文生图扩散模型中的互动控制生成

摘要

近年来扩散模型在图像生成工作中的应用越来越广泛，图像控制生成使得模型能根据用户给出的条件生成符合要求的图像，其中一种控制生成模型是布局生成，它通过文本和对应的布局框同时控制图像生成，所复现的文章观察到在现存的大多数布局生成工作中，它们大多致力于对对象的定位、姿势和图像轮廓等因素的控制，但是忽略了对象之间的交互动作的问题。这篇文章最大的创新点和工作是专注于对象之间的动作生成，目的是为了在布局生成中让对象之间的动作更自然，逼真。

关键词： 交互控制；扩散模型；

1 引言

良好控制生成图像中的交互可以产生有意义的应用，例如创建具有交互角色的真实场景。接下来将介绍文章在生成效果上的优势。如下图 1 所示，Stable Diffusion 直接根据 Caption 生成图像，可以看到，这一方法无法让用户控制图像中各个实例的位置，只能对其内容作一个大致的控制；GLIGEN 方法可用于布局生成，它在获取 caption 的基础上，还引入了 Caption 中的对象的名词以及它们的布局框，实现了对实例的位置控制，从图 1 中可以看出，GLIGEN 方法使得各个实例之间的位置都能满足布局框输入的给出的约束，但是实例之间的动作有部分失真的现象，实例间的动作并不自然；而 InteractDIffusion 额外使用了 Caption 中的动词，并且让其作为动作的标签，进一步地实现动作控制从而让对象间的交互更逼真更自然。

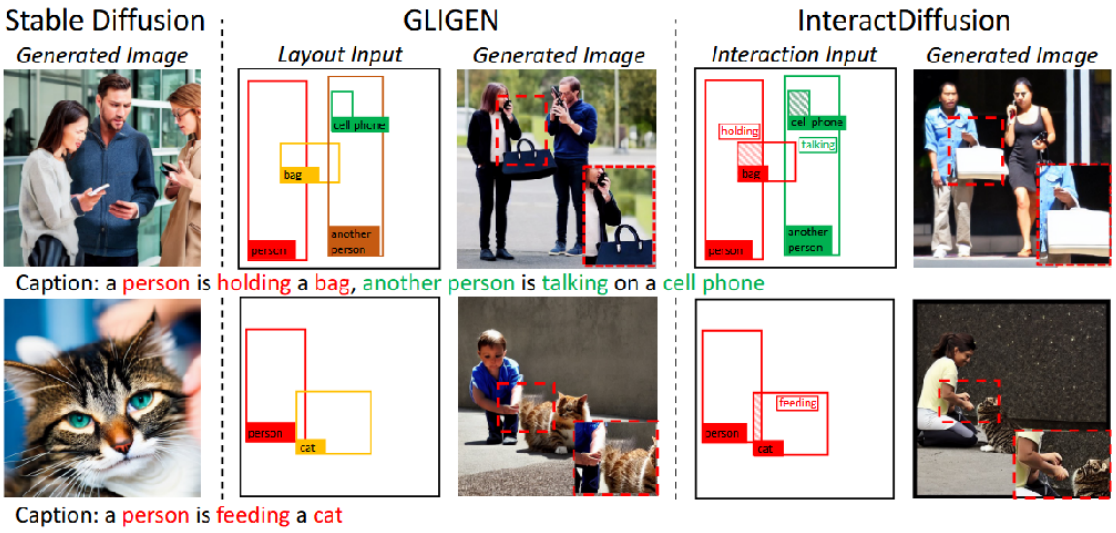

图 1. 方法对比图

2 相关工作

随着深度神经网络的不断发展成熟以及计算设备算力的提升，生成模型逐渐走进大众的视野，与已知样本求解标签的识别任务相反，生成任务在已知标签（准确来说应称为条件）的情况下求解满足标签的样本。GAN [1] 通过精心设计的生成网络和识别网络来实现生成任务。后来，大型文生图模型主要可以被分为两类：自回归模型和扩散模型。对于自回归模型，使用像 VQ-VAE [7] 这样的图像标记器将图像转换为标记，然后训练以文本标记为条件的自回归转换器 [8] 来预测图像标记。然而，自回归模型通常需要大量参数和计算资源来生成高质量图像，如 Parti [10] 中所示。StableDiffusion(SD) 建立在潜在扩散模型 [6] 的基础上，该模型在潜在空间而不是像素空间上运行，使得 SD 能够仅使用文本控制扩散模型生成高分辨率图像。除了文本控制，人们还引入了图像控制生成模型，对典型的工作有 ControlNet 和 IP-Adapter，图像控制可以有效地控制生成内容的细节，但是通常只能生成固定模式的图像，并且对用户来说，图像条件可能难以获取。GLIGEN [3] 添加布局作为条件来帮助指定对象的位置，但控制对象之间的关系或交互仍然是一个难题。这也是本文攻克的难题。

