# Lab 4 - Conditional VAE for Video Prediction

Zong-lin Gao

- Assignment release: 2023/8/3 18:30
- Assignment announce: 2023/8/8 18:30
- Deadline: **2023/8/17 12:00**
- Demo: 2023/8/17 start at 13:20
- Format: Whole source code directory, report (.pdf). Zip all the file and named it in **LAB4_{studentID}_{YOUR_NAME}.zip**

## 1. Introduction

In this assignment, we need to implement conditional video prediction in a VAE-based model. Before we go through this LAB, I want to mention that the topic about this LAB is highly correlated to the ICCV paper [1] which makes the prediction in a GAN-based model and ICML paper [2] which makes the video prediction in the VAE-based model. It might be helpful to study these papers [1], [2] before doing your work.

In [1], the author posts an interesting approach that predicts the video clips in GAN model structure. By taking a pose image generated by a pre-trained pose estimation network and a previous frame, the model can generate the next consecutive frame with a comparable subjective quality. Further details can be found in [1].

Figure 1. Taking the pose image (left down) that is generated by an off-the-shelf pose estimation network as an input to generate the next consecutive frame ( right image).

In [2] , the author used a VAE-based model which combined the LSTM module to predict future frames in RNN manner. By taking two reference frames, the model has the ability to predict the following future frames. If you have NO ideas about how to perform video generation, reading this paper [2] will provide you great insights.

## 2. Lab Description

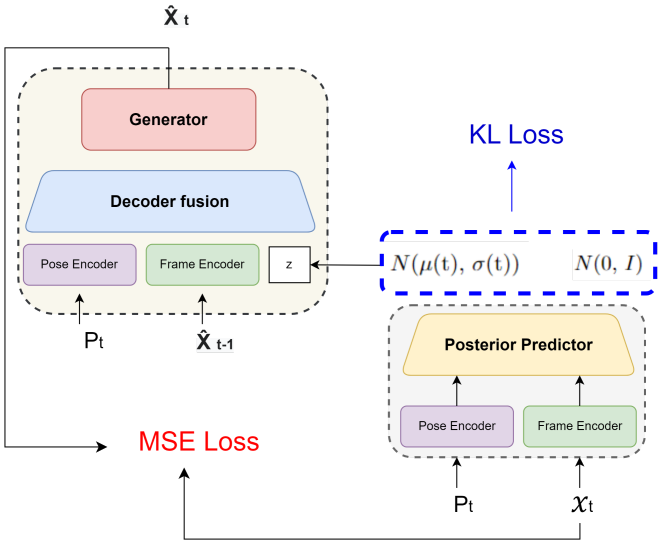



Figure 2. (a)

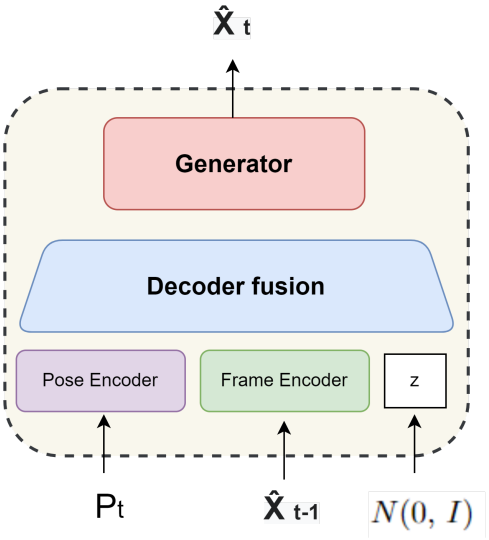



Figure 2. (b)

Figure 2. (a) give you a coarse version of the training protocol. Posterior Predictor which takes the current frame and current label as input to generate the distribution. The generator which takes the current label, last generated frame, noise sampled by the distribution predicted by Posterior Predictor as inputs to generate the current frame. Figure 2. (b) is the video generation protocol in inference time that the noise will directly sample from the prior distribution.

