InteractDiffusion: Interaction Control in Text-to-Image Diffusion Models

摘要

目前，大规模文本到图像 (T2I) 扩散模型已展示出生成连贯图像的惊人能力，最近的进展已经引入了对象定位、姿势和图像轮廓等因素的控制，但在控制生成内容中对象之间交互的能力方面仍然存在关键差距。在这项工作中，作者研究了使用人-物交互 (HOI) 信息调节 T2I 扩散模型的问题，该信息由三元组标签（人、动作、物体）和相应的边界框组成。文章提出了一个可插入的交互控制模型，称为 InteractDiffusion，它扩展了现有的预训练 T2I 扩散模型，使它们能够更好地适应交互。具体来说，作者对 HOI 信息进行标记并通过交互嵌入学习它们的关系。训练条件自注意力层以将 HOI 标记映射到视觉 tokens，从而更好地调节现有 T2I 扩散模型中的视觉 tokens。该模型实现了控制现有 T2I 扩散模型上的交互和位置的能力，其 HOI 检测得分以及保真度均大大优于现有基线。

关键词： 扩散模型; 人-物交互; 条件控制; 图像生成

1 引言

近年来，扩散生成模型成为了图像生成的主流框架。扩散模型可以生成各种高质量的图像来重建原始数据分布，但控制生成的内容非常重要。此后，有大量研究通过类 [6]、文本 [15]、图像（包括边缘、线条、涂鸦和骨架） [2] 和布局 [4] 来控制扩散模型的图像生成。然而，这些不足以有效地表达细微的意图和期望的结果，特别是物体之间的相互作用。该工作在图像生成中引入了另一个重要的控制：交互。

交互是指两个实体或个体之间的相互动作。这些交互是描述我们日常活动不可或缺的一部分。然而，现有的扩散模型在静态图像上效果很好，但在生成涉及交互的图像方面面临巨大挑战。例如，GLIGEN [11] 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悬而未决的难题。文本到图像扩散模型中交互级别的控制有无数应用，例如游戏、电子商务、交互式故事讲述等。

该研究以交互为条件的图像生成，即如何在图像生成过程中指定交互。它面临三个主要挑战：1）交互表示：如何以有意义的 token 表示形式表示交互信息。2）复杂的交互关系：具有交互的对象之间的关系复杂，生成连贯的图像仍然是一个巨大的挑战。3）将条件集成到现有模型中：当前的 T2I 扩散模型在图像生成质量方面表现出色，但缺乏交互控制。可以无缝集成到其中的可插拔模块势在必行。

为了解决上述问题，本文研究 [8] 提出了一个称为 InteractDiffusion 的交互控制模型，作为现有 T2I 扩散模型的可插入模块，旨在实施交互控制。首先，为了向扩散模型提供条件信

息，作者将每个交互对视为一个 HOI 三元组，并将其信息转换为有意义的 token 表示，其中包含有关位置、大小和类别标签的信息。具体而言，每个 HOI 三元组都生成了三个不同的 token，即主体、动作和客体标记。主体和客体 tokens 都包含有关位置、大小和对象类别的信息，而动作 token 包括交互的位置及其类别标签。

其次，表示复杂交互的挑战在于对多个交互的 token 之间的关系进行编码，其中 token 来自不同的交互实例，并且在交互实例中具有不同的角色。为了应对这一挑战，该研究提出了实例嵌入和角色嵌入来对同一交互的 token 进行分组并在语义上嵌入它们的角色。第三，由于现有的 Transformer 块由自注意力和交叉注意力层组成，作者在两者之间添加了一个新的交互自注意力层，以将交互 token 合并到现有的 T2I 模型中。这有助于在训练期间保留原始模型，同时合并额外的交互条件信息。

该研究的主要贡献概括如下：

（i）解决了现有 T2I 模型中的交互不匹配问题，并提出了一个新的挑战：控制 T2I 扩散模型中的交互。所提出的框架 InteractDiffusion 可插入现有的 T2I 模型。它将交互信息作为训练交互可控的 T2I 扩散模型的附加条件，从而提高生成图像中交互的精度。

（ii）引入了一种新方法，将〈主体、动作、客体〉的定位和类别信息标记化为三个不同的标记，然后通过嵌入框架将这些标记分组在一起并指定其交互角色，以有效捕捉复杂的交互关系。这种创新方法增强了复杂交互的表示。

（iii）该工作首次尝试将交互控制引入扩散模型，InteractDiffusion 在 HOI 检测分数方面明显优于基线方法，并且保持了生成质量，FID 和 KID 指标均略有改善。

2 相关工作

2.1 人-物交互

人物交互的最新进展主要集中在检测图像中的 HOI。它旨在通过边界框定位交互的人与物对，并以三元组形式对这些物及其交互进行分类。虽然最近的 HOI 检测工作 [4] 显示出不错的结果，但数据稀缺阻碍了罕见交互的检测性能。相反，HOI 检测的逆任务 HOI 图像合成相对较少被探索。InteractGAN [7] 提出了通过人体姿势以及人和物体的参考图像来生成 HOI 图像。然而，这种方法很复杂，因为它需要一个姿势模板池以及人和物体的参考图像。更密切相关的工作是基于布局的方法 [9]，它专注于根据 HOI 三元组进行场景布局提案以合成图像，但仅限于生成“物体放置”提案。我们的工作重点是解决一个新问题，即使用简单的边界框和交互关系以端到端的方式控制现有 T2I 扩散模型中的交互，而无需人体姿势和参考图像。这种方法有效地解决了 HOI 检测任务中的数据稀缺问题，并开辟了广泛的应用范围。

