

NKU 深度学习（高阶课）实验报告



实验名称：_____GAN_____

学 院：_____网络空间安全学院_____

姓 名：_____邢清画_____

学 号：_____2211999_____

专 业：_____物联网工程_____

二〇二五年五月

1 实验要求

1.1 原始版本 GAN

1.1.1 网络结构

生成器（Generator）结构优化：

1. 采用了多层线性层的深度结构（ $128 \rightarrow 256 \rightarrow 512 \rightarrow 1024$ ）
2. 引入了残差连接（Residual Connection），有助于解决深度网络训练中的梯度消失问题
3. 使用了 BatchNorm1d 进行归一化，稳定训练过程
4. 加入了 Dropout 层（ $\text{dropout_rate} = 0.2$ ），减少过拟合
5. 使用 LeakyReLU 激活函数和 Tanh 输出层，有利于生成更自然的图像

判别器（Discriminator）结构优化：

1. 采用了谱归一化（SpectralNorm）技术，这是一种先进的权重归一化方法，能够稳定 GAN 训练
2. 使用了较高的 Dropout 率（0.3）来增强鲁棒性
3. 将网络分为特征提取和分类两个部分，便于进行特征匹配等高级训练技巧

```
Generator(  
  (main): Sequential(  
    (0): Linear(in_features=100, out_features=128, bias=True)  
    (1): LeakyReLU(negative_slope=0.2, inplace=True)  
    (2): Linear(in_features=128, out_features=256, bias=True)  
    (3): BatchNorm1d(256, eps=0.8, momentum=0.1, affine=True, track_running_stats=True)  
    (4): LeakyReLU(negative_slope=0.2, inplace=True)  
    (5): Dropout(p=0.2, inplace=False)  
    (6): Linear(in_features=256, out_features=512, bias=True)  
    (7): BatchNorm1d(512, eps=0.8, momentum=0.1, affine=True, track_running_stats=True)  
    (8): LeakyReLU(negative_slope=0.2, inplace=True)  
    (9): Dropout(p=0.2, inplace=False)  
    (10): Linear(in_features=512, out_features=1024, bias=True)  
    (11): BatchNorm1d(1024, eps=0.8, momentum=0.1, affine=True, track_running_stats=True)  
    (12): LeakyReLU(negative_slope=0.2, inplace=True)  
    (13): Dropout(p=0.2, inplace=False)  
  )  
  (shortcut): Sequential(  
    (0): Linear(in_features=100, out_features=1024, bias=True)  
    (1): BatchNorm1d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
  )  
  (output_layer): Sequential(  
    (0): Linear(in_features=1024, out_features=784, bias=True)  
    (1): Tanh()  
  )  
)
```

(a) 生成器结构

```
Discriminator(  
  (feature_extractor): Sequential(  
    (0): SpectralNorm(  
      (module): Linear(in_features=784, out_features=512, bias=True)  
    )  
    (1): LeakyReLU(negative_slope=0.2, inplace=True)  
    (2): Dropout(p=0.3, inplace=False)  
    (3): SpectralNorm(  
      (module): Linear(in_features=512, out_features=256, bias=True)  
    )  
    (4): LeakyReLU(negative_slope=0.2, inplace=True)  
    (5): Dropout(p=0.3, inplace=False)  
  )  
  (classifier): Sequential(  
    (0): SpectralNorm(  
      (module): Linear(in_features=256, out_features=1, bias=True)  
    )  
    (1): Sigmoid()  
  )  
)
```

(b) 判别器结构

图 1: 生成对抗网络中的生成器与判别器结构示意图

1.1.2 训练结果

图 2 展示了 GAN 模型在训练过程中的 loss 曲线。初期生成器与判别器的损失值存在剧烈波动，随着训练的深入，两者的 loss 均趋于稳定。其中，判别器的损失波动幅度较小，说明其具备稳定的真伪样本识别能力；生成器的损失尽管存在轻微起伏，但整体呈下降趋势，表明其生成样本质量逐步提升，模型整体训练效果良好。

