Disentangling Propagation and Generation for Video Prediction

Authors: Hang Gao, Huazhe Xu, Qi-Zhi Cai, Ruth Wang, Fisher Yu, Trevor Darrell

Venue: International Conference on Computer Vision (ICCV 2019)

Presenter: Tanuj Jain

CAP 6412- Advanced Computer Vision, Spring 2020

Outline of this presentation

- Video Prediction
- Existing Approaches to solve the problem
- Proposed method
- Datasets
- Results
- Conclusion

Video Prediction

- Given a sequence of frames, need to predict the next frame.
- Such an easy task for humans, but not for computers, Why?

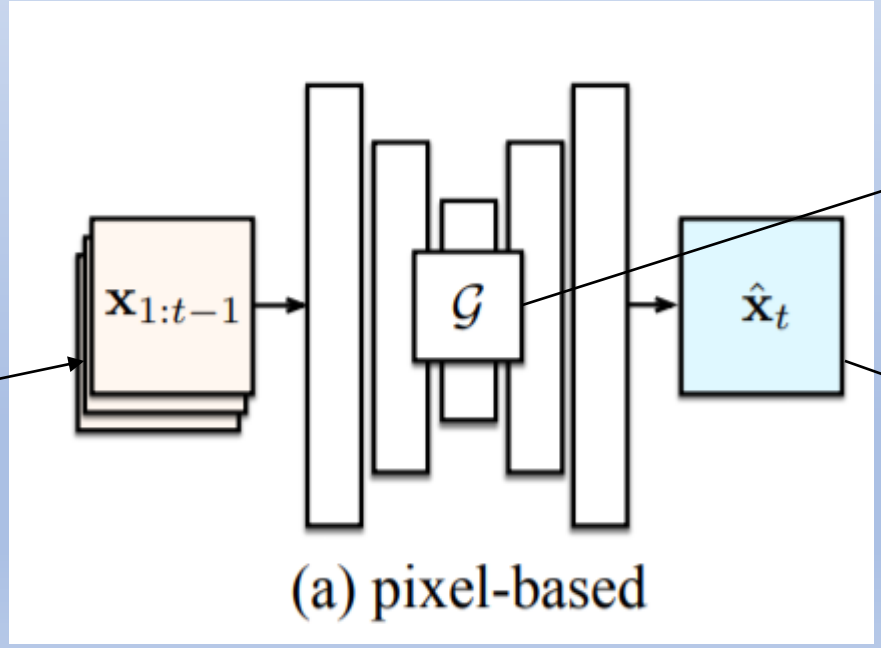


Existing Approaches to solve the problem

- There are mainly 3 approaches for Video prediction.
- Pixel Based Approach(PredNet, MCNet)
- Motion Based Approach(DVF)
- Linear Fusion (CtrlGen)

Pixel Based Approach

- Generate target frame pixel by pixel from scratch
- Use implicit motion representations.



