arXiv:2512.17650v1 [cs.CV] 19 Dec 2025

用于指令式视频编辑的区域约束上下文生成*

Zhongwei Zhang†、Fuchen Long§、Wei Li†、Zhaofan Qiu§、Wu Liu†、Ting Yao§和Tao Mei§

†中国科学技术大学 §HiDream.ai公司{zhwzhang, weili2023}@mail.ustc.edu.cn, {longfuchen, qiuzhaofan}@hidream.ailiuwu@live.cn, {tiyao, tmei}@hidream.aihttps://zhw-zhang.github.io/ReCo-page/

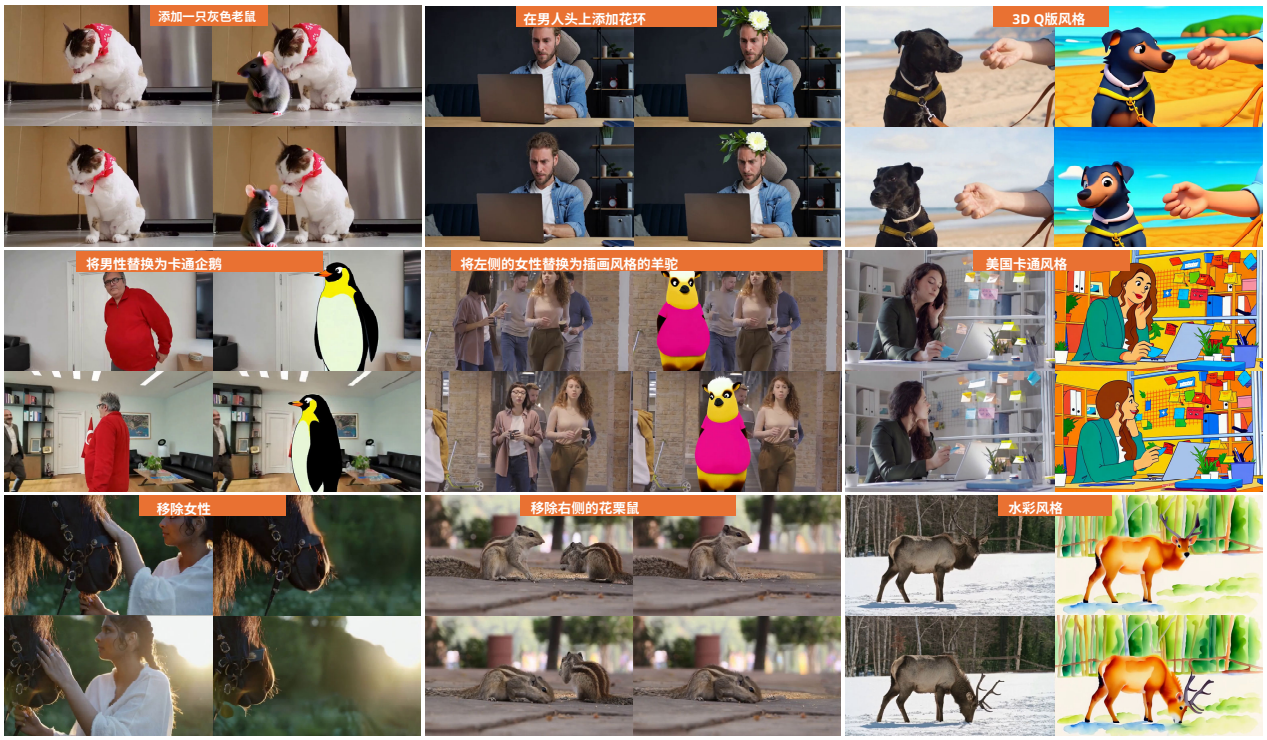


图 1. 我们的 ReCo 可仅凭文本指令实现视频编辑，精准且高保真地修改视频内容。ReCo 能灵活处理各种具有挑战性的视频编辑任务，包括局部对象编辑和全局风格迁移。

摘要

近期，上下文内生成范式在指令式图像编辑方面展现出强大能力，兼具数据效率和合成质量。然而，将这种上下文学习方式应用于基于指令的视频编辑并非易事。若未指定编辑区域，结果可能出现编辑区域定位不准确的问题，并且在去噪过程中，编辑区域与非编辑区域之间还会产生词元干扰。为解决这些问题，我们提出 *ReCo*，一种新型指令式视频

一种新颖的编辑范式，深入探索上下文生成过程中编辑区域与非编辑区域之间的约束建模。从技术上讲，*ReCo* 沿宽度方向拼接源视频与目标视频，以进行联合去噪。为了校准视频扩散学习，*ReCo* 利用两种正则化项，即潜变量正则化和注意力正则化，分别作用于单步反向去噪潜变量和注意力图。前者增大源视频与目标视频中编辑区域的潜变量差异，同时减小非编辑区域的差异，从而突出对编辑区域的修改，并缓解区域外的意外内容生成。后者则抑制编辑区域中的词元对源视频对应区域中词元的注意力，

*本工作是在 HiDream.ai 完成的。

从而减轻这些词元在目标视频中新对象生成过程中的干扰。此外，我们提出了一个大规模、高质量的视频编辑数据集 *ReCo-Data*，其中包含 *50* 万组指令-视频对，可用于促进模型训练。在四项主要的基于指令的视频编辑任务上开展的大量实验表明，我们提出的方法具有优越性。

1. 引言

随着扩散模型的快速发展， [17, 21, 36, 37, 40, 42, 44, 48, 49, 67, 70, 71, 74], 基于指令的图像和视频视觉编辑受到了广泛关注。近期的基于指令的图像编辑模型 [33, 73]能够根据自然语言指令编辑输入图像，无需额外条件。然而，要在基于指令的视频编辑领域复制图像编辑取得的成功并非易事。一些有前景的视频编辑方案[24, 26]通常需要输入掩码来定位编辑区域，或需要针对特定任务进行配置，这限制了它们在现实世界中的实用性。仅依据文本指令来引导视频编辑，仍是文献中尚未充分探索的问题。

受上下文内生成范式在图像编辑中兼具数据效率与生成质量这一成功经验的启发 [25, 72]，我们构建了一个用于基于指令的视频编辑的源视频-目标视频联合视频扩散框架。由于视频固有的时间复杂性，在为视频生成构建上下文学习时面临两大挑战：1）仅有文本指令时，如何准确定位编辑区域？2）如何进一步减少源视频编辑区域中的内容对目标视频中新对象生成的干扰？遵循视觉处理中的区域约束建模方法 [23] ，我们通过对视频潜变量和注意力图上的区域关系进行建模来解决这两个问题。针对第一个问题，我们增大源视频与目标视频在编辑区域内的潜变量差异，同时减小非编辑区域内的差异，从而促使编辑区域中的内容重新生成，同时保持背景一致性。为缓解第二个问题，我们抑制编辑区域内词元对源视频同一区域词元的注意力，减轻原始内容带来的词元干扰。这一做法还鼓励新对象生成更多利用目标视频背景中的词元信息，从而与背景实现更好的连贯性。

通过整合区域约束上下文生成的思想，我们提出了一种名为 ReCo 的新框架，用于基于指令的视频编辑。在技术层面，ReCo 首先沿左-右面板拼接源视频与目标视频，并进行联合视频去噪以实现编辑生成。在每个训练步骤中，首先通过单步反向扩散过程估计成对的视频潜变量。随后，ReCo 计算源视频潜变量与目标视频潜变量之间的潜变量差异，并进一步施加成对约束，以增大编辑区域的潜变量差异，同时减小非编辑区域的潜变量差异。类似的正则化项也被应用于 DiT模块的注意力图，以抑制编辑区域中的词元集中关注源视频中同一区域的词元。此外，还增强编辑区域词元对目标视频自身背景的注意力，从而实现新对象与背景之间的和谐构图。整个框架通过流匹配扩散损失和两个区域约束正则项进行联合优化。

本文的主要贡献是提出了一种用于基于指令的视频编辑的新型区域约束上下文生成范式。除了架构设计之外，我们还精心构建了一个大规模、高质量的视频编辑数据集 ReCo-Data，其中包含 50 万个指令-视频对，涵盖广泛的编辑任务，以促进社区对指令式视频编辑的研究。大量实验进一步验证了 ReCo 在编辑准确性和质量方面的有效性。

2. 相关工作

基于指令的图像编辑。 近年来，文本到图像生成 [5, 9, 17、19–21, 30, 32, 36, 42, 42, 44, 48, 49, 58, 65, 67] 领域取得的显著进展，推动了指令引导图像编辑的发展。作为该领域的代表性工作之一，InstructPix2Pix [9], 建立了高效的图像编辑数据构建流程，并取得了令人鼓舞的编辑成果。后续工作将这一数据流程作为原型加以改进，为训练强大的基于指令的编辑器提供更多数据。沿用这一方案，Emu Edit [50], OmniGen [64], ICEdit [72],HiDream-E1 [11], Flux-Kontext [33], Qwen-Image [61],和 Nano-Banana [1] 等多模态模型进一步解锁了局部编辑和场景变换等复杂能力，即使无需专门微调也能实现。然而，要将图像编辑的成功经验复现到指令式视频编辑领域并非易事。视频编辑的挑战不仅在于数据匮乏，还在于必须同时处理空间和时间标记之间错综复杂的依赖关系。本文通过上下文生成范式和区域约束建模来应对这些挑战，并以我们新构建的 ReCo-Data 为支撑。