## 3. Requirements

a. Training details implementation
   i. Implement Video prediction protocol in (train) stage
   ii. Implement reparameterization tricks
   iii. KL annealing implementation. (a) Cyclical. (b) Monotonic

b. Teacher forcing strategy
   i. Setup teacher forcing ratio, and plot the diagram in training
   ii. Teacher forcing ratio: 0~1

c. Plot diagrams
   i. Plot the loss curve while training
   ii. Plot PSNR-per frame diagram while validation

d. Analysis
   i. Compare the result in the loss curve if you apply different KL annealing strategies or even without KL annealing. Which one is better ?

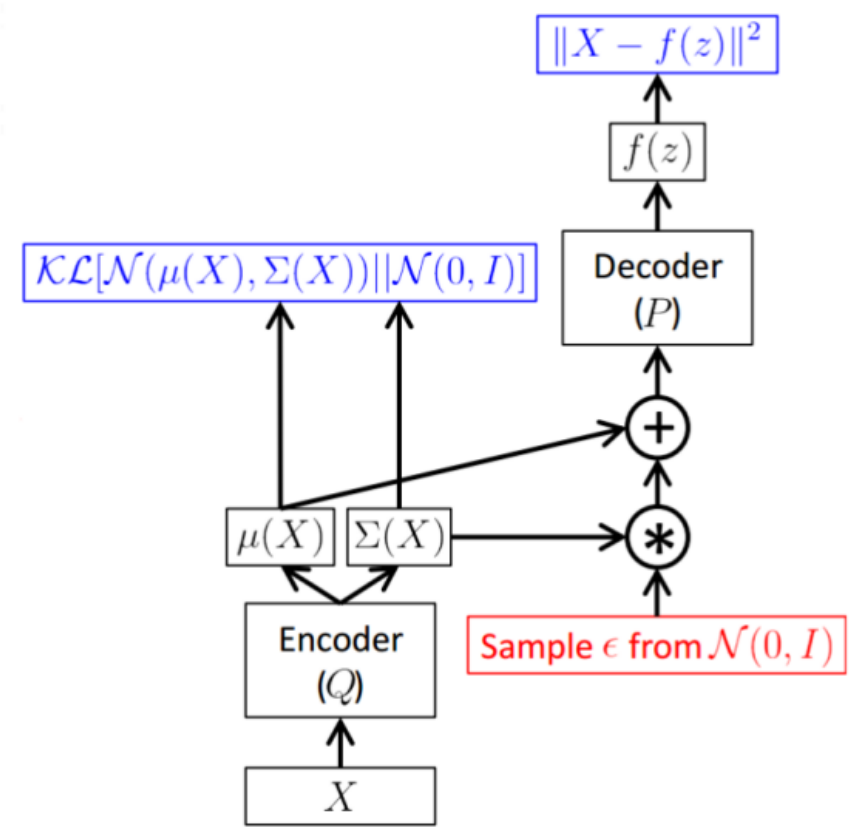
e. Validate your result and make it into gif file
   i. Pick one video clips in testing dataset and make the frames generated by your model into a gif file (This should be shown to TAs in LAB4 demo)

f. Derivation of Conditional VAE
   i. Hand write the derivation of conditional VAE in your report

# 4. Implementation Details

## a. VAE recap

In the DL lecture, VAE has been explained thoroughly. While training VAE N2N, we need to adopt a reparameterization trick. For the purpose of stable training, the output of the reparameterization trick should be log variance instead of variance directly.




$$\underbrace{E_{\boldsymbol{Z}\sim q(\boldsymbol{Z}|\boldsymbol{X};\boldsymbol{\theta}')} \log p(\boldsymbol{X}|\boldsymbol{Z};\boldsymbol{\theta})}_{\text{Re-parameterization for end-to-end training}} \quad -\mathrm{KL}(q(\boldsymbol{Z}|\boldsymbol{X};\boldsymbol{\theta}')||p(\boldsymbol{Z}))$$

Figure 2.1 The illustration of the reparameterization trick.

a. **In one training step**: There will be 16 frames as a training video sequence {x1, x2, …… , x16}, and 16 label frames as conditional signals {P1, P2, …… , P16}. First frame is the past frame which is provided to predict the consecutive 15 future frames. Example is provided in Figure 3.