3 本文方法

3.1 布局框标签的扩展

如下图 2 本文的 pipeline，它是在预训练的 StableDiffusion 的 Unet 中的自注意力层和交叉注意力层之间引入了新的动作融合模块，本文的交互模块，即 Interaction Module.

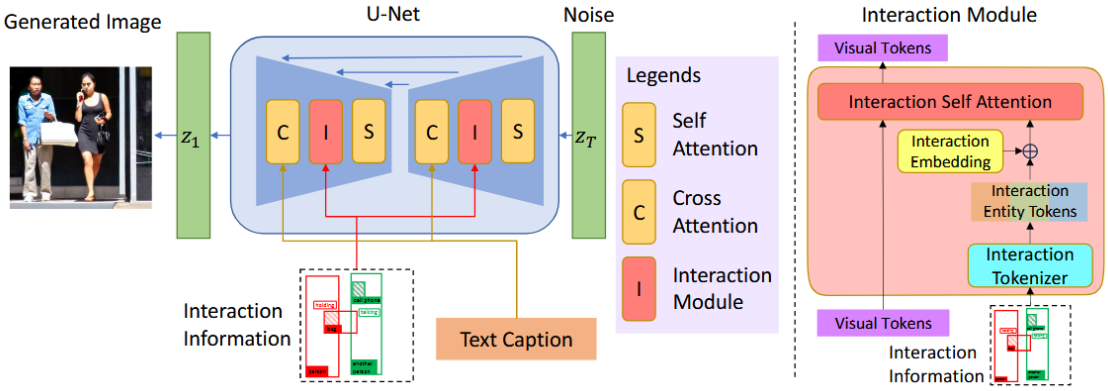

图 2. 方法流程图

首先，本文的贡献是对动作生成进行监督，从而使得生成的图像中实例之间有逼真的动作交互，这就需要在训练数据中让动作跟实例一样，即不仅有文本标签信息，还有动作方式位置的布局信息。但是由于原始的训练数据是不包含后者的，所以本文采用如下公式来计算动作的布局框位置：

$$\begin{aligned} \mathbf{b}_a &= \mathbf{b}_s \,\text{between}\, \mathbf{b}_o \\ &= [R_2(\alpha_i), R_2(\beta_i)], [R_3(\alpha_i), R_3(\beta_i)] \end{aligned} \tag{1}$$

这个公式的意思是动作布局框 $\mathbf{b}_a$ 取决于两个实例之间的布局框，上式中 α_i 指的是两个实例的布局框的四个横坐标，β_i 指两个实例的布局框的四个纵坐标，而 R_j 表示坐标组由小到

大排序的第 j 个值。所以以上公式的意思是，动作布局框的左上角的横（纵）坐标分别为四个横（纵）坐标的由小到大排序第二的值，右下的横（纵）坐标分别为四个横（纵）坐标的由小到大排序第三的值。得到动作的布局框后，我们接下来就获得了<主体，动作，对象>三元组标签，其中主体是施行动作的实例，对象是接受动作的实例，三元组中的每个元素又是包含文本标签和布局框标签的。即:

$$\begin{aligned}\mathcal{D} = [\mathbf{d}_1, ..., \mathbf{d}_N] = [&(s_1, a_1, o_1, \mathbf{b}_{s1}, \mathbf{b}_{a1}, \mathbf{b}_{o1}), ..., \\ &(s_N, a_N, o_N, \mathbf{b}_{sN}, \mathbf{b}_{aN}, \mathbf{b}_{oN})]\end{aligned} \tag{2}$$

下图 3 是关于上述公式计算得到的动作布局框的示例:

