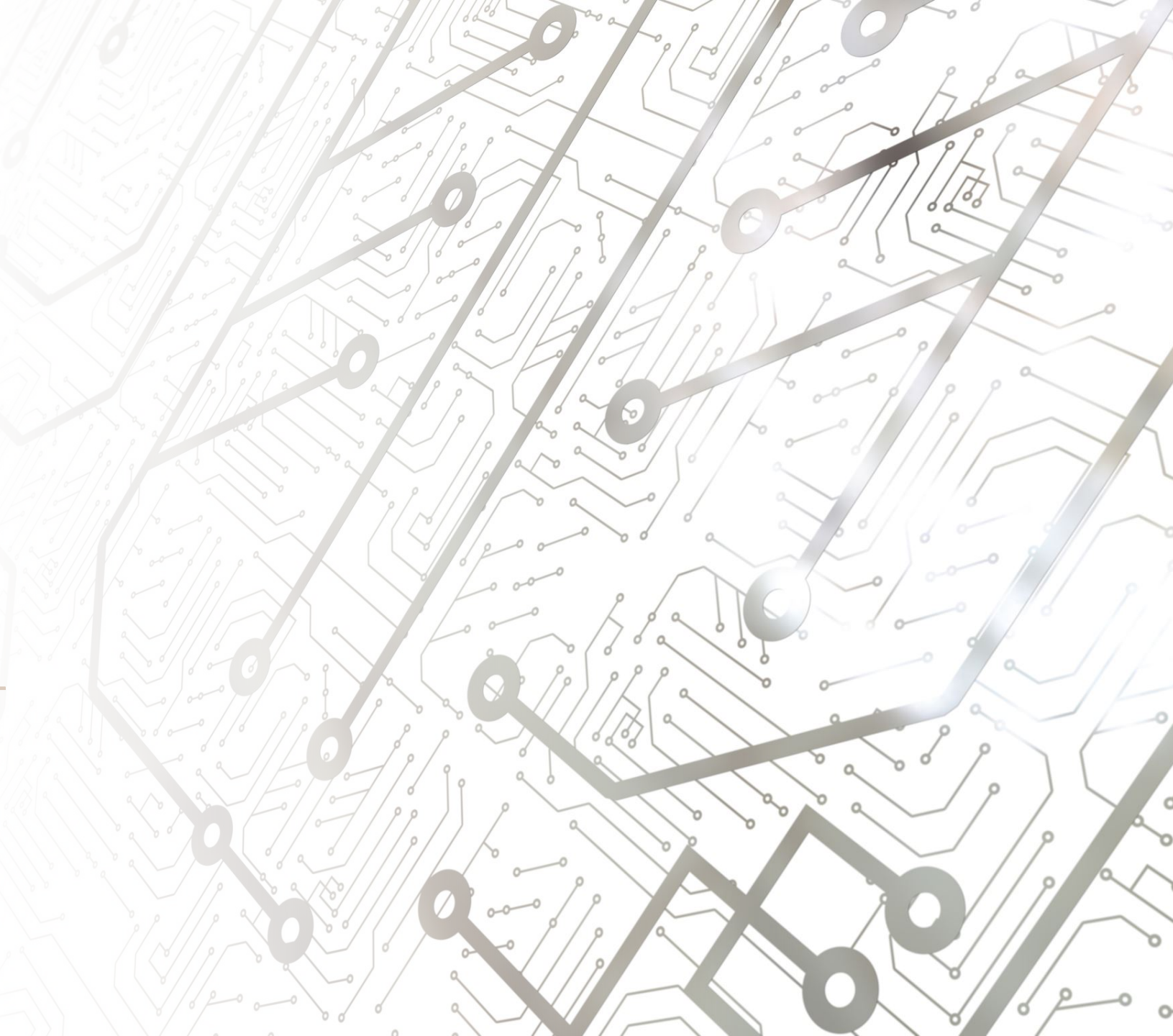
Disentangling Propagation and Generation for Video Prediction

