1. Introduction

   In this assignment, we need to implement conditional video prediction in a CVAE-based model. CVAE allows us to generate future frames of a video sequence conditioned on both the past frames and some additional conditional information, such as labels or attributes. The architecture of a CVAE for video prediction consists of an encoder network, a latent space, and a decoder network. Training a CVAE involves maximizing the Evidence Lower Bound (ELBO) objective, which consists of two main components: the reconstruction loss and the KL divergence. The reconstruction loss measures how well the generated frames match the ground truth future frames. The KL divergence measures the difference between the learned latent distribution and a Gaussian distribution in the latent space.

2. Implementation

   (1) How do you write your training protocol

   x: image, p: label

   First, I put x1 into RGB_Encoder, and p2 into label_Encoder to extract features from the input data. Besides, put x2, p2 into Gaussian_Predictor to generate z(latent variables), mu, logvar. Second, put the encoder output and z into Decoder_Fusion to generate reconstructed data. Third, let Generator to output the predicted image. From the second loop to the end, if teacher forcing is true, use the ground truth to be the input image, otherwise, use the generative image to be input. In each epoch, use mu, logvar to do kl criterion, ground truth and the generative image to do mse criterion. And compute the loss by the formula mse + kl * beta. The beta is generate by kl_annealing. Finally, backward to update the parameter weight.

   (2) How do you implement reparameterization tricks

   Sample exp from a standard normal distribution.

   Reparameterize z = mu + logvar * exp.

   By reparameterizing z in this way and backpropagate gradients through mu and logvar without being hindered by the sampling operation.

   (3) How do you set your teacher forcing strategy

   I use the adaptive Teacher Forcing to monitor the model's performance during training and dynamically adjust the teacher forcing ratio. If the model's performance is improving, reduce the teacher forcing rate to encourage the model to rely on its own predictions. Conversely, if the performance degrades, you can increase teacher forcing to stabilize training.

   (4) How do you set your kl annealing ratio

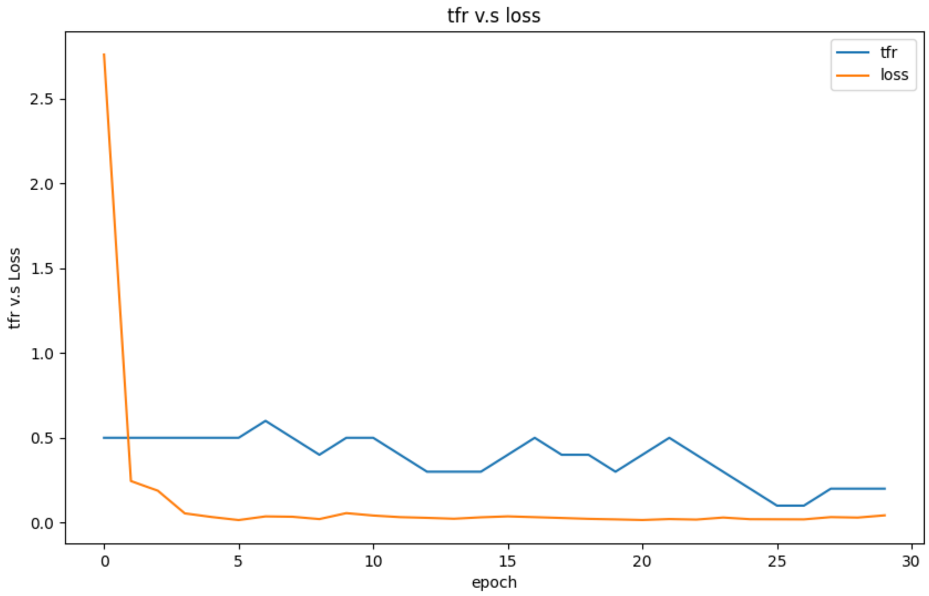
   Cycle linear: periodically and regularly adjust the ratio within a cycle, gradually increasing it in each cycle. This helps the model gradually adapt to the influence of KL divergence in each cycle.