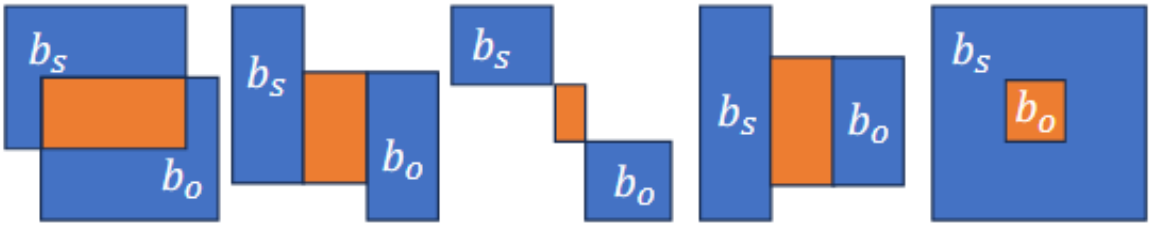

图 3. token 计算流程

3.2 动作及实例编码模块

下图 4 为对上一小节标签的编码模块，三元组标签中的文本标签将被 CLIP 文本 [5] 编码器编码，而布局框标签将被傅里叶编码器 [4] 处理后得到对应的编码。

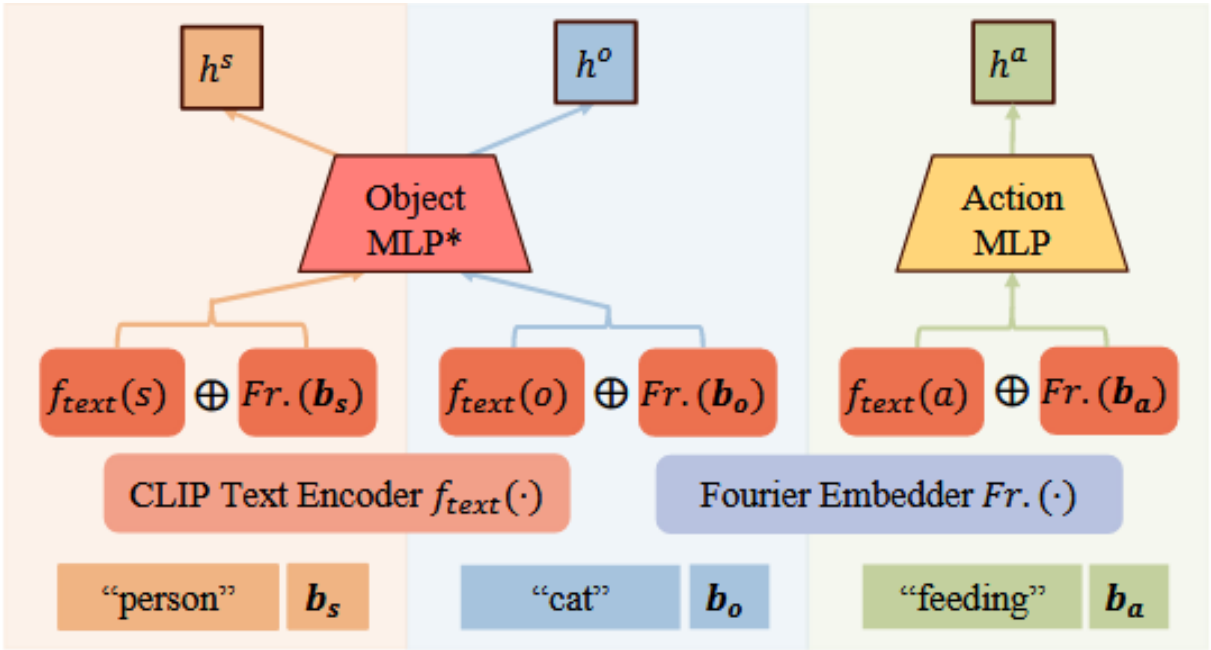

图 4. 动作布局框示例

将主体、动作和对象各自的文本编码和布局框坐标编码经过处理后进行叠加，然后传入

多层感知机网络，经过网络处理融合后就得到了对应的 token。即

$$
\begin{aligned}
h^s &= \text{ObjectMLP}([f_{text}(s), Fourier(\mathbf{b_s})]) \\
h^o &= \text{ObjectMLP}([f_{text}(o), Fourier(\mathbf{b_o})]) \\
h^a &= \text{ActionMLP}([f_{text}(a), Fourier(\mathbf{b_a})]) \\
h &= (h^s, h^a, h^o) = \text{InToken}(s, a, o, \mathbf{b_s}, \mathbf{b_a}, \mathbf{b_o})
\end{aligned} \tag{3}
$$

