DynaDrag：基于运动预测的动态拖拽式图像编辑

Jiacheng Sui YujieZhou 上海交通大学 上海交通大学 中国上海 中国上海jcsui01@sjtu.edu.cn yujieouo@sjtu.edu.cnLi Niu 上海交通大学 中国上海ustcnewly@sjtu.edu.cn

arXiv:2601.00542v1 [cs.CV] 2 Jan 2026

摘要

为实现像素级图像操控，使用点或轨迹作为条件进行图像编辑的拖拽式图像编辑正受到广泛关注。大多数以往方法遵循移动与跟踪框架，其中跟踪遗漏和跟踪模糊是不可避免的棘手问题。其他框架下的方法则存在各种问题，例如源图像与目标编辑图像之间存在巨大差距，以及不合理的中间点可能导致可编辑性较低。为避免这些问题，我们提出了 *DynaDrag*，这是首个基于预测并移动框架的拖拽方法。在 *DynaDrag* 中，运动预测和运动监督以迭代方式执行。在每次迭代中，运动预测首先预测操控点应移动到的位置，然后运动监督据此拖拽操控点。我们还提出动态调整有效控制点，以进一步提升性能。在面部和人体数据集上的实验展示了其相较于以往工作的优越性。

1. 引言

生成模型的图像编辑 [2, 8, 11, 31, 37, 39]因其广泛的应用场景而受到广泛关注。近年来，越来越多的研究聚焦于像素级操作 [3, 7, 14, 16, 26, 28, 29, 32, 34–36,42–44, 52]。其中，DragGAN [36] 是最早开展相关研究的工作之一。DragGAN 提出了拖拽式图像编辑的概念：用户提供一张图像，以及成对的操控点和目标点。拖拽式图像编辑的目标是将操控点处的内容“拖拽”到目标点。用户还可以提供掩码来指定可编辑区域，而模型需要保持未被掩码覆盖的区域不变。DragGAN 提出了一个由两个迭代步骤组成的移动与跟踪框架：运动监督和点跟踪。DragDiffusion [42] 在图1(a)中采用了与 DragGAN 相同的框架，但 DragDiffusion 使用扩散模型[40]而不是生成对抗网络。

尽管 DragGAN 和 DragDiffusion 已展现出显著的效果，但它们仍面临图1（b）中的跟踪遗漏以及图1（c）中的跟踪模糊等挑战。在点跟踪步骤中，这些问题会导致图像质量下降。为了解决 DragGAN [36] 和 DragDiffusion[42], FreeDrag [25] 在图 1（d）中提出了一种无需点跟踪的框架，本文称之为搜索-移动框架。在实验过程中，我们观察到，FreeDrag 图像质量的提升是以牺牲图像的可编辑性和编辑精度为代价的，这是由图1（e）中不合理的中间点造成的。

拖拽式图像编辑也有 DragonDiffusion [32]等单步编辑方法。尽管单步方法可以避免点跟踪带来的问题，但如图 1(f) 所示，由于待编辑图像与目标图像之间存在较大差异，这些方法可能会出现可编辑性和真实性较差的问题。鉴于单步拖拽存在这些问题，拖拽式编辑过程应拆分为多个步骤，以降低任务难度。点跟踪被动地确定操控点的新位置，而我们则主动预测中间点的位置。我们认为，光流可以可靠地指示中间点的位置。基于这一认识，我们提出了一种名为“预测并移动”的拖拽式图像编辑新框架，并提出了该框架下的首个方法，它结合了运动预测（MP）和运动监督（MS）。我们的框架如图所示——

图 1(g) 所示。

具体而言，我们的方法执行两个迭代步骤：运动预测和运动监督。在运动预测步骤中，我们根据给定图像以及成对的操控点和目标点，预测操控点的运动，然后相应地进行运动监督步骤。上述框架不仅直接避免了点跟踪带来的棘手问题，提高了中间点的合理性，还使动态调整有效操控点成为可能。通过运动预测，可以确定操控点的运动，从而精确识别相关的中间点。因此，我们能够动态自动选择合适的操控点，以取得更好的效果。在实践中显然并非所有用户提供的操控点都能产生积极作用，因此我们应选择可靠的操控点，以实现更好的图像编辑。

总体而言，我们框架中的运动预测通过预测操控点的“下一步运动”，将拖拽式图像编辑任务分而治之，从而避免了点跟踪中的烦人问题。它能够缩小待编辑图像与目标图像之间的差距，提高中间点的合理性，最终增强图像的可编辑性以及编辑后的真实性。总结而言，我们的主要贡献如下：

- 我们提出了一种用于拖拽式图像编辑的新型预测并移动框架。在该框架下，我们提出了 DynaDrag，它能够预测控制点的下一步运动，并根据预测的运动对其运动进行监督。
- 我们提出动态选择有效控制点，以帮助稳健地完成编辑任务并保持图像质量。
- 大量实验表明，DynaDrag 在拖拽式图像编辑中具有稳定性和优越性。

2. 相关工作

2.1. 扩散模型

扩散模型最早于 [46] 提出。近年来，随着计算能力和数据规模的增长，DDPM 模型 [13] 和 DDIM 模型 [47] 使合成高质量图像成为可能，并将扩散模型引入各种任务 [18, 21, 33, 49, 50]。这些模型通过迭代地对随机噪声进行去噪来合成图像。为提升时间和内存效率，研究者探索了潜扩散模型（LDM）（例如 Stable Diffusion（SD） [40]）。SD 包含一个自编码器和一个用作去噪器的 UNet。自编码器负责在像素空间与潜空间之间转换图像。LDM 的加噪和去噪过程都在潜空间。

2.2. 拖拽式图像编辑

现有的拖拽式图像编辑方法，根据所采用的框架，大致可分为三类。

移动与跟踪框架（图1（a）：DragGAN[36], DragDiffusion [42], StableDrag [5], DragNoise [26] EasyDrag [14], GoodDrag [52] 和 AdaptiveDrag [3]。DragGAN 提出了一种基于 StyleGAN 的拖拽式图像编辑新方法和新思路 [19, 20]。DragGAN 包含两个交替进行的步骤：运动监督和点跟踪。基于生成对抗网络的图像编辑方法受限于生成对抗网络模型的能力 [10, 20]。DragDiffusion [42]首次提出了基于扩散模型的拖拽式图像编辑方法。EasyDrag 和 StableDrag 改进了点跟踪算法，而 DragNoise 和 GoodDrag 则改变了优化目标。AdaptiveDrag[3]通过聚类语义相似的区域来提升编辑效果。

搜索与移动框架（图1（d）：FreeDrag[25]指出了移动与跟踪框架下方法的缺点，可概括为跟踪遗漏和跟踪模糊。为解决这些问题，FreeDrag 将线搜索与回溯相结合，在拖拽过程中更新中间目标，并通过自适应更新模板特征来提升稳定性。

单步框架：DragonDiffusion [32], DragA- Part [23], FastDrag [53], LightningDrag [43], LucidDrag[4]和 InstantDrag [44]。DragonDiffusion 依靠分类器引导，将拖拽式图像编辑操作转换为梯度。FastDrag 将潜在翘曲函数引入此任务，而 LucidDrag 利用大型视觉语言模型和大语言模型取得更好的效果。DragAPart 和 LightningDrag 将编辑点转换为条件，再将其注入扩散过程。InstantDrag[44]训练了一个光流预测模型，并利用预测的光流作为条件，一步生成编辑后的图像。

此外，DiffUHaul [1] 专为拖拽式图像编辑中的物体拖拽而设计。DiffUHaul 主要对注意力机制进行修改和改进。CLIPDrag [16]主要致力于将文本输入与拖拽信号结合起来，以消除编辑过程中的歧义。RegionDrag [28]引入基于区域的编辑，以提高拖拽式图像编辑的准确性并消除歧义。