2.2 扩散模型

扩散概率模型在像素空间中训练和评估扩散模型成本高昂且速度慢，尽管 DDIM [18] 在训练和采样方法上对其进行了进一步改进，其在高分辨率图像上训练仍需要计算昂贵的梯度。潜在扩散模型 (LDM) [17] 将图像压缩为低维潜在表示，并在潜在空间中执行扩散过程以减少计算，并进一步扩展到稳定扩散。我们的工作为稳定扩散模型添加了交互控制。

2.3 控制图像生成

T2I 扩散模型通常使用预训练语言模型（如 CLIP [16]）来指导图像扩散过程。这允许通过提供的文本标题来控制生成图像的内容。但是，仅有一个文本标题不足以控制生成的内容，特别是旨在创建特定内容（例如对象位置和布局、场景深度图、人体姿势、边界线和交互）时。为了解决这个问题，几个模型提出了不同的方法来控制生成的内容，包括对象布局 [10]、分割图 [1] 和图像 [14]。虽然通过对象布局和图像控制图像生成通常可以产生更好的结果，但图像的一个重要方面却被很大程度上忽略了，即对象之间的交互。文本的工作通过加强对生成内容中交互的控制来扩展当前 T2I 模型的功能。

3 本文方法

本文的模型整体框架如图 1 所示。模型的核心是一个可插拔的交互模块 I，其主要功能是将交互信息转换成 token，并将交互信息无缝地集成到现有的 T2I 扩散模型中。比如，在本文的实验中，此交互模块嵌入在了 U-Net 模型中的自注意模块和交叉注意力模块中间。它包括四个部分：(a) 交互 tokenizer，将交互条件转换为 token，(b) 交互嵌入，链接交互三元组的 token 之间的关系，(c) 交互 transformer，在图像块和交互信息之间构建注意力，以及 (d) 交互条件扩散模型，生成具有交互条件的图像。

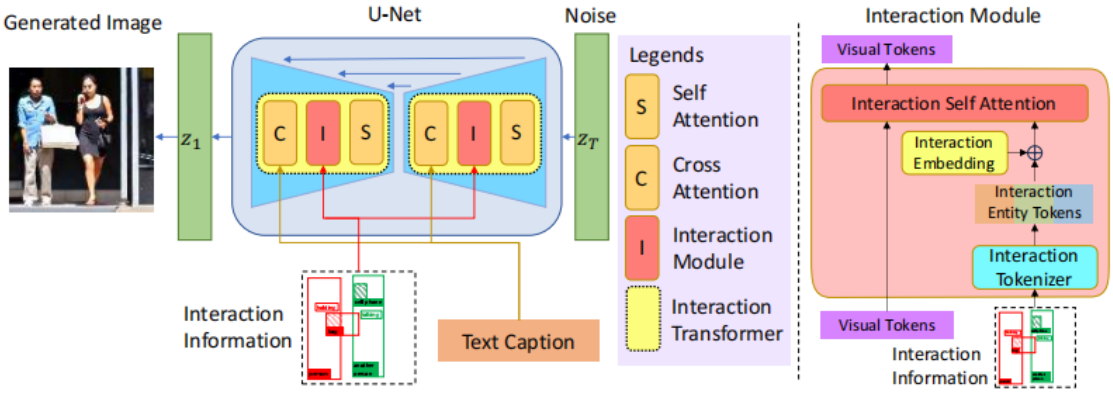

图 1. InteractDiffusion 的整体框架。可插入式交互模块 I 将交互信息无缝集成到现有的 T2I 扩散模型中（左）。提出的模块 I（右）由将交互信息转换为有意义 tokens 的交互 tokenizer、包含复杂交互关系的交互嵌入和将交互控制信息集成到现有 T2I 模型的视觉 tokens 中的交互自注意力组成。

3.1 准备工作

该研究将交互条件与文本标题条件一起纳入现有 T2I 扩散模型的问题。其目标是训练一个扩散模型 $f_\theta(z, c, d)$ 来生成以交互和文本标题为条件的图像，其中 z 是初始噪声。

Stable Diffusion 目前是最好的模型之一，它是 LDM [17] 的扩展，具有更大的模型和数据大小。与其他扩散模型不同，LDM 分为两个阶段以降低计算复杂度。它首先学习双向投影，将图像 x 从像素空间投影到潜在空间作为潜在表示 z，然后在潜在空间中用潜在 z 训练扩散模型 $f_\theta(z, c)$。LDM 学习长度为 T 的固定马尔可夫链的逆过程。它可以解释为去噪自动编码器的等权重序列 $\epsilon_\theta(z_t, t); t = 1, ..., T$，它们被训练来预测其输入 z_t 的去噪版本，其中 z_t 是输

入 z 的噪声版本。无条件目标可以视为:

$$\min_{\theta} \mathcal{L}_{\text{LDM}} = \mathbb{E}_{z,\epsilon\sim\mathcal{N}(0,\mathbf{I}),t}\left[\|\epsilon - \epsilon_\theta(z_t, t)\|_2^2\right] \tag{1}$$

其中 t 从 $1, \cdots, T$ 均匀采样。该模型迭代地从噪声 z_T 到 $z_{T-1}, z_{T-2}, \cdots, z_0$ 产生噪声较小的样本，其中模型 $\epsilon_\theta(z_t, t)$ 由 UNet 实现。通过将潜在空间中的 z_0 投影回图像空间，通过第一阶段训练的解码器一次性获得最终图像。

在 LDM 中，为了用文本说明等各种模态来调节扩散模型，在 UNet 主干之上添加了交叉注意机制。各种模态的条件输入表示为 y，并使用特定域的编码器 $\tau_\theta(\cdot)$ 将 y 投影到中间标记表示 $\tau_\theta(y)$。