需要注意的是，主体和对象使用共享的多层感知机，而动作使用独立的多层感知机。原因是虽然主体和对象在三元组标签中承担不一样的角色，但是它们本质上都是实例，它们对应的标签种类和规律都是一样的，所以使用了相同的多层感知机，而动作编码则使用单独的多层感知机以适应不同规律的动作信息。

3.3 角色标记嵌入

前文提到，主体和对象在计算 token 中的操作是一样的，意味着目前来说模型仍然是无法区分一个 token 是来自哪一动作对中的主体还是对象的。所以本小节将引入特殊的标记来区分各个 token 的角色。文章给每个 token 加入了两个标记，如下图 5

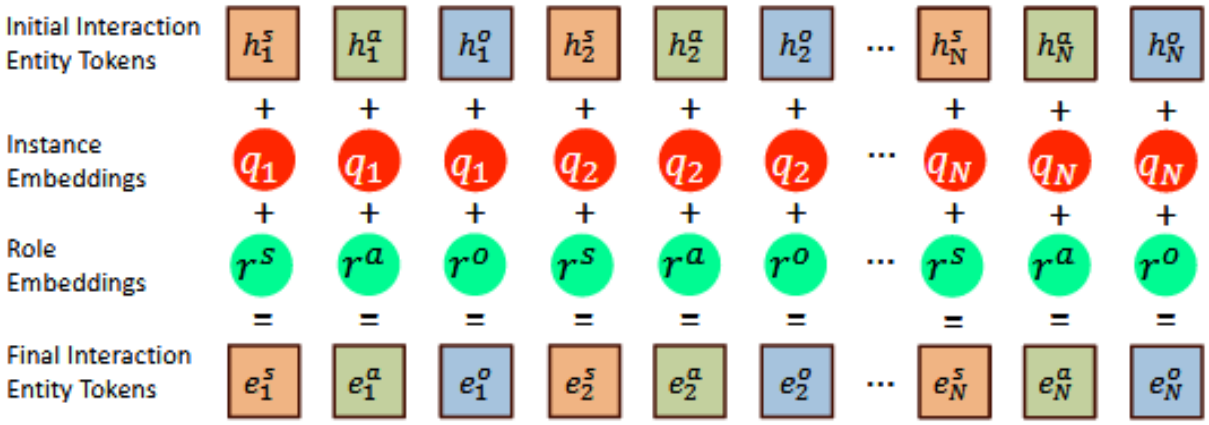

图 5. 标记示意图

上图中，红色的圆圈指代的标记提供了这个 token 是属于哪一组动作对的，而绿色的圆圈指代的是该 token 是来自于主体、动作还是对象。具体公式如下：

$$
\begin{aligned}
e_i &= h_i + q_i + r \\
&= (h_i^s + q_i + r^s, h_i^a + q_i + r^a, h_i^o + q_i + r^o)
\end{aligned} \tag{4}
$$