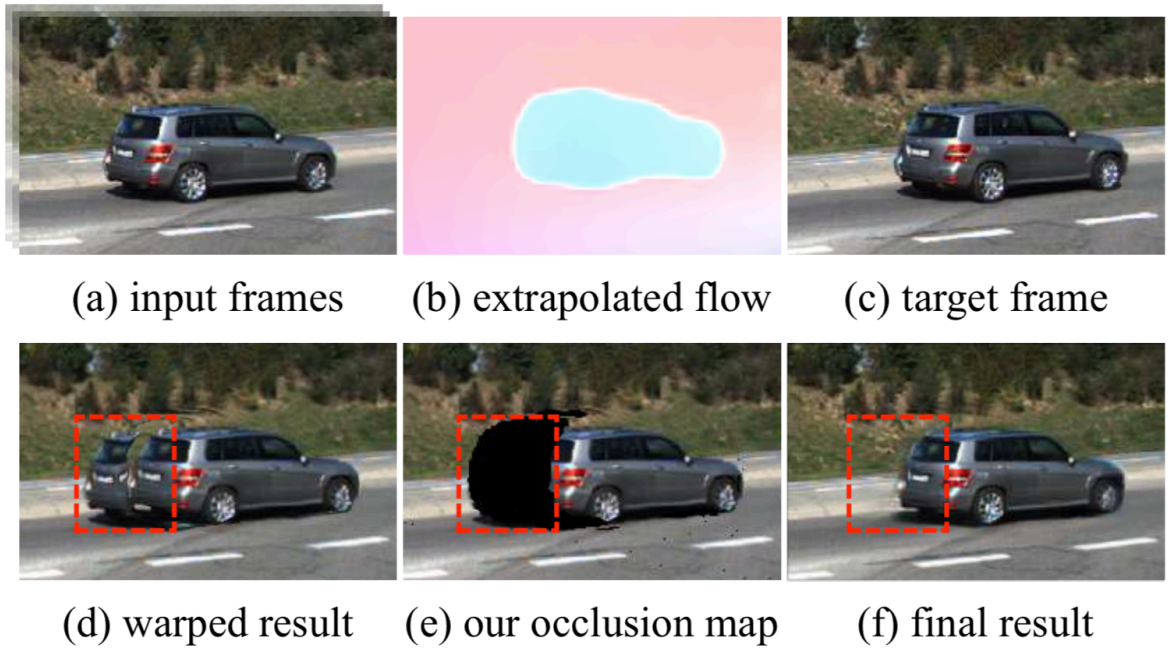
UC Berkeley; Sinovation Ventures AI Institute; Columbia University;

ICCV 2019



Introduction

Goal: Given video frames 1 to i-1, predict frame i



(a) input frames (b) extrapolated flow (c) target frame

(d) warped result (e) our occlusion map (f) final result

Video frames:

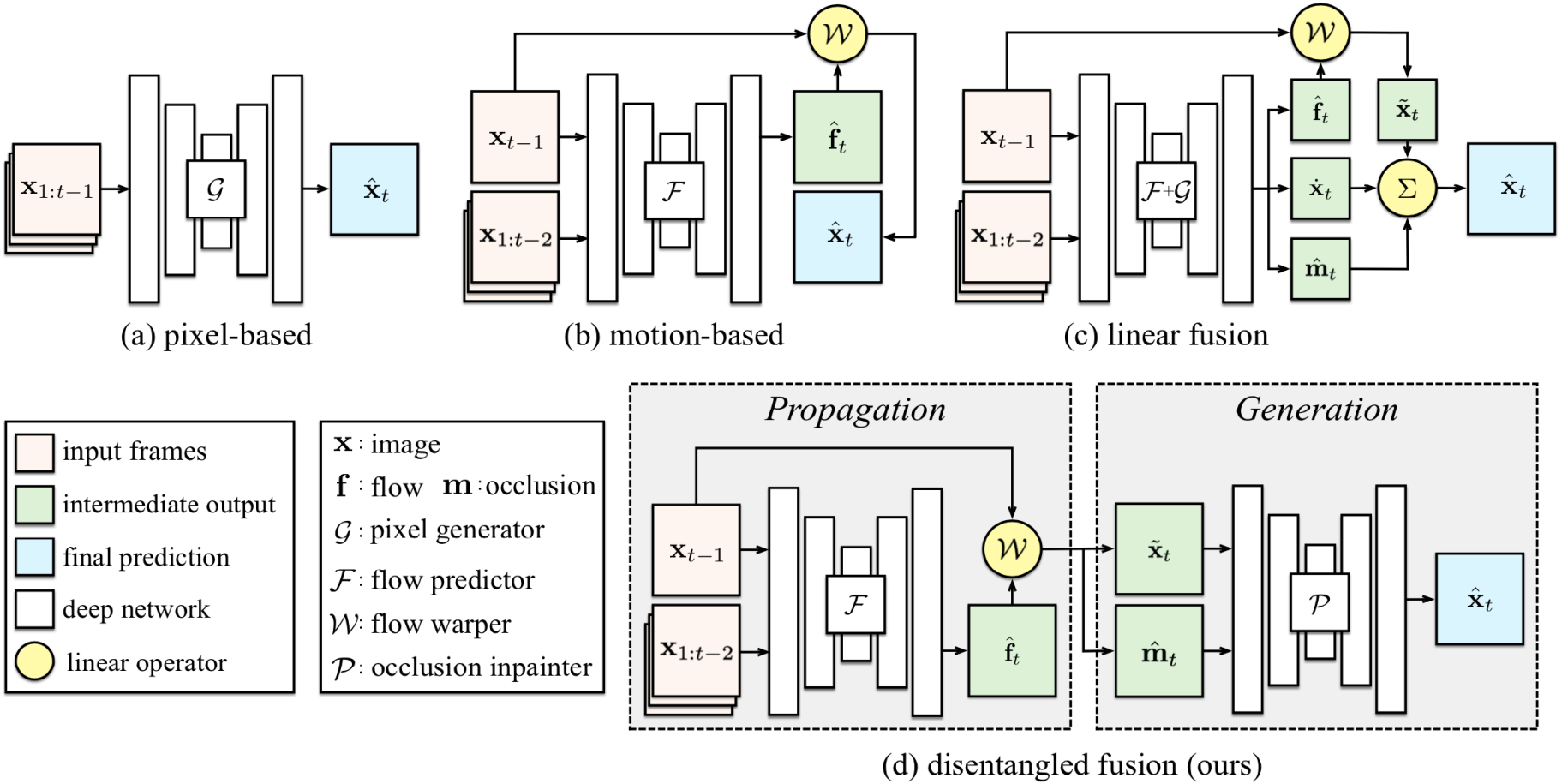
Non-occluded regions

Move fluidly; can be predicted

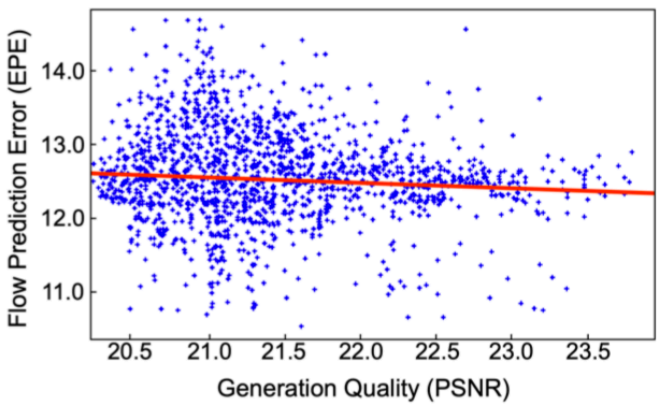
Occluded/disoccluded/exposed regions

Can not be extrapolated

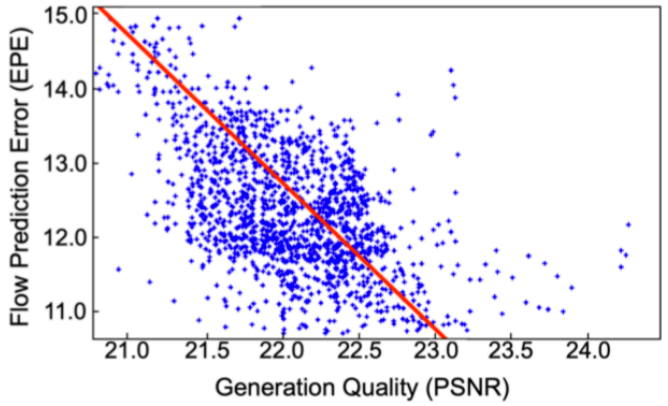
Previous work and The model here



Is sequential model better?