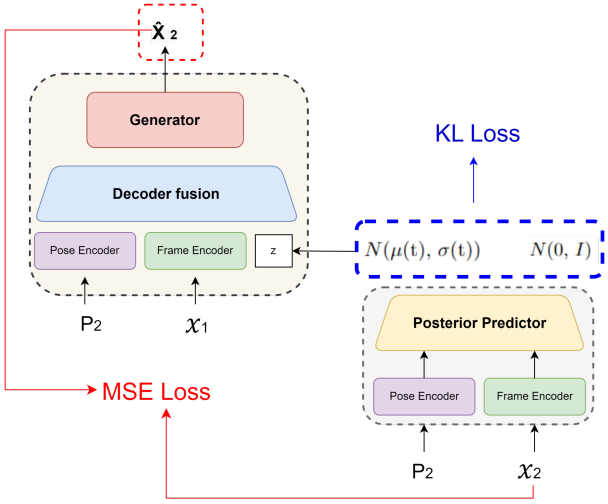



Figure 3. (a)

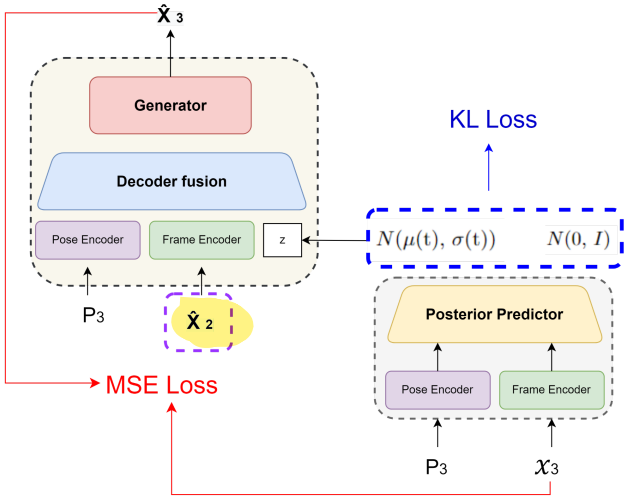



Figure 3. (b)

Figure 3. (a), and (b) give you some examples of how to generate the future frames. In (a), the task is to generate frame X2. However, we have no last frame to be taken as Decoder fusion input in scenario (a). Hence, X1 will be provided in the dataset to be the first input of the generative system. In (b), the task is to generate x3. Decoder fusion takes the frame generated by the last step (red dash box in fig. 3(a) ) as input to make the prediction.

b. **In inference time**: { X1, P2, Z } will first be taken as input to generate the second frame. After the second frame is generated, it will be taken as an input to generate the next consecutive frame. Further detail can be found in Figure 4.

