HW4

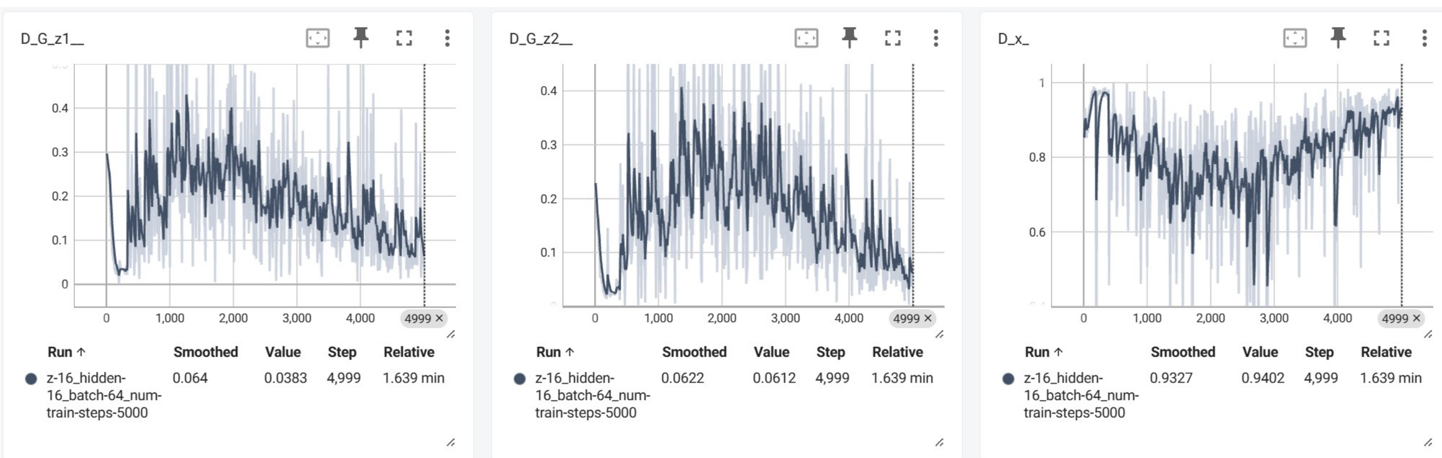
张驰 2022010754 zhang-ch22@mails.tsinghua.edu.cn

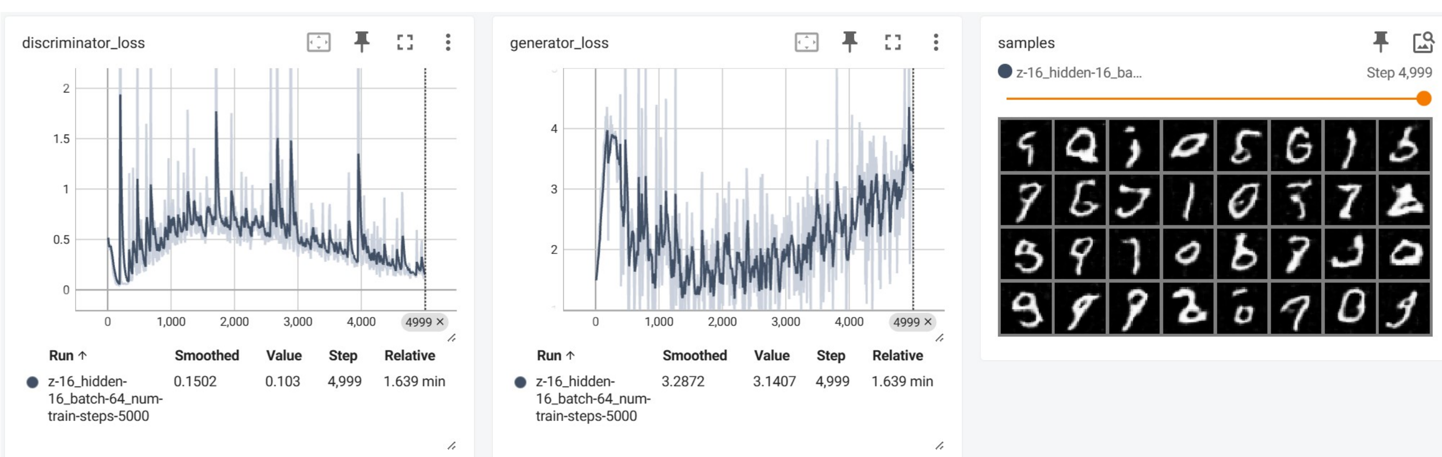
1. Model Training

I trained the GAN model with latent dimension 16/100 and hidden dimension 16/100 respectively. The training curves are as follows:

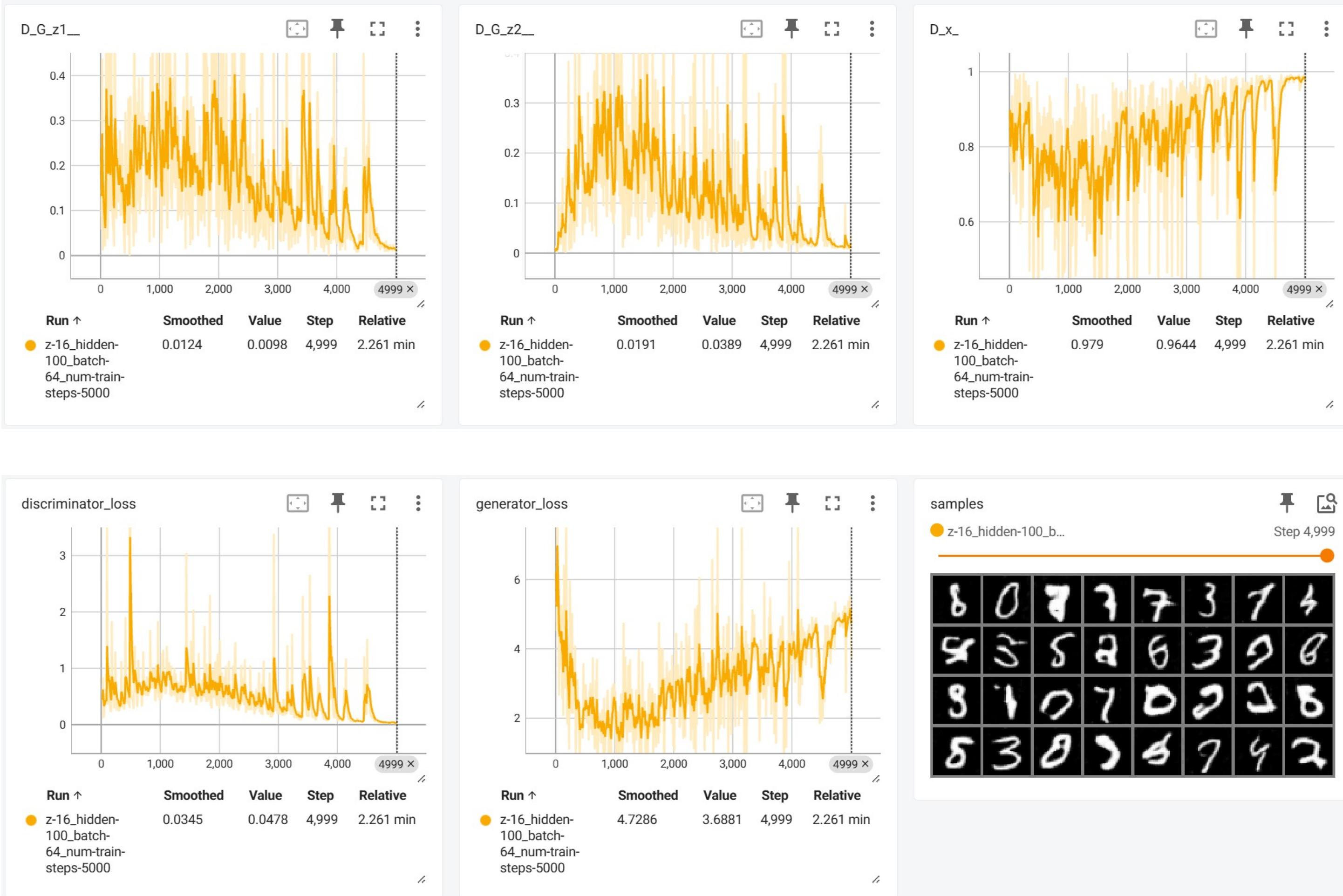
> All training curves and generated figures used models with `num_train_steps = 5000`