(a) linear fusion [11]



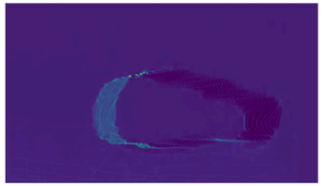
(b) disentangled fusion (ours)

Figure 3: The correlation between pixel and flow prediction tasks evaluated by different models. We show scatter plots about generation quality versus flow prediction error on KITTI Flow dataset [24]. Pixel and flow prediction tasks do not actually correlate significantly with each other on previous models using multi-task learning. Our disentangled fusion pipeline shows that after factorizing these tasks apart, both tasks could be better learned.

Approaches

(a) $t = 0$, flow.

(b) $t = 1$, warping result



(c) $t = 1$, density map.

(d) $t = 1$, occlusion map.

The pixel density can be viewed as an energy map: initially, it is uniformly filled for all pixels, and the motion propagation changes its distribution and make the energy map become dense or sparse on different region.

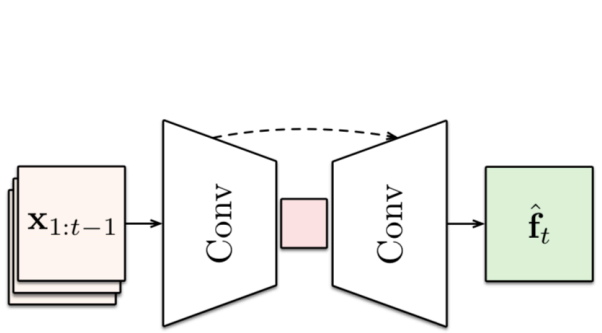
1. If $\mathbf{E}^2_{x,y} = 0$, there is no pixel moving to this coordinate, which indicates it will be *dis-occluded*;

2. If $\mathbf{E}^2_{x,y} > 2$, there are at least two pixels in the first frame compete for the same location, which suggests it will be *occluded*.

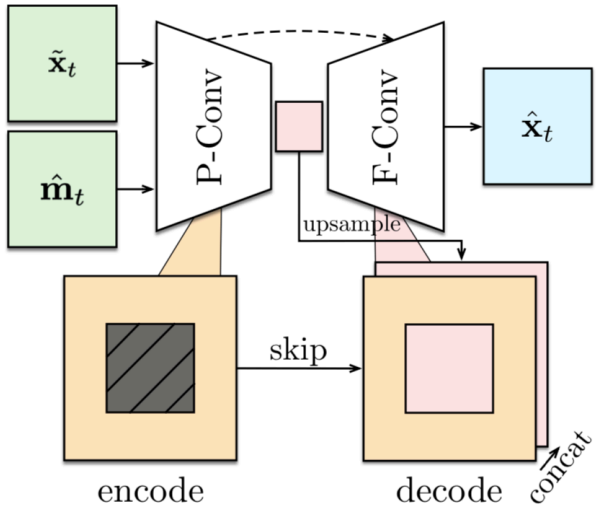
After complete definition of occluded and dis-occluded regions, we can then get the occlusion map according to

$$\hat{\mathbf{m}}_{i,j} = \begin{cases} 1 & 0 < \mathbf{E}^2_{i,j} < 2, \\ 0 & otherwise. \end{cases} \tag{6}$$

Approaches



(a) Flow predictor $\mathcal{F}$.