通过给 token 加入标记嵌入，模型将正确地识别布局框信息的来源。

3.4 信息融合

下图 6 展示了模型的融合机制

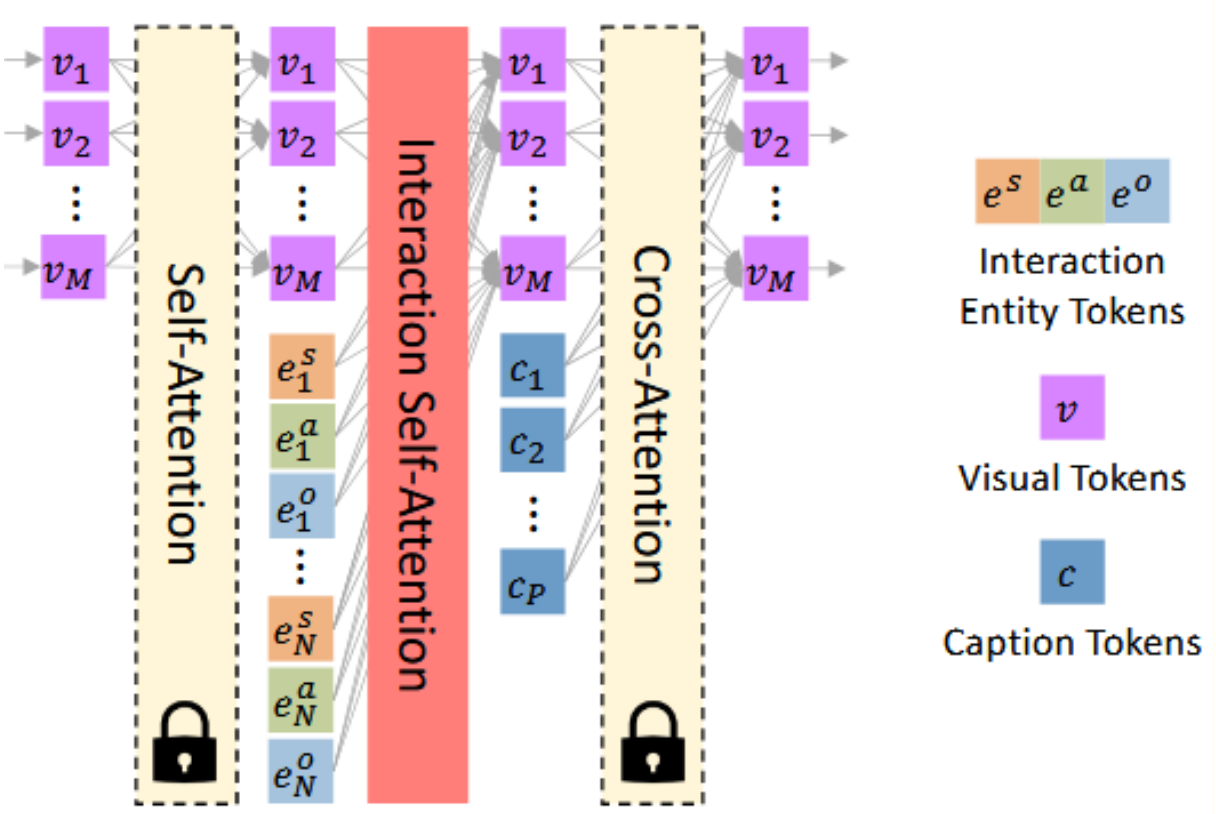

图 6. token 融合示意图

再基模型 SD 中，Unet 去噪网络中的各下采样层中存在一个自注意力层和交叉注意力层，其中交叉注意力层用于融合原来的文本提示信息。文章在自注意力层和交叉注意力层之间引入了交互自注意力层，以融合上文中提到的布局信息。具体计算过程如下：

$$v = v + \eta \tanh \gamma \cdot \mathrm{TS}(\mathrm{SelfAttn}([v, e^s, e^a, e^o])) \tag{5}$$

其中 TS(·) 是一个 token 切片操作，用于仅保留视觉 token 的输出并去掉其他 token。而 η 是超参数，γ 是可训练的参数，两者共同决定了动作信息对模型的控制能力。一个典型的控制例子是在采样过程中，η 可以有如下取值策略：

$$\eta = \begin{cases} 1, t \leq \omega * T \\ 0, t > \omega * T \end{cases} \tag{6}$$

其中 T 是采样的总时间步，而 ω 是使用布局框信息的比例，这个采样策略的含义是，在采样的前 ω 部分，模型使用提取的动作信息，而在后续采样阶段则不使用这些信息。以上即为这篇文章的所有创新计算流程。

4 复现细节

4.1 与已有开源代码对比

本人在复现这篇文章的时候使用了文章 [2] 作者提供的源代码 https://jiuntian.github.io/interactdiffusion. 在此基础上，本人使用了 12 天的时间成功训练了一个 InteractDiffusion 模型，

下一章中的所有生成图像结果都是来源于本人训练的结果。此外，本人复现了另一篇文章 InstanceDiff [9] 的生成效果与之比较，凸显了 InteractDiffusion 在互动生成上的优势。进一步，本人在复现过程中可视化了模型的布局框，展示了模型所生成的实例以及它们之间的动作是符合布局框约束的。本人选择了诸多不同的布局框和文本进行实验，得到了在各种情况下 InteractDiffusion 的生成结果。

4.2 实验环境搭建

在代码实现上，本文在 python 3.9.20，accelerate 1.0.1,torch 2.5.0，torchvision 0.20.0 以及 cuda 12.4 中完成了实验，在硬件上，训练时采用了 3 个 P100 显卡进行分布式训练，在推理阶段使用了 1 个 P100 显卡进行推理。