与上述方法不同，我们提出的预测并移动框架通过多个步骤编辑图像，并主动预测中间点的位置。这种方法简化了编辑过程，使其更加合理。

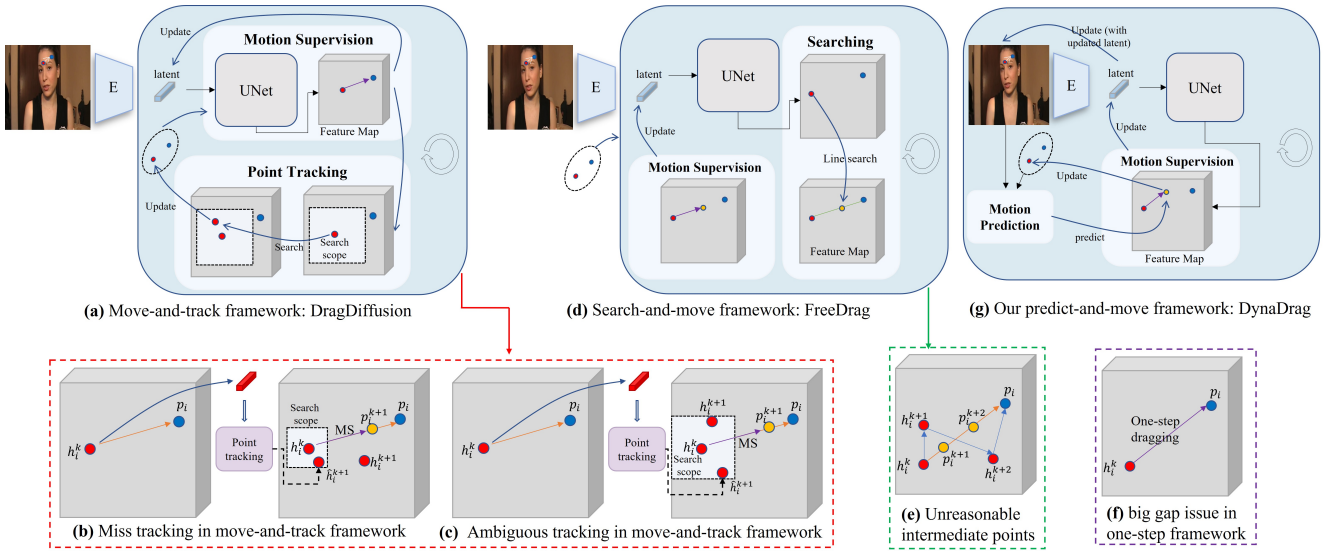


图1. 现有框架及其缺点与我们框架的示意图。为简洁起见，省略了这些框架的细节。(b、c)中的 MS 表示运动监督。 h_i^k 是第 k 次迭代中的第 i个操控点， p_i 表示其对应的目标点。 p_i^{k+1} 是第 k次运动监督迭代中的优化目标点，$\hat{}h_i^{k+1}$ 是通过点跟踪搜索得到的新操控点，其实际位置应为 h_i^{k+1} 。

3. 方法

3.1. 方法概述

我们提出的 DynaDrag 旨在优化特定的扩散潜变量，以实现拖拽式图像编辑；用户选择操控点，并将其拖至各自的目标点。为实现这一目标，我们首先在扩散模型上微调 LoRA [15] ，以重建输入图像，从而在图像编辑过程中更好地保留原始图像内容，正如 [42] 中所述。LoRA 微调后，我们使用 DDIM反演 [47] 对输入图像添加噪声，以获得特定时间步 t的扩散潜变量。接下来，为了优化第 t步的扩散潜变量，我们反复应用图 2(b) 所示的运动预测（MP）和图 2 (c) 所示的运动监督（MS）。我们的方法的详细信息和完整流程如图 2(a) 所示。

为便于描述，在介绍我们的方法之前，先定义一些符号。我们将第 k 次预测并移动迭代中的 n 操控点记为 $\{h_i^k = (x_i^k,\ y_i^k)|_{i=1}^n\}$，其中 $\{h_i^0|_{i=1}^n\}$ 是用户提供的操控点及其对应的目标点，记为 $\{p_i = (x_i,\ y_i)|_{i=1}^n\}$。用户输入图像记为 I。由于运动预测和运动监督是迭代进行的，用户输入图像也可以记为 I_0，表示第一次迭代的输入图像；第 k 次迭代图像的潜在代码记为 z_0^k；通过 DDIM反演得到的 z_0^k 的第 t 步潜在代码记为 z_t^k；将第 t 步潜在代码 z_t^k 作为输入时，最后一个 UNet模块的特征图记为 $F(z_t^k)$。

3.2. 运动预测

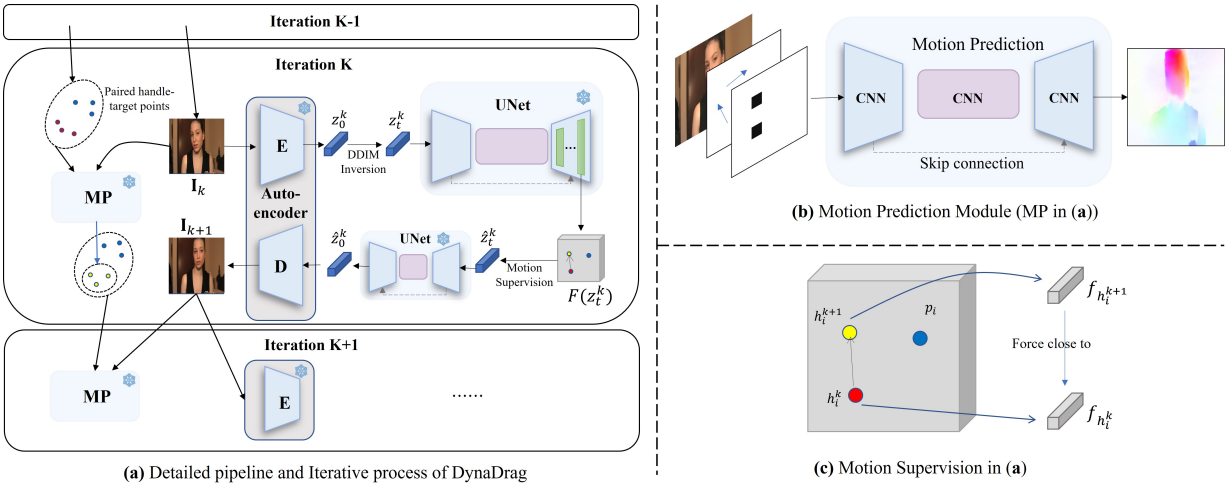
本节将介绍运动预测的具体细节，以及如何动态选择有效的操控点。

3.2.1. 运动预测的目标与训练细节。

我们将把操控点拖动到目标点的过程视为一系列子过程，具体而言，操控点需要经过多个中间点运动，才能到达指定的目标点。运动预测旨在根据第 k次迭代的图像 I_k （由用户提供I_0 ）、第 k次迭代的操控点 h_i^k 及其目标点 p_i，预测第 $k+1$ 次迭代时操控点 h_i^{k+1} 的位置。

具体而言，我们将三通道 RGB 图像 $I_k \in \mathcal{R}^{3\times H\times W}$、双通道增量图 $D_k \in \mathcal{R}^{2\times H\times W}$（表示操控点与其对应目标点之间的距离），以及单通道热力图$G_k \in \mathcal{R}^{1\times H\times W}$ （表示操控点）拼接在一起，作为运动预测模块的输入。在增量图D_k中，每个操控点 h_i^k 处的值表示该操控点与其对应目标点之间的距离（$\Delta x = x_i^k - x_i,\ \Delta y = y_i^k - y_i$），其他所有位置的值均为零。考虑到增量图 D_k 非常稀疏，我们遵循 [6, 30], 的做法，将单通道热力图G_k 拼接为输入，其中，每个第 k 次迭代操作点周围邻域（$r=4$）内各点的值为 1，其余位置的值为 0。

总体而言，我们最终得到一个包含 6 个通道的张量 $T_k \in \mathcal{R}^{6\times H\times W}$ 作为输入，其中 3 个是 RGB 图像，其余 3 个



(a) Detailed pipeline and Iterative process of DynaDrag

(b) Motion Prediction Module (MP in (**a**))

(c) Motion Supervision in (**a**)

图2。我们方法的详细信息及整体流程。 h_i^k （c）中表示第 k次迭代中的第 i个操控点， p_i 表示第 i个目标点，而 $f_{h_i^k}$是 h_i^k处的特征向量。

通道用于指示操控点和目标点的位置。我们采用 SimVP [9], 这一简单但有效的视频预测方法作为运动预测模块的结构，并稍作修改以适配我们的任务。运动预测模块的输出是一个与输入图像大小相同的双通道光流图 I_k，用于表示 I_k中每个点的位置偏移量。接下来，我们将详细说明如何为运动预测模块准备训练数据。

我们按照以下步骤构建数据集：1）我们使用 Unimatch [48] 预测给定视频 $v \in \mathcal{R}^{S\times H\times W\times 3}$的光流 $f \in \mathcal{R}^{(S-1)\times H\times W\times 3}$。2）我们从 S 帧中随机选择一帧 v_s 作为起始帧，然后通过分割或检测方法提取编辑区域。3）我们在 v_s的编辑区域内随机采样1 $\sim$ 7 个操控点，其采样概率与该处光流f_s的幅值成正比。4）根据 Unimatch 预测的光流，我们在 $v_{s+15} \sim v_{s+55}$ 中随机选择一帧作为结束帧 v_e，并计算操控点在结束帧 v_e 中的终止位置，将其作为目标点。5）训练运动预测模块时，将使用光流图 f_s 作为真实值。根据成对操控点和目标点的位置，我们推导出相应的增量图 D_s 和热力图 G_s。有关数据集构建的更多细节，请参见补充材料。

