A unified model for continuous conditional video prediction Supplementary Material

Xi Ye* Guillaume-Alexandre Bilodeau
Polytechnique Montréal
{xi.ye, gabilodeau}@polymtl.ca

1. Table of important acronyms and notations

NPVP:	Neural process for conditional video prediction
VFI:	Video frame interpolation
VFP:	Video future frame prediction
VPE:	Video past frame extrapolation
VRC:	Video random missing frames completion
NPs:	Neural processes
INRs:	Implicit neural representations
FFN:	Fourier feature network
SIREN:	Sinusoidal representation networks
MLP:	Multiple layer perceptron
CNN:	Convolutional neural network
ConvLSTMs:	Convolutional-LSTMs
V_C:	Context video frames
V_T:	Target video frames
X_C:	Context coordinate representations
Y_C:	Context video frame features
X_T:	Target coordinate representations
Y_T:	Target video frame features
M_C:	Output feature of $\mathcal{T}_E$ given X_C and Y_C
M_T:	Output feature of $\mathcal{T}_E$ given X_T and Y_T
z_e:	event variable
$\mathcal{T}_E$:	Transformer encoder
$\mathcal{T}_D$:	Transformer decoder
E_C:	Context event CNN encoder
E_T:	Target event CNN encoder

Table 1. Table of important acronyms and notations

2. Implementation details

3. Datasets

KTH. KTH dataset includes grayscale videos of 6 different human actions. Following the experimental setup of previous work, we take persons 1-16 as training set, and persons 17-25 as test set. Random horizontal flips and vertical flips are applied to each video clip as data augmentation.

BAIR. BAIR dataset includes RGB video clips of a robot arm randomly moving over a table with small objects. The training and test sets are defined by the creators of BAIR. Random horizontal flips and vertical flips are applied to each video clip as data augmentation.

SM-MNIST. Stochastic Moving MNIST (SM-MNIST) is a synthetic dataset includes videos of two randomly moving MNIST characters within a square region. There is no data augmentation for SM-MNIST during training.

Cityscapes. Cityscapes dataset includes high-resolution urban traffic videos of many cities. Note that we do not use any annotation provided by Cityscapes, for example, object classes or segmentation masks. Same as previous work, we use the raw video clips from the "$leftImg8bit\_sequence\_trainvaltest.zip$" of Cityscapes. The frames are firstly center-cropped to be square, then we resize the frames to be the resolution of 128×128. There is no data augmentation for the Cityscapes dataset during training.

KITTI. KITTI dataset includes traffic videos across multiple scenarios, including city, residential, road etc. We follow the experimental setup of previous works [1], i.e., randomly select 4 sequences from the raw data of KITTI for testing and use the remaining videos for training. The frames are firstly center-cropped and then resize to be the resolution of 128×128. Random horizontal flips and vertical flips are applied to each video clip as data augmentation.

3.1. Training details

Training of the autoencoder. For all datasets, the dimension of visual features is set to be $H = 8, W = 8, D = 512$. For input with a resolution of 64×64, the frame encoder includes 3 downsampling blocks and 2 residual blocks. For input with a resolution of 128×128, the frame encoder includes 4 downsampling blocks and 3 residual blocks. The number of upsampling blocks for the frame decoder equals to the number of downsampling blocks in the corresponding frame encoder. An Adam optimizer with

*Corresponding author

a learning rate of $1e^{-4}$ is used for the training.

Training of the NPs-based predictor. For all datasets, $\gamma = 0.01$. For BAIR and SM-MNIST, $\beta = 1e^{-6}$. For KTH, $\beta = 1e^{-8}$. The predictors are trained by AdamW, we take a cosine annealing learning rate scheduler with warm restarts [2] at every 150 epochs, the maximum learning rate is $1e^{-4}$ and the minimum learning rate is $1e^{-7}$. Gradient clipping is applied to $\mathcal{T}_E$ and $\mathcal{T}_D$ during training. Please visit `https://npvp.github.io` for the code.

3.2. Architecture of VidHRFormer block

For the convenience of the readers, we have redrawn the detail architecture of VidHRFormer block [4] and the VPTR decoder block in Figure 1.