Monotonic: gradually increasing the KL annealing ratio over the course of training epochs. This allows you to initially emphasize the reconstruction error, subsequently shifting focus to the KL divergence term in the loss function.

3. Analysis & Discussion

(1) Plot Teacher forcing ratio

a. Analysis & compare with the loss curve



In the initial five epochs, I set a 50% probability of using teacher forcing, aiming to rapidly reduce the initially high loss. After the fifth epoch, if 'teacher forcing' is set to true, adjustments will be made based on whether the validation PSNR improves compared to the previous epoch. If there's an improvement, the teacher forcing ratio will be decreased by 0.1. This adjustment signifies that the current model's performance surpasses the previous one, indicating the need to reduce the probability of relying on teacher forcing. This allows the predicted images to drive the subsequent model refinements. And because of using teacher forcing, the loss will decrease. The reason is that when training model, if we use the ground truth to be the input, the output will close to the ground truth.

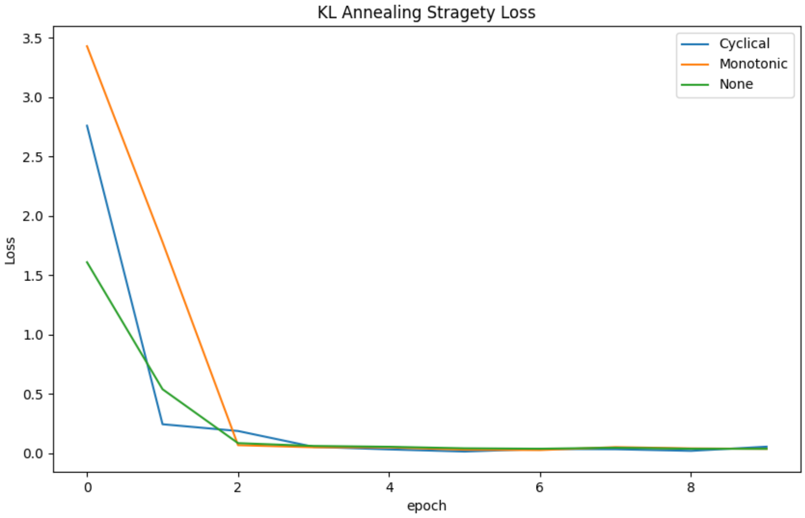
Discuss:

Why does using teacher forcing result in a low loss during training, but a poor PSNR during testing?

Idea:

Using teacher forcing might expose the model to different scenarios during training and testing. With teacher forcing, the decoder's input comes from the real target sequence, while during testing, the decoder relies on its own predictions. This exposure bias can make it challenging for the model to handle situations different from training. Besides, Training and testing sequences may have varying lengths, posing different challenges to the model during testing. Teacher forcing might facilitate handling shorter sequences during training, but generating longer sequences during testing could pose difficulties for the model.

(2) Plot the loss curve while training with different. Analysis the difference between them



The total loss typically consists of two components: the reconstruction loss and the KL divergence. The reconstruction loss measures the similarity between the data generated by the model and the actual data, while the KL divergence quantifies the difference between the distribution of latent variables and a prior distribution. The purpose of the KL annealing strategy is to gradually increase the impact of the KL divergence during the training process, aiding the model in better learning the distribution of latent variables. This strategy is implemented using a weight parameter, which is gradually multiplied by the KL divergence term, gradually incorporating it into the loss calculation.

a. With KL annealing (Monotonic)

The annealing process might involve gradually increasing the weight, then decreasing it, and repeating this cycle multiple times during training. This approach can be useful for avoiding local optima and aiding the model in escaping suboptimal solutions.