在 StableDiffusion 中，用 y 表示的文本说明用于调节模型。它使用 CLIP 编码解码器 $\tau_\theta(\cdot)$ 将文本说明 y 投影到 77 个文本嵌入中，即 $\tau_\theta(y)$。具体而言，StableDiffusion 的条件目标可以视为:

$$\min_{\theta} \mathcal{L}_{\text{LDM}} = \mathbb{E}_{z,\epsilon\sim\mathcal{N}(0,\mathbf{I}),t}\left[\|\epsilon - \epsilon_\theta(z_t, t, \tau_\theta(y))\|_2^2\right] \tag{2}$$

其中 $\tau_\theta(\cdot)$ 代表 CLIP 文本编码器，y 代表文本说明。

3.2 交互 Tokenizer

本文将交互 d 定义为由〈主体 s、动作 a 和对象 o〉组成的三元组标签，以及它们对应的边界框，分别表示为 $b_s\ b_a\ b_o$。我们使用主体和对象边界框来描述它们的位置和大小，并引入动作边界框来指定动作的空间位置。例如，主体（例如女孩、男孩）对特定对象（例如手提包、球）执行特定动作（例如携带、踢）。为了获得动作边界框，这里定义了一个 “between” 操作，应用于主体和对象边界框。假设 b_s 和 b_o 由它们的角坐标 $[\alpha_i, \beta_i], i = 1, 2, 3, 4$，对 b_s 和 b_o 执行 “between” 操作以获得 b_a 的公式是:

$$\mathbf{b}_a = \mathbf{b}_s \text{ between } \mathbf{b}_o \tag{3}$$

$$= [R_2(\alpha_i), R_2(\beta_i)], [R_3(\alpha_i), R_3(\beta_i)], \tag{4}$$

其中 $R_k(\cdot)$ 是其参数的第 k 个升序。图 2 展示了一些 “between” 操作结果的示例。由此将图像的交互条件输入为:

$$D = [d_1, ..., d_N] = [(s_1, a_1, o_1, \mathbf{b}_{s_1}, \mathbf{b}_{a_1}, \mathbf{b}_{o_1}), ..., (s_N, a_N, o_N, \mathbf{b}_{s_N}, \mathbf{b}_{a_N}, \mathbf{b}_{o_N})] \tag{5}$$

其中 N 是交互实例的数量。

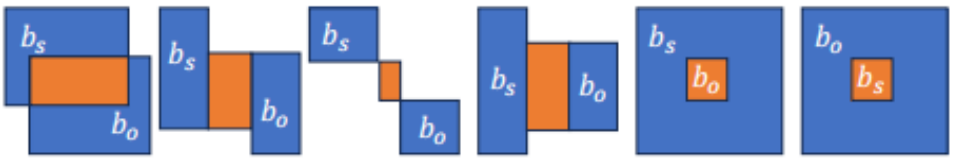

图 2. “Between” 操作获得主体和对象边界框之间的动作焦点区域（以橙色高亮显示）。

主客体 tokens 首先将文本标签和边界框预处理为中间表示。具体来说，使用预训练的 CLIP 文本编码器将主体、动作和客体的文本编码为代表性文本嵌入，并使用傅里叶嵌入 [13]

按照 GLIGEN 对它们各自的边界框进行编码。为了生成主体和客体 tokens h_s, h_o，这里使用多层感知器 ObjectMLP$(\cdot)$ 将它们融合为：

$$h^s = \text{ObjectMLP}([f_{\text{text}}(s), \text{Fourier}(\mathbf{b}_s)]) \tag{6}$$

$$h^o = \text{ObjectMLP}([f_{\text{text}}(o), \text{Fourier}(\mathbf{b}_o)]) \tag{7}$$

动作 tokens 对于动作 token，这里额外训练了一个单独的多层感知器 ActionMLP$(\cdot)$，因为动作在语义上与主体和客体是分开的，其表示为：

$$h^a = \text{ObjectMLP}([f_{\text{text}}(a), \text{Fourier}(\mathbf{b}_a)]) \tag{8}$$

对于每个交互，最终可以将交互条件输入 d 转换为三元组 token h：

$$h = (h^s, h^a, h^o) = \text{InToken}(s, a, o, \mathbf{b}_s, \mathbf{b}_a, \mathbf{b}_o) \tag{9}$$