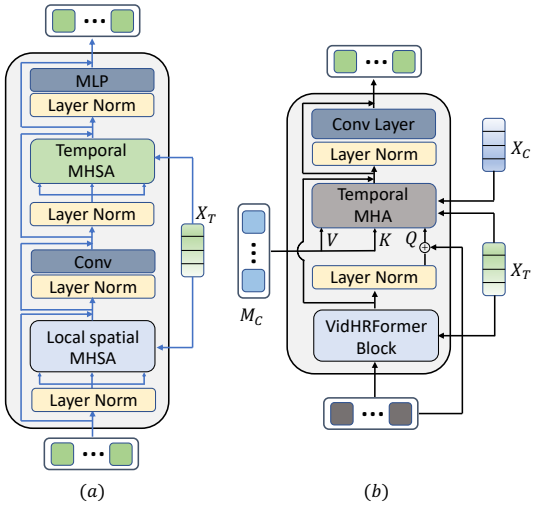

Figure 1. (a) VidHRFormer block [4]. (b) Decoder block of VPTR [4].

3.3. Architecture of Event encoder E_C and E_T

E_C and E_T share the same architecture, see Figure 2. They are implemented by a small neural network with three $Conv - BN - ReLU$ layers and two $Conv$ heads to output μ and σ respectively.

4. Qualitative examples

4.1. Unified model

Here we show another example (see Figure 3) of the unified model on Cityscapes dataset for all four different conditional video prediction tasks. In order to demonstrate the continuous prediction ability of NPVP, we take the trained unified model to solve different tasks with different rates, please visit `https://npvp.github.io` for video examples of a unified model for KTH dataset.

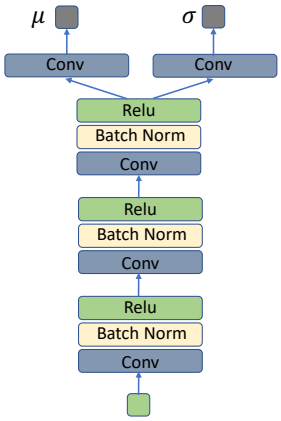

Figure 2. Architecture of the Event encoders.

4.2. Task-specific VFI

We present uncurated VFI examples of KTH, SM-MNIST and BAIR datasets by task-specific *NPVP* models, see Figure 4, Figure 5 and Figure 6. As there is little stochasticity for VFI on KTH and SM-MNIST, we only show the example with the best SSIM from 100 random examples. We also present the VFI results of MCVD [3] on KTH and SM-MNIST datasets for qualitative comparison. Please visit `https://npvp.github.io` for video examples.

4.3. Task-specific VFP

We present VFP examples by task-specific *NPVP* models, see Figure 7, Figure 8 and Figure 9. For Cityscapes dataset, we show the results of MCVD for comparison. Please visit `https://npvp.github.io` for video examples.