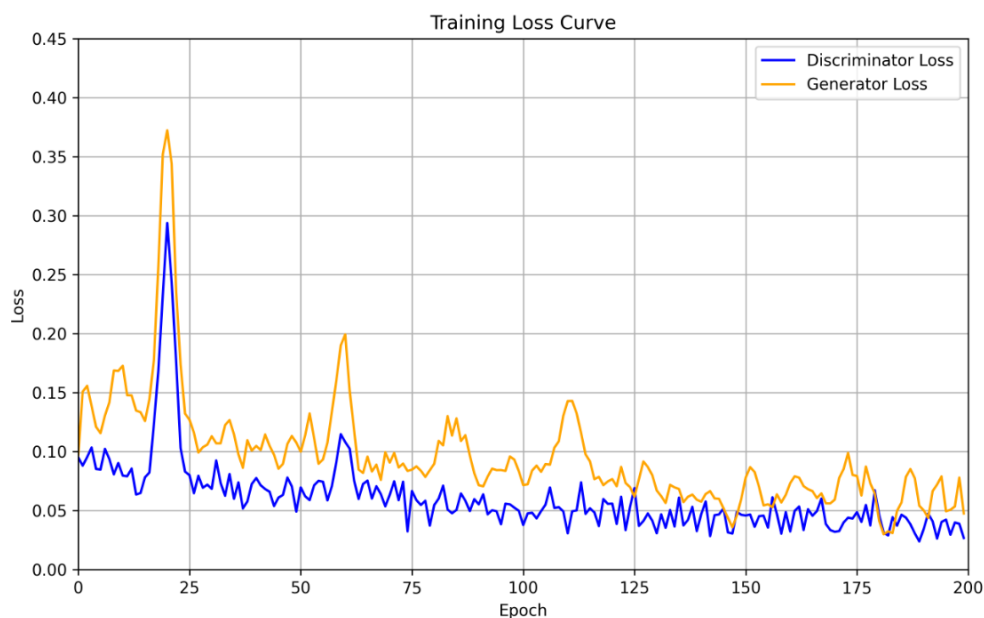


图 2: GAN 模型训练 loss 曲线

自定义一组随机数，生成 8 张图：



图 3: 自定义 8 个随机数原图像及对应的生成图像

针对自定义的 100 个随机数，自由挑选 5 个随机数，查看调整每个随机数时，生成图像的变化（每个随机数调整 3 次，共生成 15x8 张图），总结调整每个随机数时，生成图像发生的变化：

