta b = b_1 - b_2.$$

3: 局部二阶展开：

$$\Delta E_{00}^2 = (\Delta L)^2 + (\Delta a)^2 + (\Delta b)^2 + O\big(\|\Delta\ell\|^3\big)\,.$$

4: **结果：** 二阶泰勒近似

$$\boxed{\Delta E_{00} \approx \sqrt{(\Delta L)^2 + (\Delta a)^2 + (\Delta b)^2}}$$

5: 安全二次型：

$$\boxed{Q = (\Delta L)^2 + (\Delta a)^2 + (\Delta b)^2.}$$

附录 2：图像修复扩散算法

算法 7 推理时颜色引导下的图像修复扩散（潜变量微调）——第一部分**要求：** 管线 $\mathcal{P} =$（变分自编码器，U-Net，调度器）；画布 $X \in [0,1]^{3\times H\times W}$；感兴趣区域掩码 $M \in \{0,1\}^{1\times H\times W}$；嵌入（$e_{\text{cond}}$，$e_{\text{uncond}}$）；步数 N；无分类器引导比例 s_{cfg}；微调步长 η；掩ter 权重 λ；窗口比例（$f_{\text{start}}, f_{\text{stop}}$）；开关：线性 RGB、条件风险价值、锚定；目标颜色$c^{\star}_{\text{sRGB}} \in [0,1]^3$；超参数 Θ。确保：最终潜变量 z 和日志（损失函数、梯度范数、感兴趣区域均值、快照）。1：调度器：Scheduler.set timesteps(N)；$\{t_i\}_{i=0}^{N-1} \leftarrow$ Scheduler.timesteps。2：$i_{\text{start}} \leftarrow \lfloor f_{\text{start}} N \rfloor$，$i_{\text{stop}} \leftarrow \lfloor f_{\text{stop}} N \rfloor$。3：画布编码：$z_0 \leftarrow \alpha_{\text{vae}} \cdot$ VAE.encode(X)，$\alpha_{\text{vae}} = 0.18215$。4：潜变量掩码：$M \leftarrow$ Resize(M, shape(z_0)$_{H,W}$)。5：掩码图像潜变量：$X_{\text{mask}} \leftarrow X \odot (1 - M)$，$z_{\text{mask}} \leftarrow \alpha_{\text{vae}} \cdot$ VAE.encode(X_{mask})。6：初始化噪声/潜变量：$\epsilon \sim \mathcal{N}(0, I)$，$z \leftarrow \sigma_0 \epsilon$，$\sigma_0 =$ Scheduler.init noise sigma。7：目标值（一次性）：$c^{\star}_{\text{lin}} \leftarrow$ sRGB $\rightarrow$ LinRGB($c^{\star}_{\text{sRGB}}$)，$c^{\star}_{\text{lab}} \leftarrow$ sRGB $\rightarrow$ Lab($c^{\star}_{\text{sRGB}}$)。8：$T_0 \leftarrow t_0$。9：对于 $i = 0$ 到 $N-1$ 执行 10：$t \leftarrow t_i$。11：（1）修复去噪（CFG）：$\tilde{\ }$ $\tilde{\hat{}}z \leftarrow$ Scheduler.scale model input($[z; z], t$) $u \leftarrow [\hat{}z;\ [M;\ M];\ [z_{\text{mask}};\ z_{\text{mask}}]]$.$\hat{}\epsilon \leftarrow$ UNet($u, t, [e_{\text{uncond}}; e_{\text{cond}}]$)，$\hat{}\epsilon = \hat{}\epsilon_u \oplus \hat{}\epsilon_c$，$\hat{}\epsilon_{\text{cfg}} \leftarrow \hat{}\epsilon_u + s_{\text{cfg}}(\hat{}\epsilon_c - \hat{}\epsilon_u)$。$z \leftarrow$ Scheduler.step($\hat{}\epsilon_{\text{cfg}}, t, z$)。12：如果启用锚定，则 13：（可选）背景锚定：$\tilde{\ }$ $\tilde{}z_{\text{bg}}(t) \leftarrow$ Scheduler.add noise(z_0, ϵ, t) $z \leftarrow (1 - M) \odot z_{\text{bg}}(t) + M \odot z$. 14：结束 if。15：（2）引导门控：应用 $\leftarrow (i \geq i_{\text{start}}) \wedge (i < i_{\text{stop}})$，用于 $\leftarrow$ 线性 RGB $\vee$条件风险价值。16：如果未应用或未启用，则 17：记录零值；继续。18：结束 if。