其中 InToken$(\cdot)$ 是公式 6 至 8 的组合，如图 3 所示。

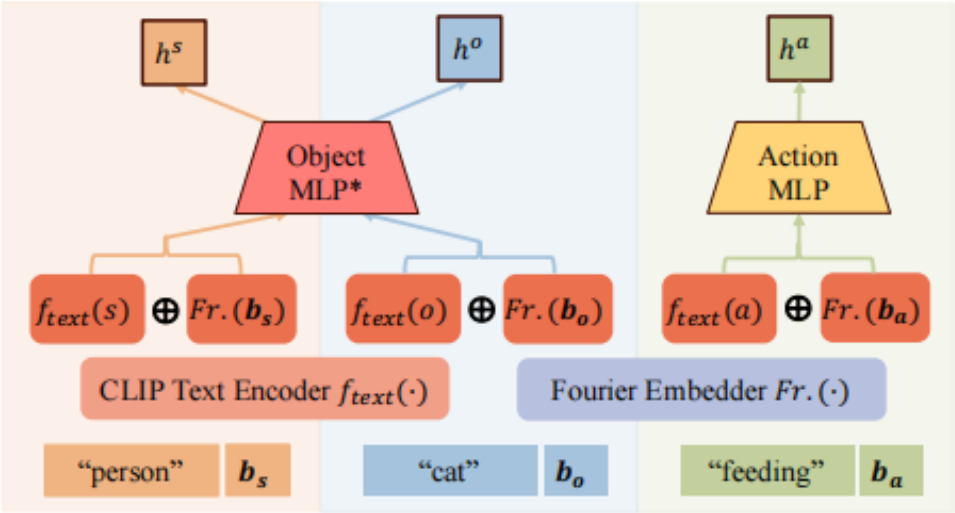

图 3. 交互 Tokenizer

3.3 交互嵌入

交互是主体、客体及其动作之间的复杂关系。从公式 9 可知，tokens h_s, h_a, h_o 是单独嵌入的（如图 1 所示）。对于多个交互实例，所有 tokens $h_i^s, h_i^a, h_i^o; i = 1, \cdots, N$ 都是单独嵌入的。因此，有必要按交互实例对这些 tokens 进行分组，并指定 token 在交互实例中的不同角色。段嵌入 [5] 已证明其在捕获文本序列中段之间关系方面的有效性，它通过向 token 添加可学习的嵌入来将一系列单词分组为段。在这项工作中，扩展了这个概念，将 token 分组为三元组，即向交互实例 $\mathbf{h} \in \{\mathbf{h}_1, \cdots, \mathbf{h}_N\}$ 添加一个新的实例嵌入，记为 $q \in q_1, \cdots, q_N$ 如下所示：

$$e_i = h_i + q_i \tag{10}$$

其中，同一实例中的所有 token 共享相同的实例嵌入。这会将所有 token 分组为交互实例或三元组。

此外，三元组中的每个 token 都有不同的角色信息。因此，我们用三个角色嵌入 $r \in \{r^s, r^a, r^o\}$ 嵌入它们的角色信息，以形成最终的实体 token e_i：

$$\mathbf{e}_i = \mathbf{h}_i + q_i + r \tag{11}$$

$$= (h_i^s + q_i + r^s, h_i^a + q_i + r^a, h_i^o + q_i + r^o) \tag{12}$$

其中 r^s, r^a 和 r^o 分别表示主体、动作和客体的角色嵌入。从公式 12 中我们可以看出，所有实例中相同角色的 token 共享相同的角色嵌入。将实例和角色嵌入添加到交互实体 token h_i（如图 4 所示）可对复杂的交互关系进行编码，即指定 token 的角色和交互实例，从而显著改善图像生成，尤其是在具有多个交互实例的场景中。

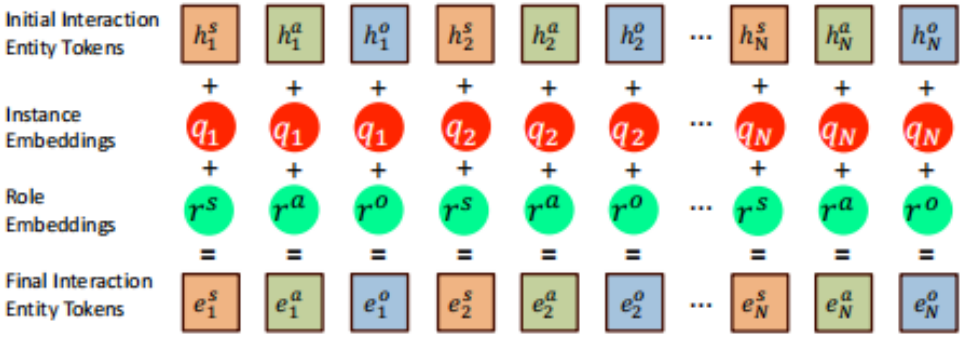

图 4. 交互嵌入。可学习的实例嵌入 q 和角色嵌入 r 被添加到 token 中，以表示主体 s、动作 a 和对象 o 之间复杂的交互关系。

3.4 交互 Transformer

目前主流 T2I 模型已经在大规模的图像-文本对上进行了训练，比如 Stable Diffusion，其在生成逼真的图像方面表现出了卓越的能力。因此，本文的目标是以最小的成本将交互控制整合到这些 T2I 模型中，则保存其中嵌入的宝贵知识至关重要。

在 LDM 模型中，Transformer 块由两个注意层组成，即 (i) 用于视觉 tokens 的自注意层和 (ii) 用于模拟视觉 tokens 和文本 tokens 之间注意的交叉注意层：

$$v = v + \text{SelfAttn}(v); \quad v = v + \text{CrossAttn}(v, c) \tag{13}$$

其中 $v = [v_1, \cdots, v_M]$ 表示为图像的视觉特征 token，c 表示为文本 tokens，且 $c = \tau_\theta(y)$。

交互自注意力 在该模块中，首先需要冻结两个原始注意力层，并在它们之间引入一个新的门控自注意力层，即交互自注意力（见图 5）。这是将交互条件添加到现有的 Transformer 块上，其对视觉和交互 tokens $[v, e^s, e^a, e^o]$ 的连接执行自注意力，重点关注交互关系：

$$v = v + \eta \cdot \tanh \gamma \cdot \text{TS}(\text{SelfAttn}([v, e^s, e^a, e^o])) \tag{14}$$

其中 TS(·) 是一个 Token Slicing 操作，用于仅保留视觉 tokens 的输出并切掉其他 tokens，如图 1 所示；η 是用于控制交互自注意力激活的预定采样的超参数；γ 是一个零初始化的可学习尺度，可逐渐控制门的流量。其中，公式 14 在公式 13 的两个部分之间执行。总而言之，交互自注意力层将交互信息（包括交互、主体和客体边界框）转换为视觉 tokens。