基于指令的视频编辑。 早期尝试 [14, 18, 28, 31, 35, 38, 41, 45, 69] 基于指令的视频编辑的方法通常采用免训练推理范式，

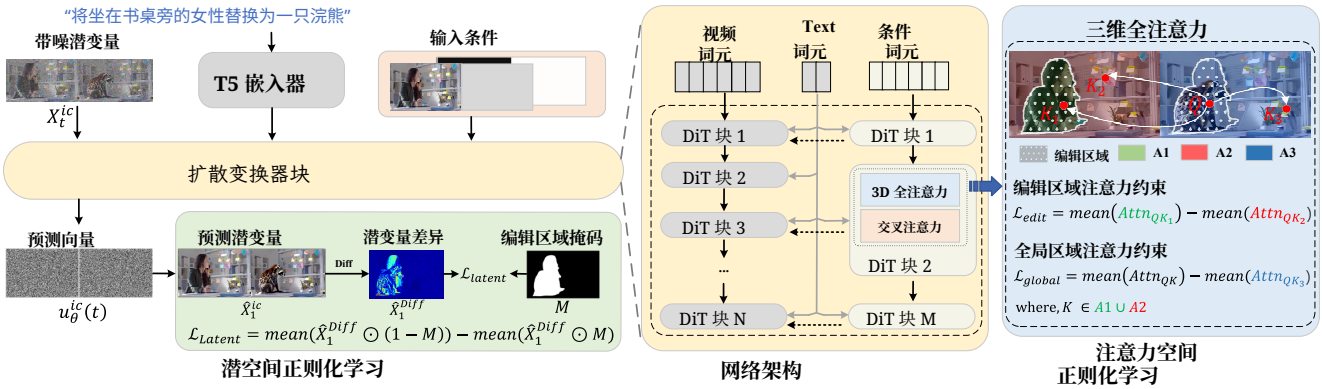


图 2. ReCo 框架概览。我们将指令视频编辑任务重新表述为上下文生成范式，并以源视频和指令提示为引导。我们通过将源视频输入辅助视频条件分支，将其作为显式条件。为突出编辑修改并缓解编辑区域与非编辑区域之间的词元干扰，ReCo 引入了两种基于区域的约束：(1) 潜空间正则化，通过增加源视频与目标视频中编辑区域的潜变量差异，同时降低非编辑区域的潜变量差异来实现。(2) 注意力空间正则化，抑制目标编辑区域对源视频中对应区域的注意力，从而减轻固有的词元干扰，同时增强其对自身生成内容的注意力。

利用预训练的文本到图像扩散模型进行逐帧视频编辑。例如，FateZero [45]通过 DDIM [53] 扩散反演来编辑视频帧。然而，缺乏时间建模会导致时间不一致问题。为了解决这一问题，后续方法（如 TokenFlow [18], VidToMe [35], FLATTEN [14],和 RAVE [28]）采用令牌合并或相似性约束来增强时间连贯性。另有一些方法（如 FlowEdit [32] 和 FlowDirector [34]）利用先进的文本到视频扩散模型，实现更准确的扩散反演。尽管免训练范式具有灵活性，视频质量和模型泛化能力仍是其固有局限。

训练式视频编辑发展的主要障碍，是大规模、高质量配对训练数据极度匮乏。早期方法通过一次性调优技术克服了这一困难，例如 Tune-A-Video [62] 和 Video-P2P [38],，但仍难以实现理想的编辑效果。另一方面，一些研究 [12, 39, 63] 从数据集构建和框架设计两方面推动了视频编辑的发展。例如，GenProp [39] 和 Senorita [75] 将编辑建模为图像传播机制：先编辑首帧，再将内容修改传播到其他帧。最近，Lucy-Edit [56] 和 Ditto [2] 提出直接使用源视频、目标视频和文本指令训练视频编辑模型。Lucy-Edit 将源视频潜变量与去噪后的视频潜变量拼接作为条件，而 Ditto 则通过 ControlNet 的方式学习条件。与此同时，上下文学习 [25, 72] 在图像编辑中已被证实兼具数据效率和生成质量优势。在我们的工作中，我们借鉴了这一思路，并进一步探讨区域级约束的设计，以促进精准的视频编辑。

总而言之，我们为指令式视频编辑设计了一种新颖的区域约束上下文生成范式。所提出的 ReCo 不仅研究如何精准定位编辑区域，还探讨如何进一步减少编辑区域与非编辑区域之间的词元干扰，从而实现连贯的视觉修改。

3. 我们的方法

这里将介绍 ReCo，这是一种用于指令式视频编辑的新型区域约束的上下文视频生成框架。整体架构如图2所示。给定一对源视频与目标视频，ReCo 将生成过程重新表述为上下文学习范式，通过沿宽度方向拼接两个视频来进行联合去噪。同时，为确保忠实保留源视频信息，我们采用额外的视频条件分支，探索对源视频进行条件学习。在训练阶段，我们引入两个正则化项，即潜变量正则化和注意力正则化，以便在不预先指定编辑区域的情况下学习精准的视频编辑。潜变量正则化在单步反向去噪潜变量上进行，以增强区域级修改并保持背景一致性。与此同时，注意力正则化项抑制目标视频中新生成对象对源视频编辑区域的注意力，从而减少原始视觉内容造成的词元干扰。

3.1. 预备知识：视频DiT训练

为了利用预训练视频生成模型的先验知识 [5, 6, 8, 10, 19, 22, 30, 52, 58, 65, 68],，我们采用先进的视频差异 Transformer，即 Wan-

T2V-1.3B [58], 作为 ReCo 的主干架构。为了便于清晰理解我们的方案，我们首先回顾视频 DiT 的训练过程。通常，大多数视频 DiT 模型都建立在流匹配 [17, 36]理论之上，该理论为学习连续时间生成过程提供了严格的理论框架。其目标是学习一个向量场，将样本从简单的先验分布 P_0（例如高斯分布 $\mathcal{N}(0,1)$）平滑地输送至目标数据分布 P_1。

给定训练中的视频潜变量 x_1，从 logit-normal 分布中采样一个随机噪声样本 $x_0 \sim \mathcal{N}(0, 1)$ 和一个时间步 $t \in [0,1]$。随后，基于 Rectified Flow [17]，通过前向扩散过程将 x_0 与 x_1 结合，得到中间带噪潜变量 x_t：

$$x_t = tx_1 + (1-t)x_0. \tag{1}$$

然后，计算真实速度向量：

$$v_t = \frac{dx_t}{dt} = x_1 - x_0. \tag{2}$$

训练视频 DiT 模型以通过以下方式估计该向量：

$$u_\theta(t) = u(x_t, c, t; \theta), \tag{3}$$

其中，x_t 是带噪潜变量，θ 表示模型参数，c 是输入条件集合。对于指令视频编辑任务，c 由文本指令和源视频组成。因此，训练目标 $\mathcal{L}$ 定义为模型输出与真实速度 v_t 之间的均方误差（MSE）：

$$\mathcal{L} = \mathbb{E}_{x_0, x_1, c, t}\left\|u(x_t, c, t; \theta) - v_t\right\|^2. \tag{4}$$

该目标表明，任意给定时间步 t 的目标向量（即瞬时速度）只需表示为 $x_1 - x_0$。目标非常明确且稳定，神经网络能够轻松学习，从而实现高质量的视频生成。

3.2. 视频编辑的上下文内生成

近期，上下文内生成范式在图像编辑方面展现出显著优势，[25, 51, 55,72], 尤其体现在数据效率和生成质量方面。受此启发，我们将视频编辑过程重新表述为上下文生成。具体而言，给定视频潜变量对（即源视频 x_1^{src}和目标视频 x_1^{tar}），我们沿宽度方向 将二者拼接，形成单个上下文视频潜变量 x_1^{ic}，如下所示：

$$x_1^{ic} = [x_1^{src}, x_1^{tar}]. \tag{5}$$

在模型训练期间，从高斯分布中采样一个噪声潜变量 x_0^{ic}，然后将其添加到 x_1^{ic}中以造成破坏，生成带噪潜变量x_t^{ic}，并将其输入视频 DiT，对源视频和目标视频进行联合去噪：

$$x_t^{ic} = tx_1^{ic} + (1-t)x_0^{ic}. \tag{6}$$

真实速度向量可重新表述为：

$$v_t^{ic} = \frac{dx_t^{ic}}{dt} = x_1^{ic} - x_0^{ic}. \tag{7}$$

因此，我们将训练目标 Eq.(4)调整为适用于上下文内生成范式的形式 $\mathcal{L}_{ic}$，如下所示：

$$u_\theta^{ic}(t) = u(x_t^{ic}, c, t; \theta), \tag{8}$$

$$\mathcal{L}_{ic} = \mathbb{E}_{x_0^{ic}, x_1^{ic}, c, t}\left\|u(x_t^{ic}, c, t; \theta) - v_t^{ic}\right\|^2. \tag{9}$$

在我们的上下文生成场景中，预测向量 $u_\theta^{ic}(t)$需要同时学习源视频的重建和编辑后视频的生成。由于源视频与目标视频在编辑过程中高度相关，这种联合学习有助于实现强大的词元交互，从而提升视频编辑性能。同时，我们采用视频条件分支（如图2所示），确保充分学习视频条件，以校准视频去噪。此外，我们利用低秩适配（LoRA）技术，实现高效、稳定的视频 DiT 微调。

3.3. 潜在空间中的区域约束

上下文生成有助于源视频与目标视频之间的词元交互，从而实现更好的指令式视频编辑。然而，与以往需要预先指定编辑区域的视频编辑方法相比，仅依赖文本指令仍可能导致编辑区域不准确的问题。为缓解这一局限，我们在潜在空间中引入区域约束。该机制旨在增大源视频与目标视频之间编辑区域的潜变量差异，同时减小非编辑区域的潜变量差异，从而分别强化编辑区域的修改并减少编辑区域之外的意外内容生成。

给定视频 DiT 在时间步 t 估计的速度向量 $u_\theta^{ic}(t)$，我们首先根据 Rectified Flow 的定义推导出一步反向去噪后的潜变量 $\hat{x}_1^{ic}$：

$$\hat{x}_1^{ic} = x_t^{ic} + (1-t)u_\theta^{ic}(t). \tag{10}$$

随后，沿宽度维度对得到的去噪后的视频潜变量 $\hat{x}_1^{ic}$ 进行拆分，得到源视频和目标视频部分，如下所示：

$$[\hat{x}_1^{src}, \hat{x}_1^{tar}] = \hat{x}_1^{ic}. \tag{11}$$

接下来，我们通过以下方式计算$\hat{x}_1^{src}$与$\hat{x}_1^{tar}$之间的潜在差异向量$\hat{X}_1^{Diff}$：

$$\hat{X}_1^{Diff} = \left|\hat{x}_1^{tar} - \hat{x}_1^{src}\right|. \tag{12}$$

为了成功完成编辑，我们假设源视频与目标视频在编辑区域内的潜变量差异应较大，而非编辑

区域应保持不变。为此，我们引入潜在空间区域约束 $\mathcal{L}_{\text{latent}}$，以调节 DiT 的训练。令 M 为指示编辑区域的二值潜在掩码（其中 $M = 1$ 表示应编辑的区域）。$\mathcal{L}_{\text{latent}}$ 旨在最小化非编辑区域内的平均差异，同时最大化编辑区域内的平均差异，其计算方式如下：

$$\mathcal{L}_{\text{latent}} = \text{mean}\left(\hat{X}_1^{Diff} \odot (1 - M)\right) - \text{mean}\left(\hat{X}_1^{Diff} \odot M\right). \tag{13}$$