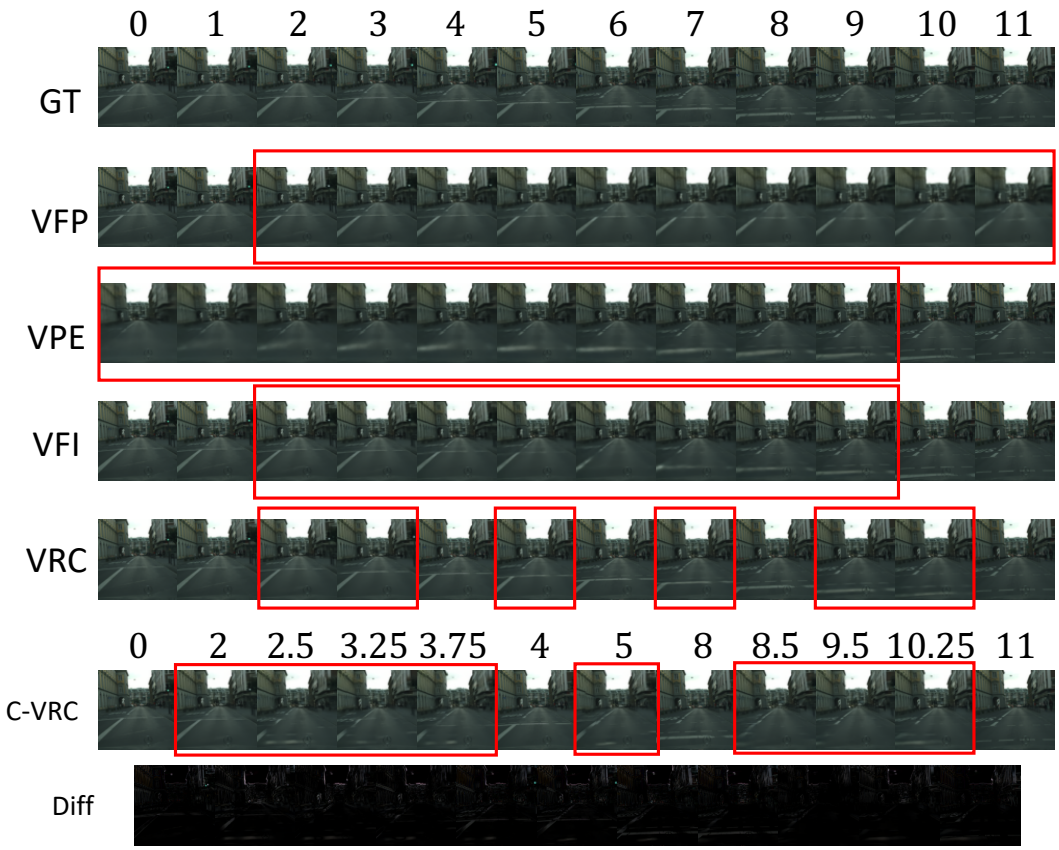

Figure 3. One model for all tasks. Frames inside the red boxes are target frames generated by the model. C-VRC denotes continuous VRC. Diff are the difference images between neighboring frames of C-VRC to show that they are all different and that the temporal coordinates are taken into account.