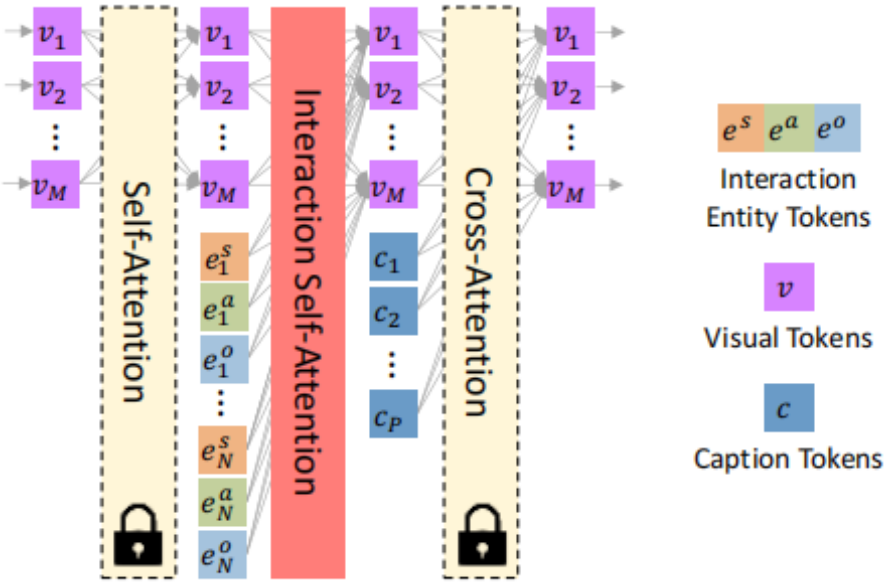

图 5. 交互 Transformer。在视觉 token 自注意力和视觉-文本交叉注意力之间添加了交互自注意力，以结合交互条件。

计划采样 本文作者在对模型进行训练和推理时，令公式 14 中 $\eta = 1$。然而，在某些特殊情况下，新添加的交互自注意力层可能会导致现有 T2I 模型的效果不佳。因此，作者在交互自注意力层上加入了对采样间隔的控制，这可以平衡文本标题和交互控制的水平。从技术上讲，该计划采样方案在推理时间内由超参数 $\omega \in [0, 1]$ 控制。它定义了受交互控制影响的扩散步骤比例，如下所示：

$$\eta = \begin{cases} 1, & t \leq w * T \quad \# \text{ 文本 + 交互} \\ 0, & t > w * T \quad \# \text{ 仅文本} \end{cases} \tag{15}$$

其中 T 是扩散过程的总步数。

3.5 基于交互条件的扩散模型

将交互 Tokenizer、交互嵌入和交互 Transformer 结合起来形成可插入式交互模块，从而实现现有 T2I 扩散模型中的交互控制。采用 LDM 训练目标（公式 2）。将新添加的参数表示为 θ'，扩散模型现在定义为 $\epsilon_{\theta,\theta'}(\cdot)$，其中额外的交互信息由交互标记器 $\tau_{\theta'}(\cdot)$ 处理。因此，模型的总体训练目标是：

$$\min_{\theta'} \mathcal{L}_{\text{InteractDiffusion}} = \mathbb{E}_{z,\epsilon\sim\mathcal{N}(0,\mathbf{I}),t}\left[\|\epsilon - \epsilon_{\theta,\theta'}(z_t, t, \tau_\theta(y), \tau_{\theta'}(\mathcal{D}))\|_2^2\right] \tag{16}$$

4 复现细节

4.1 创新点

交互嵌入优化 在节 3.3 中提到，原文对于交互嵌入的操作是通过将 token 与实例嵌入与角色嵌入相加得出。对于角色嵌入，作者认为所有主体都有一个共同的角色嵌入（人），客

体（物）如同。但实际上，这是有局限性的，所有的主体或客体不一定为同一类角色，主体也可以是物，客体也可以是人（比如：猫跳到人身上）。

因此，我认为可以先通过一个简单的分类器，判别主体是人还是物，来选择添加相应的角色嵌入。通过这种方法，可以更具体地表示某一类角色，能够使用更精确的角色嵌入对每个 token 进行表示，其公式可表示为：

$$r_i^s = R_{so}[\text{Classifier}(c^s)] \tag{17}$$

$$r_i^o = R_{so}[\text{Classifier}(c^o)] \tag{18}$$

其中，R_{so} 表示为主客体角色嵌入的集合，其包括了人物角色嵌入 r^h 和物体角色嵌入 r^o。另外，对于动作角色嵌入，我认为也可以进行细节区别，比如该动作将主客体紧密相连（拿着，吃着），或者该动作使主客体间断相隔（看着，指挥）。其公式可表示为：

$$r_i^o = R_a[\text{Classifier}(c^a)] \tag{19}$$

其中，R_a 表示为动作角色嵌入的集合，其包括了相连动作角色嵌入 r^{cn} 和相隔动作角色嵌入 r^{sp}。因此，最终的实体 token e_i 可表示为：

$$e_i = (h_i^s + q_i + r_i^s, h_i^a + q_i + r_i^a, h_i^o + q_i + r_i^o) \tag{20}$$

通过更改后的交互嵌入方式，我们不仅能够得到良好的人与物的交互，还能更加灵活地拓展到人与人之间的交互，物与人之间的交互，甚至是物与物之间的交互。

交互 Transformer 优化 在节 3.4 中，原文使用了自注意力机制将视觉 tokens 与交互 tokens 进行了组合，但是交互 tokens 作为辅助引导图像生成的部分，使用自注意力机制会过分强调交互的作用，从而影响图像生成质量。尽管作者加入了门控机制，但也难以避免自注意力机制带来的副作用。

因此，我认为这里最好的改进是将自注意力层改为交叉注意力层，将交互 token 作为查询（Query），视觉 token 作为键（Key）和值（Value）。交叉注意力机制会计算交互 token 与视觉 token 之间的相关性权重，得到与交互相关的视觉特征表示，从而生成既包含交互语义又包含图像视觉信息的融合特征。可以得到计算公式：

$$v = v + \eta \cdot \tanh\gamma \cdot \text{TS}(\text{CrossAttn}([v, e^s, e^a, e^o])) \tag{21}$$