算法 8 使用推理时颜色引导的修复扩散（潜变量微调）——第二部分

(3) 解码 +感兴趣区域均值（监控）：

$$x \leftarrow \mathrm{VAE.decode}(z/\alpha_{\mathrm{vae}}), \quad \ell \leftarrow \mathrm{VAE}\to\mathrm{Lab}(x), \quad r \leftarrow \mathrm{VAE}\to\mathrm{sRGB}(x), \quad n \leftarrow \sum_{h,w} M_{1,h,w}.$$

$$\bar{\ell} \leftarrow \frac{1}{n+\varepsilon}\sum_{h,w} \ell_{:,h,w} M_{1,h,w}, \qquad \bar{r} \leftarrow \frac{1}{n+\varepsilon}\sum_{h,w} r_{:,h,w} M_{1,h,w}.$$

追加 $(\ell, \bar{r})$ 以及可选快照。

(4) 损失函数 + 调度计划： $L_{\mathrm{lin}} \leftarrow 0$, $L_{\mathrm{cvar}} \leftarrow 0$。

如果 线性 RGB**则**

$$x_{\mathrm{lin}} \leftarrow \mathrm{VAE}\to\mathrm{LinRGB}(x), \quad w_{\mathrm{lin}}(i) \leftarrow \frac{1}{i+1}, \quad L_{\mathrm{lin}} \leftarrow w_{\mathrm{lin}}(i)\,\lambda_{\mathrm{lin}}\,\mathcal{L}_{\mathrm{LinRGB}}(x_{\mathrm{lin}}, M, c^{\star}_{\mathrm{lin}}).$$

结束如果

如果 条件风险价值 **且** $k \neq 0$ **则**

$$w_{\mathrm{cvar}}(t) \leftarrow \frac{\max(0, T_0 - t)}{k}, \qquad L_{\mathrm{cvar}} \leftarrow w_{\mathrm{cvar}}(t)\,\mathcal{L}_{\mathrm{CVaR}}(x, t, \{t_i\}, f_{\mathrm{start}}, c^{\star}_{\mathrm{lab}}, M; \Theta).$$

结束 if

$$L \leftarrow \lambda\,(L_{\mathrm{lin}} + L_{\mathrm{cvar}}).$$

记录 L_{lin}， L_{cvar} 以及（如果可用）条件风险价值子项。

(5) 潜变量微调（仅限感兴趣区域）：

如果非 有限(L) **或** $L \leq 0$ **则**

记录零值；**继续**。

结束 if

计算梯度 $g \leftarrow \nabla_z L$; $g \leftarrow \mathrm{clip}(g, -1, 1)$; $a \leftarrow \|g\|_2$。

如果 $a \leq 0$ **则**

记录并**继续**。

结束如果

$$\tilde{g} \leftarrow \frac{g}{a+\varepsilon}, \qquad z \leftarrow z - \eta\,\tilde{g} \odot \tilde{M}.$$

如果存在 NaN/Inf，则将其替换；记录 L和 a。

结束循环

返回最终值 z （包含日志和运行时间）。

附录 3：实验结果（条件风险价值 +线性 RGB）

我们展示了所提出的条件风险价值 + 线性 RGB 损失配置的实验验证。结果表明，复合目标函数能够有效引导潜变量生成，使其在感兴趣区域（ROI）内趋向目标颜色。

运行摘要

- **设备：** Tesla T4（运行时间：39.9 秒）
- **步数：** 80（强制器激活：20%–100%）

- **参数：** $\eta = 0.009$，$\lambda_{\text{引导}} = 100$，$\lambda_{\text{主控}} = 0.07$，$s_{\text{CFG}} = 8$，$k = 2$
- **切换项：** 线性 RGB=True，Angad=True，背景锚点=True

目标颜色与最终颜色

指标	L	a	b
目标颜色（Lab）	69.97	47.40	11.54
目标颜色（sRGB）	[1.00, 0.53, 0.60]		
最终颜色（Lab）	62.71	25.60	-6.48
误差（ΔE_{00}）	**29.19**（ROI 均值）		

表 1：目标颜色与实际颜色统计

逐像素 ΔE_{00} 统计

ROI 内逐像素误差分布分析：

- **最小值：** 0.36
- **最大值：** 72.05
- **均值：** 52.95
- **标准差：** 19.36
- **中位数：** 62.68