一般而言，运动预测模块的训练损失如下：

$$\mathcal{L}_{mp} = MSE(f_s, MP(v_s, D_s, G_s)), \quad (1)$$

其中， $MP(\cdot)$ 表示运动预测模块。

与以往的拖拽式图像编辑方法不同，我们完全摒弃了点跟踪，并将运动预测引入这项任务，从而能够动态调整有效控制点。

3.2.2. 动态调整有效控制点。

在实际应用场景中，用户提供的操控点并非都适用于拖拽式图像编辑。有些操控点对编辑的影响可能微乎其微，甚至会产生负面作用，从而显著影响图像的真实性和可编辑性。本节将介绍如何动态选择有效控制点。

我们发现，在 UNet 特征图中，操控点与目标点的特征向量相似度越低，越有助于提高图像可编辑性，并降低生成 $F(z_t^k)$ 人工伪影的可能性，从而提升编辑后图像的真实性。一种可能的解释如下：操控点与目标点的特征向量相似度较高，意味着它们在语义、形状及其他属性上的差异很小。因此，弱监督信号可能会误导模型，使其认为已经进行了充分调整。相反，在更具挑战性的场景中，如果其他操控点与目标点之间的相似度较低，模型往往会保留相似度较高的操控点的位置，从而损害图像的真实性和可编辑性。例如，如图3下方子图所示，下方一对编辑点旨在将嘴唇向下编辑，此处操控点与目标点之间的语义差距较大。然而，上方一对编辑点之间的语义差距相对较小，导致网络误以为该处已编辑到位。因此，如编辑后图像所示，绿色点几乎保持不动，而红色点则得到了充分且显著的编辑。

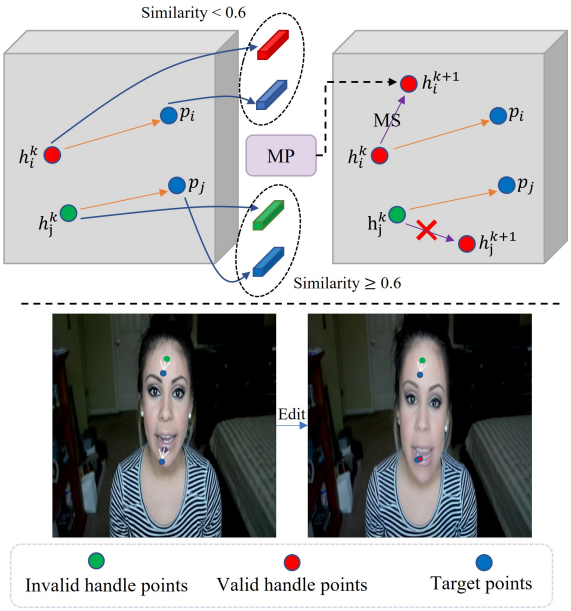


图 3. 动态选择策略。提取控制点的特征向量及其对应目标点的特征向量，并计算这两个特征向量之间的余弦相似度。仅保留相似度低于 0.6 的配对。

因此，我们会筛选用户提供的操控点和目标点配对，仅保留相似度低于 0.6 的点对作为有效点对（如图3左侧子图所示）。如果用户提供的操控点中没有相似度低于 0.6 的点，则使用相似度最低的操控点进行拖动。

一般而言，运动预测避免了点跟踪带来的问题，并通过预测点来缩小源图像与目标图像之间的差距，使中间点更加合理。此外，运动预测支持动态调整有效控制点。

3.3. 运动监督

运动监督 [36, 42] 的设计非常简单，其核心是强制 UNet 最后一张特征图 $F(z_t^k)$ 中目标点处的特征向量接近控制点处的特征向量。运动监督的目标函数如下：

$$\mathcal{L}_{\text{ms}}(z_t^k) = \sum_{i=1}^{n} \left\| F_{\Omega_{r_1}(h_i^{k+1})}(z_t^k) - \text{sg}(F_{\Omega_{r_1}(h_i^k)}(z_t^k)) \right\|_1 + \lambda \left\| (z_{t-1}^k - \text{sg}(z_{t-1}^0)) \odot (1-M) \right\|_1, \quad (2)$$

其中，z_t^k是运动监督优化前第 k次迭代中的第 t步潜在代码（$z_t^0 = z_t$），$\Omega_{r_1}(h_i^k) = \{(x, y): |x - x_i^k| \le r_1, |y - y_i^k| \le r_1\}$ 表示一个邻域

其中心位于控制点 $h_i^k = (x_i^k, y_i^k)$，边长为 $2r_1 + 1$；$F_{h_i^k}(z_t^k)$表示最后一个UNet特征图 $F(z_t^k)$中控制点处的特征向量h_i^k；$F_{\Omega_{r_1}(h_i^k)}(z_t^k)$表示 $F(z_t^k)$中以控制点 h_i^k为中心、边长为 $2r_1 + 1$ 的特征块；h_i^{k+1} 表示由运动预测模块预测的控制点 h_i^k下一次迭代的位置；M 表示用户提供的二值掩码，用于定义图像中可编辑的区域；$sg(\cdot)$表示梯度停止运算符，例如，$F_{\Omega_{r_1}(h_i^k)}(z_t^k)$的反向传播将在项sg($F_{\Omega_{r_1}(h_i^k)}(z_t^k)$)处停止。值得注意的是，在基于用户提供的掩码进行优化的公式（2）第二项中，我们使用 $t-1$ 步的扩散潜在编码进行监督，而不是使用UNet特征图。最后，经过5步梯度下降优化，我们得到用于后续预测并移动迭代的扩散潜在编码$\hat{z}_t^k$：

$$\hat{z}_t^{k,i+1} = \hat{z}_t^{k,i} - \eta \cdot \frac{\partial \mathcal{L}_{\text{ms}}(\hat{z}_t^{k,i})}{\partial \hat{z}_t^{k,i}}, i = (0, 1, 2, 3, 4), \quad (3)$$

其中 $\hat{z}^{k,0}_t = z_t^k$，$\hat{z}_t^k = \hat{z}^{k,5}_t$，$\eta$ 为学习率。

每次迭代完成运动监督后k，我们将优化后的潜在代码 $\hat{z}_t^k$ 送入 UNet 进行去噪，并通过解码器获得编辑后的图像 I_{k+1}。然而，仅使用 DDIM 处理优化后的潜在代码，可能会导致图像质量下降，表现为人工伪影增多，从而损害图像的真实性。我们推测，这一现象可能是由于优化过程中原始图像的一部分潜在代码遭到破坏所致。为尽量减轻这一问题的影响，根据 [32, 42], , 替换键值似乎有助于维持编辑前后图像的一致性。具体而言，我们使用 z_t^0的去噪过程来引导 $\hat{z}_t^k$的去噪过程，从而保留原始图像中的信息。正如 [32, 42], 所建议的，在 UNet 去噪的推理阶段，我们将从 $\hat{z}_t^k$ 中获取的自注意力模块中的键和值，替换为 z_t^0中相应的部分。

3.4. 运动预测与运动监督的迭代过程

如图所示，运动预测和运动监督以迭代方式进行。2（a）。当用户提供图像 I_0 及其配对的控制点 h_i^0和目标点 p_i时，预测得到的光流图 f^{pred} 首先通过运动预测模块获得，并计算下一次迭代步骤中控制点的对应位置：

$$h_i^{k+1} = h_i^k \oplus f_{h_i^k}^{pred}, i = 0, 1 \cdots n, \quad (4)$$

其中，$f_{h_i^k}^{pred}$是光流图 f^{pred}中像素位置 h_i^k对应的双通道向量（Δx，Δy），$\oplus$ 表示按位相加。



图4. 运动预测的泛化研究。在 FaceForensics 上训练的运动预测++数据集 [41] 用于在 DragBench 上进行测试。

图像 I_0 经编码器编码为 z_0^0，并通过DDIM反演加噪为 z_t^0。接下来，运动监督优化扩散潜在编码 z_t^0，使控制点h_i^0 朝运动预测模块预测的 h_i^1 移动，从而得到$\hat{}$ z_t^0。之后，我们将$\hat{}z_t^0$ 去噪为$\hat{}z_0^0$，并将$\hat{}z_0^0$ 输入自编码器的解码器，得到首次编辑后的图像 I_1。值得一提的是，在将$\hat{}z_t^0$去噪为$\hat{}z_0^0$的过程中，会启用替换键值策略。此时，I_1 和 h_i^1已替换 I_0 和 h_i^0的位置，并将进入下一轮运动预测与运动监督循环。上述过程将持续进行，直到控制点足够接近目标点或达到最大迭代次数。