3.4. 注意力空间中的区域约束 ce

除了对潜在空间施加区域约束外，要稳健地进行上下文生成，还需要在注意力层面缓解编辑区域与非编辑区域之间固有的词元干扰。例如，为了更好地保持连贯性，应减少对源视频中编辑区域原始内容的关注，并更多关注该区域自身生成的背景。为了描述注意力中的这些相对关联，我们从两个角度对注意力图学习加以规范，即编辑区域内部的相对关系，以及整个视频区域范围内的相对关系。

如图 2右侧所示，我们首先将源视频-目标视频对的整个区域划分为三个不同的关键区域：源视频的编辑区域（A1）、源视频的非编辑区域（A2）以及整个目标视频区域（A3）。为了描述编辑区域内部与注意力学习相关的相对关系，来自目标编辑区域的词元（查询 Q）应减少对相应源编辑区域（键 K_1）的注意力。我们将其定义为编辑注意力损失 $\mathcal{L}_{edit}$：

$$\mathcal{L}_{edit} = \text{mean}(Attn_{QK_1}) - \text{mean}(Attn_{QK_2}), \tag{14}$$

其中 $Attn_{QK}$ 是查询 Q与键 K之间的相似度分数。此外，为确保生成内容与背景协调融合，查询Q 应减少对整个源视频（例如， $A1 \cup A2$中的键 K ）的整体依赖，同时更加关注目标视频中与上下文相关的区域（例如， $A3$中的键K_3 ）。因此，此类约束被表述为全局注意力损失 $\mathcal{L}_{global}$：

$$\mathcal{L}_{global} = \text{mean}(Attn_{QK}) - \text{mean}(Attn_{QK_3}), \tag{15}$$

因此，注意力空间区域约束定义为这两个组成部分之和：

$$\mathcal{L}_{attn} = \mathcal{L}_{edit} + \mathcal{L}_{global}. \tag{16}$$

最后，我们 ReCo 的整体训练目标被表述为多任务损失，通过整合基础的上下文

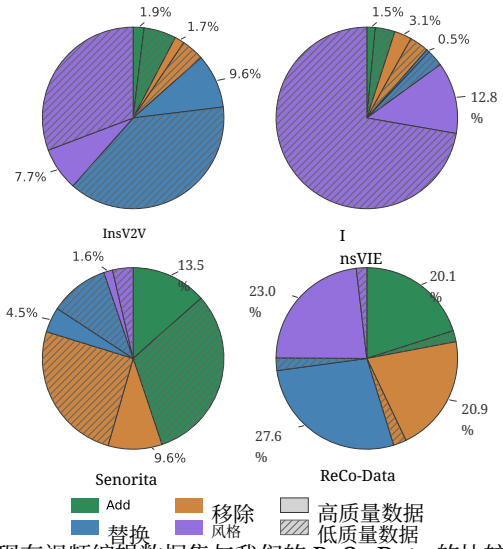


图3. 现有视频编辑数据集与我们的 ReCo-Data 的比较。我们的数据集具有最均衡的数据分布，且高质量样本的比例更高。

流匹配损失 $\mathcal{L}_{ic}$ 以及潜在空间 $\mathcal{L}_{\text{latent}}$ 和注意力空间 $\mathcal{L}_{attn}$ 中的两项区域约束：

$$\mathcal{L} = \mathcal{L}_{ic} + \lambda_1 \mathcal{L}_{\text{latent}} + \lambda_2 \mathcal{L}_{attn}, \tag{17}$$

其中， λ_1 和 λ_2 是权衡参数。这两个约束强调更准确的编辑区域以及对正确词元关系的学习，从而减轻词元干扰，实现更自然的视频内容生成。

4. 实验

4.1. 实验设置

数据集。 尽管指令式视频编辑近年来取得了巨大进展，但仍存在一个重大瓶颈：缺乏大规模、高质量的训练数据集。为应对这一挑战，我们推出了**ReCo-Data**，该数据集经过精心整理，旨在支持四项主要的视频编辑任务：实例级对象添加、移除和替换，以及全局视频风格化。我们的数据构建流程包括六个主要阶段：(1) 原始数据预处理；(2) 对象分割；(3) 使用视觉大语言模型（即 Gemini-2.5-Flash-Thinking [1]）生成指令；(4) 构建条件对；(5) 使用 VACE [26]进行视频合成；(6) 使用视觉大语言模型进行视频筛选和重新生成描述。更多详情见补充材料。最终，我们构建了 ReCo-Data，其中包含 50 万个高质量指令-视频对。每个视频片段包含 81 帧，分辨率为 480×832。视频时长为 5.0 秒。

我们从高质量样本占比的角度，将 ReCo-Data 与现有视频编辑数据集进行比较，该指标反映了数据集的可用性和整体质量。具体而言，我们从所有数据集中每项编辑任务中随机抽取 200 个视频编辑对

表1。在四项视频编辑任务（即添加对象、替换对象、移除对象和风格迁移）上的性能比较。我们将源视频与目标视频对输入 Gemini-2.5-Flash-Thinking，以评估视频编辑质量 [1]，并要求 VLLM 从三个主要角度给出评分：（1）编辑准确性包括语义准确性（SA）、范围精确度（SP）和内容保留（CP）三个子维度；（2）视频自然度包括外观自然度（VN）、尺度自然度（SN）和运动自然度（MN）；（3）视频质量包括视觉保真度（VF）、时间稳定性（TS）和编辑稳定性（ES）。每个评估子维度的得分范围为0至10（得分越高越好）。此外，我们还报告了各类别得分（即 S_{EA}、S_{VN}、S_{VQ}），其计算方式为对每个主要角度下所有子维度得分取几何平均值，并给出总体平均得分 S。

Task	方法	编辑准确性（EA）			视频自然度（VN）			视频质量（VQ）			平均得分			
		SA	SP	CP	AN	SN	MN	VF	TS	ES	S_{EA}	S_{VN}	S_{VQ}	S
Add	InsViE [63]	2.60	2.79	2.78	2.33	3.98	3.74	3.71	3.91	3.58	2.60	3.10	3.46	3.05
	Lucy-Edit [56]	6.27	6.32	7.75	4.63	7.08	6.08	6.31	6.82	7.57	6.47	5.70	6.77	6.31
	Ditto [2]	7.46	7.24	6.30	6.30	**8.85**	**8.30**	**8.13**	8.55	9.03	6.70	**7.57**	8.41	7.56
	ReCo	**8.65**	**8.40**	**9.22**	**6.39**	8.78	8.28	8.02	**8.61**	**9.61**	**8.54**	7.55	**8.61**	**8.23**
替换	InsViE [63]	1.89	2.38	2.48	2.58	5.25	5.05	3.76	4.00	3.52	2.10	3.91	3.49	3.17
	Lucy-Edit [56]	6.57	7.49	7.73	5.13	7.46	6.65	6.32	6.64	8.08	7.08	6.21	6.88	6.72
	Ditto [2]	4.95	4.83	4.79	5.81	8.63	8.10	7.55	7.95	8.71	4.56	7.21	7.96	6.58
	ReCo	**9.38**	**9.43**	**9.59**	**7.07**	**8.87**	**8.47**	**8.19**	**8.65**	**9.67**	**9.43**	**8.01**	**8.77**	**8.74**
移除	InsViE [63]	2.53	2.49	2.44	2.63	4.87	4.72	3.41	3.67	3.40	2.44	3.76	3.29	3.16
	VACE [26]	4.58	4.58	4.56	4.96	6.09	5.89	5.48	5.50	5.57	4.57	5.43	5.56	5.19
	ReCo	**7.43**	**7.43**	**7.17**	**6.20**	**7.43**	**7.30**	**6.48**	**6.63**	**7.68**	**7.28**	**6.90**	**6.82**	**7.00**
风格	InsViE [63]	7.59	8.86	8.49	6.77	9.14	9.28	7.13	6.40	8.99	8.17	8.21	7.35	7.91
	Lucy-Edit [56]	3.73	5.59	5.39	4.20	5.88	5.88	4.44	4.17	5.87	4.65	4.67	5.17	4.83
	Ditto [2]	9.10	9.36	9.26	8.25	9.51	9.58	8.33	8.33	9.77	9.20	9.07	8.77	9.01
	ReCo	**9.11**	**9.82**	**9.54**	**8.43**	**9.55**	**9.70**	**8.61**	**8.35**	**9.87**	**9.42**	**9.19**	**8.90**	**9.17**

，并邀请 10 名评估人员对视频编辑质量进行定性评估。如图3所示，现有数据集（即 InsV2V [12], InsVIE [63], 和 Senorita [75]）中的高质量样本占比通常较低（$17.9\% \sim 29.2\%$）。这表明这些数据集尚未经过严格的数据清洗，大量低质量样本使其不适合用于训练高性能的指令式视频编辑模型。此外，重新清洗数据的成本极高，而由于以往数据集的帧率低、分辨率低且合成质量差，潜在收益却微乎其微。相比之下，我们的 ReCo-Data 中高质量样本占比极高（91.6%），且不同任务间的数据分布均衡。该数据集无需任何预处理即可直接用于模型训练。我们的模型训练结果也验证了 ReCo-Data 的可用性。

基准测试。我们构建了一个视频编辑评估基准测试，包含 480 个视频-指令对，四种视频编辑任务各占 120 对。由于传统指标通常难以从多个维度准确、全面地评估视频编辑，我们借鉴图像编辑领域的进展 [60], , 采用 VLLM 作为评审进行评估。考虑到视频数据固有的复杂性，我们扩展了图像编辑指标 [60]，并构建了一套针对视频编辑的多样化评估维度。我们从三个主要方面衡量视频编辑：（1）编辑准确性，其子维度包括语义准确性（SA）、范围精确度（SP）和内容保留（CP）；（2）视频自然度，包括外观自然度（AN）、尺度自然度（SN）和运动自然度（MN）；（3）视频质量，包括视觉保真度（VF）、时间稳定性（TS）和编辑稳定性（ES）三个维度。我们通过计算各主要方面下所有子维度得分的几何平均值，得到类别得分（即 S_{EA}、S_{VN}、S_{VQ}）。总体平均得分（S）为三个类别得分的算术平均数。具体而言，我们将源视频与生成视频组成的视频对，以及预定义的系统指令输入 Gemini-2.5-Flash-Thinking [1], , 并要求其从全部九个子维度对视频编辑进行评分。关于基准测试构建和完整评估流程的更多细节，请参阅补充材料。