视觉分析

图 4 展示了最终输出与目标的对比，以及一张 ΔE_{00} 热力图，显示误差的空间分布。图 5 详细展示了采样过程中的损失轨迹、梯度范数和分量子损失。图 6 分析了达到不同 ΔE_{00} 阈值的感兴趣区域像素百分比。

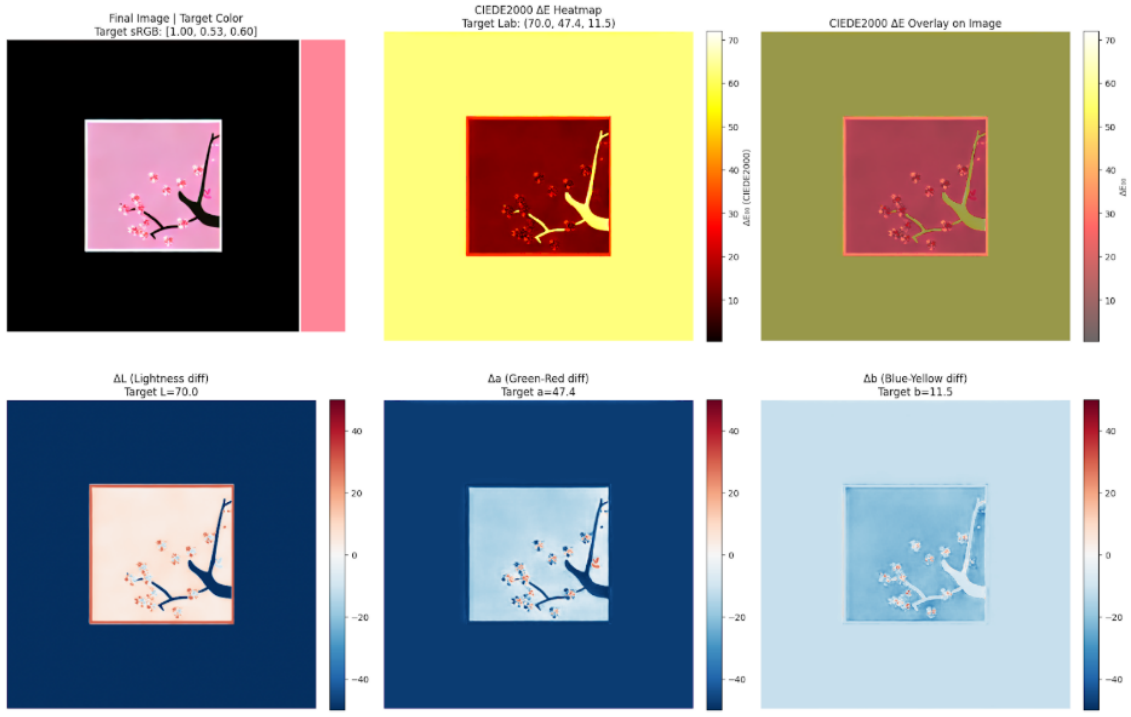


图 4：**定性结果与误差热力图**。 上排：最终图像与目标色块、 ΔE_{00} 热力图及叠加图的对比。下排： ΔL、 Δa 和 Δb 的分量差异图。

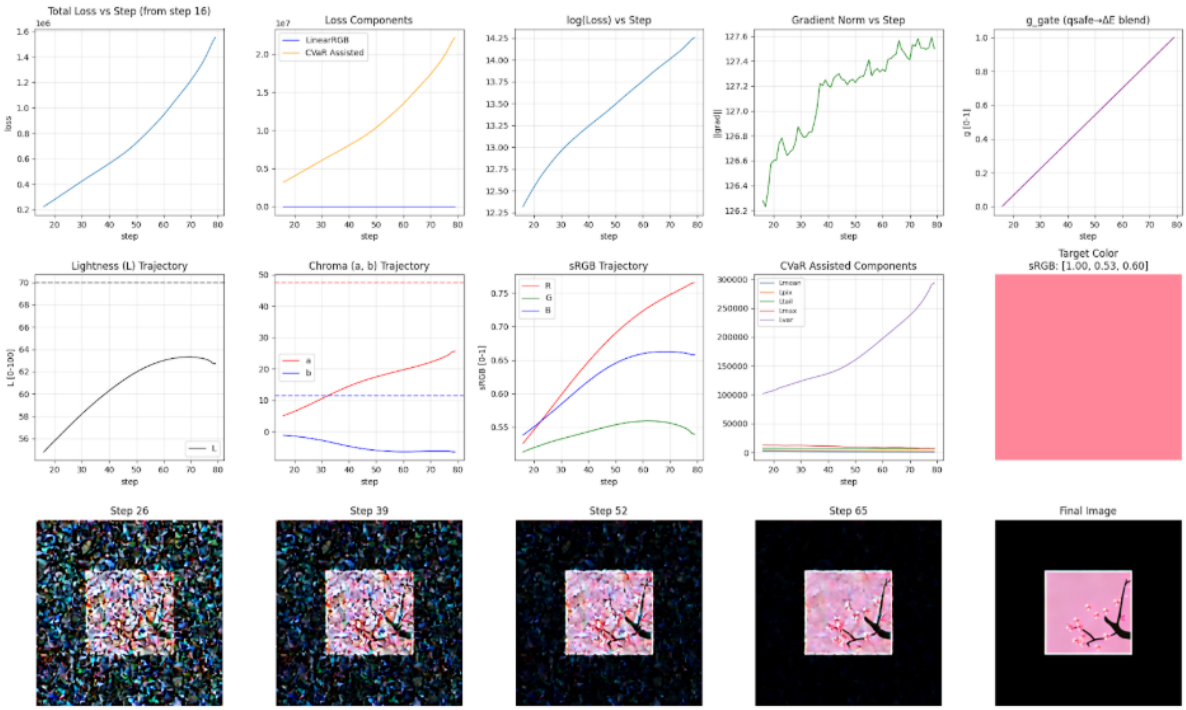


图 5：**损失与优化动态**。 总损失、分量损失（线性 RGB 与条件风险价值）、梯度范数及 sRGB 轨迹在 80 个采样步骤中的变化。



图 6：**阈值分析**。 满足 $\Delta E_{00} < T$ 的感兴趣区域像素可视化，阈值为 $T \in \{2, 5, 10, 20, 50\}$。绿色表示满足条件。

附录 4：实验结果（线性 RGB 均方误差）

我们还展示了仅使用线性 RGB 均方误差损失（不含分布感知的条件风险价值项）的配置结果。该基线体现了基于均值的目标在捕捉感知色彩准确度方面的局限性。

运行摘要

- **设备：** Tesla T4（运行时间：37.8 秒）
- **步数：** 80（强制器已启用：20%–100%）
- **参数：** $\eta = 0.009$， $\lambda_{\text{引导}} = 100$， $\lambda_{\text{主控}} = 0.07$， $s_{\text{CFG}} = 8$， $k = 2$
- **开关：** 线性 RGB=开启，Angad=关闭，背景锚点=开启

目标颜色与最终颜色

指标	L	a	b
目标颜色（Lab）	69.97	47.40	11.54
目标颜色（sRGB）	[1.00, 0.53, 0.60]		
最终颜色（Lab）	55.67	17.32	-7.26
误差（ΔE_{00}）	**38.24**（ROI 均值）		

表 2：目标颜色与实际颜色统计（仅线性 RGB）

逐像素 ΔE_{00} 统计

ROI 内逐像素误差分布分析：

- **最小值：** 3.87
- **最大值：** 68.50
- **均值：** 55.12
- **标准差：** 15.41
- **中位数：** 62.68