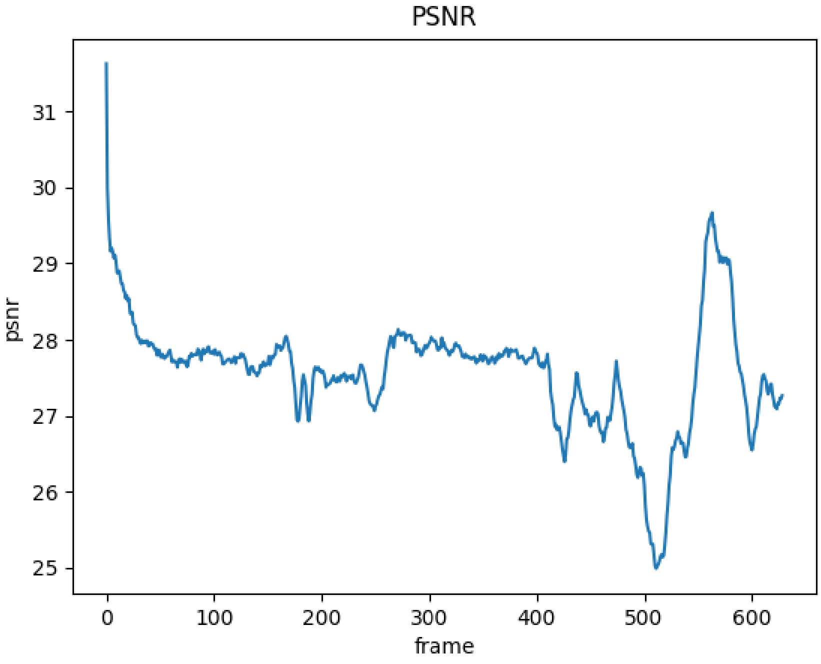
b. With KL annealing (Cyclical)

The weight might be increased or decreased in a consistent manner without any cyclical pattern. This type of annealing might be employed when a gradual and steady transition is desired without the fluctuations of cyclical annealing.

c. Without KL annealing

the weight of the KL divergence term remains constant or is not explicitly adjusted during training. Without KL annealing, the model's learning process relies solely on the initial configuration of the KL divergence term, which might affect the convergence behavior and the quality of the learned latent space representation.

(3) Plot the PSNR-per frame diagram in validation data



At the beginning, the PSNR value is very high. The reason is that we use the ground truth image to be the input, but later the PSNR decreases very fast, this is caused by we only can use the predicted image as the input.

Discuss:

Are there frames where the model performs exceptionally well or poorly? What factors could contribute to these variations?

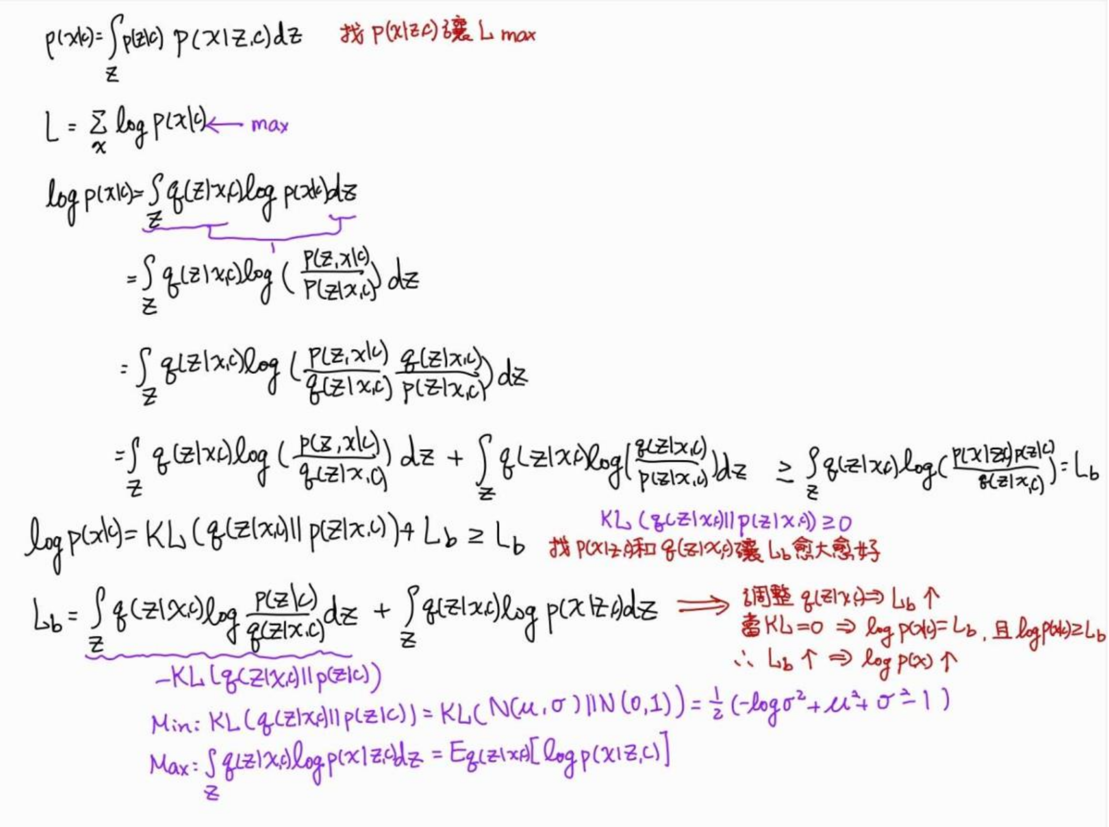
Idea:

Good performance: Certain frames may inherently contain blurriness or noise in the original data, making the generated predictions more similar to the ground truth frames and resulting in higher PSNR values. The other reason is that the video sequence contains simple motion patterns, the model may find it easier to predict these frames accurately, leading to higher PSNR values.

Poor performance:

Frames with low contrast or subtle grayscale variations may pose difficulties for the model to capture fine details accurately. The another reason is that some frames might require contextual information for accurate prediction, and the model may struggle to incorporate the necessary context. Other, The model's performance can be influenced by the training data. If the training data lacks certain types of motion or scenes, the model may exhibit poor performance in those scenarios.

(4) Derivate conditional VAE formula



$$p(x|c)=\int_z p(z|c)\,p(x|z,c)\,dz$$ 找 $P(x|z,c)$ 讓 L max

$$L=\sum_x \log p(x|c) \leftarrow$$ max

$$\log p(x|c)=\underbrace{\int_z q(z|x,c)\log p(x|c)\,dz}$$

$$=\int_z q(z|x,c)\log\left(\frac{P(z,x|c)}{P(z|x,c)}\right)dz$$

$$=\int_z q(z|x,c)\log\left(\frac{P(z,x|c)}{q(z|x,c)}\frac{q(z|x,c)}{p(z|x,c)}\right)dz$$

$$=\int_z q(z|x,c)\log\left(\frac{p(z,x|c)}{q(z|x,c)}\right)dz+\int_z q(z|x,c)\log\left(\frac{q(z|x,c)}{p(z|x,c)}\right)dz \geq \int_z q(z|x,c)\log\left(\frac{p(x|z,c)p(z|c)}{q(z|x,c)}\right)=L_b$$

$$\log p(x|c)=KL(q(z|x,c)\|p(z|x,c))+L_b\geq L_b$$

$KL(q(z|x,c)\|p(z|x,c))\geq 0$

找 $P(x|z,c)$和 $q(z|x,c)$讓 L_b愈大愈好

$$L_b=\underbrace{\int_z q(z|x,c)\log\frac{P(z|c)}{q(z|x,c)}dz}_{-KL(q(z|x,c)\|p(z|c))}+\int_z q(z|x,c)\log p(x|z,c)\,dz \Longrightarrow$$

調整 $q(z|x,c)\Rightarrow L_b\uparrow$

當 $KL=0\Rightarrow\log p(x|c)=L_b$，且 $\log p(x|c)\geq L_b$

$\therefore L_b\uparrow\Rightarrow\log p(x)\uparrow$

Min: $KL(q(z|x,c)\|p(z|c))=KL(N(\mu,\sigma)\|N(0,1))=\frac{1}{2}(-\log\sigma^2+\mu^2+\sigma^2-1)$

Max: $\int_z q(z|x,c)\log p(x|z,c)\,dz=E_{q(z|x,c)}[\log p(x|z,c)]$