4. 实验

4.1. 数据集

本文在面部数据集 Face-Forensics++ [41]和人体数据集 TED演讲 [45]上进行实验。选择这些数据集主要有两个原因。从理论上说，人体和面部的变体最为丰富。与建筑物、树木等自然场景以及苹果等自然物体相比，人体和面部具有灵活性，不仅会发生位置和姿态变化，其大小和形状也会变化。因此，在面部和人体数据集上比较各种方法，是评估其编辑能力最有效的方式。从应用角度来看，用户对编辑面部和人体的大小、形状和姿态有着最高需求，而且这类编辑有明确的工业应用场景。

为了评估运动预测的泛化性能，我们还在 DragBench 上进行了实验 [42]。详细的泛化研究见第4.7节。

4.2. 实现细节

在所有实验中，我们采用 SimVP [9]作为运动预测模型，并进行少量修改以适配我们的任务（SimVP 的详细配置和修改见补充材料）。我们使用 Stable Diffusion 1.5 [40]作为运动监督的扩散模型。

编辑图像之前，我们首先训练一个 LoRA [15]，并将其注入 Stable Diffusion 去噪器中的所有注意力模块。我们将 LoRA 的秩设为 16，并使用 AdamW [27] 优化器以2^{-4} 作为学习率微调 LoRA，共进行 200 步。在运动监督过程中优化潜在代码时，我们为 DDIM [47]安排 50 步，并优化第 50 步的潜在代码（去噪时的第一个潜在代码）。我们将提示设为空字符串，并使用 Adam [22] 优化器，学习率设为 0.01。运动预测和运动监督的最大迭代次数均设为 25。公式（ r_1 2）中的超参数 λ 公式（2）中的参数由用户设定，默认值为 0.1。用户可以增大 λ，以更好地保留未掩码区域的内容。

4.3. 定性评估

如图 6所示，我们提出的 DynaDrag 展现出卓越的可编辑性和编辑准确度，能够精确地将操控点移动至对应的目- 标点，以实现特定的编辑目标（例如重新调整头部方向、调整 Face- Forensics++ 数据集中的面部表情，以及改变 Ted- talks 数据集中的手部位置）。此外，我们的方法能够有效减轻可能严重降低图像保真度的伪影。值得注意的是，与 DragDif- fusion、DragDiffusion 等移动与跟踪框架下的方法相比，一步式方法 +PIPS2（其使用 PIPS2 跟踪替代了 DragDiffusion 中的点跟踪 [54]）、FreeDrag 和我们的方法，往往会产生严重的伪影；而移动与跟踪框架下的方法则会遇到跟踪遗漏和跟踪模糊的问题，从而导致不理想的编辑结果。总体而言，与以往方法相比，我们的方法在编辑准确度和图像保真度方面展现出了优越性。

4.4. 定量评估

本节介绍了为评估我们的方法及现有方法的性能而开展的实验。我们在 FaceForensics++[41]和 TED演讲 [45]数据集上，将我们的方法与八种基线方法进行比较：DragDiffusion [42], DragDiffusion +PIPS2、DragonDiffusion [32], FreeDrag [25], EasyDrag [14], DragNoise [26], Instant- Drag [44], LightningDrag [43]。DragGAN [36] 基于 StyleGAN [19, 20], , 但这些数据集没有可用的预训练检查点。具体而言，我们使用 FID [12]评估编辑后图像的图像质量。均方误差、LPIPS

数据集 指标	FaceForensics++ FID↓	MSE↓ ($\times 10^3$)	LPIPS↓	CLIP 相似度↑	TED演讲 FID↓	MSE↓ ($\times 10^3$)	LPIPS↓	CLIP 相似度↑
DragDiff [42]	51.37	1.304	0.1564	0.9133	91.77	2.377	0.3908	0.8254
DragDiff +PIPS2[54]	55.93	1.617	0.1913	0.9134	69.55	1.785	0.3534	0.8682
DragonDiffusion [32]	79.22	1.575	0.2045	0.8332	87.46	1.350	0.3176	0.7777
FreeDrag [25]	52.27	1.470	0.1674	0.9160	51.80	1.144	0.2577	0.9181
EasyDrag [14]	59.15	1.232	0.1675	0.8948	63.81	1.135	0.2834	0.8579
DragNoise [26]	58.61	**1.124**	0.1685	0.9106	56.33	**0.922**	**0.2433**	0.8997
InstantDrag [44]	56.48	1.382	0.1696	**0.9221**	64.35	1.707	0.3215	0.9022
LightningDrag [43]	57.00	1.244	0.1496	0.8930	77.38	1.107	0.3031	0.8082
Ours	**48.04**	1.147	**0.1397**	0.9205	**51.51**	1.224	0.2649	**0.9223**

表1。FaceForensics++ 数据集和 Ted-talks 数据集上的定量评估。FID 分数越低，图像保真度越高；均方误差和 LPIPS 越低、CLIP 相似度分数越高，图像操控就越精准。



Iteration 1 Iteration 3 Iteration 6 Iteration 10 Iteration 16

图5。在每次迭代中，蓝点表示操控点，红点表示目标点，绿点表示此前某些迭代步骤中的中间点（为更清楚地展示轨迹，我们没有绘制所有中间点），黄色线条则表示由中间点形成的轨迹。可以观察到，中间点的轨迹并非直线，这与典型的移动与跟踪框架不同。运动预测使操控点能够以更平滑、更自然的方式朝目标点移动（运动预测生成的轨迹更贴合实际视频轨迹，从而降低潜在代码编辑的难度。因此，潜在代码之间的过渡更加自然、平滑）。

数据集 指标	FaceForensics++ FID↓	MSE↓ ($\times 10^3$)	LPIPS↓	CLIP 相似度↑
我们的模型（无 DS）	51.85	1.231	0.1529	0.9152
我们的模型（采用 RS）	49.30	1.171	0.1426	0.9180
我们的模型（采用 FDS）	49.41	**1.136**	0.1400	0.9204
我们的模型（采用 ADS）	**48.04**	1.147	**0.1397**	**0.9205**

表 2. 动态选择（DS）的消融研究。

[51] 以及编辑后图像与目标图像之间的 CLIP 相似度 [38] 用于评估编辑准确性。

如表 1所示，我们提出的 DynaDrag 在 FaceForensics++ 数据集和 Ted-talks 数据集的大多数测试指标上都取得了更好的表现，这意味着在精确的像素级拖拽式图像编辑和图像保真度方面，它优于以往的基线方法。

4.5. 消融研究

本节中，我们对动态选择（DS）在操控点上的有效性进行了消融实验。根据参与程度，动态选择可分为两种类型。首次动态选择（FDS）是指仅在编辑的第一次迭代中，根据特征向量相似度选择有效控制点，并在之后的迭代中保持所选控制点有效，同时舍弃其他控制点。全程动态选择（ADS）是指在编辑的每次迭代中都应用动态选择，这意味着不同迭代中的有效控制点可能不同。

如表2所示，动态选择确实有助于提升图像的可编辑性和质量。需要注意的是，DS 会减少拖动编辑图像时有效控制点的数量。DragGAN 中的实验 [36] 表明，增加控制点可能不利于保持图像的真实性。为排除控制点数量的影响