视觉分析

图7展示了定性结果，与目标相比，可以明显看出显著的颜色偏差。图 8展示了损失演化，图9则突出显示了阈值满足率。

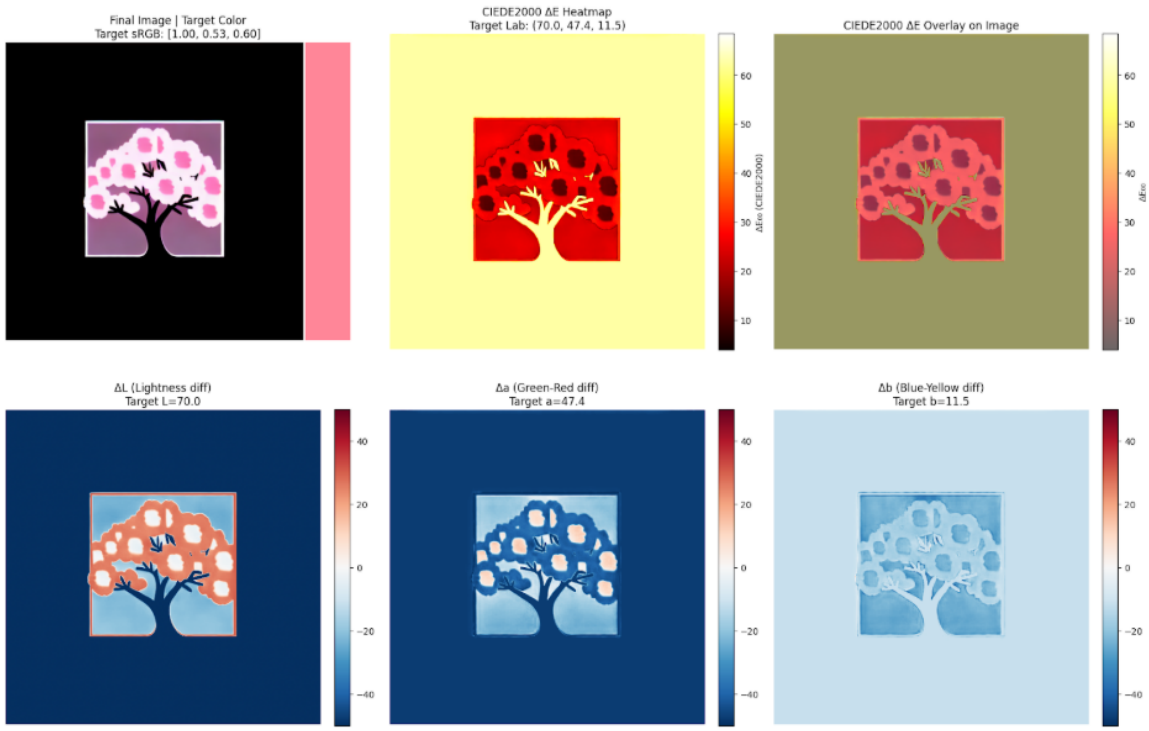


图7：**定性结果（线性RGB最佳努力）**。请注意，与条件风险价值引导的结果相比，饱和度降低且色相发生偏移。

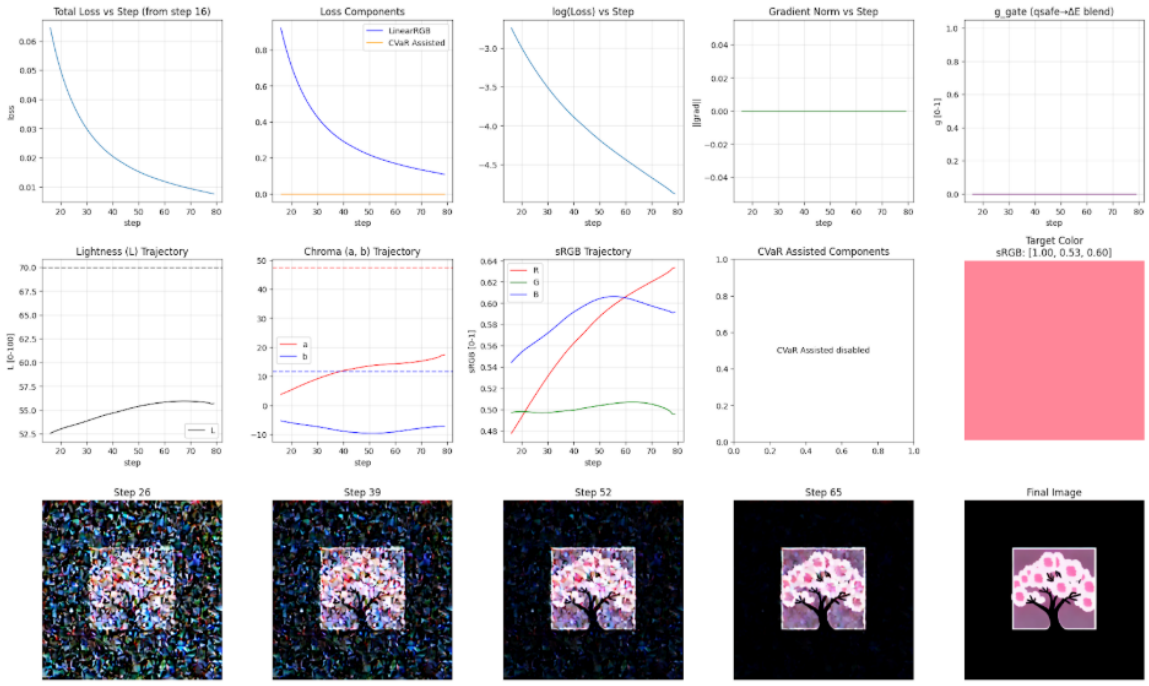


图 8：**损失动态（仅线性 RGB）**。损失平稳下降，但未能执行严格的感知约束。



图 9：**阈值分析（线性 RGB）**。满足率显著低于经条件风险价值辅助的运行结果，其中 0% 的像素达到 $\Delta E_{00} < 10$ 阈值。

附录 5：实验结果（无引导基线）

最后，我们展示未启用推理时引导的配置结果。这作为阴性对照，凸显主动引导对于颜色约束的必要性。

运行摘要

- **设备：**Tesla T4（运行时间：11.6 秒）
- **步数：**80（强制器启用：20%–100%）
- **参数：** $\eta = 0.009$， $\lambda_{\text{引导}} = 100$， $\lambda_{\text{主控}} = 0.07$， $s_{\text{CFG}} = 8$， $k = 2$
- **切换项：**线性 RGB=False，Angad=False，背景锚点=True

目标颜色与最终颜色

指标	L	a	b
目标颜色（Lab）	69.97	47.40	11.54
目标颜色（sRGB）	[1.00, 0.53, 0.60]		
最终颜色（Lab）	52.87	21.41	-11.49
误差 (ΔE_{00})	**38.71** (ROI 均值)		

表 3：目标颜色与实际颜色统计（无引导）

逐像素 ΔE_{00} 统计

感兴趣区域内逐像素误差分布分析：

- **最小值：** 5.65
- **最大值：** 73.28
- **均值：** 55.22
- **标准差：** 16.84
- **中位数：** 62.68

视觉分析

图10展示了基线性能。在没有引导的情况下，模型生成的通用粉色与目标存在显著偏差。图11显示缺少主动损失组件。图12证实，近87%的像素具有一个 $\Delta E_{00} \geq 20$，表明颜色“非常不同”。