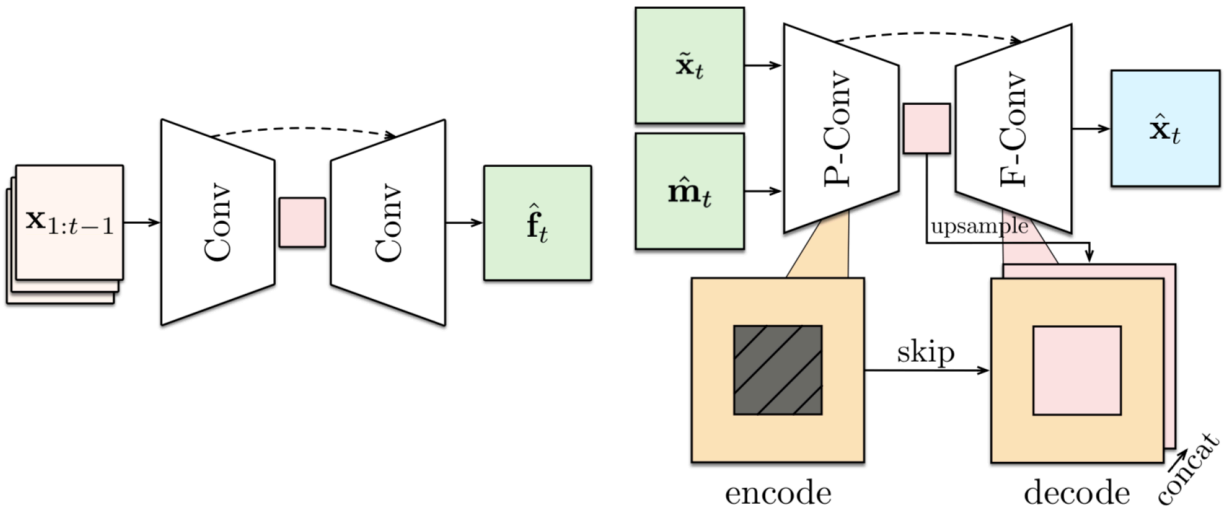


(b) Occlusion inpainter $\mathcal{P}$.

$$\mathcal{L}_p(\tilde{\mathbf{x}}_t, \mathbf{x}_t; \hat{\mathbf{m}}_t) = \alpha \frac{1 - SSIM(\tilde{\mathbf{x}}_t \odot \hat{\mathbf{m}}_t, \mathbf{x}_t \odot \hat{\mathbf{m}}_t)}{2} + (1-\alpha)\|\tilde{\mathbf{x}}_t \odot \hat{\mathbf{m}}_t - \mathbf{x}_t \odot \hat{\mathbf{m}}_t\|_1 \quad (3)$$

$$\mathcal{L}_{smt}(\hat{\mathbf{f}}_t, \mathbf{x}_t) = \sum_{i,j} |\nabla \hat{\mathbf{f}}_t(i,j)| \cdot (e^{-|\nabla \mathbf{x}_t(i,j)|})^T, \quad (4)$$

Approaches



(a) Flow predictor $\mathcal{F}$. (b) Occlusion inpainter $\mathcal{P}$.

1. The pixel reconstruction loss

$$\mathcal{L}_{pix} = \mathcal{L}_p(\hat{\mathbf{x}}_t, \mathbf{x}_t; \hat{\mathbf{m}}_t) + \beta\mathcal{L}_p(\hat{\mathbf{x}}_t, \mathbf{x}_t; 1 - \hat{\mathbf{m}}_t),$$

where $\mathcal{L}_p$ is defined in Equation 3.

2. The perceptual and style losses in VGG's n-dimensional latent space $\{\mathbf{\Psi}^n\}$ as in [34].

$$\mathcal{L}_{prc} = \frac{1}{n}\sum_n \| [\psi(\tilde{\mathbf{x}}) - \psi(\mathbf{x})] \odot \hat{\mathbf{m}}_t\|_1 + \beta\| [\psi(\tilde{\mathbf{x}}) - \psi(\mathbf{x})] \odot (1 - \hat{\mathbf{m}}_t)\|_1,$$

$$\mathcal{L}_{sty} = \frac{1}{n}\sum_n \|(\psi(\tilde{\mathbf{x}}) - \psi(\mathbf{x}))^T(\psi(\tilde{\mathbf{x}}) - \psi(\mathbf{x})) \odot \hat{\mathbf{m}}_t\|_1 + \beta\|(\psi(\tilde{\mathbf{x}}) - \psi(\mathbf{x}))^T(\psi(\tilde{\mathbf{x}}) - \psi(\mathbf{x})) \odot (1 - \hat{\mathbf{m}}_t)\|_1.$$

