ControlNet 论文阅读

一、Introduction

1. 拟解决问题：数据少，算力大

二、Related Works

1. Diffusion

DDPM → DDIM → Latent Diffusion Model.

score-based　　解决算力

2. Fine-tuning

(1) HyperNetwork：在预训练模型后再加几层神经网络.

(2) zero convolution

3. Text-to-Image Diffusion

(1) CLIP：基于对比学习的文本和图像的多模态模型.
（对图像打标签、对文本编码）

(2) Disco Diffusion

4. Control of Pretrained Diffusion Model

(1) img2img (Stable Diffusio)：color-level detail variations

(2) inpainting：对图中某个区域进行修改.

5. Img2Img

(1) Taming Transformer.

三、Method.

1. 设计指导

以前控制扩散模型用微调/HyperNetwork，所以采用保护与扩展的思想.（保护Stable Diffusion的权重）.

遗忘 ×　　很难做到逐像素对齐
↑　　↑
微调　　空间控制

2. ControlNet 网络搭建

(1) 令 SD 的 U-Net 里的一个神经网络层记作 F，参数为 Θ.

① 输入是 x，输出是 y.

$$y = F(x;\Theta)$$

② 一个分支是将 Θ 锁定 → Θ_{lock}

另一分支复制相同结构 → $\Theta_{trainable}$。为了可以接收额外输入，该分支的输入为 $x + Z(c)$。 ← Z 是简单的特征提取器.

边缘图 c ↑

③ Zero Convolution.

(二) · 不能直接将 trainable copy 的输出加回去.

· 引入零卷积层 Z → $Z(\cdot;\Theta_{zero})$

☆其中 W、B 初始化为 0.

$$\Downarrow$$

$$y_c = F(x;\Theta_{lock}) + Z(F(x + Z(c);\Theta_{trainable});\Theta_{zero})$$

(2) Why Zero Convolution?

在训练的第一步，零卷积的输出是零 ⇒ $y_c = F(x;\Theta_{lock}) + 0 = y$

which means，在训练开始时，ControlNet 对模型的影响为 0，等价于原本的 SD ⇒ 保证了训练的稳定性.

↑ 逐步调整 Θ_{zero} 和 $\Theta_{trainable}$，逐步注入控制信息.

(3) 复制了什么?

· 只复制了 Encoder 和 Middle Block（共 13 个模块），并没有复制 Decoder. ← 为了省参数

· 注入点：加到 SD 的 decoder 的每一层上。

。ControlNet 的每一层输出，会作为残差加到相应的 SD 层的输出上，然后一起送入下一层的 SD-Decoder 中.

六、训练策略.

(1) 数据构建：(原图 x_0, Prompt, Condition Map (c))

↑ 自动生成

(2) Loss

$$L = E_{x_0, t, c, \varepsilon}[\| \varepsilon - \varepsilon_\theta(x_t, t, c) \|^2]$$

· SD 权重锁死，梯度只会回传到 copy 层和零卷积层。

(3) 突然收敛/顿悟 (Grokking) 现象.

(4) 空文本训练 (Classifier-Free Guidance Support)

· 为了让 ControlNet 在推理时支持调节 Prompt 对生成结果产生影响，训练时必须采用 Dropout 策略：50% 将 Text Prompt 替换为 "" 字符串.

→ 迫使不仅仅依赖文本，学会从 Condition Map 寻找线索.

(5) Zero convolution 中的 baekpropogation.

① 前向：$y = wx + b \xrightarrow[\text{step}]{\text{first}} 0 \cdot x + 0 = 0$ ⇒ 传给 SD 的是 0.

② 反向：· 需要更新权重 w. (根据 L)

Chain Law: $\frac{\partial L}{\partial w} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial w} = \frac{\partial L}{\partial y} \cdot x$

⇒ 只要输入不是 0，并且模型有误差，w 就会获得 ∇，从而更新.

ControlNet 的

(4) ControlNet 的头部有一个卷积网络，用来做降维和匹配通道数与 U-Net 输入一致.

卷积与全连接的联系

- $\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} & w_{15} \\ w_{21} & \cdots & \cdots & \cdots & \\ w_{31} & & & & \\ w_{41} & & & & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}$ ⟶ 计算 y_i 需用到 $x_1 \sim x_5$ 的所有输入信息.

- $[a, b]$ 的卷积核:

$y_1 = ax_1 + bx_2$

$y_2 = ax_2 + bx_3$

$\vdots$

$y_4 = ax_4 + bx_5$

⟶ $\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} a & b & 0 & \\ & a & b & 0 \\ & 0 & a & b \\ & & & a \quad b \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}$

⇒ 区别 ⟨ 稀疏连接 ⇐ 局部感受野 ⇐ Attention 改进
⟨ 参数共享 ⇒ 平移不变性 (对特征位置不敏感)

Conclusion: 卷积是一种被限制了自由度, 但换来了极高 ⟨ 参数少 效率和泛化能力的全连接层.