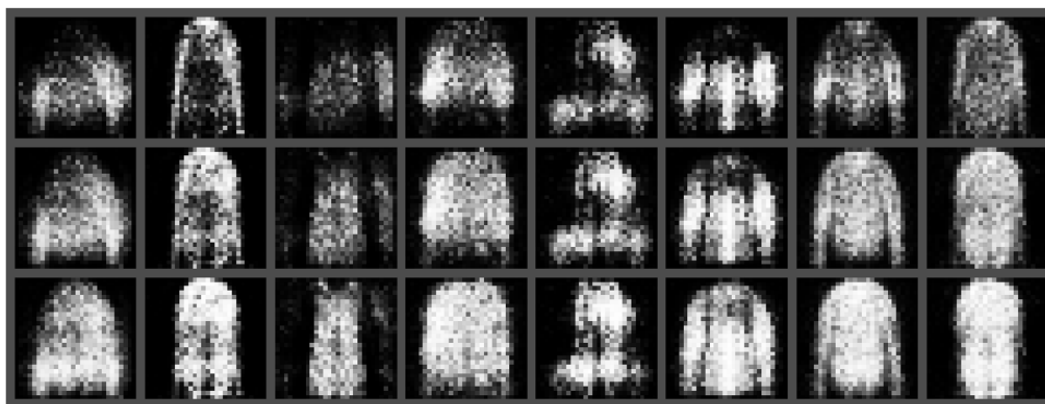


图 4: 维度 5 调整值-3,0,3

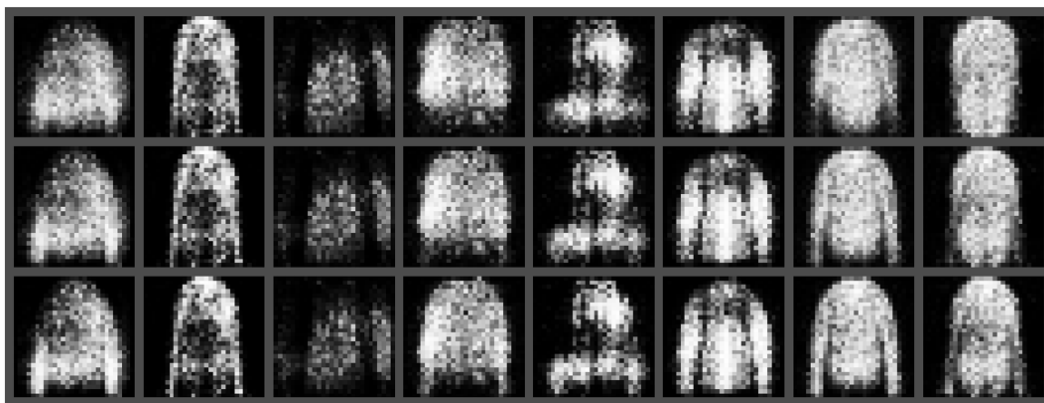


图 5: 维度 20 调整值-3,0,3

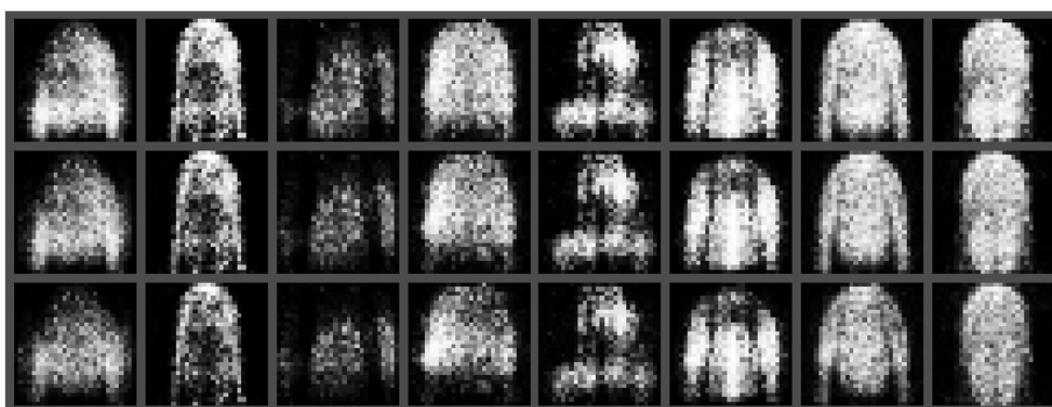


图 6: 维度 40 调整值-3,0,3

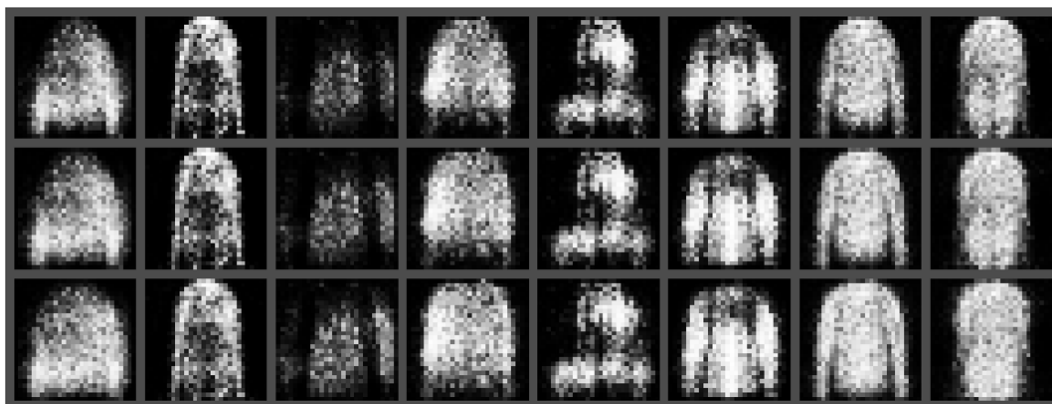


图 7: 维度 60 调整值-3,0,3

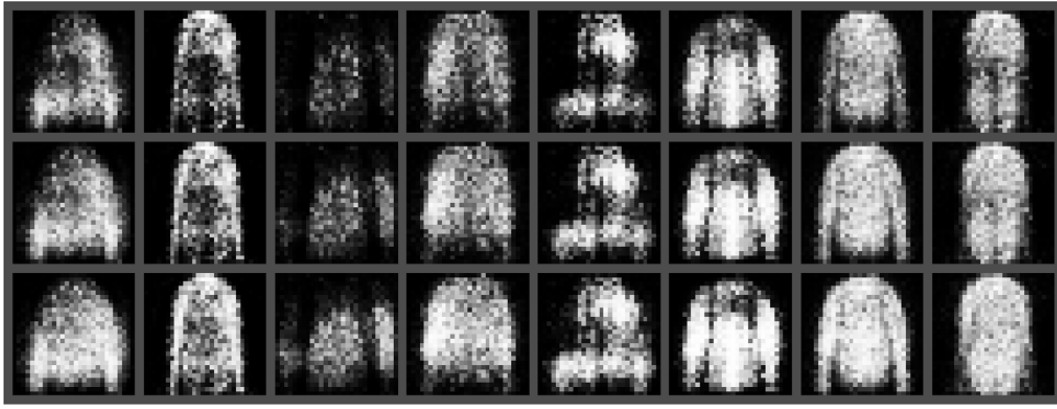


图 8: 维度 80 调整值-3,0,3

1.1.3 潜在向量扰动实验分析

从固定的 100 维潜在向量中，随机选取了 5 个具有代表性的维度（第 5、20、40、60 和 80 维），并分别赋予 -3 、 0 和 3 三种数值，对每组包含 8 个样本的潜在向量进行扰动。

- **维度调节效果差异显著：**

- **第 5 维**对图像的 **整体轮廓和结构感**影响显著。低值时图像边缘模糊、形状不完整，高值时轮廓清晰、主体对称，表明该维度可能编码了“形状完整性”信息。
- **第 20 维**主要调控图像的 **亮度与对比度**。随着数值增大，图像变得更加明亮，背景干净，信息细节更丰富，体现出该维度与图像的光照相关。
- **第 40 维**敏感于 **局部纹理与细节**，如衣领、褶皱、边缘线条等，表明其潜在地控制了局部结构的清晰度。
- **第 60 维**的变化常导致 **风格和背景变化**，可能关联的是整体图像风格或背景噪声控制。
- **第 80 维**控制 **衣物形态的比例变化**：低值生成的服饰偏修长，高值趋向宽大甚至外套样式，反映出某种高层语义的形状调整功能。

- **潜在空间具有良好的连续性与可控性：**

从图像变化中可以看出，潜在向量中多数维度的扰动在图像上引起的是 **平滑且可解释的连续变化**，表现出 GAN 模型在这些方向上具备较好的生成稳定性与语义解耦能力。

- **存在响应异常或训练盲区：**

在部分样本中，扰动某些维度后图像会出现严重模糊、饱和或失真现象，如第 3 列与第 4 列，说明模型对某些潜在方向尚未充分学习，存在 **模式崩溃**或 **响应不稳定的问题**。

1.2 DCGAN

1.2.1 网络结构

```
DCGANGenerator(  
    (11): Sequential(  
      (0): Linear(in_features=100, out_features=6272, bias=True)  
    )  
    (conv_blocks): Sequential(  
      (0): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
      (1): Upsample(scale_factor=2.0, mode='nearest')  
      (2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))  
      (3): BatchNorm2d(128, eps=0.8, momentum=0.1, affine=True, track_running_stats=True)  
      (4): LeakyReLU(negative_slope=0.2, inplace=True)  
      (5): Upsample(scale_factor=2.0, mode='nearest')  