3. The total-variance loss to encourage similar texture in occlusion boundaries

$$\mathcal{L}_{var} = \sum_{i,j}\sqrt{\|\mathbf{x}_{t(i,j+1)} - \mathbf{x}_{t(i,j)}\|_2^2 + \|\mathbf{x}_{t(i+1,j)} - \mathbf{x}_{t(i,j)}\|_2^2}.$$

4. The extra semantic loss to enforce layout consistency between segmentation masks, distilled by a pretrained segmentation network [32] $\mathbf{\Phi} : \mathcal{X} \to \mathcal{Y}$, since unconditional image inpainting tends to remove the foreground objects,

$$\mathcal{L}_{seg} = CE(\mathbf{\Phi}(\hat{\mathbf{x}}_t) \odot \hat{\mathbf{m}}_t, \mathbf{y}_t \odot \hat{\mathbf{m}}_t) + \beta CE\big(\mathbf{\Phi}(\hat{\mathbf{x}}_t) \odot (1 - \hat{\mathbf{m}}_t), \mathbf{y}_t \odot (1 - \hat{\mathbf{m}}_t)\big).$$

The total loss for the inpainting module is

$$\mathcal{L}_{\mathcal{P}} = \mathcal{L}_{pix} + \lambda_{prc}\mathcal{L}_{prc} + \lambda_{sty}\mathcal{L}_{sty} + \lambda_{var}\mathcal{L}_{var} + \lambda_{seg}\mathcal{L}_{seg}. \quad (7)$$

Results

Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) :disagree with the human perception
Learned Perceptual Image Patch Similarity (LPIPS) :Realism

CalTech Pedestrian dataset [7]

- The training, validation and testing sets consists of 3738, 14, and 1948 clips
- Every 11 consecutive frames are divided and sampled as a training clip in which the first 10 frames are fed into the model as the input, and the 11th frame is used as prediction target.

Method	Family	PSNR↑	SSIM↑	LPIPS↓ ($\times 10^{-2}$)
Repeat	–	23.3	0.779	5.11
BeyondMSE [21]	*P*	–	0.881	–
PredNet [20]	*P*	27.6	0.905	7.47
ContextVP [4]	*P*	**28.7**	0.921	6.03
DVF [19]	*M*	26.2	0.897	5.57
Dual Motion GAN [17]	*F*	–	0.899	–
CtrlGen [11]	*F*	26.5	0.900	6.38
DPG (Ours)	*F*	28.2	**0.923**	**5.04**

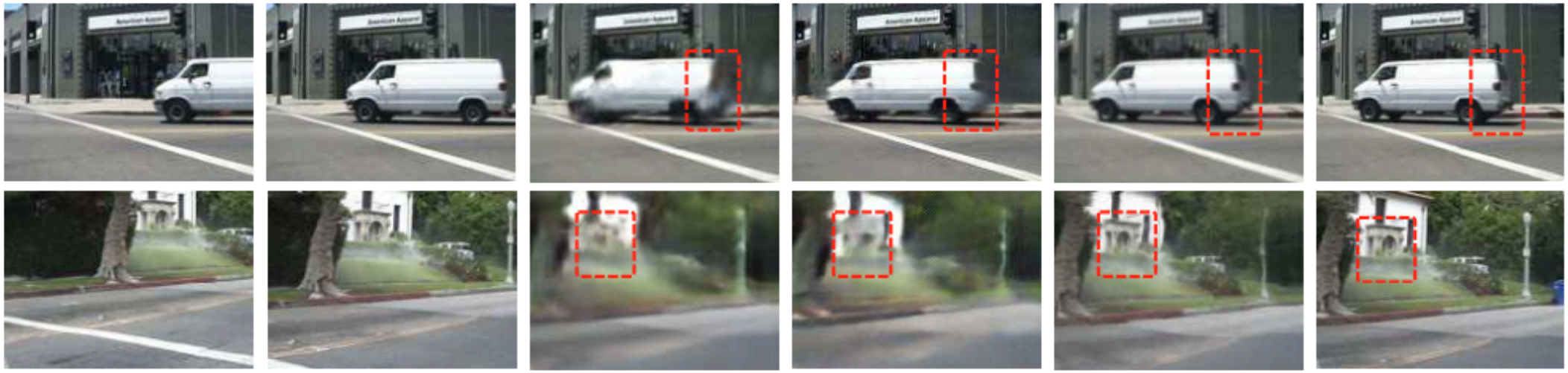
KITTI Flow dataset [23] :