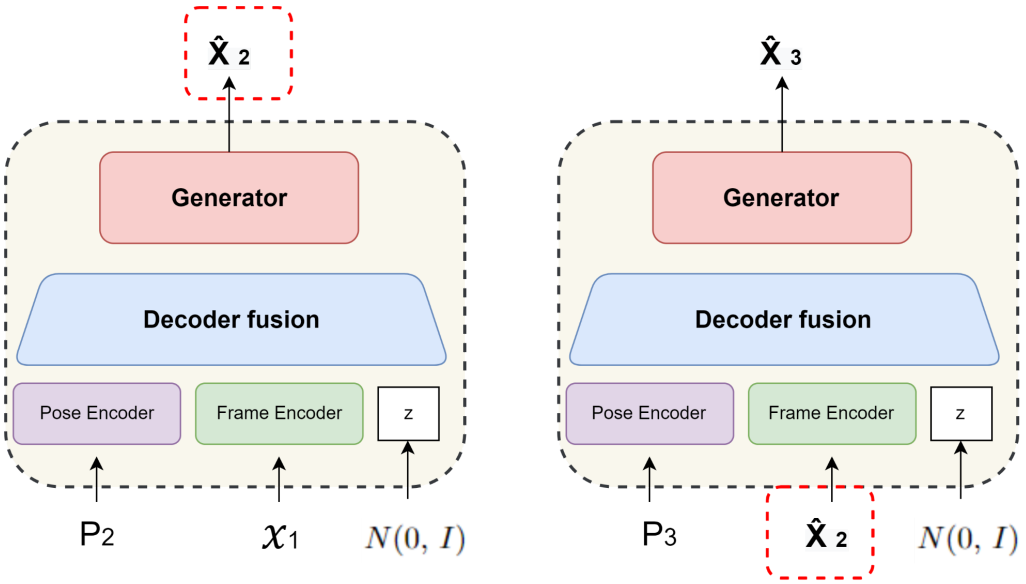



Figure 4 Example of how inference works

There would be 630 consecutive frames in the validation dataset. By taking one past frame, your model should predict **the remaining 629 frames**. After the prediction, it is suggested that you convert these 629 frames into a gif file, and see the quality of your prediction.

c. **KL annealing strategy:** A variable weight is added to the KL term in the loss function. The following experiment results should be conducted and make the comparison in your report. It is suggested that you plot the loss curve in training while applying different strategies. Further details can be found in [3].

   i. Cyclical
   ii. Monotonic
   iii. Without KL annealing strategy

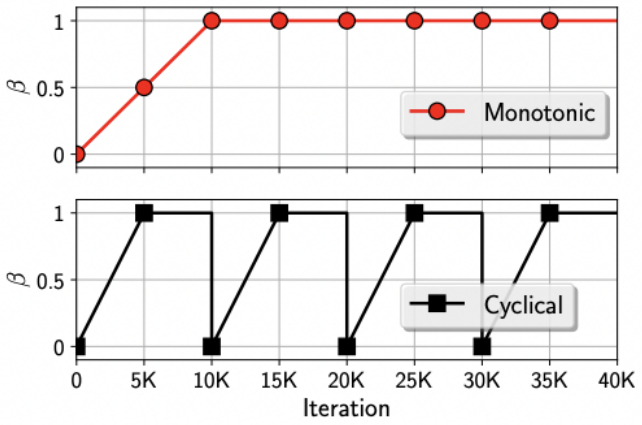



Figure 5. weight in different KL annealing strategies

## 5. Dataset

a. Training dataset
   i. train_img: 23410 png files
   ii. train_label: 23410 png files

b. Valadition dataset
   i. val_img: 630 png files
   ii. val_label: 630 png files

c. Testing dataset
   i. 6 video sequences are given. Each video sequence contains one first frame and 630 label frames.

d. Download dataset
   i. ON your own machine

```
?> sftp -P 10051 pp037@140.113.215.196 (passwd: pp037123)
?> get LAB4_dataset.zip
```

   ii. ON Provided machine

```
?> sftp pp037@192.128.201.51 (passwd: pp037123)
?> get LAB4_dataset.zip
```

6. Model Configurations
    a. Input images and labels will be resized to (32, 64) due to memory limitation.
    b. All modules e.g.
        i. frame encoder
        ii. pose encoder
        iii. posterior generator
        iv. decoder fusion
        v. generator

    are provided in directory named modules. You can change the structure of the modules mentioned above but **DO NOT** change the modules' name.

    c. Recommended command

- Training command

```
python Trainer.py --DR {YOUR_DATASET_PATH}
--save_root {PATH_TO_SAVE_YOUR_CHECKPOINT}
--fast_train
```