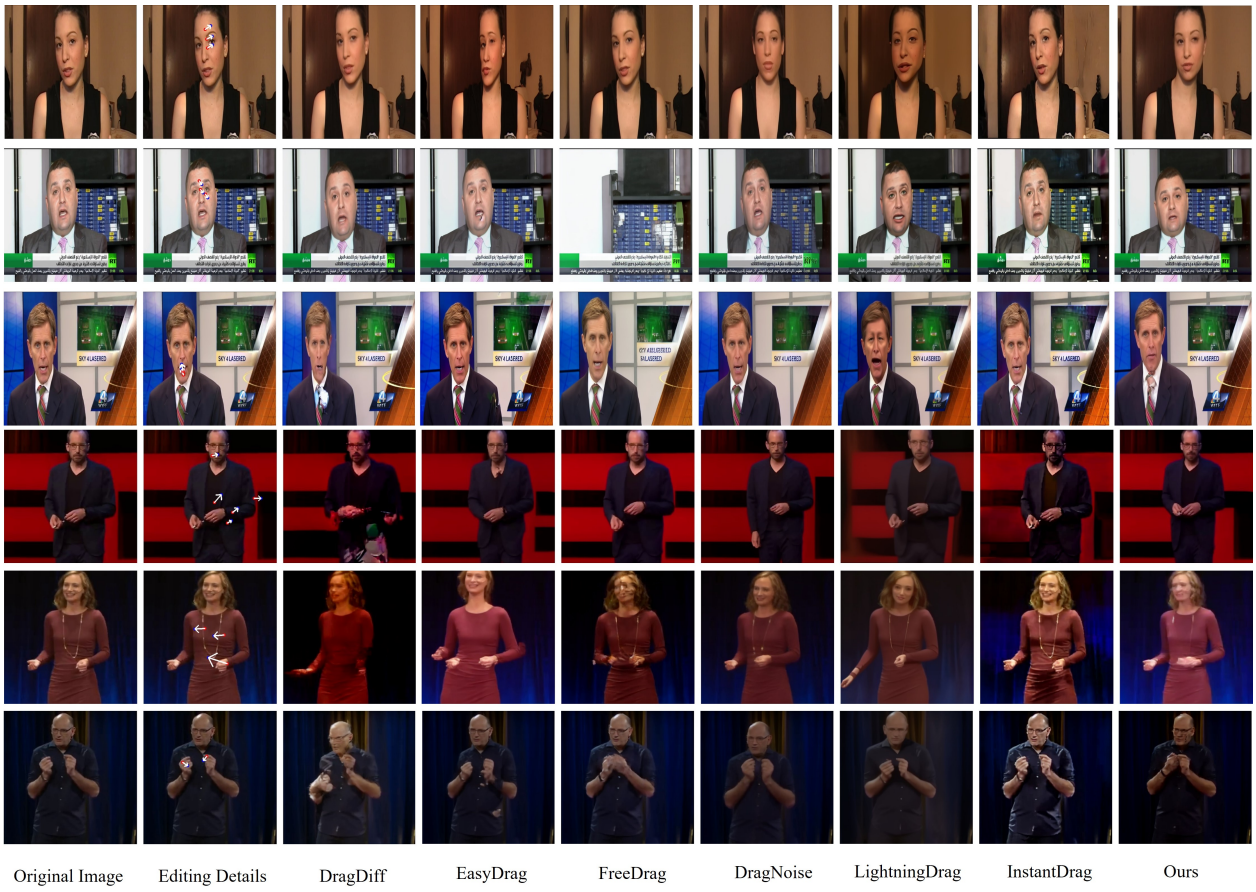


图6. 在 FaceForensics++ 数据集（上方子图）和 Ted-talks 数据集（下方子图）上的定性评估。与基线方法相比，我们提出的 DynaDrag 在编辑准确度和图像保真度方面表现更优。

，并进一步证明动态选择带来的改进，我们按照本文方法 ADS 所采用的有效控制点数量进行随机选择（RS）实验，结果如表2第二行所示。在 FaceForensics++数据集上测试的几乎所有指标中，FDS 和 ADS 的表现都优于 RS，这验证了动态选择的优越性。

4.6. DynaDrag 的有效性分析

本节分析了 DynaDrag 为何优于基线方法。在图 5中，我们展示了 FaceForensics 的一帧上的迭代编辑过程 ++ 数据集[41]。如图 5所示，图中描绘了一位演讲者，其手势和手部动作增强了演讲的生动性。与移动并跟踪的方法不同，这类方法会在操控点和目标点之间沿直线放置中间点，而 DynaDrag 会在移动操控点之前，预测该操控点的下一个中间点。这种方法能够更合理、自然地放置中间点，并使操控点的运动轨迹更加平滑。对中间点的准确预测还有助于操控点在每次迭代中朝预期的中间点自然移动，最终到达目标点。

4.7. 泛化研究

我们在 DragBench 上进行了泛化实验 [42]。DragBench 是一个用于拖拽式图像编辑任务的基准数据集，涵盖了广泛的场景，包括人物、动物和自然景观。我们通过直接采用在 FaceForensics++数据集 [41] 上训练的运动预测模型，研究了我们的方法的泛化能力。实验表明，运动预测展现出强大的泛化能力。例如，如图4所示，尽管模型是在 FaceForensics++ 数据集上训练的，且 FaceForensics++ 数据集与 DragBench 之间存在显著差异，运动预测在许多场景中仍然有效。

5. 结论

本文提出了 DynaDrag，这是一种采用全新“预测并移动”框架的拖拽式图像编辑方法，可在编辑给定图像时动态调整有效控制点。全面的定量评估、定性评估和消融实验展现了所提方法的出色性能与优势。

DynaDrag：通过运动预测实现动态拖拽式图像编辑

补充材料

在本补充材料中，我们将在第 6 节介绍运动预测的架构，在第 7 节分析我们方法的编辑耗时，在第 8 节说明数据集构建的详细信息，在第 9 节展示更多定性结果，在第 10节展示动态选择消融研究的可视化案例，并在第 11 节讨论我们方法的局限性。

6. 运动预测架构

如图 2 所示，我们采用 SimVP [9], 这一种简洁而有效的视频预测方法，作为运动预测模块的基础模型，并进行少量修改以适应我们的任务。

鉴于 SimVP 专注于视频预测任务，其设计必须兼顾图像内部的空间关系和连续帧之间的时间关系。SimVP 由编码器、转换器和解码器组成，三者均采用卷积神经网络结构，各司其职：编码器提取空间特征，转换器捕捉时间演变，解码器则整合时空细节以预测未来帧。具体而言，输入形状为 (B, T, C, H, W) 的 T 个相邻帧输入后，编码器会将其重塑为形状为 $(B \times T, C, H, W)$ 的张量，并对 (H, W) 应用卷积，使用 C 个通道提取每帧内的空间信息，得到形状为 $(B \times T, C_e, H_e, W_e)$ 的张量。随后，转换器将编码器的输出重塑为形状为 $(B, T \times C_e, H_e, W_e)$ 的张量，并对 (H_e, W_e) 应用卷积，使用 $T \times C_e$ 个通道捕捉帧间的时间关系。最后，解码器整合来自转换器的时间信息，通过跳跃连接融合来自编码器的空间信息，最终生成预测的视频帧。

在我们的任务中，我们根据给定的图像、操控点和目标点，预测图像中每个像素的运动方向。由于不需要时间特征，我们只需提取空间特征。为此，我们将 3 通道 RGB图像、增量图和热力图直接拼接成形状为 $(B, 6, H, W)$ 的 6 通道张量作为输入。我们移除了转换器中的重塑步骤，并保留编码器与解码器之间的跳跃连接。此外，我们将解码器的输出通道数从 3 改为 2，以表示图像中每个像素的运动方向。SimVP 的其他默认配置（如隐藏通道数）在我们的任务中保持不变。

7. 编辑时间分析

为研究我们方法的编辑耗时，我们使用 NVIDIA RTX 3090 GPU 对每个数据集中的 500 张、分辨率为 512×512 的图像进行了测试，并计算了平均编辑耗时。如表 3 所示，LoRA微调过程每张图像约需 57 秒。编辑阶段包含运动预测和运动监督：FaceForensics ++ 数据集 [41] 每张图像需 406.8 秒，而 Ted-talks 数据集 [45]每张图像需 326.6 秒。由于运动预测在像素空间中进行，因此每次运动监督步骤后都需要进行 DDIM [47]采样。实验结果表明，DDIM 采样占据了大部分编辑时间，而运动监督步骤中的优化过程相对较快。

8. 数据集构建

我们使用 FaceForensics++ [41] 和 Ted-talks 数据集 [45] 构建用于运动预测模块的训练集，以及用于评估现有方法的测试集。FaceForensics++ 是一个由 1000 段原始视频序列组成的数据集，这些视频序列经过自动人脸操纵方法处理。FaceForensics++ 中的每段视频都包含一张可跟踪、基本为正面且没有遮挡的人脸，因此可以轻松地通过对视频应用人脸解析方法，准确定位人脸区域。Ted-talks 数据集包含 3000 多段从互联网上采集的演讲视频，记录了人物进行演讲的过程。

如图8所示，我们通过以下步骤构建数据集：1）使用 Unimatch [48] 预测给定视频的光流 $f \in \mathcal{R}^{S-1 \times H \times W \times 3}$ $v \in \mathcal{R}^{S \times H \times W \times 3}$。2）从 v_s 帧中S 随机选取一帧作为起始帧，然后使用分割方法或检测方法提取编辑区域。3）根据光流幅值作为概率，在 $1 \sim 7$ 的编辑区域内随机采样 v_s 个操控点 f_s 。4）根据 Unimatch 预测的光流，在 $v_{s+15} \sim v_{s+55}$ 中随机选取一帧作为结束帧 v_e，并计算操控点在结束帧中的最终位置 v_e 作为目标点。训练运动预测模块时，将光流图 f_s 用作真实值。根据成对操控点和目标点的位置，生成对应的增量图 D_s 和热力图 G_s。至于编辑区域的提取，对于 FaceForensics++数据集，我们使用人脸解析方法 RTTN [24] 提取人脸区域作为编辑区域；对于 Ted-talks 数据集，则使用 YOLOv8[17] 检测人体区域作为编辑区域。