- Higher resolution, larger motion and more occlusions (More challenging)
- It contains 3823 examples for training, 378 for validation and 4167 for testing.
- In addition, we sample video clips of 5 frames (4-in 1-out) from the dataset using a sliding window.

Method	Family	PSNR↑	SSIM↑	LPIPS↓ ($\times 10^{-2}$)
Repeat	–	16.5	0.489	19.0
PredNet [20]	*P*	17.0	0.527	26.3
SVP-LP [6]	*P*	18.5	0.564	20.2
MCNet [37]	*P*	18.9	0.587	23.7
MoCoGAN [36]	*P*	19.2	0.572	18.6
DVF [19]	*M*	22.1	0.683	16.3
CtrlGen [11]	*F*	21.8	0.678	17.9
DPG (Ours)	*F*	**22.3**	**0.696**	**11.4**

Results

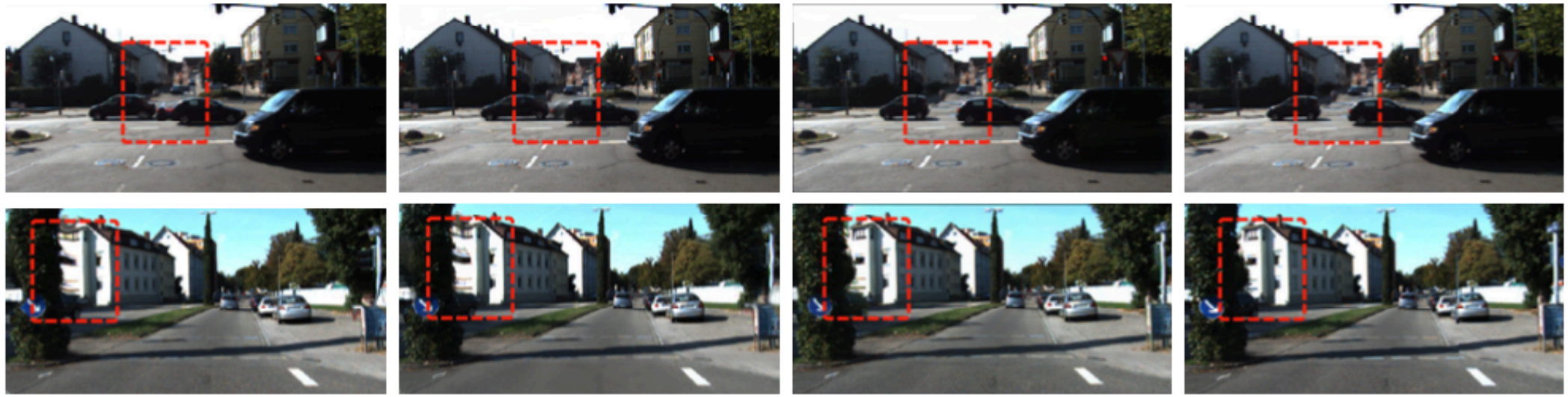
CalTech Pedestrian dataset [7]



(a) $\mathbf{x}_{t-5}$ (b) $\mathbf{x}_{t-2}$ (c) PredNet (d) ContextVP (e) DPG (Ours) (f) $\mathbf{x}_t$