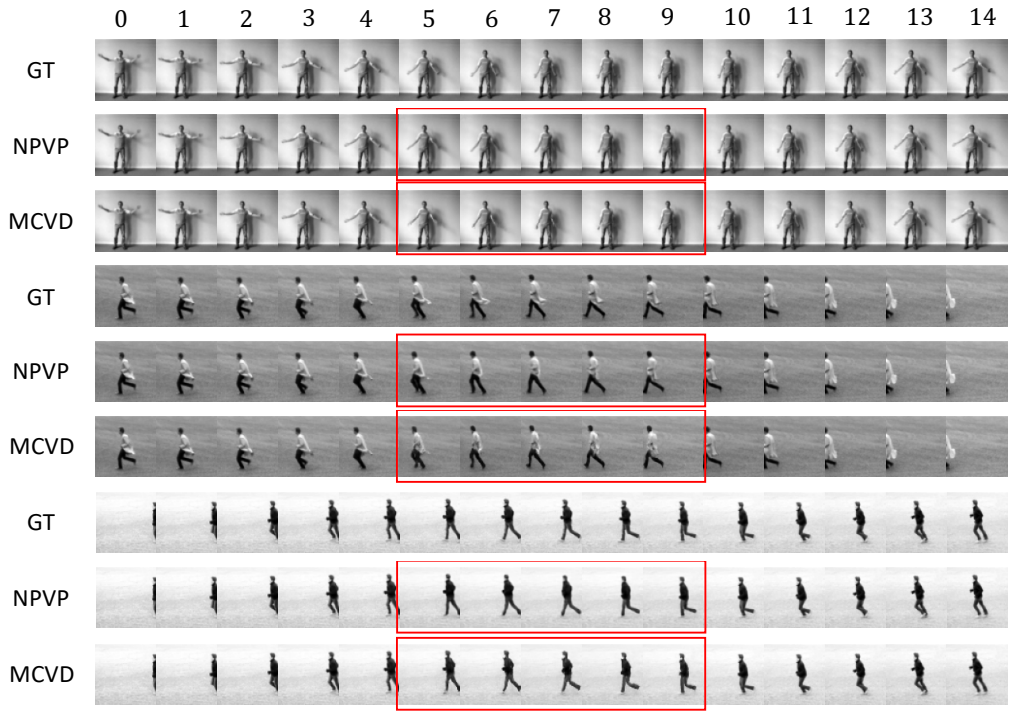

Figure 4. VFI examples on KTH by a Task-specific *NPVP* (*10→5*) model. Frames inside the red boxes are target frames generated by the models. Compared with MCVD [3], predicted moving arms or legs by *NPVP* are more realistic and more similar to the ground-truth.