模块可移植性研究 交互模块的集成允许对生成过程进行细粒度的交互控制，而无需进行大量的再训练。原文中的模块是在 Stable Diffusion v1-5 上训练与推理的，论文中展示了其在 Stable Diffusion 上的交互性比其它模型的优势，却对该模块移植到其它模型的方案鲜有提及。为此，我尝试将该模块移植到 Anythingv5 模型中，研究该模块的可移植性，如节 5 所示。实验结果发现，模块在成功保持了个性化模型独特的风格属性的前提下，同时提供了改进的交互可控性。

4.2 与已有开源代码对比

本文代码参考了开源代码 https://github.com/jiuntian/interactdiffusion，根据节 4.1 所提出的创新点，其在代码层面的主要改进如下：

```python
class ObjectTypeClassifier(nn.Module):
    def __init__(self, input_dim):
        super(ObjectTypeClassifier, self).__init__()
        self.classifier = nn.Sequential(
            nn.Linear(input_dim, 64),
            nn.ReLU(),
            nn.Linear(64, 2)
        )

    def forward(self, x):
        return self.classifier(x)  # 输出类型人或物

class ActionTypeClassifier(nn.Module):
    def __init__(self, input_dim):
        super(ObjectTypeClassifier, self).__init__()
        self.classifier = nn.Sequential(
            nn.Linear(input_dim, 64),
            nn.ReLU(),
            nn.Linear(64, 2)
        )

    def forward(self, x):
        return self.classifier(x)  # 输出动作类型
```

图 6. 定义了两个新的分类器，用于判别主客体的类型以及动作的类型。

```python
class HOIPositionNetV5(nn.Module):
    def __init__(self, in_dim, out_dim, fourier_freqs=8, max_interactions=30):
        super().__init__()
        self.O_classifier = ObjectTypeClassifier(in_dim) # 物体分类器
        self.A_classifier = ActionTypeClassifier(in_dim) # 动作分类器
```

图 7. 在 HOI 模块中对分类器进行初始化。

```python
# 获取分类结果 (0: 人, 1: 物; 2: 相连动作, 3: 相隔动作)
subject_labels = torch.argmax(self.O_classifier(objs_subject), dim=1)
object_labels = torch.argmax(self.O_classifier(objs_object), dim=1)
action_labels = torch.argmax(self.A_classifier(objs_action), dim=1) + 2

# 根据分类结果进行嵌入
objs_subject = objs_subject + self.position_embedding.emb(torch.tensor(subject_labels).to(objs_subject.device))
objs_object = objs_object + self.position_embedding.emb(torch.tensor(object_labels).to(objs_object.device))
objs_action = objs_action + self.position_embedding.emb(torch.tensor(action_labels).to(objs_action.device))
```

图 8. 根据分类结果对 token 进行角色嵌入。

```python
class GatedSelfAttentionDense(nn.Module):
    def __init__(self, query_dim, context_dim, n_heads, d_head):
        super().__init__()

        self.linear = nn.Linear(context_dim, query_dim)

        #self.attn = SelfAttention(query_dim=query_dim, heads=n_heads, dim_head=d_head)
        self.attn = CrossAttention(query_dim=query_dim, heads=n_heads, dim_head=d_head)

        self.ff = FeedForward(query_dim, glu=True)

        self.norm1 = nn.LayerNorm(query_dim)
        self.norm2 = nn.LayerNorm(query_dim)

        self.register_parameter('alpha_attn', nn.Parameter(torch.tensor(0.)))
        self.register_parameter('alpha_dense', nn.Parameter(torch.tensor(0.)))
        self.scale = 1

    def forward(self, x, objs):
        N_visual = x.shape[1]
        objs = self.linear(objs)
        # x = x + self.scale * torch.tanh(self.alpha_attn) * self.attn(self.norm1(torch.cat(
        #                                                     [x, objs],dim=1)))[:,0:N_visual, :]

        # 改为交叉注意力机制
        x = x + self.scale * torch.tanh(self.alpha_attn) * \
            self.attn(self.norm1(x), self.norm1(objs))[:,0:N_visual, :]

        x = x + self.scale * torch.tanh(self.alpha_dense) * self.ff(self.norm2(x))

        return x
```

图 9. 使用交叉注意力代替自注意力

4.3 实验环境搭建

- 系统：ubuntu16.04.1
- GPU：8 张 P100
- Cuda：12.2
- Python == 3.10.16
- Pytorch == 2.5.2
- 其它必要的包（详见 environment.yml）

4.4 界面分析与使用说明

下载 interactiondiffusion 模型 https://huggingface.co/jiuntian/interactiondiffusion weight/blob/main/interact-diffusion-v1-2.pth

训练 准备数据集（HICO-DET [3]）和预训练模型（Stable Diffusion v1-4 或 v1-5），执行以下命令运行：

CUDA_VISIBLE_DEVICES = 0,1 torchrun –nproc_per_node = 2 main.py –yaml_file configs/hoi_hico_text.yaml –ckpt <existing_gligen_checkpoint>–name test –batch_size = 4 –gradient_accumulation_step 2 –total_iters 500000 –amp true –disable_inference_in_training true –official_ckpt_name <existing SD v1.4/v1.5 checkpoint>

推理 将 inference_batch.py 中的 ckpt.pth 更改为训练的模型。执行以下命令进行推理：

python inference_batch.py –batch_size 1 –folder generated_output –seed 489 –scheduled -sampling 1.0 –half