      (6): Conv2d(128, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))  
      (7): BatchNorm2d(64, eps=0.8, momentum=0.1, affine=True, track_running_stats=True)  
      (8): LeakyReLU(negative_slope=0.2, inplace=True)  
      (9): Conv2d(64, 1, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))  
      (10): Tanh()  
    )  
  )  
)
```

图 9: DCGAN 生成器网络结构

```
DCGANDiscriminator(  
    (model): Sequential(  
      (0): Conv2d(1, 16, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))  
      (1): LeakyReLU(negative_slope=0.2, inplace=True)  
      (2): Dropout2d(p=0.25, inplace=False)  
      (3): Conv2d(16, 32, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))  
      (4): LeakyReLU(negative_slope=0.2, inplace=True)  
      (5): Dropout2d(p=0.25, inplace=False)  
      (6): BatchNorm2d(32, eps=0.8, momentum=0.1, affine=True, track_running_stats=True)  
      (7): Conv2d(32, 64, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))  
      (8): LeakyReLU(negative_slope=0.2, inplace=True)  
      (9): Dropout2d(p=0.25, inplace=False)  
      (10): BatchNorm2d(64, eps=0.8, momentum=0.1, affine=True, track_running_stats=True)  
      (11): Conv2d(64, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))  
      (12): LeakyReLU(negative_slope=0.2, inplace=True)  
      (13): Dropout2d(p=0.25, inplace=False)  
      (14): BatchNorm2d(128, eps=0.8, momentum=0.1, affine=True, track_running_stats=True)  
    )  
    (adv_layer): Sequential(  
      (0): Linear(in_features=512, out_features=1, bias=True)  
      (1): Sigmoid()  
    )  
  )  
)
```