微调 LoRA	编辑阶段							
	FaceForensics++ [41]				TED演讲 [45]			
57s	MP	MS	DDIM	总计	MP	MS	DDIM	总计
	55.6秒	27.5秒	291.7秒	406.8秒	44.7秒	23.6秒	236.2 秒	326.6 秒

表 3. 在每个数据集的 500 张图像上测试的平均编辑时间。每张图像需要 57 秒来微调 LoRA。FaceForensics++ 数据集 [41] 的编辑阶段每张图像耗时 406.8 秒，Ted-talks 数据集 [45]则耗时 326.6 秒。MP 表示运动预测所需时间，MS 和 DDIM 分别表示运动监督和 DDIM 采样所需时间。



图 7. 在 FaceForensics++数据集 [41]上测试的视觉消融结果。

我们分别使用 FaceForensics++ 和 Ted-talks 训练集视频序列的 95% 训练两个运动预测模型，并使用 FaceForensics ++和 Ted-talks 测试集中的其余视频序列进行测试。

9. 更多定性结果

本节中，我们将进一步对比 DragDiffusion [42], EasyDrag [14], FreeDrag [25],DragNoise [26], LightningDrag [43], InstantDrag [44] 与我们提出的 DynaDrag。如图 10所示，我们提出的 DynaDrag 能够出色地满足编辑需求，包括 Ted-talks 数据集中的身体姿态调整、身体朝向调整和整体身体运动。在 Face-Forensics++ 数据集 [41]中，如图 11所示，我们提出的方法能够有效修改面部表情、面部朝向和整体面部位置。与先前的方法相比，测试结果表明，该方法在两个数据集上均提升了可编辑性和编辑准确度。Ted-talks 数据集 [45] 和 FaceForensics++ 数据集[41] 中的更多定性结果再次展现了 DynaDrag 相较于先前方法的优越性。

10. 动态选择的视觉消融

本节展示了关于动态选择的消融研究可视化结果。如图7中的第一个子图所示，蓝线表示应将鼻子处的控制点拖动到的最终位置。值得注意的是，只有 ADS 和 FDS 有效实现了这一调整。此外，通过考察该子图中的其他几组编辑点，我们发现，在编辑过程中，鼻子处的控制点最能体现用户的编辑意图，且相似度最低。不使用动态选择或采用随机选择时，鼻子处的点可能会受到其他点的影响而被过度拖动。然而，启用动态选择后，ADS 能够在编辑后的中间结果中持续优化有效点，最终生成比 FDS 更符合用户拖动要求的结果。如图7中的第二个子图所示，采用 ADS 和 FDS 的方法比不使用动态选择以及采用随机选择的方法产生的伪影更少。在第三个示例中，采用 ADS 的方法能够更准确地将控制点编辑到目标点。

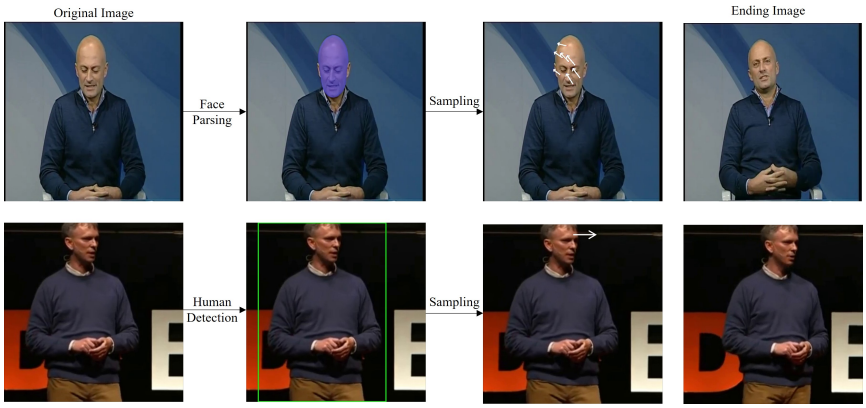


图 8. 数据集构建示意图。我们使用 RTTN 提取 FaceForensics [24] 中每个视频序列的编辑区域++，并使用 YOLOv [41] 处理 TED演讲8 [17] 数据集。随后，我们仅在每帧的编辑区域内采样操控点。



图 9. 失败案例，其中操控点（红色）应拖动至目标点（蓝色）。然而，运动预测未能准确预测每个操控点的运动，导致编辑效果不理想。

11. 局限性

尽管大量实验已证明，我们提出的方法优于以往的拖拽式图像编辑方法 [32, 42]，但也必须承认，我们的方法存在固有局限。通过引入预测并移动框架，我们的方法无需进行点跟踪，从而避免了跟踪方法带来的问题。然而，运动预测模块也带来了自身的挑战，尤其是准确性问题。在某些情况下，运动预测模块无法准确预测每个控制点的运动；在极端情况下，它甚至会预测与实际方向相反的运动，导致编辑后的图像不符合用户预期（如图9所示）。此外，每个运动预测模型都针对单一场景的特定数据集进行训练，这限制了预训练模型的通用性。