实现细节。在 ReCo 中，我们采用 Wan-T2V-1.3B [58] 作为基础架构。每个训练样本都是一个 81 帧的视频片段，帧率为 16 fps，分辨率为 480×832。为了使掩码生成与视频潜变量的分辨率对齐，我们先通过变分自编码器 [29]对编辑掩码进行编码，然后应用 k-均值聚类将其二值化。我们将 LoRA 的秩设为 128。ReCo 使用 AdamW 优化器进行训练，并采用两阶段学习率计划：模型首先以 1×10^{-4} 的学习率进行训练，以实现稳定收敛；随后进入微调阶段，使用较低的 2×10^{-5}学习率进一步优化。所有实验均在 24 张 NVIDIA A800 GPU 上进行，迷你批次大小为 24。

4.2. 与最先进方法的比较

我们将 ReCo 与多种最先进的指令视频编辑方法进行比较，包括 InsViE [63],Ditto [2], Lucy-Edit [56], 和VACE [26], , 在我们的 VLLM-

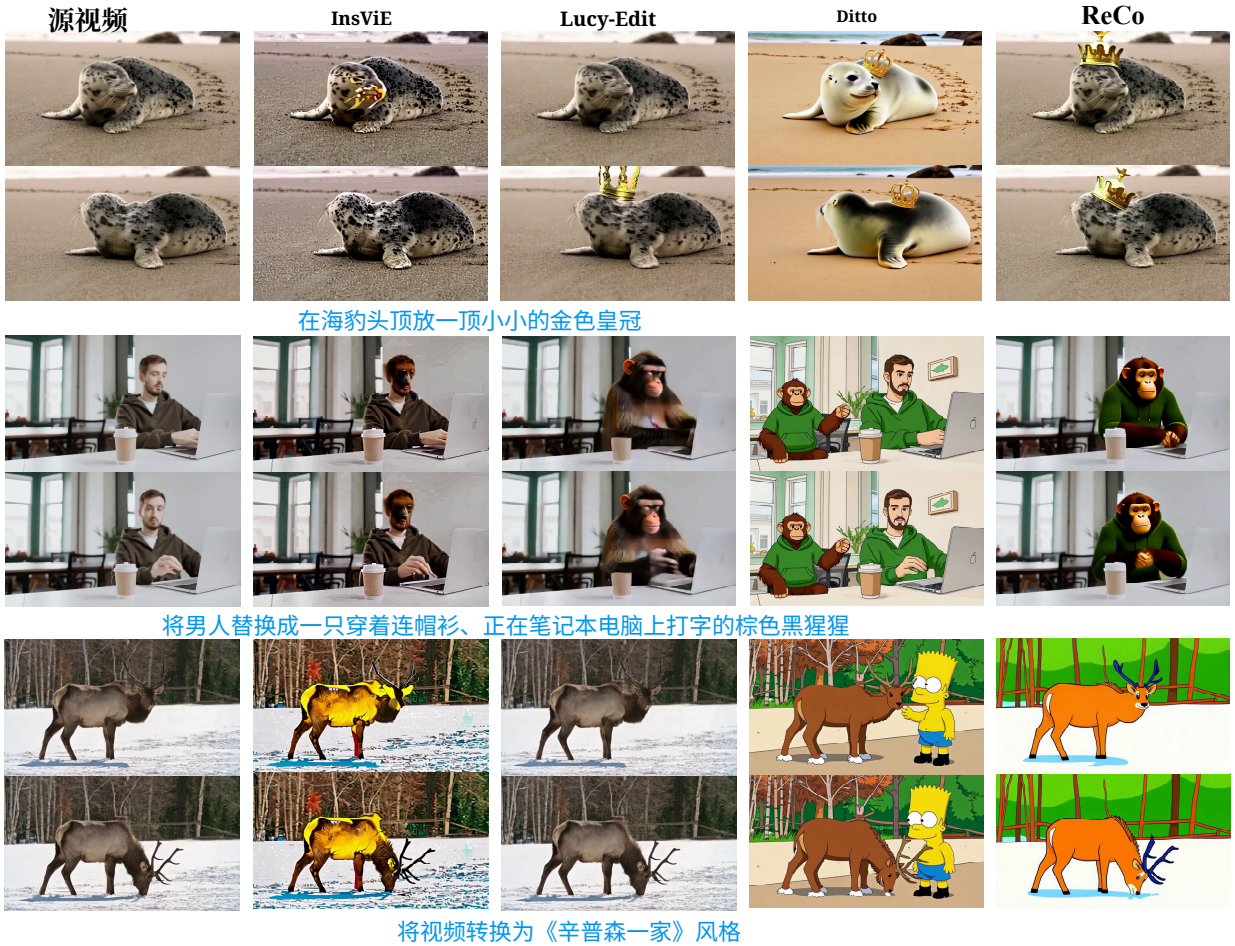


图 4. 不同方法的视频编辑（即添加对象、替换对象和风格迁移）结果示例。

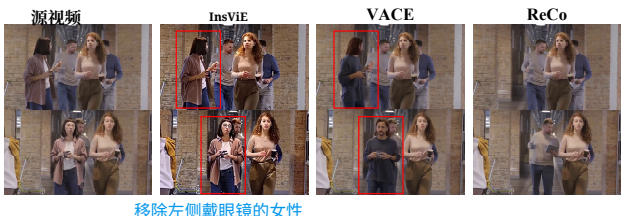


图 5. 物体移除任务的视觉对比。

基准测试中。表 1总结了四项视频编辑任务上的性能比较结果。总体而言，ReCo 在总分上始终优于现有基线方法 S，并在所有任务中均取得更好的表现。尤其是在局部视频编辑任务中，ReCo 在 S Add 上的总分为 8.23，在 *Replace*上的总分为 8.74，分别超过强劲竞争对手 Ditto（7.56）和 Lucy-Edit（6.72）0.67 和 S_{EA}。在编辑准确性方面（即 S_{VN}），也可以观察到显著的性能趋势。结果表明，我们的 ReCo 不仅能够准确遵循指令提示，正确定位编辑区域，还能保留未编辑区域的内容。在视频自然度方面（即 S_{VN}），其表现更优

我们的模型所取得的成果验证了将编辑对象自然融入源视频的有效性。尽管 S_{VN}ReCo 的表现略低于 Ditto 在添加任务上的表现，但如图4所示，与 Ditto 倾向于将整个视频重新渲染为不同色彩风格相比，我们的方法能更好地保留原始视频内容。这一现象也从 Ditto 较低的 S_{EA} 得分（6.70）得到印证。此外，视频质量（S_{VQ}）方面的最佳表现进一步表明，ReCo 生成的视频几乎没有视觉伪影或质量退化。即使在多任务训练设置下（即统一局部编辑和全局风格化），这可能会在模型优化过程中引发一些冲突，ReCo 仍展现出强大的视频风格迁移能力，并在总分9.17 中获得 S。所有这些结果基本上验证了在指令式视频编辑的上下文生成中采用区域约束建模的优势。

图4 和5进一步展示了四项任务上的视频编辑结果。总体而言，与其他基线方法相比，ReCo 编辑的视频具有更好的指令遵循能力、更高的视频质量和更好的背景一致性。例如，InsViE 往往会生成带有伪影的视频，并且

表2. ReCo 的不同变体在四种视频编辑任务上的性能比较。

模型	Add				替换				移除				风格			
	S_{EA}	S_{VN}	S_{VQ}	S	S_{EA}	S_{VN}	S_{VQ}	S	S_{EA}	S_{VN}	S_{VQ}	S	S_{EA}	S_{VN}	S_{VQ}	S
ReCo$_{LC-}$	8.05	7.44	8.59	8.03	9.01	8.01	8.67	8.56	6.90	6.83	**6.91**	6.88	9.09	9.10	8.84	9.01
ReCo$_{AC-}$	8.33	7.37	8.01	7.90	9.23	7.94	8.46	8.54	7.11	6.75	6.70	6.85	9.21	9.08	8.81	9.03
ReCo	**8.54**	**7.55**	**8.61**	**8.23**	**9.43**	**8.01**	**8.77**	**8.74**	**7.28**	**6.90**	6.82	**7.00**	**9.43**	**9.19**	**8.90**	**9.17**

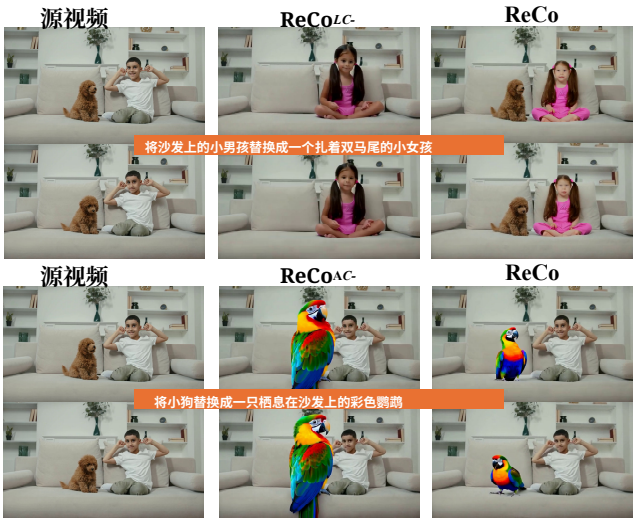


图 6. ReCo 各变体在替换任务上的编辑结果。

通常会出现编辑失败。近期的 Lucy-Edit 指令遵循能力较差，无法准确呈现指定属性（例如，穿着连帽衫的棕色黑猩猩）。尽管 Ditto 在添加 任务中能够生成外观自然的对象，但难以保持非编辑区域的背景一致性，也难以准确定位编辑区域（例如，在海豹背部添加皇冠）。与此同时，Ditto 的指令遵循能力不如我们的方法，它错误地在男子身旁合成了一只新猴子，而非替换该男子。我们推测，Ditto 的这些问题源于直接使用文本指令对 VACE 进行微调时，缺乏针对上下文生成的区域相关性建模 [26] 。相比之下，我们的 ReCo 通过区域级约束来规范上下文生成学习，从而强调编辑区域定位，并同时缓解跨区域令牌干扰。因此，ReCo 修改后的视频既呈现了准确的编辑结果，也实现了新对象与原始视频背景的自然融合。