Results

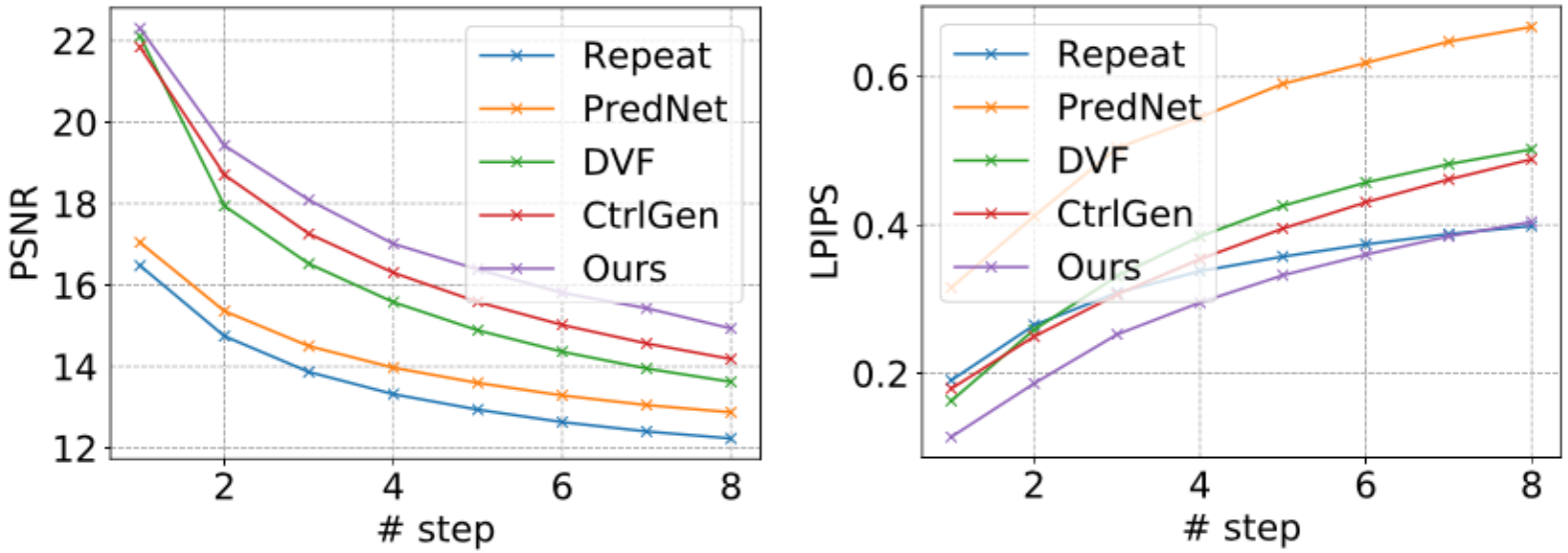
KITTI Flow dataset [23] :



(a) DVF (b) CtrlGen (c) DPG (Ours) (d) $\mathbf{x}_t$

Results

Take in 4 frames as input and recursively predict next 8 frames



(a) PSNR↑ (b) LPIPS↓

Results

Method	PSNR↑	SSIM↑	LPIPS↓
Warped only [19]	21.6	0.684	0.139
No occlusion map (auto-encoder)	21.3	0.674	0.120
Learned occlusion map [11]	21.8	0.679	0.179
Sparse occlusion map (Ours)	**22.3**	**0.696**	**0.114**

Table 3: Ablation study on different design choice for occlusion map.

Method	Learned occlusion map [11]	Sparse occlusion map (Ours)
IoU↑	0.0411	24.91

Table 4: Evaluation for learned occlusion map and our sparse occlusion map against ground-truth occlusion map.

Method	PSNR↑	SSIM↑	LPIPS↓
$\beta = 0$ (unmasked pixel only)	20.7	0.663	0.159
$\beta = 0.1$	20.9	0.671	0.134
$\beta = 1$ (same as unmasked pixel)	21.2	0.675	0.123
$\beta = \infty$ (masked pixel only)	18.4	0.514	0.206
$\beta = 10$ (Ours)	**22.3**	**0.696**	**0.114**

Table 5: Ablation study on different masking weight β.

Conclusion

- We propose a disentangled fusion pipeline:
  - gates two tasks into two separate modules;
  - directly evaluate pixel density map after warping for sparse occlusion maps.
- Experiments on synthetic and real datasets show our effectiveness against prior works.

Thanks!