5 实验结果分析

首先选择了几个布局框及其标签的例子来考察其生成结果，这些例子具有一定的代表性和多样性，旨在全面地测试相关模型的性能。其中，第一个例子是 “两个人坐在一张长椅上”，这是一个较为常见的场景，涉及到人与人以及人与长椅之间的位置关系，需要模型精确地呈现出两个人在长椅上的坐姿，以及他们可能会有的动作，比如是并肩而坐还是前后而坐。第二个例子是 “摩托赛车手正在驾驶一辆摩托赛车”，该场景涉及速度与激情，摩托赛车手作为主体，处于驾驶的动态状态，他需要牢牢地握住车把，身体可能前倾以减少风阻，眼睛专注地注视着前方，而摩托赛车作为对象，在模型的生成结果中，其外观和姿态也十分重要，例如车轮的转动、车身的倾斜角度以及尾气的排放等细节，都应该得到合理的展现。第三个例子是 “自行车手正在骑自行车”，这个场景同样是动态的，自行车手的动作细节需要着重考虑，比如脚蹬踏板的动作、手臂的摆动，以及他在骑行过程中的平衡感，自行车的状态也需要合理呈现，包括车轮的转动、链条的传动，以及车把的转向等，模型需要呈现出自行车手和自行车之间的协调配合，让人们感受到骑行的流畅性。“人正在喂牛” 这个例子则展现了人与动物之间的交互，模型需要考虑人的动作，比如手的动作，可能拿着草料或饲料，伸向牛的嘴巴，人的表情也许是温和的、耐心的，而牛作为被喂养的对象，其反应也很重要，可能会低头吃草，或者抬起头看着喂食的人，两者之间形成一种互动。最后一个例子是 “若干个人坐在一张桌子上吃饭”，在这个场景中，人们围坐在桌子旁，模型要考虑每个人的座位安排，有的人可能正在夹菜，有的人可能在交谈，桌子上摆放的餐具和食物也需要呈现出来，包括碗筷、盘子、菜肴的摆放，这些都需要模型准确地展现出来。实验结果如下图 7 所示：