4.3. 区域约束的消融研究

我们研究了 ReCo 中的两种区域约束如何影响最终的基于指令的视频编辑。表 2总结了 ReCo 不同变体的视频编辑性能。我们还进行了两次额外实验，即 ReCo$_{LC-}$ 和 ReCo$_{AC-}$，分别移除了 ReCo 中的潜在空间区域约束和注意力区域约束。具体而言，当舍弃潜在空间中的区域约束时，在 S_{EA}上的性能会大幅下降，表明编辑准确性显著降低。S_{VN} 和 S_{VQ} 的得分也略有下降，但仍具有可比性。这些结果凸显了潜在区域约束学习在提升编辑区域准确定位方面的有效性。图 6的上半部分进一步展示了 ReCo$_{LC-}$ 和 ReCo 中的一个视频编辑案例。给定“将沙发上的小男孩替换为一个扎双马尾的小女孩”的指令，ReCo$_{LC-}$ 能够替换男孩，但错误地移除了附近的狗。

移除注意力空间中的正则化项后，ReCo$_{AC-}$ 在视频自然度方面表现更差（即 S_{VN}），如表 2所示。我们还在图 6的下半部分展示了一个编辑示例。如图所示，ReCo$_{AC-}$ 生成了一只体型很大的鹦鹉，与周围环境的比例不协调。借助注意力正则化，减少编辑区域对词元的干扰，并增强生成新对象时与背景的交互，ReCo 合成的鹦鹉大小自然，整体协调性也更好。

5. 结论

我们提出了 ReCo，用于塑造基于指令的视频编辑中的上下文生成。具体而言，我们研究了如何在扩散训练中融入编辑区域与非编辑区域之间的区域约束建模。为实现这一构想，ReCo 根据自然语言指令，对按宽度拼接的源视频-目标视频对进行联合去噪，并采用两项正则化，以强调单步反向去噪潜变量和注意力图中的区域间关系。为减少非编辑区域中的意外内容生成，潜在空间中的正则化项旨在缩小源视频与目标视频之间非编辑区域的潜变量差异，同时增大编辑区域的差异。与此同时，ReCo 会抑制编辑区域中的词元对源视频相同部分中词元的注意力，从而减轻原编辑区域词元对新对象生成的干扰。此外，我们精心构建了高质量视频编辑数据集 ReCo-Data，其中包含 50万组指令-视频对，覆盖范围广泛的编辑任务。在四种编辑任务上开展的大量实验验证了 ReCo 相较于最先进方法的优越性。

参考文献

[1] Gemini Team Google：Rohan Anil、Sebastian Borgeaud、Jean-Baptiste Alayrac、Jiahui Yu、Radu Soricut、Johan Schalkwyk、Anja Hauth、Andrew M. Dai、Katie Millican 等。Gemini：功能强大的多模态模型系列。arXiv 预印本 arXiv:2312.11805，2023。2、5、6、1、4[2] Qingyan Bai、Qiuyu Wang、Hao Ouyang、Yue Yu、Hanlin Wang、Wen Wang、Shuailei Ma、Ka Leong Cheng、Yanhong Zeng、Zichen Liu、Yinghao Xu、Yujun Shen 和 Qifeng Chen。利用高质量合成数据集扩展基于指令的视频编辑。arXiv:2510.15742，2025。3、6[3] Shuai Bai、Yuxuan Cai、Ruizhe Chen、Keqin Chen、Xionghui Chen、Zesen Cheng、Lianghao Deng、Wei Ding、Chang Gao、Chunjiang Ge 等。Qwen3-VL 技术报告。arXiv:2511.21631，2025。2[4] Shuai Bai、Keqin Chen、Xuejing Liu、Jialin Wang、Wenbin Ge、Sibo Song、Kai Dang、Peng Wang、Shijie Wang、Jun Tang 等。Qwen2.5-VL 技术报告。arXiv 预印本 arXiv:2502.13923，2025。1[5]Fan Bao、Chendong Xiang、Gang Yue、Guande He、Hongzhou Zhu、Kaiwen Zheng、Min Zhao、Shilong Liu、Yaole Wang 和 Jun Zhu。Vidu：采用扩散模型的高一致性、动态且功能强大的文生视频生成器。arXiv 预印本 arXiv:2405.04233，2024。2、3[6] Omer Bar-Tal、Hila Chefer、Omer Tov、Charles Herrmann、Roni Paiss、Shiran Zada、Ariel Ephrat、Junhwa Hur、Guanghui Liu、Amit Raj、Yuanzhen Li、Michael Rubinstein、Tomer Michaeli、Oliver Wang、Deqing Sun、Tali Dekel 和 Inbar Mosseri。Lumiere：用于视频生成的时空扩散模型。arXiv 预印本 arXiv:2401.12945，2024。3[7] Yuxuan Bian、Zhaoyang Zhang、Xuan Ju、Mingdeng Cao、Liangbin Xie、Ying Shan 和 Qiang Xu。VideoPainter：借助即插即用上下文控制实现任意长度的视频修复与编辑。载于 CVPR，2025。4、1[8] Andreas Blattmann、Robin Rombach、Huan Ling、Tim Dockhorn、Seung Wook Kim、Sanja Fidler 和 Karsten Kreis。对齐你的潜变量：利用潜在扩散模型进行高分辨率视频合成。载于 CVPR，2023。3[9] Tim Brooks、Aleksander Holynski 和 Alexei A. Efros。InstructPix2Pix：学习遵循图像编辑指令。载于 CVPR，2023。2[10] Tim Brooks、Bill Peebles、Connor Holmes、Will DePue、Yufei Guo、Li Jing、David Schnurr、Joe Taylor、Troy Luhman、Eric Luhman、Clarence Ng、Ricky Wang 和 Aditya Ramesh。作为世界模拟器的视频生成模型。2024。3[11] Qi Cai、Jingwen Chen、Yang Chen、Yehao Li、Fuchen Long、Yingwei Pan、Zhaofan Qiu 等。HiDream-I1：高效图像生成基础模型，采用稀疏扩散变换器。arXiv 预印本 arXiv:2505 22705，2025。2。[12] Jiaxin Cheng、Tianjun Xiao 和 Tong He。使用合成数据集实现一致的视频到视频迁移。载于 ICLR，2024。3、6

[13] christophschuhmann。improved-aesthetic-predictor。improved-aesthetic-predictor 实验室，2024。1[14] Yuren Cong、Mengmeng Xu、Christian Simon、Shoufa Chen、Jiawei Ren、Yanping Xie、Juan-Manuel Perez-Rua、Bodo Rosenhahn、Tao Xiang 和 Sen He。FLATTEN：用于一致性文本到视频编辑的光流引导注意力。载于 ICLR，2024。2、3[15] Cheng Cui、Ting Sun、Manhui Lin、Tingquan Gao、Yubo Zhang、Jiaxuan Liu、Xueqing Wang、Zelun Zhang、Changda Zhou、Hongen Liu、Yue Zhang、Wenyu Lv、Kui Huang、Yichao Zhang、Jing Zhang、Jun Zhang、Yi Liu、Dianhai Yu 和 Yanjun Ma。PaddleOCR 3.0 技术报告。arXiv 预印本 arXiv:2507.05595，2025。1[16] PySceneDetect 开发者。PySceneDetect。2024。1[17] Patrick Esser、Sumith Kulal、Andreas Blattmann、Rahim Entezari、Jonas M¨uller、Harry Saini、Yam Levi、Dominik Lorenz、Axel Sauer、Frederic Boesel 等。用于高分辨率图像合成的整流流变换器扩展。载于 ICML，2024。2、4[18] Michal Geyer、Omer Bar-Tal、Shai Bagon 和 Tali Dekel。TokenFlow：用于一致性视频编辑的一致性扩散特征。载于 ICLR，2024。2、3[19] Yoav HaCohen、Nisan Chiprut、Benny Brazowski、Daniel Shalem、Dudu Moshe、Eitan Richardson、Eran Levin、Guy Shiran、Nir Zabari、Ori Gordon、Poriya Panet、Sapir Weiss-buch、Victor Kulikov、Yaki Bitterman、Zeev Melumian 和 Ofir Bibi。LTX-Video：实时视频潜在扩散。arXiv 预印本 arXiv:2501.00103，2025。2、3[20] Amir Hertz、Ron Mokady、Jay Tenenbaum、Kfir Aberman、Yael Pritch 和 Daniel Cohen-or。使用交叉注意力控制的提示到提示图像编辑。载于 ICLR，2023。[21] Jonathan Ho、Ajay Jain 和 Pieter Abbeel。去噪扩散概率模型。载于 NeurIPS，2020。2[22] Jonathan Ho、William Chan、Chitwan Saharia、Jay Whang、Ruiqi Gao、Alexey Gritsenko、Diederik P. Kingma、Ben Poole、Mohammad Norouzi、David J. Fleet 和 Tim Sal-imans。Imagen Video：使用扩散模型进行高清视频生成。载于 CVPR，2022。3[23] Hanzhe Hu、Jinshi Cui 和 Liwei Wang。用于语义分割的区域感知对比学习。载于 ICCV，2021。2[24] Teng Hu、Zhentao Yu、Zhengguang Zhou、Sen Liang、Yuan Zhou、Qin Lin 和 Qinglin Lu。HunyuanCustom：用于定制视频生成的多模态驱动架构。arXiv 预印本 arXiv:2505.04512，2025。2、4[25] Lianghua Huang、Wei Wang、Zhi-Fan Wu、Yupeng Shi、Huanzhang Dou、Chen Liang、Yu Liu、Yutong Feng 和 Jingren Zhou。用于扩散变换器的上下文内 LoRA。arXiv 预印本 arxiv:2410.23775，2024。2、3、4[26] Zeyinzi Jiang、Zhen Han、Chaojie Mao、Jingfeng Zhang、Yulin Pan 和 Yu Liu。VACE：一体化视频创作与编辑。载于 ICCV，2025。2、4、5、6、8、1、3[27] Zhao Jixin、Zhou Shangchen、Wang Zhouxia、Yang Peiqing 和 Loy Chen Change。ObjectClear：通过物体-效果注意力实现完整对象移除。arXiv 预印本 arXiv:2505.22636，2025。1