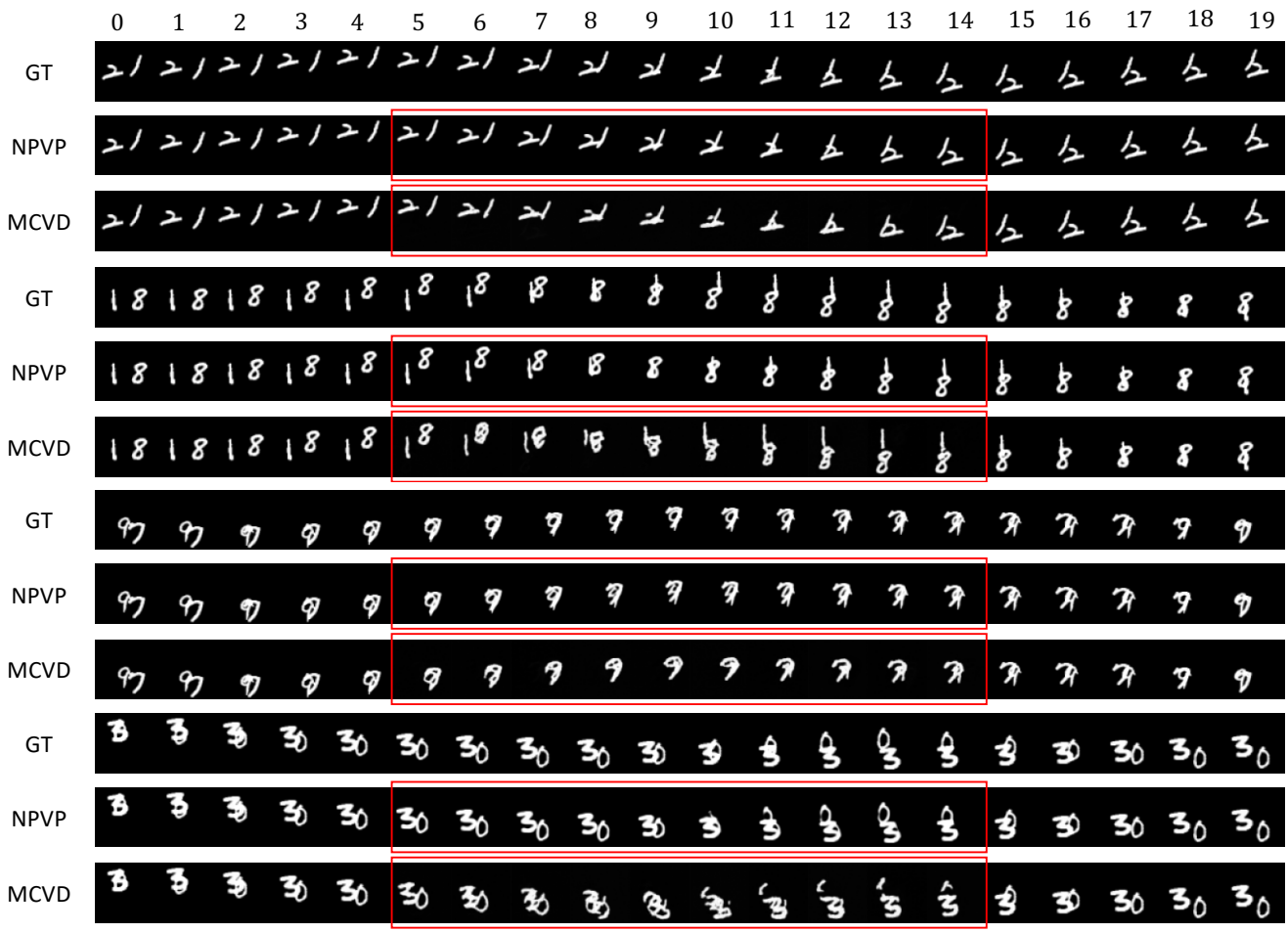

Figure 5. VFI examples on SM-MNIST by a Task-specific *NPVP* (*10→10*) model. Frames inside the red boxes are target frames generated by the model. Compared with MCVD [3], the interpolation quality of *NPVP* is better as it captures the shape and motion of MNIST characters for missing frames.

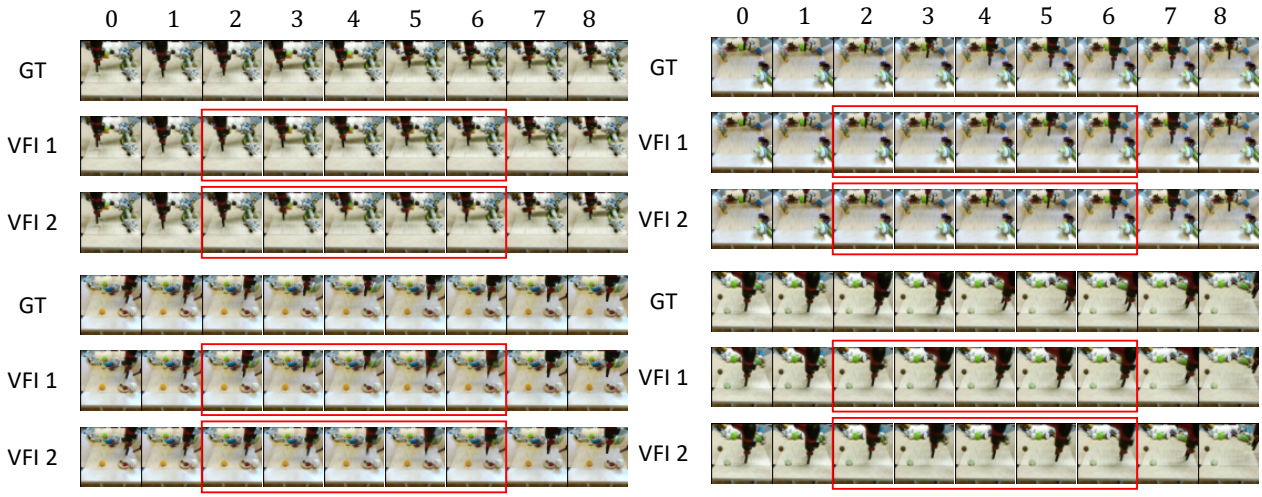

Figure 6. VFI examples on BAIR by a Task-specific *NPVP* (*4→5*) model. Frames inside the red boxes are target frames generated by the model. VFI 1 and VFI 2 denote two different random interpolations given the same contexts.