Latent dimension 16, hidden dimension 16

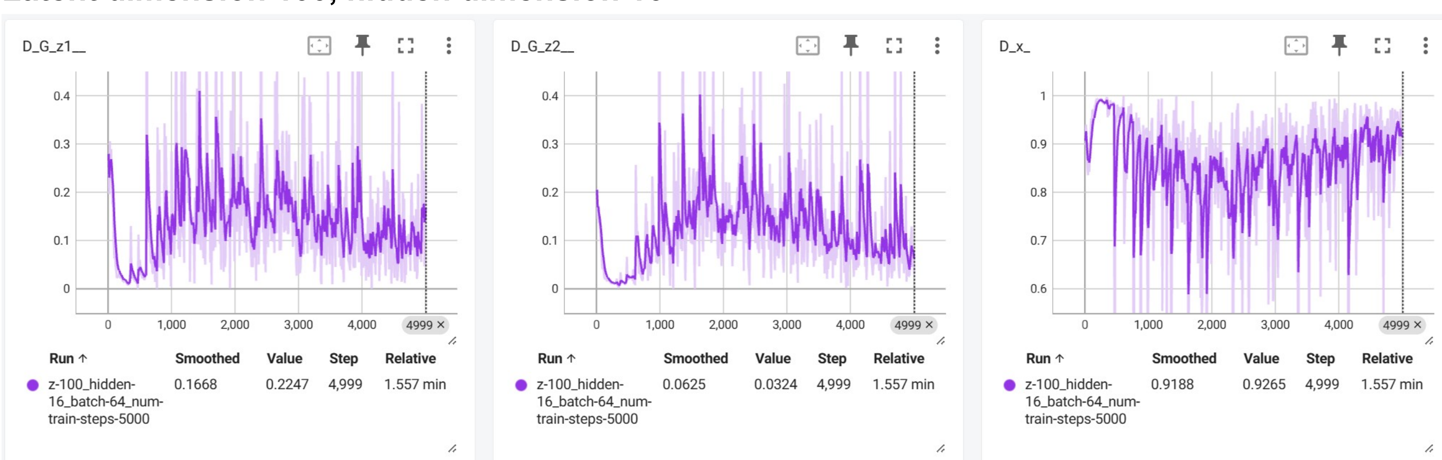


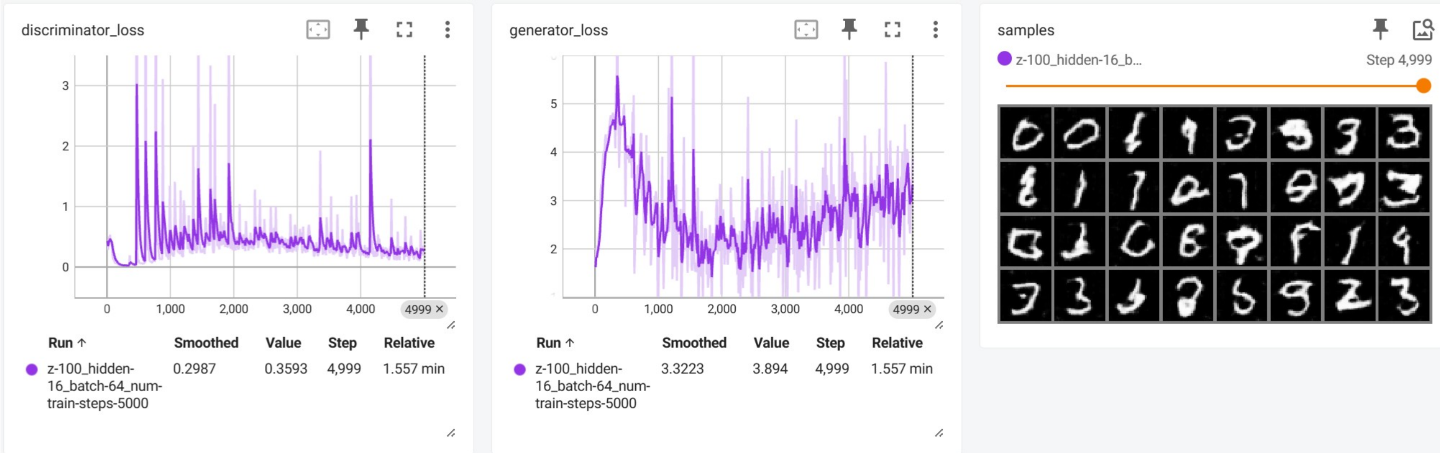


Latent dimension 16, hidden dimension 100

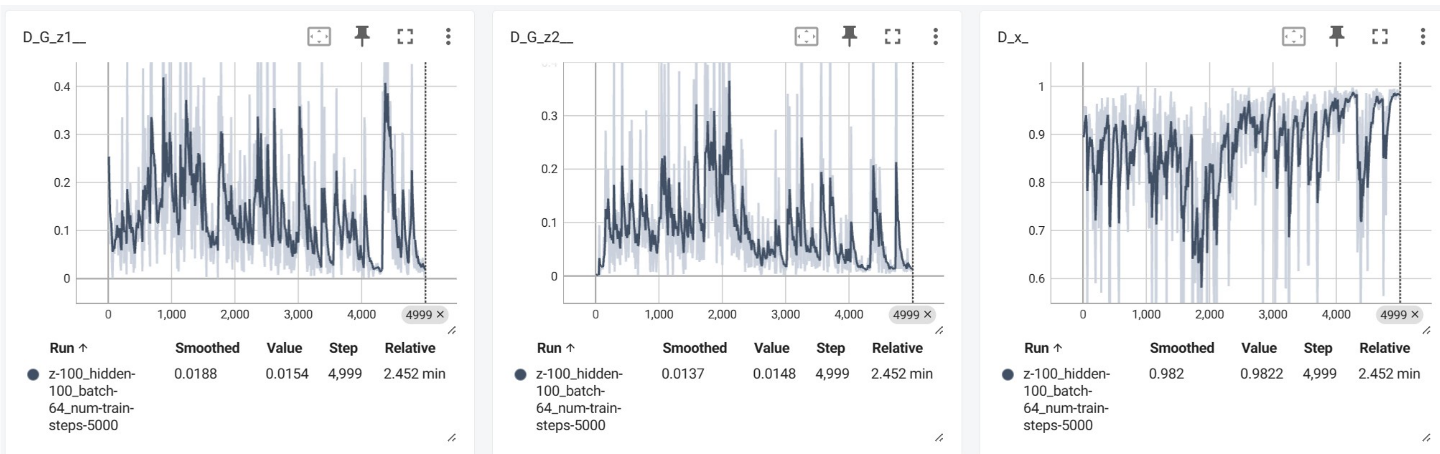


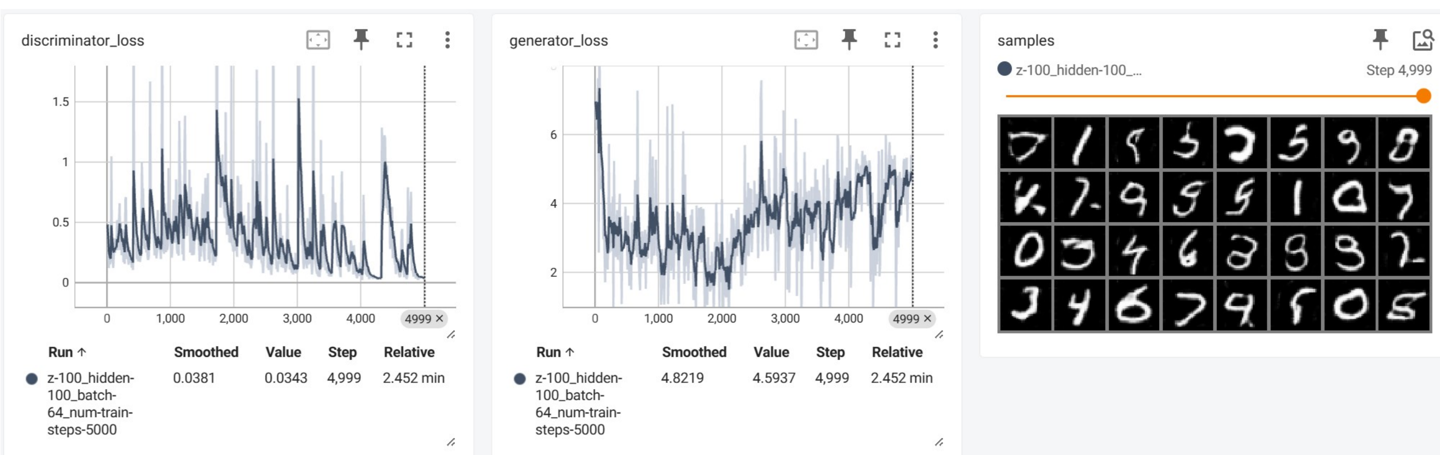
Latent dimension 100, hidden dimension 16





Latent dimension 100, hidden dimension 100





From the graphs we can see, unlike the previous neural networks trained in homework, The loss of GAN displays a significant degree of oscillation, and convergence isn't significant. (In the ideal case, D_G_z and D_x should be similar, as the discriminator cannot distinguish the generated data from the real data.)