[28] Ozgur Kara、Bariscan Kurtkaya、Hidir Yesiltepe、James M. Rehg 和 Pinar Yanardag。RAVE：使用扩散模型，通过随机噪声洗牌实现快速且一致的视频编辑。载于 CVPR，2024。2、3[29] Diederik P Kingma 和 Max Welling。自动编码变分贝叶斯。arXiv 预印本 arXiv:1312.6114，2013。6[30] Weijie Kong、Qi Tian、Zijian Zhang、Rox Min、Zuozhuo Dai、Jin Zhou、Jiangfeng Xiong、Xin Li、Bo Wu、Jianwei Zhang、Kathrina Wu、Qin Lin、Junkun Yuan、Yanxin Long、Aladdin Wang、Andong Wang 等。混元视频：大规模视频生成模型的系统化框架。arXiv 预印本 arXiv:2412.03603，2024。2、3[31] Max Ku、Cong Wei、Weiming Ren、Harry Yang 和 Wenhu Chen。AnyV2V：适用于各种视频到视频编辑任务的免调优框架。载于 TMLR，2024。2[32]Vladimir Kulikov、Matan Kleiner、Inbar Huberman-Spiegelglas 和 Tomer Michaeli。FlowEdit：使用预训练流模型进行无反演文本编辑。载于 ICCV，2025。2、3[33] Black Forest Labs、Stephen Batifol、Andreas Blattmann、Frederic Boesel、Saksham Consul、Cyril Diagne、Tim Dockhorn、Jack English、Zion English、Patrick Esser、Sumith Kulal 等。FLUX.1 Kontext：潜在空间中的上下文内图像生成与编辑的流匹配。arXiv:2506.15742，2025。2[34] Guangzhao Li、Yanming Yang、Chenxi Song 和 Chi Zhang。FlowDirector：用于精准文本到视频编辑的免训练流引导。arXiv:2506.05046，2025。3[35] Xirui Li、Chao Ma、Xiaokang Yang 和 Ming-Hsuan Yang。VidToMe：用于零样本视频编辑的视频令牌合并。载于 CVPR，2024 2, 3 。[36] Yaron Lipman、Ricky T. Q. Chen、Heli Ben-Hamu、Maximilian Nickel 和 Matt Le。生成建模的流匹配。载于 ICLR，2023。2、4[37] Kai Liu、Wei Li、Lai Chen、Shengqiong Wu、Yanhao Zheng、Jiayi Ji、Fan Zhou、Rongxin Jiang、Jiebo Luo、Hao Fei 和 Tat-Seng Chua。JavisDiT：具有分层时空先验同步机制的联合音视频扩散变换器。arXiv 预印本 arXiv:2503.23377，2025。2[38] Shaoteng Liu、Yuechen Zhang、Wenbo Li、Zhe Lin 和 Jiaya Jia。Video-P2P：通过交叉注意力控制进行视频编辑。载于 CVPR，2024。2、3[39] Shaoteng Liu、Tianyu Wang、Jui-Hsien Wang、Qing Liu、Zhifei Zhang、Yijun Li、Joon-Young Lee、Bei Yu、Zhe Lin、Soo Ye Kim 和 Jiaya Jia。生成式视频传播。载于 CVPR，2025。3[40]Fuchen Long、Zhaofan Qiu、Ting Yao 和 Tao Mei。VideoStudio：生成内容一致的多场景视频。载于 ECCV，2024。2[41] Yue Ma、Yingqing He、Xiaodong Cun、Xintao Wang、Siran Chen、Ying Shan、Xiu Li 和 Qifeng Chen。Follow Your Pose：利用无姿态视频进行姿态引导的文本到视频生成。载于 AAAI，2023。2[42] Alex 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逼真图像生成与编辑。载于 ICML，2022。2

[43] Ravi Nikhila、Gabeur Valentin、Hu Yuan-Ting、Hu Rong-hang、Ryali Chaitanya、Ma Tengyu、Khedr Haitham、R”adle Roman、Rolland Chloe、Gustafson Laura、Mintun Eric、Pan Junting、Alwala Kalyan Vasudev、Carion Nicolas、Wu Chao-Yuan、Girshick Ross、Doll’ar Piotr 和 Feichtenhofer Christoph。《SAM 2：图像与视频中的万物分割》。载于 ICLR，2025。1[44]Dustin Podell、Zion English、Kyle Lacey、Andreas Blattmann、Tim Dockhorn、Jonas M¨uller、Joe Penna 和 Robin Rombach。《SDXL：改进潜在扩散模型，实现高分辨率图像合成》。载于 ICLR，2024。2[45] Chenyang Qi、Xiaodong Cun、Yong Zhang、Chenyang Lei、Xintao Wang、Ying Shan 和 Qifeng Chen。《FateZero：融合注意力，实现零样本文本视频编辑》。载于 ICCV，2023。2, 3[46] Ren´e Ranftl、Katrin Lasinger、David Hafner、Konrad Schindler 和 Vladlen Koltun。《迈向稳健的单目深度估计：混合数据集实现零样本跨数据集迁移》。IEEE TPAMI，2020。2[47] Tianhe Ren、Shilong Liu、Ailing Zeng、Jing Lin、Kunchang Li、He Cao、Jiayu Chen、Xinyu Huang、Yukang Chen、Feng Yan、Zhaoyang Zeng、Hao Zhang、Feng Li、Jie Yang、Hongyang Li、Qing Jiang 和 Lei Zhang。《Grounded SAM：整合开放世界模型，服务多样化视觉任务》。arXiv 预印本 arXiv:2401.14159，2024。1[48]Robin Rombach、Andreas Blattmann、Dominik Lorenz、Patrick Esser 和 Bj¨orn Ommer。《利用潜在扩散模型实现高分辨率图像合成》。载于 CVPR，2022。2[49]Chitwan Saharia、William Chan、Saurabh Saxena、Lala Li、Jay Whang、Emily Denton、Seyed Kamyar Seyed Ghasemipour、Burcu Karagol Ayan、S. Sara Mahdavi、Rapha Gontijo Lopes、Tim Salimans、Jonathan Ho、David J Fleet 和 Mohammad Norouzi。《具备深度语言理解能力的逼真文本到图像扩散模型》。载于 NeurIPS，2022。2[50] Shelly Sheynin、Adam Polyak、Uriel Singer、Yuval Kirstain、Amit Zohar、Oron Ashual、Devi Parikh 和 Yaniv Taigman。《Emu Edit：通过识别与生成任务实现精准图像编辑》。载于 CVPR，2024。2[51] Chaehun Shin、Jooyoung Choi、Heeseung Kim 和 Sungroh Yoon。《具备图像修复能力的大规模文本到图像模型是一种零样本主体驱动图像生成器》。载于 CVPR，2025。4[52] Uriel Singer、Adam Polyak、Thomas Hayes、Xi Yin、Jie An、Songyang Zhang、Qiyuan Hu、Harry Yang、Oron Ashual、Oran Gafni、Devi Parikh、Sonal Gupta 和 Yaniv Taig-man。《Make-a-Video：无需文本视频数据的文本到视频生成》。载于 ICLR，2023。3[53] Jiaming Song、Chenlin Meng 和 Stefano Ermon。《去噪扩散隐式模型》。载于 ICLR，2021。3[54] spaCy 开发者。spaCy。2024。1[55] Zhenxiong Tan、Songhua Liu、Xingyi Yang、Qiaochu Xue 和 Xinchao Wang。《OminiControl：用于扩散变换器的极简通用控制》。载于 ICCV，2025。4[56] DecartAI 团队。《Lucy Edit：开放权重文本引导视频编辑》。2025 3, 6 。[57] Zachary Teed 和 Jia Deng。《RAFT：用于光流的循环全对场变换》。载于 ECCV，2020。1

[58] Wan团队、Ang Wang、Baole Ai、Bin Wen、Chaojie Mao、Chen-Wei Xie、Di Chen、Feiwu Yu、Haiming Zhao、Jianxiao Yang 等。《Wan：开放且先进的大规模视频生成模型》。arXiv 预印本 arXiv:2503.20314，2025。2, 3, 4, 6[59] Wenjing Wang、Huan Yang、Zixi Tuo、Huiguo He、Junchen Zhu、Jianlong Fu 和 Jiaying Liu。《VideoFactory：在时空扩散中交换注意力以实现文本到视频生成》。arXiv 预印本 arXiv:2305.10874，2023。1[60] Cong Wei、Zheyang Xiong、Xinrun Du、Weiming Ren、Ge Zhang 和 Wenhu Chen。《OmniEdit：通过专家监督构建图像编辑通用模型》。载于 ICLR，2025。6, 2[61] Chenfei Wu、Jiahao Li、Junyang Lin、Jingren Zhou、Kaiyuan Gao、Kun Yan、Sheng ming Yin、Shuai Bai、Xiao Xu、Yilei Chen、Yuxiang Chen 等。《Qwen-Image 技术报告》。arXiv:2508.02324，2025。2[62] Jay Zhangjie Wu、Yixiao Ge、Xintao Wang、Weixian Lei、Yuchao Gu、Yufei Shi、Wynne Hsu、Ying Shan、Xiaohu Qie 和 Mike Zheng Shou。《Tune-A-Video：对图像扩散模型进行一次性调优以实现文本到视频生成》。载于 ICCV，2023。3[63] Yuhui Wu、Liyi Chen、Ruibin Li、Shihao Wang、Chenxi Xie 和 Lei Zhang。《InsViE-1M：通过精细的数据集构建实现有效的基于指令的视频编辑》。载于 ICCV，2025。3, 6[64] Shitao Xiao、Yueze Wang、Junjie Zhou、Huaying Yuan、Xingrun Xing、Ruiran Yan、Chaofan Li、Shuting Wang、Tiejun Huang 和 Zheng Liu。《OmniGen：统一图像生成》。载于 CVPR，2025。2[65] Zhuoyi Yang、Jiayan Teng、Wendi Zheng、Ming Ding、Shiyu Huang、Jiazheng Xu、Yuanming Yang、Wenyi Hong、Xiaohan Zhang、Guanyu Feng、Da Yin、Xiaotao Gu、Yuxuan Zhang、Weihan Wang、Yean Cheng、Ting Liu、Bin Xu、Yuxiao Dong 和 Jie Tang。《CogVideoX：采用专家Transformer的文本到视频扩散模型》。载于 ICLR，2025。2, 3[66]Shenghai Yuan、Xianyi He、Yufan Deng、Yang Ye、Jinfa Huang、Bin Lin、Jiebo Luo 和 Li Yuan。《OpenS2V-Nexus：主体到视频生成的详细基准测试与百万级数据集》。arXiv 预印本 arXiv:2505.20292，2025。1[67] Lvmin Zhang、Anyi Rao 和 Maneesh Agrawala。《为文本到图像扩散模型添加条件控制》。载于 ICCV，2023。2[68] Yifu Zhang、Hao Yang、Yuqi Zhang、Yifei Hu、Fengda Zhu、Chuang Lin、Xiaofeng Mei、Yi Jiang、Bingyue Peng 和 Ze-huan Yuan。《Waver：挥动画笔，生成栩栩如生的视频》。arXiv 预印本 arXiv:2508.15761，2025。3[69] Zicheng Zhang、Bonan Li、Xuecheng Nie、Congying Han、Tiande Guo 和 Luoqi Liu。《利用文本到图像扩散模型实现连贯的视频编辑》。载于 NeurIPS，2023。2[70] Zhongwei Zhang、Fuchen Long、Yingwei Pan、Zhaofan Qiu、Ting Yao、Yang Cao 和 Tao Mei。《TRIP：利用图像噪声先验进行时间残差学习，以实现图像到视频的扩散模型》。载于 CVPR，2024。2