    - --fast_train: is use fewer dataset and large learning rate to speed up your training

- Testing command

```
python Tester.py --DR {YOUR_DATASET_PATH}
--save_root {PATH_TO_SAVE_YOUR_CHECKPOINT}
--ckpt_path {PATH_TO_YOUR_CHECKPOINT}
```

7. Scoring Criteria
    a. Report (50%)
        i. Introduction (5%)
        ii. Implementation details (25%)
            1. How do you write your training protocol (10%)
            2. How do you implement reparameterization tricks (5%)
            3. How do you set your teacher forcing strategy (5%)
            4. How do you set your kl annealing ratio (5%)
        iii. Analysis & Discussion (20%)
            1. Plot Teacher forcing ratio (5%)
                a. Analysis & compare with the loss curve
            2. Plot the loss curve while training with different settings. Analyze the difference between them (10%)
                a. With KL annealing (Monotonic)
                b. With KL annealing (Cyclical)
                c. Without KL annealing
            3. Plot the PSNR-per frame diagram in validation dataset (5%)

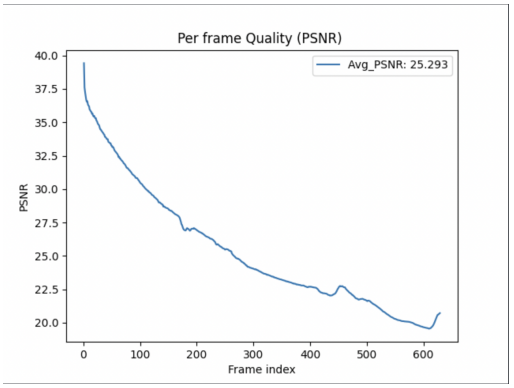



Figure 6. give you an example of what you need to do in this part

b. Demo (50%)

i. Questions (30%)

ii. Quality of your prediction result (20%)

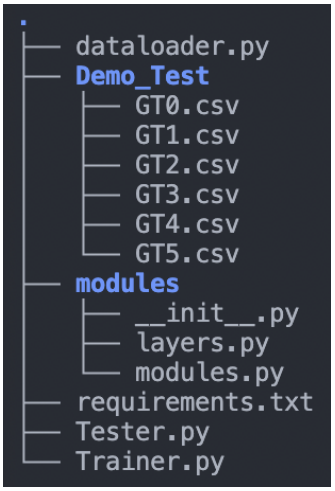
| Accuracy | Grade |
|---|---|
| $\text{PSNR} \geq 25$ | 100 % |
| $25 > \text{PSNR} \geq 24$ | 90 % |
| $24 > \text{PSNR} \geq 23$ | 80 % |
| $23 > \text{PSNR} \geq 22$ | 70 % |
| $22 > \text{PSNR} \geq 21$ | 60 % |
| $21 > \text{PSNR} \geq 20$ | 50 % |
| $\text{PSNR} < 20$ | 0 % |


Figure 7. is the output of your result after running Tester.py The red box means the average PSNR value in 6 testing video sequences, and will be taken as your scoring criteria.

## 8. Files provided

a. Code

i. LAB4.zip is given on E3. Structure would look like the picture below




9. Hints

   Friendly notification

   a. Do your work early.

   b. Read the reference papers

   c. Dataloader is provided

   d. **View the details in Tester.py** it may help while you implement your training protocol

10. Upload file format

    DLP_2023_LAB4_{studentID}.zip

    example: LAB4_311554005_高宗霖.zip

    - LAB4_311554005_高宗霖.zip
      - LAB4/
      - report.pdf

11. References

[1] C. Chan, et al., "Everybody Dance Now," ICCV, 2019

[2] E. Denton, et al., "Stochastic Video Generation with a Learned Prior," ICML, 2018
[3] H. Fu, et al., "Cyclical Annealing Schedule: A Simple Approach to Mitigating KL Vanishing," NAACL 2019