其中核心的输入数据如下：

- prompt=”a person is touching a cat” # 文本描述
- subject_phrases=[”person”] # 主体
- object_phrases=[”cat”] # 客体
- action_phrases=[”touching”] # 动作
- subject_boxes=[[0.0332, 0.1660, 0.3359, 0.7305]] # 主体边界框角坐标
- object_boxes=[[0.2891, 0.4766, 0.6680, 0.7930]] # 客体边界框角坐标

评估 安装 FGAHOI [12]，并对 HOI 检测得分进行计算。执行以下命令进行评估：

python main.py –backbone swin_tiny –dataset_file hico –resume weights/FGAHOI_Tiny.pth –num_verb_classes 117 –num_obj_classes 80 –output_dir logs –merge –hierarchical_merge –task_merge –eval –hoi_path data/id_generated_output –pretrain_model_path ”.....” –output_dir logs/id-generated-output-t

5 实验结果分析

我使用训练好的 InteractDiffusion 进行推理，该模型是基于 Stable Diffusion v1-5 进行训练的，其结果如图 10，11 第二行所示。为了拓展交互模块的使用，我另外在 Anythingv5 模型（该模型是 Stable Diffusion v1-5 的微调模型，生成的是动漫风格的图像）上嵌入了该可插拔的模块进行实验，其结果如图 10，11 第三行所示。此外，我还分别做了简单交互以及复杂交互的实验，简单交互指的是只有一对交互，复杂交互指的是有多对交互。其中，复杂交互的示例应用到了节 3.3 中的交互实例嵌入，能够更充分地体现该模块的优势。

实验结果表明，该模块能够很好地控制交互的生成，比如当我输入 “a girl is holding her boyfriend’s hands” 时，模型能够很好地生成握手这个交互；再比如当我输入多对交互 “girl is carrying backpack;boy is holding an umbrella” 时，模型也能够同时生成背包和撑伞两个交互；此外，对于 “girl watching TV” 此类相隔动作，模型也能够较好地展示其交互。这些交互的准确生成源于节 4.1 中所提出的交互嵌入优化，使得其能够准确在相应的位置生成对应的物体，对于交互的细节也能够较为清晰地呈现。