[71] Zhongwei Zhang、Fuchen Long、Zhaofan Qiu、Yingwei Pan、Wu Liu、Ting Yao 和 Tao Mei。MotionPro：用于图像到视频生成的精确运动控制器。发表于 *CVPR*，2025 年。 2[72] Zechuan Zhang、Ji Xie、Yu Lu、Zongxin Yang 和 Yi Yang。借助大规模扩散变换器中的上下文生成实现指令式图像编辑。发表于 *NeurIPS*，2025 年。 2，3、4[73] Zechuan Zhang、Ji Xie、Yu Lu、Zongxin Yang 和 Yi Yang。上下文内编辑：借助大规模扩散变换器中的上下文生成实现指令式图像编辑。发表于 *NeurIPS*，2025 年。 2[74] Rui Zhu、Yingwei Pan、Yehao Li、Ting Yao、Zhenglong Sun、Tao Mei 和 Chang Wen Chen。SD-DiT：释放扩散变换器中自监督判别的力量。发表于 *CVPR*，2025 年。 2[75] Bojia Zi、Penghui Ruan、Xianbiao Qi Marco Chen、Shaozhe Hao、Shihao Zhao、Youze Huang、Bin Liang、Rong Xiao 和 Kam-Fai Wong。Se˜norita-2M：由视频专家打造的通用视频编辑高质量指令数据集。发表于*NeurIPS*，2025 年。 3，6

用于指令式视频编辑的区域约束上下文生成

— 补充材料

补充材料包含：1）ReCo-Data 的构建流程；2）基于 VLLM 的评估基准测试的详细信息；3）基线方法和 ReCo 的实现细节；4）ReCo 的泛化能力。

1. ReCo-Data 的构建流程

尽管指令式视频编辑近年来取得了显著进展，但缺乏大规模、高质量的训练数据集仍是一大关键障碍。为此，我们推出了**ReCo-Data**，这是一个经过精心设计的数据集，旨在支持四项关键编辑任务：实例级对象添加、移除、替换以及全局视频风格化。

如图所示 7，ReCo-Data 的构建流程包含六个主要阶段：（1）原始数据预处理，根据特定质量标准筛选原始视频数据；（2）对象分割，从视频中提取对象掩码；（3）指令生成，使用 VLLM（即 Gemini-2.5-Flash-Thinking [1]）构建编辑提示；（4）条件对构建，通过首帧编辑和深度图生成来准备 VACE 所需的输入条件 [26]；（5）视频合成，使用 VACE 根据条件生成视频；（6）视频筛选和重新生成描述，再次借助 VLLM（即 Gemini-2.5-Flash-Thinking [1]）筛除低质量样本，并为保留的视频重新生成描述。

1.1. 原始数据预处理

数据收集。 为确保数据多样性，我们从多个来源收集原始视频，包括 HD-VG [59],OpenS2V-Nexus [66]，以及 Pixel 网站 [7] 上的视频。我们使用 PySceneDetect [16] 将较长的多场景视频分割成较短、便于处理的片段。

数据筛选。 我们首先根据基本元数据筛选片段，保留时长超过 5 秒、帧率高于 24 fps 且分辨率至少为 720P 的片段。随后，我们利用美学评分 [13] 和光流 [57]，筛选出美学质量高且运动幅度适宜的视频。最后，为确保视觉内容纯净，我们使用 PaddleOCR [15] 检测水印，并对帧进行空间裁剪，以排除所有置信度超过 0.7 的检测文本。

视频描述生成。 为了后续进行对象分割和构建编辑提示词，我们使用 Qwen2.5-VL- 32B [4] 获取视频中保留内容的详细描述。

1.2. 对象分割

为了利用 VACE 等视频修复模型 [26], 实现精确的实例级对象编辑（例如替换和移除），我们需要先分离目标对象。鉴于包含多个对象的场景较为复杂，我们采用系统化的分割方法。首先，我们定义分类体系，并使用命名实体识别（NER）模型 SpaCy [54], 从视频描述中提取相关的实体名词。随后，我们利用 Grounding Dino [47] 检测对象并获取其边界框。为确保候选框的质量，我们采用非极大值抑制（NMS）过滤重叠度超过 25% 的重复框，并剔除比例过大或过小的框。最后，我们将边界框作为提示词，使用 SAM 2 [43]为目标对象生成掩码序列。

1.3. 指令生成

用于准备编辑提示的流程，会因局部编辑和全局视频风格化任务而有所不同。

局部编辑。 对于局部视频编辑任务，我们向 Gemini 提供一个二元组，其中包含原始视频描述和一张具有代表性的关键帧。在这张关键帧中，目标对象区域通过红色凸包明确标出。在经过精细调优的系统提示引导下，Gemini 需要生成恰当的编辑指令，以及描述编辑后状态的目标视频描述。

视频风格化。 该过程与局部编辑类似。我们利用 Gemini 的创造力，集思广益，构思多样的风格化指令，并生成相应的目标视频描述。

1.4. 条件对构建

在此阶段，我们充分发挥现有模型的能力，构建最优条件对，并将其输入 VACE [26] 以进行编辑视频生成。针对每项编辑任务，我们采用特定策略来生成条件。

对象移除。 将掩码视频（由对象掩码生成）和目标提示输入 VACE 进行编辑视频生成，往往无法干净地移除对象，或会产生意料之外内容的幻觉。为缓解这一问题，我们采用两阶段方法进行编辑视频生成。首先，我们在首帧上使用 ObjectClear [27]，干净地移除对象。接着，将编辑后的帧与后续掩码视频帧拼接，再输入 VACE 进行视频修复，从而实现稳定、高质量的对象移除。

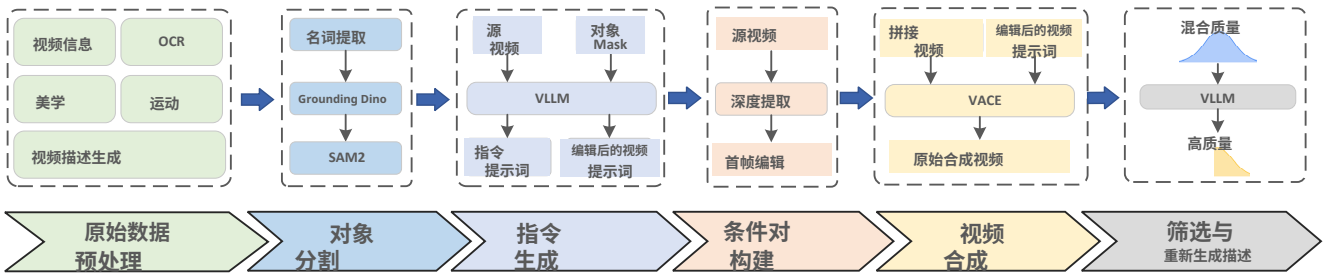


图 7. 数据构建流程概览。该流程包含六个主要阶段：原始数据预处理、对象分割、指令生成、条件对构建、视频合成，以及视频筛选和重新生成描述。

对象添加。 我们将此任务视为对象移除的逆过程。生成有效的移除视频对后，只需交换源视频与编辑后的视频，即可创建相应的对象添加任务训练对。

对象替换。VACE 在对象替换方面表现稳健。因此，我们只需将遮罩视频序列和目标视频提示输入 VACE，即可生成高质量的替换结果。

视频风格化。尽管 VACE 支持以深度图为条件进行视频风格化，但它在保持原始视频内容结构方面的表现并不理想。因此，我们采用了类似于对象移除的策略。首先，我们使用 FLUX.1 Kontext [33] 对首帧进行风格迁移。随后，我们将编辑后的帧与深度图序列（通过 MiDaS [46]提取）拼接起来，作为 VACE 的输入条件对，从而生成时间一致的风格化视频。

我们的流程专为视频风格化而设计，旨在解决 VACE 生成数据中的常见伪影，例如帧崩塌、突变、时间不一致和内容失真（如面部变形）。为确保输出质量，我们采用两阶段策略。首先，我们使用 Qwen-3-VL-Instant [3] 筛选流畅稳定的视频，剔除存在严重伪影或闪烁的低质量帧。其次，我们使用 14B Wan-2.2-T2V [58] 模型对筛选出的视频进行优化。结合这些策略，我们能够合成视觉和时间质量显著提升的风格化视频。

1.5. 视频合成

准备好所需的条件对后，我们便以大规模批次运行 VACE，合成高质量的编辑视频。为最大限度地发挥合成数据的效用并确保高效构建，我们设计了一种数据增强策略，无需额外计算成本即可生成更多训练对。

可逆替换。 对于基于单个源视频进行的对象替换，我们将这一过程视为可逆的。通过交换源视频与目标视频，我们实际上将替换数据量扩大了一倍。

跨任务增强。 事实上，对象移除和替换生成的编辑后的视频共享相同的干净背景。因此，可以将替换任务合成的视频（其中包含一个新对象）与移除任务的背景视频配对。这样，我们就能构建新的“移除”配对（新对象→背景）和“添加”配对（背景 → 新对象），从而使这两项任务的数据集规模都扩大一倍。

最终，我们为四项编辑任务共构建了约 800K 个视频对。整个数据合成过程在 NVIDIARTX 4090 上耗费约 **76,800 GPU小时** 。

1.6. 视频筛选与重新生成描述

为确保指令-视频对的高质量，我们使用 VLLM（即 Gemini-2.5-Flash-Thinking）对总计 800K 个视频对进行评估，并筛除低质量样本。我们从源视频和编辑后的视频中提取具有代表性的关键帧，并将其拼接为并排布局，以便 VLLM 进行评估。剩余的视频对由 VLLM 重新生成描述。整个描述生成流程（包括第 1.3节）总计花费约$13，600。最终，我们构建了 ReCo-Data，其中包含 500K 个高质量指令-视频对。每个视频片段包含 81帧，分辨率为 480×832 ，时长为 5秒。

2. 基于视觉大语言模型的评测基准

传统的视频生成指标往往难以准确评估视频编辑的保真度和质量。受近期图像编辑评测方案的启发 [60], , 我们提出了一种基于视觉大语言模型的评测基准，以全面、有效地评估视频编辑质量。