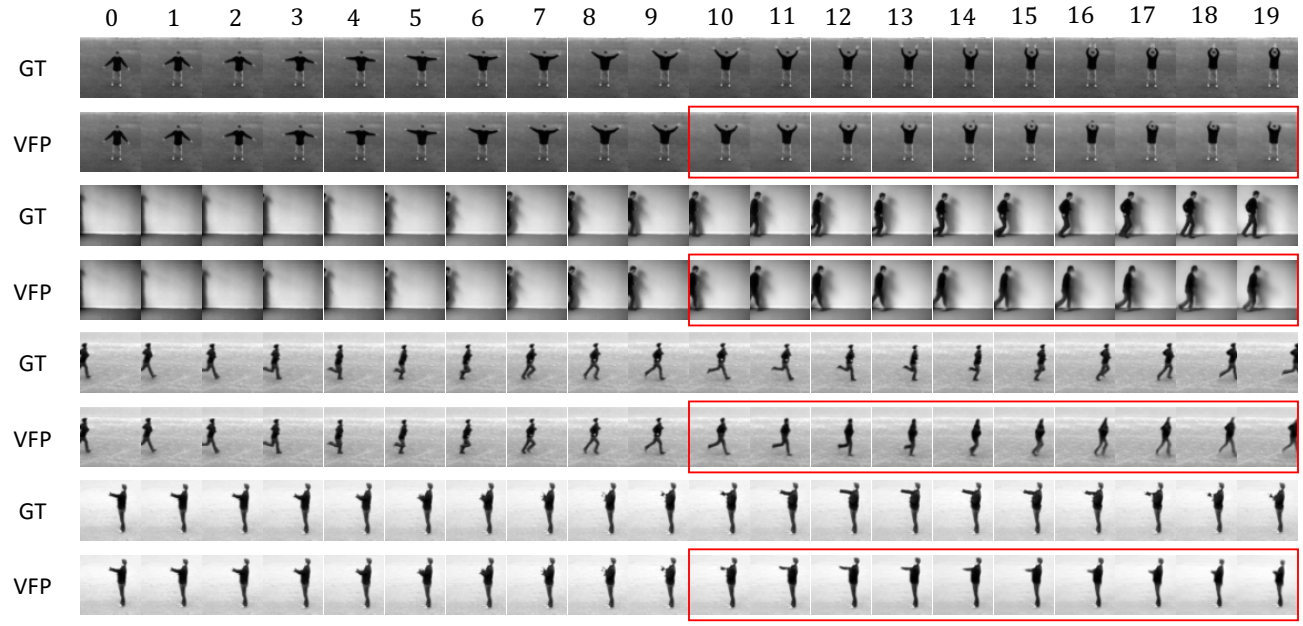

Figure 7. VFP examples on KTH by a Task-specific *NPVP* (*10→10*) model. Frames inside the red boxes are target frames generated by the model.