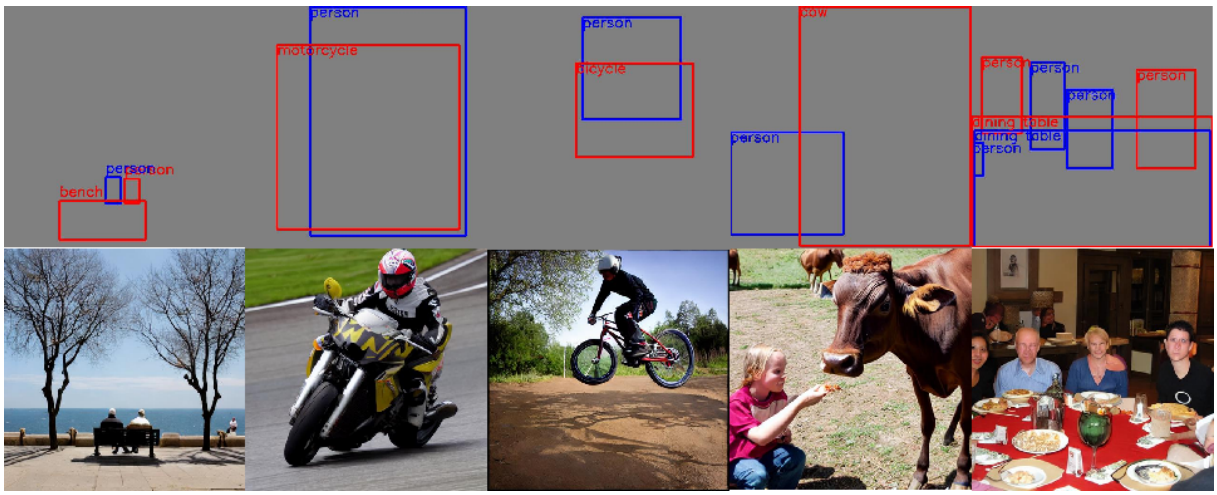

图 7. 实验 1

从实验结果中可以看出，第一个例子中由于布局框比较小，模型将人物以及长椅的位置放在比较远的位置，这可能是由于布局框的尺寸限制，但值得肯定的是，人的坐姿比较逼真。我们可以清晰地看到人物可能是自然地坐在长椅上，他们的腿部自然伸展。在例子二中，尽管摩托车手的与摩托车之间的布局框重合度很大，但是它们之间的动作是协调逼真的，也展现了在摩托车比赛过程中，车和人的稍微倾斜的合理性。摩托车手的身体紧紧地贴在摩托车上，手臂有力地握住车把，双腿夹紧车身；而摩托车则展现出高速行驶时的姿态，车轮飞速旋转，车身根据赛道的弯道或车手的操作而适当倾斜，这些细节都增大了所生成图像的逼真性。例子三中，人物在骑行自行车的动作和自行车的姿态等等因素也是较为自然的，此外，模型生成的图像还考虑了地上的阴影，让整个场景更加真实。第四个例子中，人的动作很明显看出是正准备给牛喂食，人的手向前伸出，手里可能拿着一把青草，手臂的姿势表明正在将青草递向牛的嘴边，面部表情十分专注，仿佛在轻声呼唤牛来进食；而牛则抬起头，嘴巴微微张开，眼睛盯着人手中的食物，给人一种生动的互动感。在例子五中，餐桌上的各种各样的食物也反映了模型把重心放在人与餐桌之间的交互“使用餐桌进食”这一动作上。餐桌上摆满了各式各样的美食，生动地展现出人们在餐桌前用餐时热闹的场景。这些表现都反映了模型把重心放在了实例间交互的生成。

接下来看 InteractDiffusion 和 InstanceDiffusion 之间的对比，本人做了四个例子的对比，旨在深入探究这两种扩散方法在不同场景下的表现差异。其中每个例子都展示三张图像。第一张图像是布局框可视化，它为我们提供了一个基础的框架和结构，让我们可以清晰地看到布局的基本信息，包括各种实例的位置和范围，为后续的生成结果提供了一种空间上的参考。第二张图像是 InteractDiffusion 的生成结果，该结果是基于 InteractDiffusion 算法生成的，通过独特的交互扩散以及融合机制，在实例之间的交互关系处理上有着自身的优势。第三张图像是对应的 InstanceDiffusion 的生成结果，事实上这个算法更侧重于生成实例，而不是实例间的交互。与 InteractDiffusion 的结果有所不同，让我们可以更直观地比较两种算法在相同条件下的不同表现。并且，每个例子正下方的文本是对应的文本是给出的文本提示，这些文本提示为生成图像提供了具体的指导和方向，它们在一定程度上引导着两种算法生成相应的图像，使我们能够更准确地理解算法是如何根据提示进行图像生成的。实验结果如下图 8。

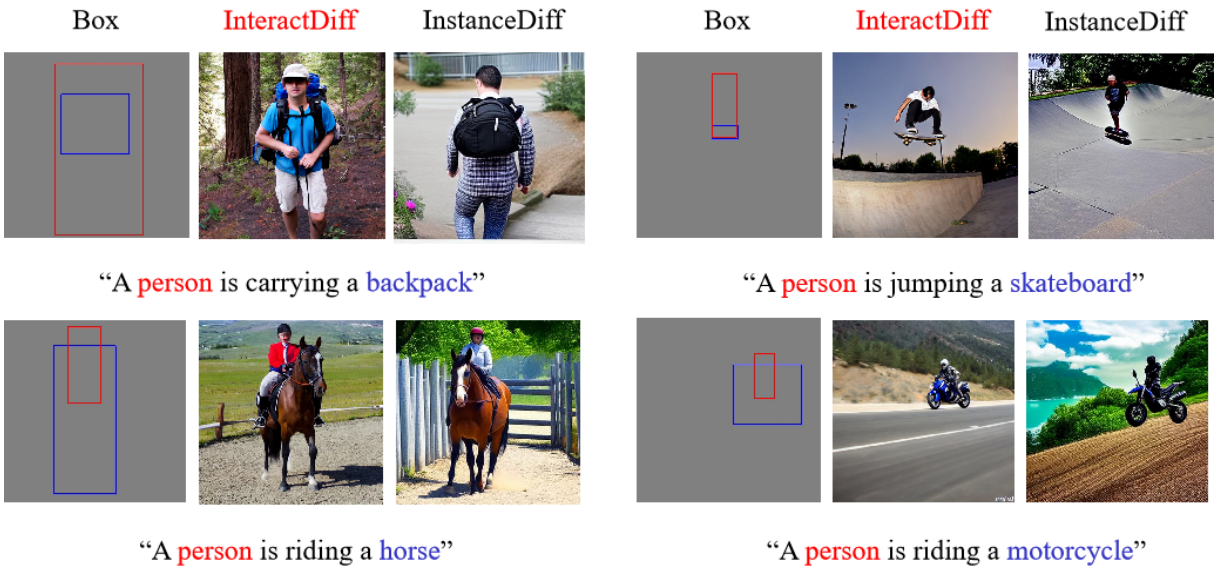

图 8. 实验 2

可以看到，在四个例子中，InteractDiffusion 生成的图像的实例之间的交互都比 InstanceDiffusion 的要逼真，在每一个例子中，相比于 InteractDiffusion，InstanceDiffusion 的生成图像更像是把实例之间的拼接。