图 10: DCGAN 判别器网络结构

1.2.2 训练结果

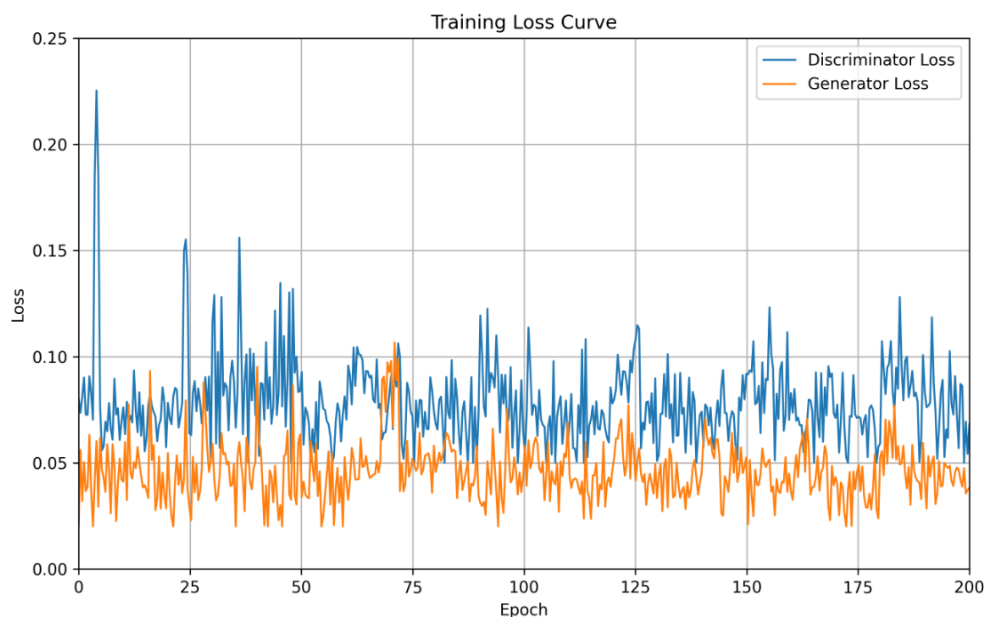


图 11: DCGAN 的训练 loss 曲线

判别器 loss 在前 20 个 epoch 内波动最大，最高逼近 0.22，随后收敛到 0.05–0.10 区间并保持小幅震荡，说明判别器很快学会区分真伪样本并维持稳定判别能力。

生成器 loss 整体更低，主要分布在 0.03–0.06 之间，虽然频繁起伏但振幅有限，表明其持续尝试迷惑判别器且未出现急剧恶化。两条曲线长期交替波动而无一方彻底压制对方，反映出 DCGAN 训练已达到较稳的对抗平衡，没有出现模式崩溃现象

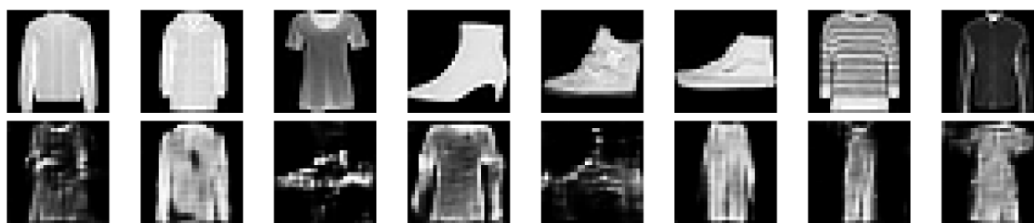


图 12: DCGAN 随机生成 8 张对比图

实验结果揭示，换成 DCGAN 框架后，模型表现的确上了一个台阶。首先，损失曲线几乎不再“大起大落”，判别器和生成器的损失曲线起伏远小于普通 GAN，说明 DCGAN 更容易找到判别器和生成器的平衡点。其次，200 epoch 时生成的样本边缘利落、纹理连贯，和真实图像放在一起也不显突兀，在轮廓锐利度、纹理细节及样式一致性方面均胜过普通 GAN；反观基础 GAN，仍会出现形变或噪点。可见在稳定性与生成质量两方面，DCGAN 都明显占优。