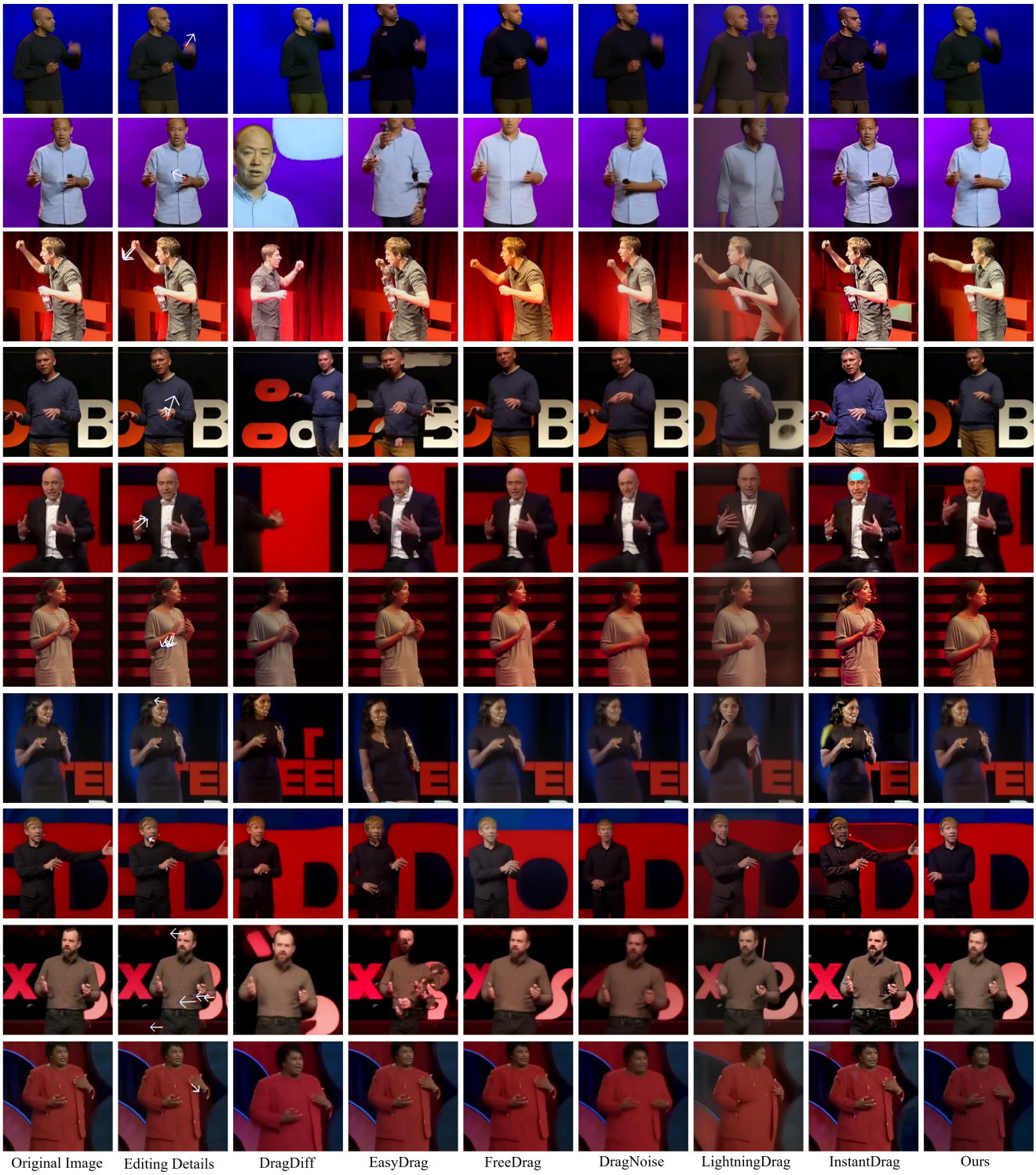


图10. 在 Ted-talks 数据集上进行测试的更多定性结果 [45]。

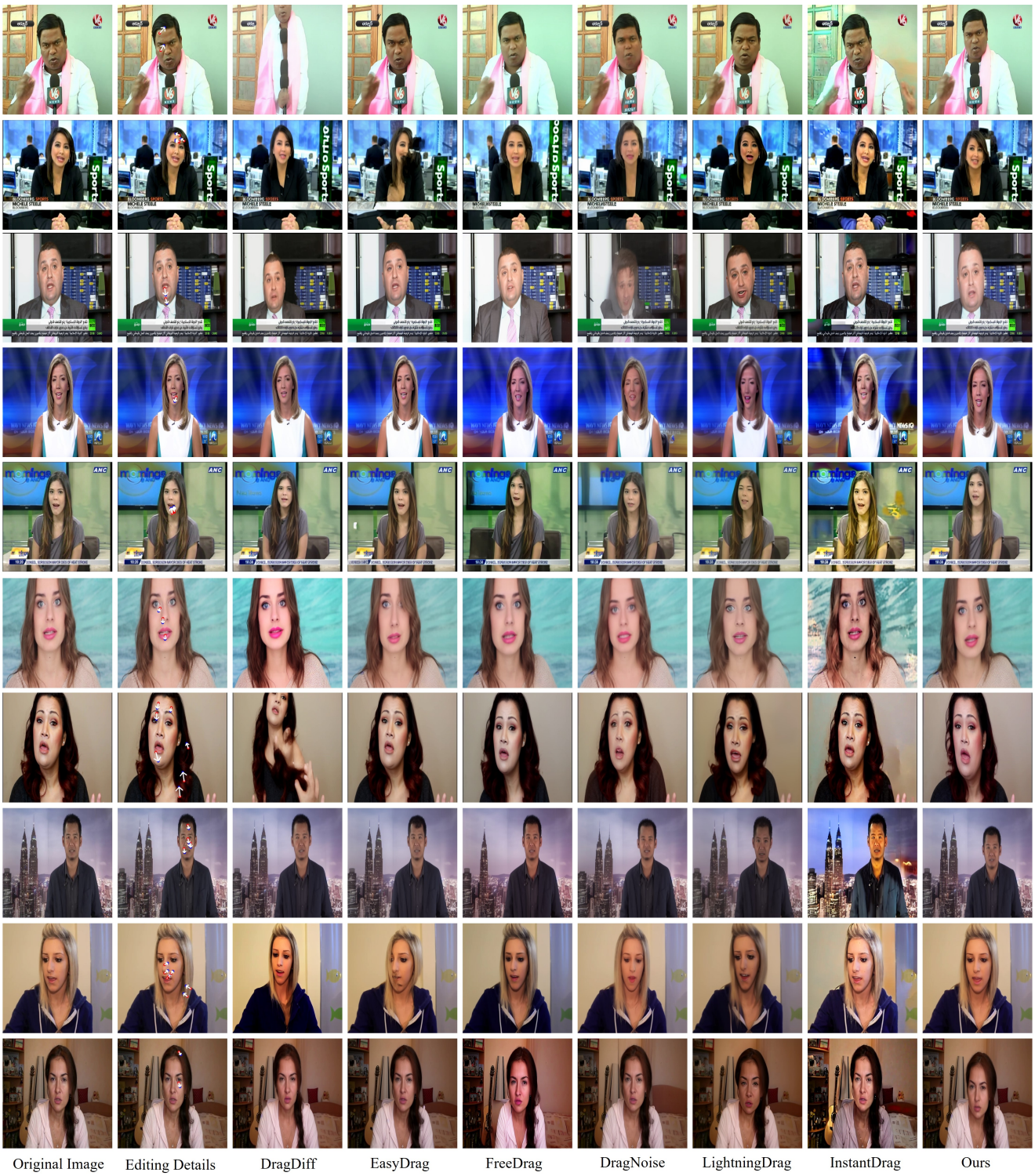


图11。在 FaceForensics++数据集 [41]上的更多定性测试结果。

参考文献

[1] Omri Avrahami、Rinon Gal、Gal Chechik、Ohad Fried、Dani Lischinski、Arash Vahdat 和 Weili Nie。DiffuHaul：一种无需训练的图像物体拖拽方法。arXiv 预印本 arXiv:2406.01594，2024年。2[2] Tim Brooks、Aleksander Holynski 和 Alexei A Efros。Instructpix2pix：学习遵循图像编辑指令。发表于 CVPR，2023年。1[3] DuoSheng Chen、Binghui Chen、Yifeng Geng 和 Liefeng Bo。AdaptiveDrag：基于扩散的图像编辑中的语义驱动拖拽。arXiv 预印本 arXiv:2410 12696，2024年 1，2。.[4] Xing Cui、Peipei Li、Zekun Li、Xuannan Liu、Yueying Zou 和 Zhaofeng He。定位、理解、协作：通过意图推理器实现语义感知拖拽。arXiv 预印本 arXiv:2406.00432，2024年。2[5] Yutao Cui、Xiaotong Zhao、Guozhen Zhang、Sheng ming Cao、Kai Ma 和 Limin Wang。StableDrag：用于基于点的图像编辑的稳定拖拽。arXiv 预印本 arXiv:2403.04437，2024年。2[6] Haoye Dong、Xiaodan Liang、Ke Gong、Hanjiang Lai、Jia Zhu 和 Jian Yin。用于姿态引导的人像图像合成的软门控变形-GAN。发表于 NeurIPS，2018年。3[7] Yuki Endo。用于 StyleGAN 图像布局编辑的用户可控潜变量变换器。Computer Graphics Forum，41(7)：395–406，2022年。1[8] Weixi Feng、Xuehai He、Tsu-Jui Fu、Varun Jampani、Arjun Akula、Pradyumna Narayana、Sugato Basu、Xin Eric Wang 和 William Yang Wang。用于组合式文本到图像合成的免训练结构化扩散引导。arXiv 预印本 arXiv:2212.05032，2022年。1[9] Zhangyang Gao、Cheng Tan、Lirong Wu 和 Stan Z Li。SimVP：更简单却更出色的视频预测。发表于 CVPR，2022年。4，6，1[10] Ian Goodfellow、Jean Pouget-Abadie、Mehdi Mirza、Bing Xu、David Warde-Farley、Sherjil Ozair、Aaron Courville 和 Yoshua Bengio。生成对抗网络。发表于 NeurIPS，2014年。2[11] Amir Hertz、Ron Mokady、Jay Tenenbaum、Kfir Aberman、Yael Pritch 和 Daniel Cohen-Or。通过交叉注意力控制实现 Prompt-to-Prompt 图像编辑。arXiv 预印本 arXiv:2208.01626，2022年。1[12] Martin Heusel、Hubert Ramsauer、Thomas Unterthiner、Bernhard Nessler 和 Sepp Hochreiter。通过双时间尺度更新规则训练的生成对抗网络收敛至局部纳什均衡。发表于 NeurIPS，2017年。6[13] Jonathan Ho、Ajay Jain 和 Pieter Abbeel。去噪扩散概率模型。发表于 NeurIPS，2020年。2[14] Xingzhong Hou、Boxiao Liu、Yi Zhang、Jihao Liu、Yu Liu 和 Haihang You。EasyDrag：基于扩散模型的高效点式操作。发表于 CVPR，2024年。1，2，6，7[15] Edward J Hu、Yelong Shen、Phillip Wallis、Zeyuan Allen-Zhu、Yuanzhi Li、Shean Wang、Lu Wang 和 Weizhu Chen。LoRA：大语言模型的低秩适配。arXiv 预印本 arXiv:2106.09685，2021年。3，6