6 总结与展望

本报告所复现的文章 InteractDiffusion 使用实例的布局框计算他们之间的交互的布局框，显示地对实例之间的交互进行监督。从而使模型把重心放在实例间交互的生成，让生成图像本身更加逼真。这是以往的所有工作都未曾关注的创新点。并且这一贡献也在现实中有很好的应用。但是本人在观察了多个生成结果后也发现了一些问题，InteractDiffusion 在生成人脸的能力上仍然存在不足，模型生成的人脸很容易出现莫名的扭曲和失真。此外，文章通过实例的布局框计算交互的布局框的方法虽然存在合理性，但是仍然过于简单，它是建立在实例间动作发生在实例之间，实例之间的距离比较小这一事实上。如果不满足这些条件，模型依然可能会给出失真的结果。此外，如果实例之间的动作范围相对于两个实例的尺寸是比较小的，那么这个计算实例间动作的公式也是存在问题的。这些都是这篇文章可以改进的问题。

本报告所复现的文章 InteractDiffusion 使用实例的布局框来计算它们之间的交互的布局框，通过这种方式，能够显式地对实例之间的交互进行监督。这样的操作可以让模型更加明确地知晓应该如何关注和处理实例之间的交互关系，从而引导模型把重心放在实例间交互的生成上，使得生成图像本身更加逼真，更具有现实感和生动性。这是以往的所有工作都未曾关注的创新点，以往的研究往往侧重于图像的其他方面，如纹理、色彩、单一物体的细节呈现等，而 InteractDiffusion 的这种对实例交互的独特关注，无疑为图像生成领域带来了新的思路和视角，并且这一贡献也在现实中有很好的应用，它可以为许多需要展现交互场景的领域提供更为优质的图像生成服务，比如在动画制作、虚拟现实、游戏开发等领域，能够帮助设计师们创造出更加真实、生动的场景和角色互动画面。但是本人在观察了多个生成结果后也发现了一些问题，InteractDiffusion 在生成人脸的能力上仍然存在不足，这是一个比较明显的缺陷。当涉及到人脸的生成时，模型生成的人脸很容易出现莫名的扭曲和失真，可能会出现

五官比例失调的情况，比如眼睛的大小和位置不准确，鼻子和嘴巴的形状奇怪，面部轮廓不自然等，这极大地影响了生成图像的整体质量和逼真度。此外，文章通过实例的布局框计算交互的布局框的方法虽然存在合理性，但是仍然过于简单，它是建立在实例间动作发生在实例之间，实例之间的距离比较小这一事实上，这种假设在某些情况下是具有局限性的。在现实世界中，很多交互场景并不完全满足这一条件，比如在一些较大的场景中，实例之间可能距离较远但仍然存在交互，或者实例之间的动作具有一定的跨越性，并非局限在较小的距离范围内。如果不满足这些条件，模型依然可能会给出失真的结果，可能会导致生成的图像中的实例无法准确地完成相应的动作，或者动作显得十分突兀，与场景不协调。此外，如果实例之间的动作范围相对于两个实例的尺寸是比较小的，那么这个计算实例间动作的公式也是存在问题的，它可能无法准确地反映出动作的细节和幅度，使得生成的动作显得僵硬、不自然，甚至可能会出现动作错误，从而无法展现出真实的交互效果。这些都是这篇文章可以改进的问题，若能解决这些问题，该方法或许能在图像生成领域取得更大的突破和进步。

参考文献

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[2] Jiun Tian Hoe, Xudong Jiang, Chee Seng Chan, Yap-Peng Tan, and Weipeng Hu. Interactdiffusion: Interaction control in text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6180–6189, 2024.

[3] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22511–22521, 2023.

[4] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[5] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[6] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[7] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[8] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[9] Xudong Wang, Trevor Darrell, Sai Saketh Rambhatla, Rohit Girdhar, and Ishan Misra. Instancediffusion: Instance-level control for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6232–6242, 2024.

[10] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2(3):5, 2022.