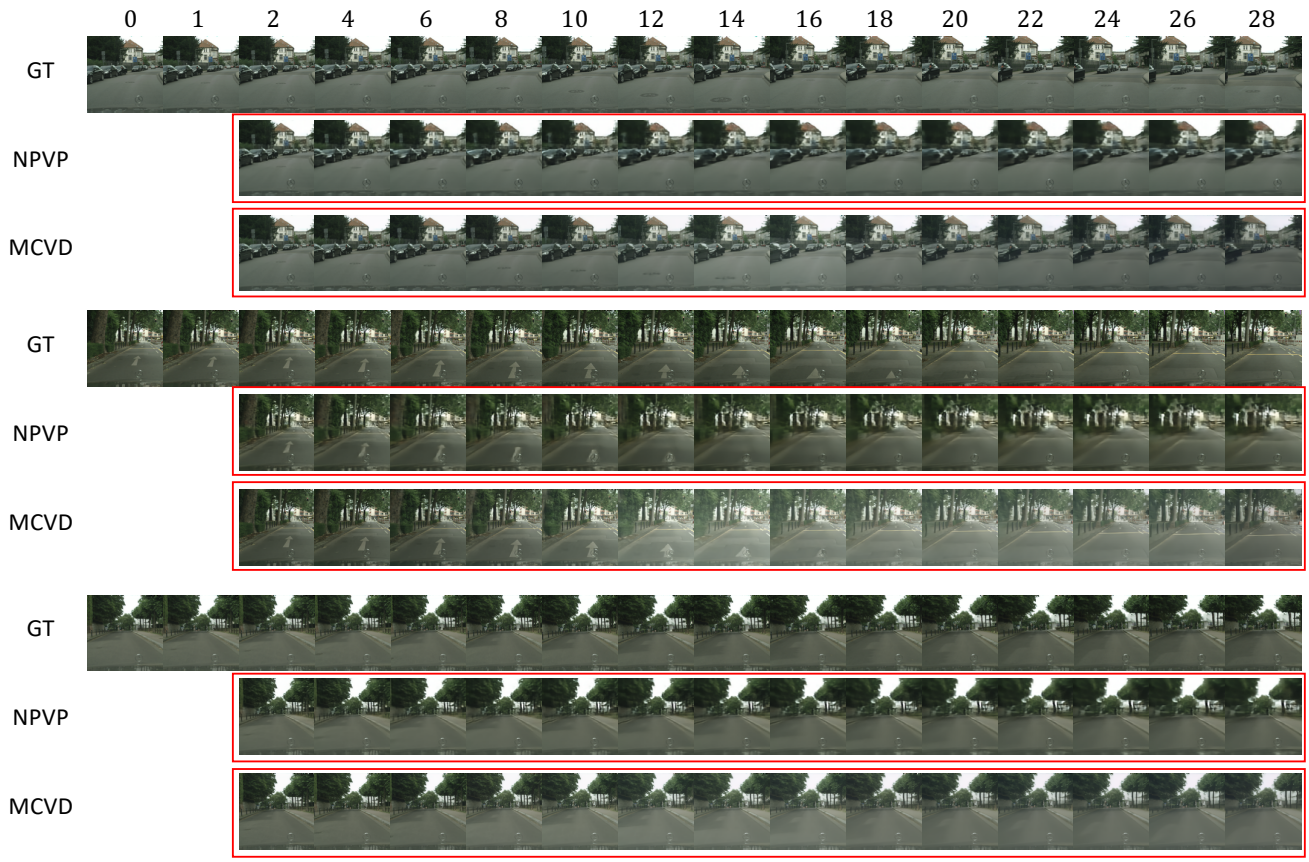

Figure 8. VFP examples on Cityscapes by a Task-specific *NPVP* (*2→28*) model. Frames inside the red boxes are target frames generated by the model. Here we also show the examples generated by MCVD [3], which suffers from a brightness-changing problem.

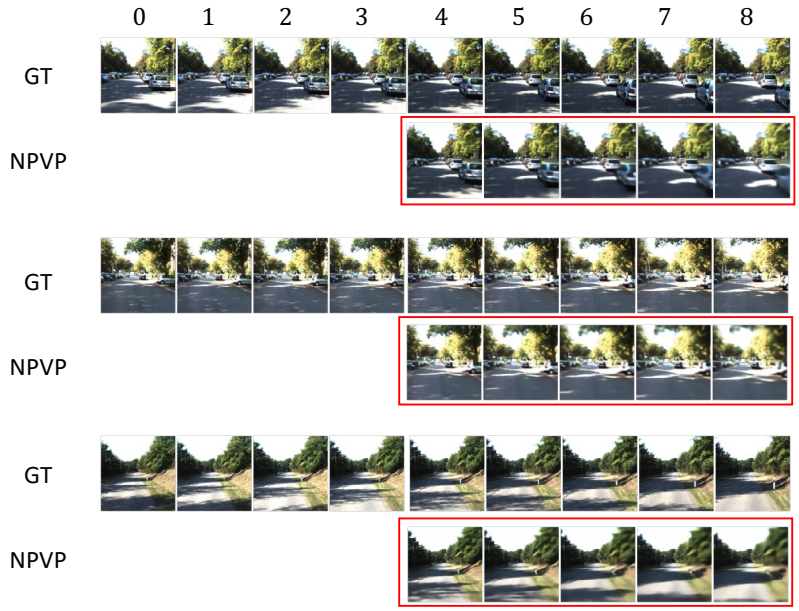

Figure 9. VFP examples on KITTI by a Task-specific *NPVP* (*4→5*) model. Frames inside the red boxes are target frames generated by the model.

References

[1] Xinzhu Bei, Yanchao Yang, and Stefano Soatto. Learning Semantic-Aware Dynamics for Video Prediction. 2021.

[2] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic Gradient Descent with Warm Restarts. In *ICLR*, 2017.

[3] Vikram Voleti, Alexia Jolicoeur-Martineau, and Christopher Pal. Masked Conditional Video Diffusion for Prediction, Generation, and Interpolation. In *Advances in Neural Information Processing Systems*, 2022.

[4] Xi Ye and Guillaume-Alexandre Bilodeau. VPTR: Efficient Transformers for Video Prediction. In *26th International Conference on Pattern Recognition (ICPR)*, 2022.