[16] Ziqi Jiang、Zhen Wang 和 Long Chen。结合基于文本和基于拖拽的编辑，实现精准灵活的图像编辑。arXiv 预印本 arXiv:2410.03097，2024年。1，2[17] Glenn Jocher、Ayush Chaurasia 和 Jing Qiu。Ultralytics YOLO，2023年。1，3[18] Animesh Karnewar、Andrea Vedaldi、David Novotny 和 Niloy J Mitra。Holodiffusion：使用 2 2023 2 d 图像训练三维扩散模型。发[19] 表于 CVPR，。Tero Karras、Samuli Laine 和 Timo Aila。用于生成对抗网络的基于风格的生成器架构。发表于 CVPR，2019年。2，6[20] Tero Karras、Samuli Laine、Miika Aittala、Janne Hellsten、Jaakko Lehtinen 和 Timo Aila。分析并提升 StyleGAN 的图像质量。发表于 CVPR，2020 2, 6 。[21] Gwanghyun Kim、Taesung Kwon 和 Jong Chul Ye。DiffusionCLIP：用于稳健图像操控的文本引导扩散模型。发表于 CVPR，2022 2 。Diederik P. [22] Kingma 和 Jimmy Ba。Adam：一种随机优化方法。发表于 ICLR，2015年。6[23] Ruining Li、Chuanxia Zheng、Christian Rupprecht 和 Andrea Vedaldi。DragAPart：为铰接物体学习部件级运动先验。arXiv 预印本 arXiv:2403 15382，2024年 2 。 .[24] Yiming Lin、Jie Shen、Yujiang Wang 和 Maja Pantic。用于野外人脸解析的 ROI 双曲正切极坐标变换器网络。《图像与视觉计算》，112:104190，2021年。1，3[25] Pengyang Ling、Lin Chen、Pan Zhang、Huaian Chen 和 Yi Jin。FreeDrag：交互式基于点的图像编辑并非只需点跟踪。arXiv 预印本 arXiv:2307.04684，2023年。1，2，6，7[26] Haofeng Liu、Chenshu Xu、Yifei Yang、Lihua Zeng 和 Shengfeng He。Drag Your Noise：通过扩散语义传播实现交互式基于点的编辑。发表于 CVPR，2024年。1，2，6，7[27] Ilya Loshchilov 和 Frank Hutter。解耦权重衰减正则化。arXiv 预印本 arXiv: *1711 05101*, 2017 6 。。[28] Jingyi Lu、Xinghui Li 和 Kai Han。RegionDrag：基于扩散模型的快速区域图像编辑。arXiv 预印本 arXiv:2407.18247，2024年。1，2[29] Grace Luo、Trevor Darrell、Oliver Wang、Dan B Goldman 和 Aleksander Holynski。Readout Guidance：从扩散特征中学习控制。arXiv 预印本 arXiv:2312.02150，2023年。1[30] Liqian Ma、Xu Jia、Qianru Sun、Bernt Schiele、Tinne Tuytelaars 和 Luc Van Gool。姿态引导的人物图像生成。发表于 NeurIPS，2017年。3[31] Ron Mokady、Amir Hertz、Kfir Aberman、Yael Pritch 和 Daniel Cohen-Or。使用引导扩散模型对真实图像进行编辑的空文本反演。发表于 CVPR，2023年。1[32] Chong Mou、Xintao Wang、Jiechong Song、Ying Shan 和 Jian Zhang。DragonDiffusion：在扩散模型上实现拖拽风格的操控。arXiv 预印本 arXiv:2307 02421，2023年 1，2，5，6，7，3 。 .[33] Chong Mou、Xintao Wang、Liangbin Xie、Yanze Wu、Jian Zhang、Zhongang Qi、Ying Shan 和 Xiaohu Qie。T2I-Adapter：学习适配器，以挖掘文生图扩散模型更强的可控能力。arXiv 预印本 arXiv:2302.08453，2023年。2

[34] Chong Mou、Xintao Wang、Jiechong Song、Ying Shan 和 Jian Zhang。Diffeditor：提升基于扩散的图像编辑的准确性和灵活性。arXiv 预印本 arXiv:2402.02583，2024年。1[35] Thao Nguyen、Utkarsh Ojha、Yuheng Li、Haotian Liu 和 Yong Jae Lee。Edit one for all：交互式批量图像编辑。arXiv 预印本 arXiv:2401.10219，2024年。[36] Xingang Pan、Ayush Tewari、Thomas Leimk¨uhler、Lingjie Liu、Abhimitra Meka 和 Christian Theobalt。Drag Your GAN：在生成图像流形上进行交互式基于点的操控。发表于 ACM SIGGRAPH，2023年。1，2，5，6，7[37] Gaurav Parmar、Krishna Kumar Singh、Richard Zhang、Yijun Li、Jingwan Lu 和 Jun-Yan Zhu。零样本图像到图像转换。发表于 ACM SIGGRAPH，2023 年。1[38]Alec Radford 、 Jong Wook Kim、Chris Hallacy、Aditya Ramesh、Gabriel Goh、Sandhini Agarwal、Girish Sastry、Amanda Askell、Pamela Mishkin、Jack Clark 等。通过自然语言监督学习可迁移视觉模型。发表于 ICML，2021年。7[39] Daniel Roich、Ron Mokady、Amit H Bermano 和 Daniel Cohen-Or。用于真实图像基于潜变量编辑的枢纽调优。《ACM图形学汇刊》（TOG），42 13，2022年 1 1 1 ( ): – 。[40]Robin Rombach、Andreas Blattmann、Dominik Lorenz、Patrick Esser 和 Bj¨orn Ommer。使用潜扩散模型进行高分辨率图像合成。发表于 CVPR，2022年。1，2，6[41] Andreas Rossler、Davide Cozzolino、Luisa Verdoliva、Christian Riess、Justus Thies 和 Matthias Nießner。FaceForensics++：学习检测篡改的人脸图像。发表于 ICCV，2019年。6，8，1，2，3，5[42] Yujun Shi、Chuhui Xue、Jiachun Pan、Wenqing Zhang、Vincent YF Tan 和 Song Bai。DragDiffusion：利用扩散模型实现交互式基于点的图像编辑。arXiv 预印本 arXiv:2306.14435，2023年。1，2，3，5，6，7，8[43] Yujun Shi、Jun Hao Liew、Hanshu Yan、Vincent YF Tan 和 Jiashi Feng。InstaDrag：从视频中诞生的闪电般快速、精准的基于拖拽的图像编辑。arXiv 预印本 arXiv:2405.13722，2024年。2，6，7[44] Joonghyuk Shin、Daehyeon Choi 和 Jaesik Park。Instant-Drag：提升基于拖拽的图像编辑的交互性。arXiv 预印本 arXiv:2409.08857，2024年。1，2，6，7[45] Aliaksandr Siarohin、Oliver J Woodford、Jian Ren、Menglei Chai 和 Sergey Tulyakov。用于铰接动画的运动表示。发表于 CVPR，2021年。6，1，2，4[46] Jascha Sohl-Dickstein、Eric Weiss、Niru Maheswaranathan 和 Surya Ganguli。利用非平衡热力学进行深度无监督学习。发表于 ICML，2015年。2[47] Jiaming Song、Chenlin Meng 和 Stefano Ermon。去噪扩散隐式模型。arXiv 预印本 arXiv:2010.02502，2020年。2，3，6，1[48] Haofei Xu、Jing Zhang、Jianfei Cai、Hamid Rezatofighi、Fisher Yu、Dacheng Tao 和 Andreas Geiger。统一光流、立体视觉和深度估计。IEEE模式分析与机器智能汇刊，2023年。4，1[49] Xingqian Xu、Zhangyang Wang、Gong Zhang、Kai Wang 和 Humphrey Shi。Versatile Diffusion：文本、图像和变体，尽在一个扩散模型中。发表于 ICCV，2023年。2[50] Lvmin Zhang、Anyi Rao 和 Maneesh Agrawala。为文生图扩散模型添加条件控制。载于 *ICCV*，2023年。 2[51] Richard Zhang、Phillip Isola、Alexei A Efros、Eli Shechtman 和 Oliver Wang。深层特征作为感知度量的惊人效用。载于 *CVPR*，2018年。 7[52] Zewei Zhang、Huan Liu、Jun Chen 和 Xiangyu Xu。Good-drag：探索使用扩散模型进行拖拽编辑的良好实践。 *arXiv* 预印本 *arXiv:2404.07206*，2024年。 1，2[53] Xuanjia Zhao、Jian Guan、Congyi Fan、Dongli Xu、Youtian Lin、Haiwei Pan 和 Pengming Feng。Fastdrag：一步操纵任意内容。 *arXiv* 预印本 *arXiv:2405 15769*，2024年2。。[54] Yang Zheng、Adam W Harley、Bokui Shen、Gordon Wetzstein 和 Leonidas J Guibas。Pointodyssey：用于长期点跟踪的大规模合成数据集。载于 *ICCV*，2023年。 6，7