2. FID Scores

In order to assess the performance of models with different parameters, I conducted training sessions with varying values for latent_dim and hidden_dim, spanning across 9, 16, 36, 64, 100, and 144. Subsequently, I evaluated their FID scores. The results are as follows:

Latent Dim \ Hidden Dim	9	16	36	64	100	144
9	102.02	109.50	70.55	54.79	60.08	43.91
16	82.83	89.38	58.78	**34.40**	41.75	54.92
36	72.70	103.91	48.44	*35.22*	68.61	42.09
64	73.12	80.78	43.76	57.49	*35.30*	*34.81*
100	97.89	80.50	44.90	43.14	48.77	43.12
144	76.79	118.63	42.45	*35.94*	40.96	39.83

3. Influence of Latent Dimension and Hidden Dimension

From the table above, we observe an overall negative correlation between hidden_dimension and FID score. This can be interpreted as follows: as the hidden_dimension increases, both the generator and discriminator have more parameters, allowing them to capture more data features and generate higher-quality images. However, in some cases, excessively large hidden_dimension values may lead to an increase in the FID score. This may be due to the unbalanced improvement of the generator and discriminator's abilities.

On the other hand, the latent_dimension does not seem to have a similarly significant impact on the FID score. When hidden_dim is small, an excessively large latent_dim can introduce too much randomness into the model, leading to instability of the model performance. Conversely, when hidden_dim is large, since latent_dim only affects the parameters of the first layer, and the total parameter count of the model is on the order of $O(hidden\_dim^2 + hidden\_dim \times latent\_dim)$, the impact of changing latent_dim becomes relatively smaller. So in both cases, increasing latent_dim does not significantly improve the model performance.

4. Nash Equilibrium

When the generator G is given, the optimal solution for the discriminator should minimize the following loss function:

$$L = log(D(x)) + log(1 - D(G(z)))$$

Where $x \sim D_{data}$ (the distribution of the real images) and $z \sim N(0,1)$. Ideally, when our x can span the entire domain, we need to minimize the following function:

$$\begin{aligned} V &= E_{x\sim D_{data}}(log(D(x))) + E_{z\sim N(0,1)}(log(1 - D(G(z)))) \\ &= E_{x\sim D_{data}}(log(D(x))) + E_{x\sim D_G}(log(1 - D(x))) \\ &= \int p_{data}(x)log(D(x))dx + \int p_G(x)log(1 - D(x))dx \\ &= \int (p_{data}(x)log(D(x)) + p_G(x)log(1 - D(x)))dx \end{aligned}$$

So, we should have:

$$\begin{aligned} \frac{\partial}{\partial D(x)}V &= \frac{\partial}{\partial D(x)}\int (p_{data}(x)log(D(x)) + p_G(x)log(1 - D(x)))dx \\ &= \int \frac{\partial}{\partial D(x)}(p_{data}(x)log(D(x)) + p_G(x)log(1 - D(x)))dx \\ &= \int (\frac{p_{data}(x)}{D(x)} - \frac{p_G(x)}{1 - D(x)})dx = 0 \end{aligned}$$

We can achieve this by setting

$$\frac{p_{data}(x)}{D(x)} = \frac{p_G(x)}{1 - D(x)}, \forall x.$$

So, we have:

$$D(x) = \frac{p_{data}(x)}{p_{data}(x) + p_G(x)}, \forall x.$$

From the training curves we can see, the discriminator easily reached its optimal solution. In the four models above and in the rest 32 models of settings in the above table (whose training graphs are not shown here), all `D_G_z` converge to near 0 and `D_x` converge to near 1. This means that the discriminator can easily tell the difference between the generated data and the real data.

When the Nash Equilibrium is reached, the generator should generate data that is indistinguishable from the real data, i.e.

$$D(G(z)) = D(x)$$

. In this case, `D_G_z` and `D_x` should be both near 0.5 as the discriminator tell the real and fake images apart. From our training curves, the system does not reach this state. The discriminator always outpowers the generator.

5. Linear Interpolation

I choose the `latent_dim = 16` and `hidden_dim = 64` model (which achieves the lowest FID score) to do linear interpolation. The results are as follows:

The interpolated figures show a smooth transition between the two digits. This indicates that linear patterns in the latent space can be reflected in the generated images, and that the generator can learn the underlying features of the images.

I also tried linear extrapolation. The results (extrapolation on range(-1,2)) are as follows:

Interesting patterns occur as the generator tries to "eliminate" the features of one digit from another. This indicates that the generator is able to "subtract" the learned features from the generated images.

6. Mode Collapse

Still using the `latent_dim = 16` and `hidden_dim = 64` model, I tried to generate 100 images randomly. The result is as follows:

I counted the occurences of each digit, and the respective counts are:

Digit	0	1	2	3	4	5	6	7	8	9
Count	15	7	6	16	2	3	10	10	12	19

Although the generated images does not collapse to one or two digits, the distribution of the generated images is not uniform. The mode collapse problem is present in this model.