测试数据。 我们收集了 480 组视频-指令对作为测试数据，平均分配给四项任务，每项任务各 120 组（即物体添加、移除、替换和视频风格化）。所有源视频均收集自 Pexels 视频平台。对于局部编辑任务（即物体添加、移除和替换），我们利用 Gemini-2.5-Flash-Thinking [1]，根据视频内容构思并生成多样化的编辑指令。为了对视频风格化进行严格评估，

我们随机选取 10 个源视频，并为每个视频应用 12 种不同的风格，从而得到 120 组评估数据。

评估指标。 以往基于图像的指标主要关注编辑准确性和静态生成质量，而视频编辑的评估则更为复杂。为此，我们构建了一套多样化的评估维度，专门针对视频进行设计。图 8展示了为 VLLM 设计的相应系统提示，该提示从三个主要方面评估性能，共包含九个子维度：

•**编辑准确性** (S_{EA})：评估结果与指令的契合程度。– 语义准确性（SA）：编辑后的视频是否正确遵循了文本指令的语义？– 范围精确度（SP）：编辑是否严格局限于目标区域，且未影响背景？– 内容保留（CP）：未编辑区域或原始细节是否得到了忠实保留？（对于风格化处理，这对应于结构保留。）

• **视频自然度** (S_{VN})：评估生成内容的真实感与连贯性。– 外观自然度（ AN ）：编辑后的视频在光照、纹理和色彩方面是否自然？– 尺度自然度（SN）：编辑后对象相对于环境的大小和比例是否合理？（对于风格化处理，此项用于评估风格化对象是否变得过大。）– 运动自然度（MN）：编辑后对象的运动（或风格渲染）是否符合物理上合理的动态规律？

• **视频质量** (S_{VQ})：评估编辑后视频的基本视觉质量。– 视觉保真度（ VF ）：视频是否清晰、锐利且没有视觉伪影？– 时间稳定性（TS）：视频在各帧之间是否没有闪烁或抖动？– 编辑稳定性（ES）：编辑内容在整个视频时长内是否始终保持一致的身份和外观？

VLLM 会对每个子维度的得分进行 0 到 10 的评分。随后，我们通过计算各自子维度得分的几何平均值，得到类别得分（即 S_{EA}、S_{VN}、 S_{VQ}）：

$$S_{EA} = \sqrt[3]{SA \cdot SP \cdot CP}, \tag{18}$$

$$S_{VN} = \sqrt[3]{AN \cdot SN \cdot MN}, \tag{19}$$

$$S_{VQ} = \sqrt[3]{VF \cdot TS \cdot ES}. \tag{20}$$

最后，总分 S 计算为以下三项类别得分的算术平均数：

$$S = \frac{1}{3}\left(S_{EA} + S_{VN} + S_{VQ}\right). \tag{21}$$

3. 基线方法与 ReCo 的实现

基线设置。 对于近期的视频编辑进展，能够同时处理全部四种编辑任务的方法为数不多。下面，我们说明基线方法的选择标准。对于**对象添加、替换和视频风格化**，我们以 InsViE [63]、Lucy-Edit [56]，和 Ditto [2] 为基准测试对象。由于 InsViE 的输入限制为 480×720 分辨率下的 49 帧，我们通过均匀时间下采样和空间尺寸调整来适配测试视频，以符合其要求。对于**对象移除**，有效的指令式基线方法十分有限。为便于进行有意义的比较，我们将 VACE [26] 纳入作为额外的基线方法。与基于指令的方法不同，VACE 需要显式提供对象掩码和目标视频提示才能执行移除。请注意，VACE 在此实现中表现出一定的不稳定性。

ReCo 的实现细节。ReCo 基于 Wan [58]架构构建，并使用 AdamW 优化器在 ReCo-Data 上进行训练。我们采用两阶段学习率计划：首先以1×10^{-4} 的学习率进行训练，以确保稳定收敛；随后以 2×10^{-5} 的学习率进行微调，以实现精确优化。关于损失权重，潜变量约束和注意力约束的初始值通常在 $[-1, 1]$范围内，而流匹配的均方误差损失约为 0。03。为平衡梯度的影响，我们将各区域约束损失的幅度缩放至均方误差损失的大约 $0.1\times$ 。因此，我们将权重系数（λ_1和 λ_2）设为 1×10^{-3}。所有实验均在一个由 24 块 NVIDIA A800 GPU 组成的集群上进行，迷你批次总大小为 24，训练耗时约 10 天。

4. ReCo 的泛化能力

有趣的是，如图9所示，我们观察到 ReCo 能够泛化到抽象和创意编辑任务。例如，它成功地在女性头上合成了光环，生成了层叠彩纸效果，在一名男子头旁放置了一个“灵感灯泡”，并制作了从计算机中冒出的烟雾。我们认为，ReCo 之所以具备这种泛化能力，是因为它有效地继承并利用了预训练视频扩散模型丰富的先验知识。

人类：
你是一位专业的数字艺术家和视频质量评估员。你的任务是从三个主要类别评估 AI生成的视频编辑：编辑准确性、视频质量和自然度。你将获得用于创建编辑的文本指令，以及并排展示的视频关键帧，其中左侧显示原始视频，右侧显示编辑版本。你必须仅按照以下 JSON 格式提供输出。不要输出其他任何内容。

{
"编辑_准确性": {"scores": [1, 1, 1], "推理": "..."},
" 视频 _ 质量 ": {"scores": [1, 1, 1], " 推理 ": "..."}, "
自然度": {"scores": [1, 1, 1], "推理": "..."} }

每条推理说明都应简洁明了，概括该类别的评分。

类别 1：编辑准确性
此类别评估 AI 对文本指令的理解和执行效果。编辑_准确性的评分列表包含三项评分：**[评分_SA，评分_SP，评分_CP]**。

- 评分 SA：语义准确性（范围：1-10）。评估编辑的核心概念是否正确（例如，添加、移除、替换或进行风格化处理的内容）。1 表示核心概念完全错误。10 表示核心概念与指令完全一致。- 评分 SP：范围精确度（范围：1-10）。评估编辑的位置、区域或范围是否正确（例如，编辑应用的位置）。1 表示位置/区域完全错误。10 表示编辑的局部或全局范围与指令完全一致。
- CP 分数：内容保留（范围：1-10）。评估 AI 是否对不应编辑的区域造成了负面影响。1 表示未编辑区域严重失真或发生变化。10 表示指定编辑范围之外的所有内容都得到了完美保留。对于全局风格化，此项评估底层结构是否得到保留。

类别 2：视频质量
此类别评估编辑后的视频在技术保真度和稳定性方面的表现。视频_质量的评分列表包含三项分数：**[分数_VF 分数_TS 分数_ES]**。

- VF 分数：视觉保真度（范围：1-10）。评估编辑后帧的整体清晰度，以及静态视觉伪影的存在情况。1 表示视频极其模糊或充满伪影。10 表示视频清晰锐利。
- TS 分数：时间稳定性（像素级）（范围：1-10）。评估视频随时间推移的低层级一致性，重点关注闪烁或沸腾纹理。1 表示视频极不稳定。10 表示视频随时间推移完全稳定。
- ES 分数：编辑效果持久性（语义层面）（评分范围：1-10）。评估预期的编辑效果（添加、移除、替换、风格）是否从头到尾保持稳定并正确延续。1 表示编辑效果在视频中途失效。10 表示预期的编辑效果始终完美稳定。糟糕的情况包括已移除的对象重新出现，或添加的对象消失。

类别 3：视觉自然度
此类别评估编辑效果是否合理且融合自然。自然度评分包括以下三项：**[分数_AN 分数_SN 分数_MN]**。

- AN 分数：外观自然度（融合度）（评分范围：1-10）。评估新增部分与原场景的光照和阴影融合得是否自然。1 表示编辑效果看起来虚假，像是贴上去的。10 表示编辑效果融合得完美无缺。
- SN 分数：空间上下文比例（评分范围：1-10）。评估编辑后对象的尺寸是否合理。1 表示比例不合逻辑。10 表示尺寸糟糕的情况包括移除后出现巨型宠物或被放大的背景纹理。
- 评分 MN：运动自然度（物理定律）（量表：1-10）。评估编辑是否遵循基本物理规律并进行合理互动。1 表示编辑违反物理规律。10 表示行为在物理上合理。糟糕的情况包括对象违背重力或无法正确跟随运动。

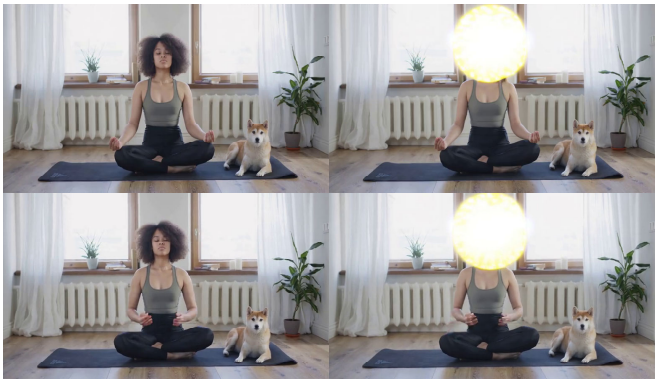
关键规则：编辑失败（相同视频）
如果编辑后的视频（右侧）与原始视频（左侧）相同，则表示彻底失败。必须将全部九项评分设为 0.
失败示例输出：

{
"edit_accuracy": { "scores": [0, 0, 0], "reasoning": "编辑未能应用。视频相同。" },
"video_quality": { "scores": [0, 0, 0], "reasoning": "编辑未能应用。视频相同。" },
"自然度": { "scores": [0, 0, 0], "reasoning": "编辑未能应用。视频相同。" }
}

编辑指令：<编辑指令>
<视频> 源视频 </视频>
<视频> 编辑后的视频 </视频>

助手：

图 8. 输入 Gemini-2.5-Flash-Thinking 的系统提示 [1] 用于视频编辑评估。我们要求 VLLM 从三个主要方面评估四项视频编辑任务，即编辑准确性、视频自然度和视频质量。



在女性头上添加一个闪烁发光的光球



在男性头旁添加一个醒目的灯泡图标



添加从屏幕顶部飘落的彩色五彩纸屑



添加从计算机中升起的烟雾

图9. ReCo 执行指令式视频编辑的四个示例，用于验证其泛化能力。我们的模型在抽象和创意编辑任务中展现出强大的泛化能力。