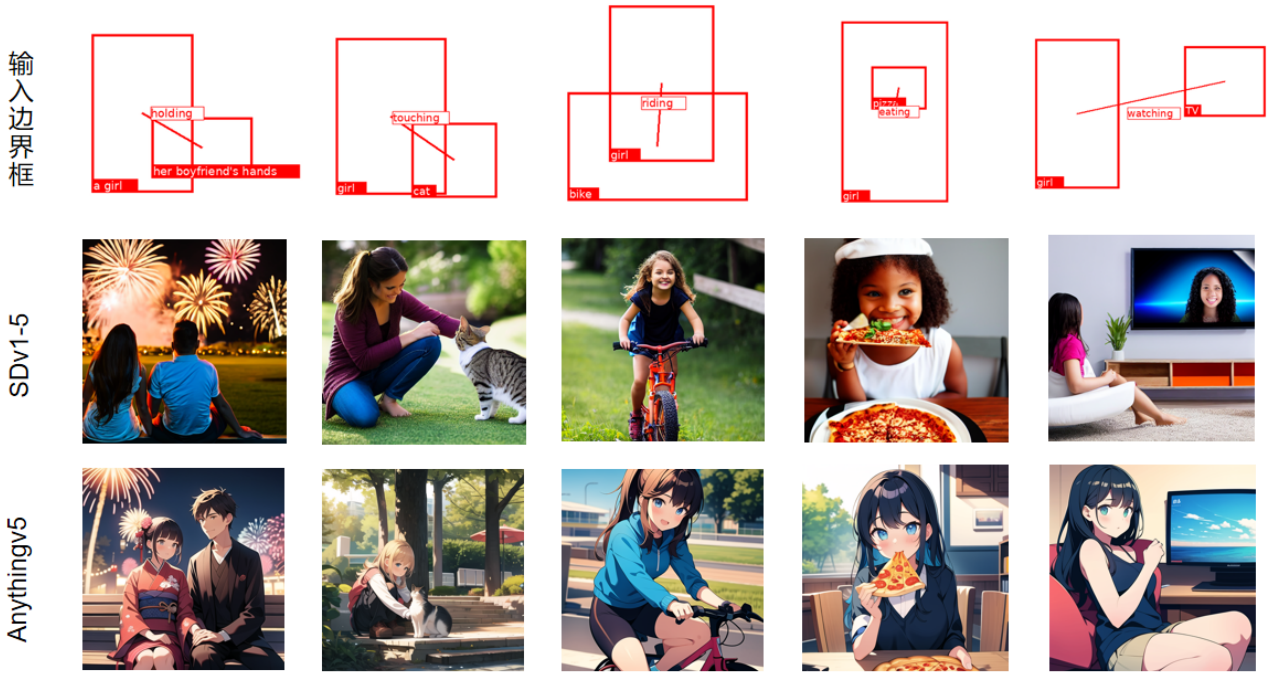

图 10. 简单交互实验结果图

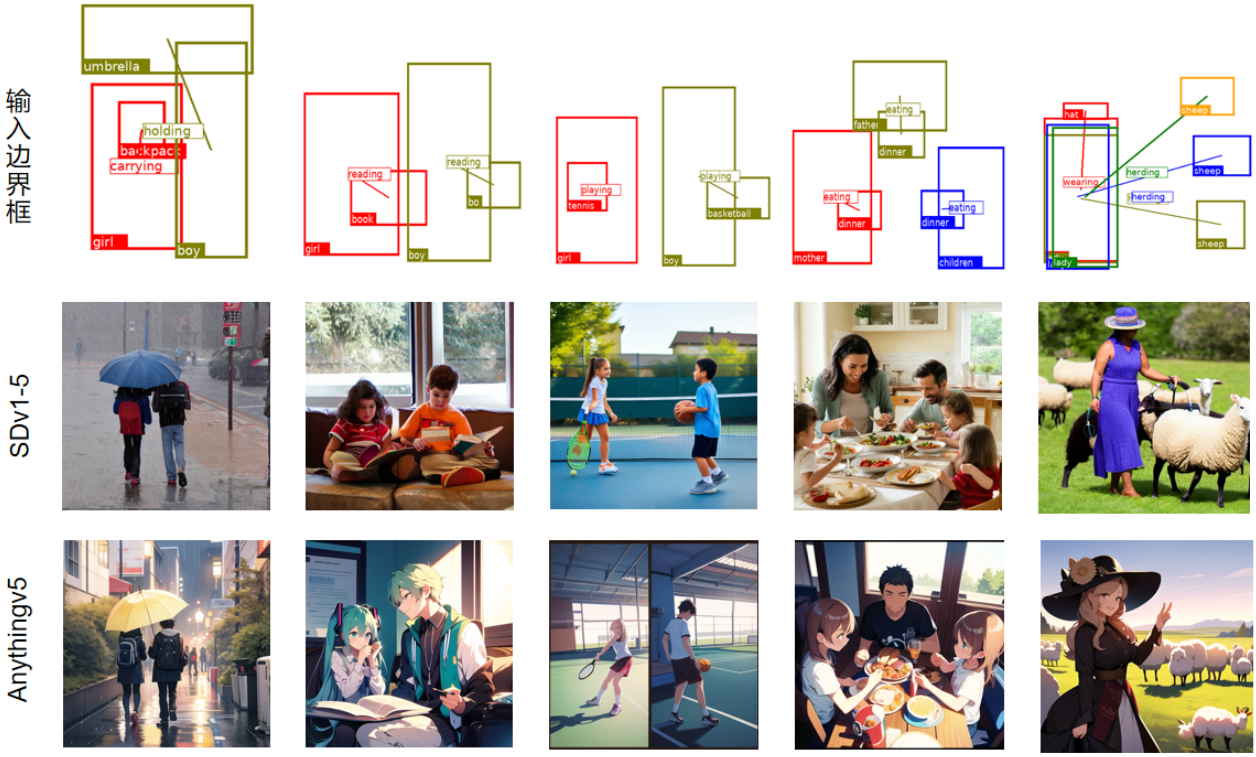

图 11. 复杂交互实验结果图

6 总结与展望

本文提出了一个以交互为条件的 T2I 扩散模型，即 InteractDiffusion，该模型解决了除文本标题之外的生成图像的条件问题。在现有的 T2I 扩散模型中，尽管已经施加了多种控制

(例如文本、图像、布局等)，但控制生成图像中的交互仍然是一项艰巨的挑战。而文章中所提出的可插入交互模块，将交互信息转换成 token，并将交互信息无缝地集成到现有的 T2I 扩散模型中，使得模型的交互能力大大提升。在改进部分，我完善了交互嵌入的更多细节，使得其不仅能够完成人-物交互任务，还能改进各类交互任务的生成质量。实验表明该方法在控制生成内容交互方面的有效性，其性能明显优于最先进的方法。

但该模块仍然存在着一定的局限性，尽管交互能力有了显著的改进，其生成的交互仍然与真实交互存在一些差异，尤其是在细节方面。此外，由于目前的大型预训练模型（CLIP、Stable Diffusion）在预训练阶段以对象为中心，因此缺乏对交互的理解，这阻碍了 InteractDiffusion 在控制交互方面的表现。对于未来的工作，希望能训练出一个包含对象和交互的更加多样化的大型模型，可以提高 InteractDiffusion 的交互可控性。

参考文献

[1] Omri Avrahami, Thomas Hayes, Oran Gafni, Sonal Gupta, Yaniv Taigman, Devi Parikh, Dani Lischinski, Ohad Fried, and Xi Yin. Spatext: Spatio-textual representation for controllable image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18370–18380, 2023.

[2] Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 843–852, 2023.

[3] Yu-Wei Chao, Yunfan Liu, Xieyang Liu, Huayi Zeng, and Jia Deng. Learning to detect human-object interactions. In *2018 ieee winter conference on applications of computer vision (wacv)*, pages 381–389. IEEE, 2018.

[4] Minghao Chen, Iro Laina, and Andrea Vedaldi. Training-free layout control with cross-attention guidance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5343–5353, 2024.

[5] Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[6] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[7] Chen Gao, Si Liu, Defa Zhu, Quan Liu, Jie Cao, Haoqian He, Ran He, and Shuicheng Yan. Interactgan: Learning to generate human-object interaction. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 165–173, 2020.

[8] 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 (CVPR), pages 6180–6189, June 2024.

[9] Tianyu Hua, Hongdong Zheng, Yalong Bai, Wei Zhang, Xiao-Ping Zhang, and Tao Mei. Exploiting relationship for complex-scene image generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1584–1592, 2021.

[10] Yunji Kim, Jiyoung Lee, Jin-Hwa Kim, Jung-Woo Ha, and Jun-Yan Zhu. Dense text-to-image generation with attention modulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7701–7711, 2023.

[11] 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.

[12] Shuailei Ma, Yuefeng Wang, Shanze Wang, and Ying Wei. Fgahoi: Fine-grained anchors for human-object interaction detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[13] 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.

[14] Sicheng Mo, Fangzhou Mu, Kuan Heng Lin, Yanli Liu, Bochen Guan, Yin Li, and Bolei Zhou. Freecontrol: Training-free spatial control of any text-to-image diffusion model with any condition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7465–7475, 2024.

[15] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

[16] 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.

[17] 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.

[18] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.