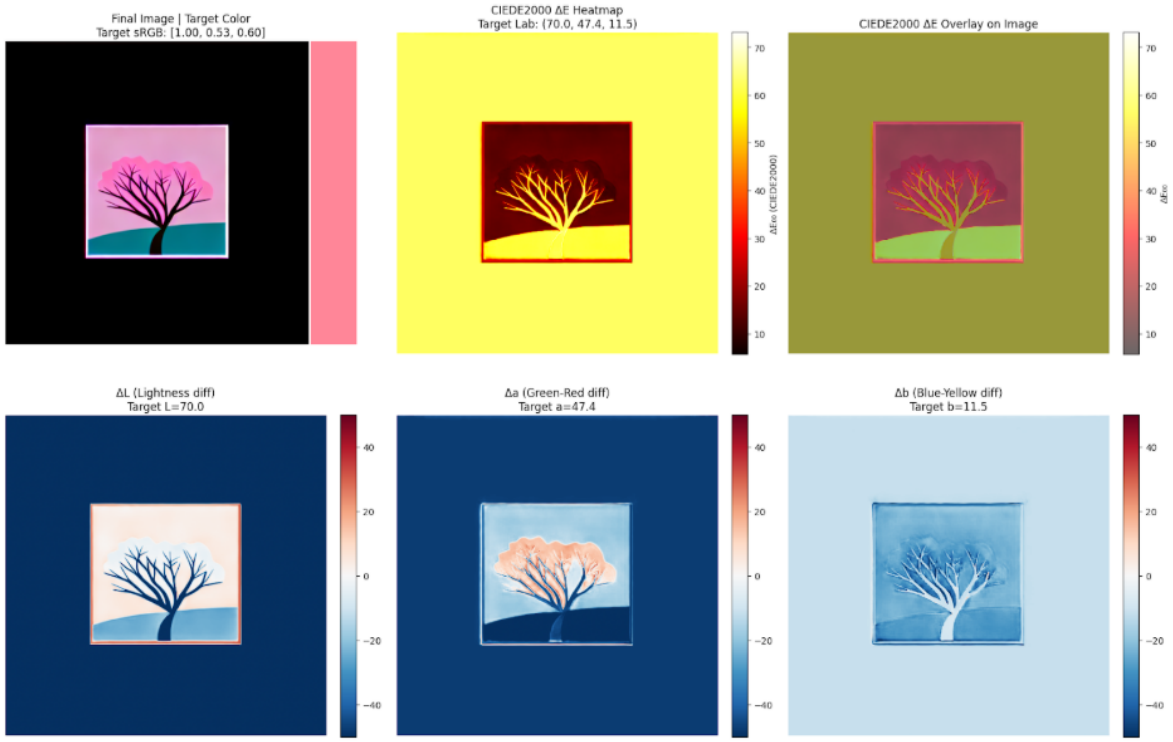


图10：**定性结果（无引导）**。输出颜色缺乏针对性，无法匹配特定的目标色块。

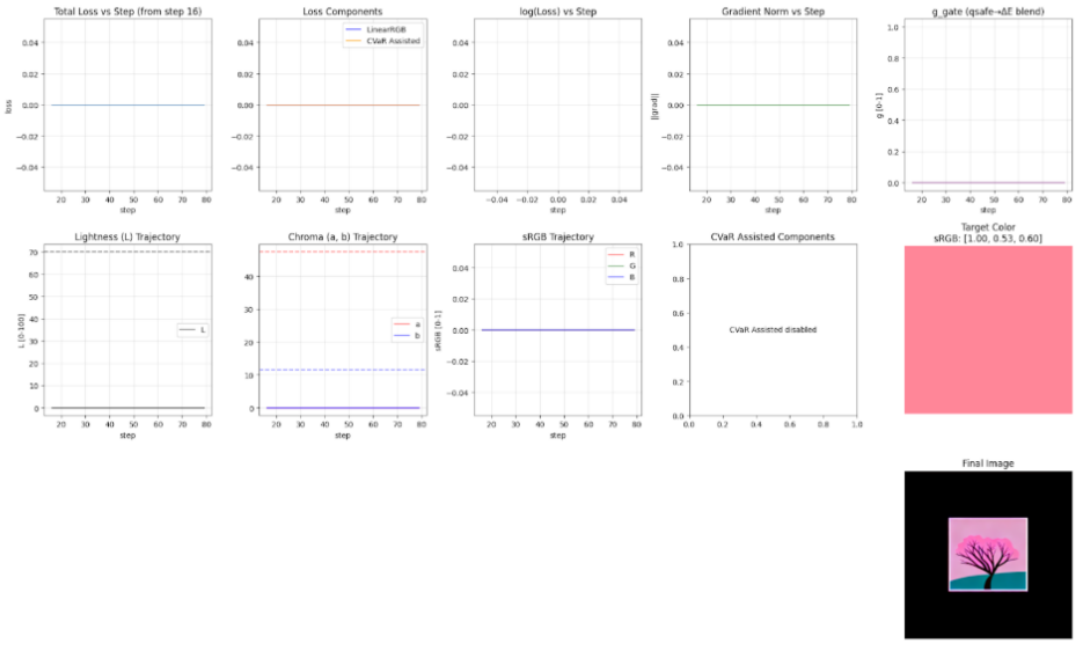


图11：**损失动态（无引导）**。平直的线条表示未计算或应用引导损失。

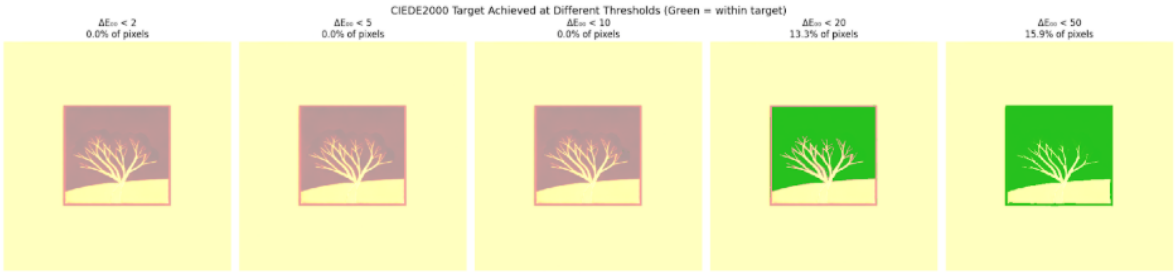


图12：**阈值分析（无引导）**。绝大多数像素都落在最高误差累积区间（$\Delta E_{00} \geq 20$）中。