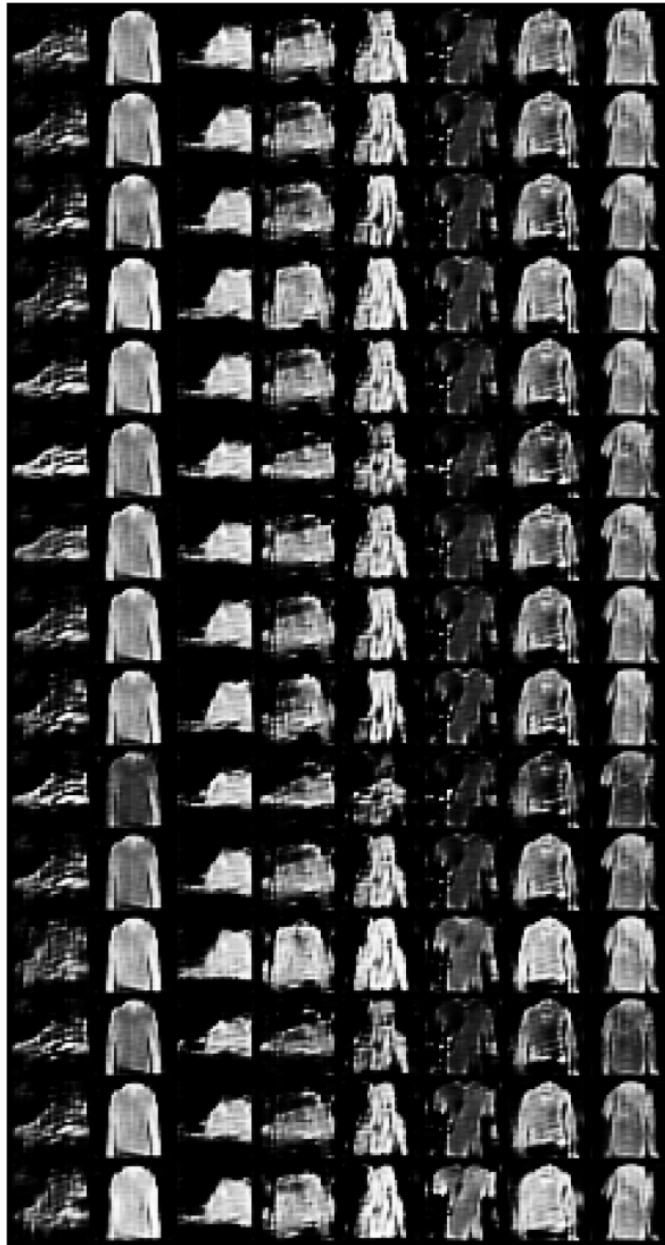


图 13: DCGAN 随机数变化生成图对比

潜在向量是 DCGAN 中生成器的输入，用于从随机噪声中提取图像的隐含语义表示。每一个维度可以理解为控制图像某种潜在属性的“调节因子”，如亮度、姿态、轮廓、纹理、形状等。不同维度之间可能存在解耦，也可能耦合成复杂的语义组合。DCGAN 生成的图像在局部结构、纹理细节、姿态或亮度上出现变化，但整体类别和轮廓通常保持稳定这表明其模型能够在保持高层语义一致的前提下，对潜在空间中的微小变化作出平滑响应，从而实现图像的细致可控生成。

1.3 实验总结：潜在向量扰动对生成图像的影响

潜在向量与维度的含义说明：

不同随机数的选择对生成图像具有决定性影响。横向每一列代表一组随机采样的潜在向量，控制了生成图像的整体类别和基础结构。而在每列内部，通过仅调整原始向量中的部分维度（即在某些维度上进行连续扰动），生成图像在局部结构、纹理细节、姿态或亮度上出现变化。

实验观察与分析结果如下：

- **不同维度的扰动在图像生成上具有明显的语义对应性：**如某个维度主导图像亮度，另一个维度可能决定服饰结构或类别，体现了模型对部分维度的语义学习能力。
- **多数维度的扰动导致图像变化连续平滑，**表现出良好的插值能力，说明潜在空间具有结构性，模型能够稳定应对局部扰动。
- **极少数维度存在变化剧烈的现象，**可能是“高语义”维度，如切换衣服款式、性别特征等，其对生成图像具有全局影响。

潜在向量各维度在 DCGAN 的生成过程中的作用并非等价，而是各自学习到了不同类型的语义控制能力。模型在局部扰动下能够产生连续、合理的图像变化，体现出一定的可解释性和可控性，为后续可控图像生成与语义因子解耦研究提供了实践基础。