Input Frames

Pixel Generator

Predicted Frame

Pixel Based Approach

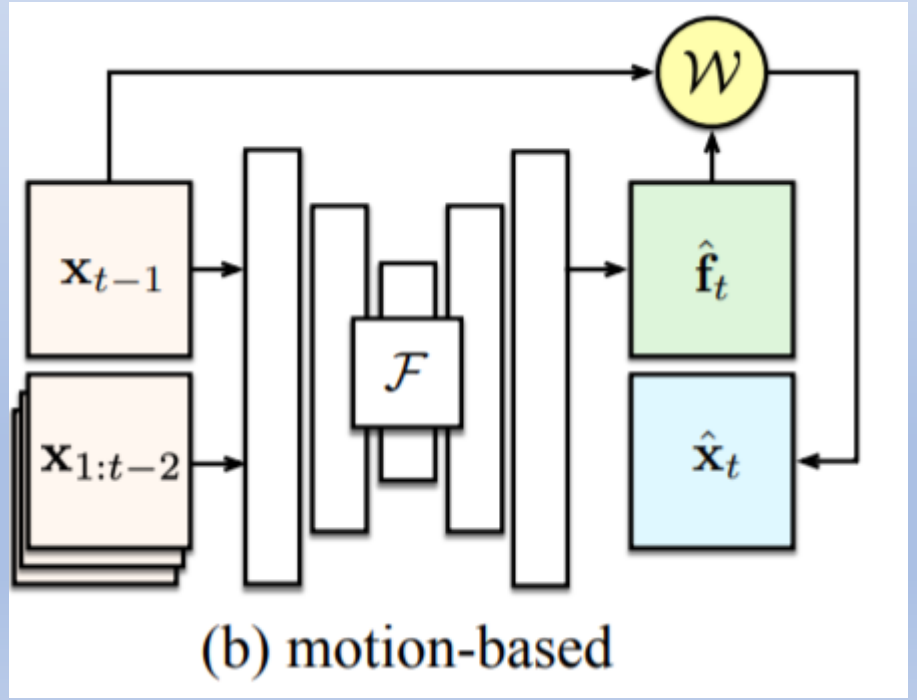
- Drawback



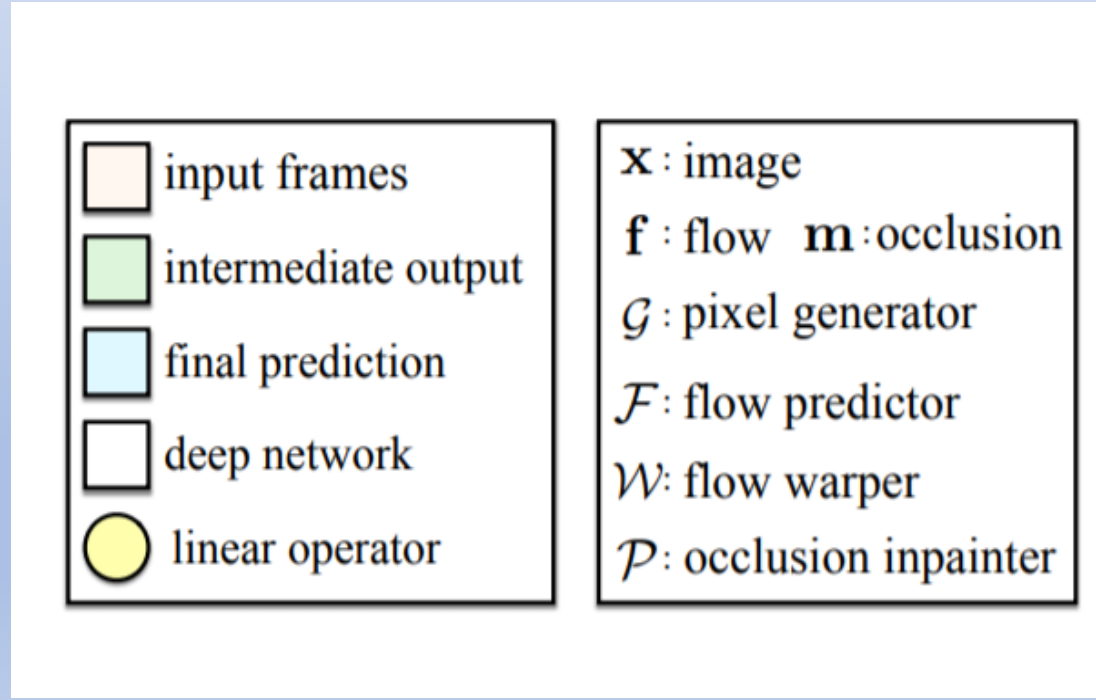
Suffers from Blurry effects

Motion Based Approach

- Predict the optical flow field
- Use of warping function

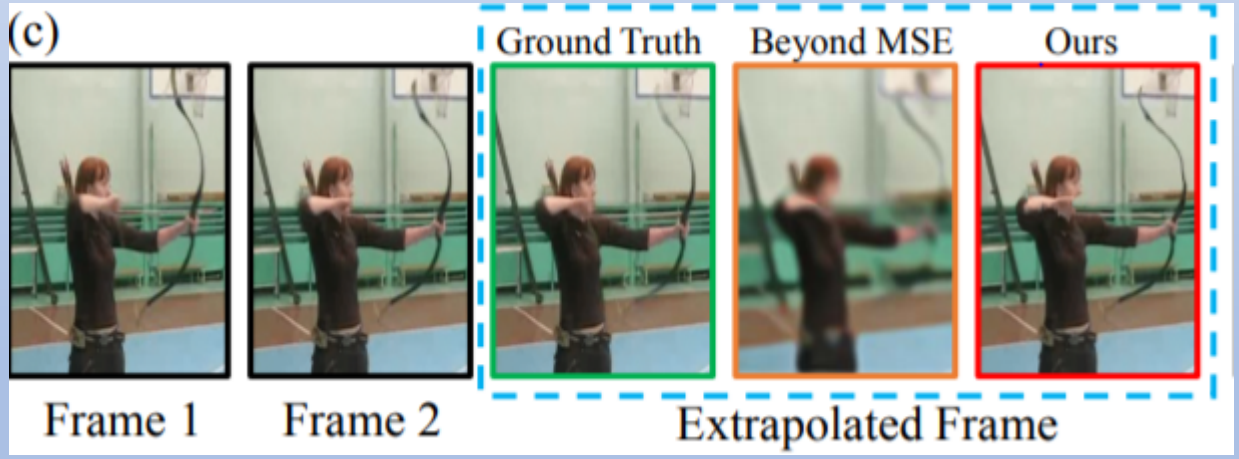


(b) motion-based



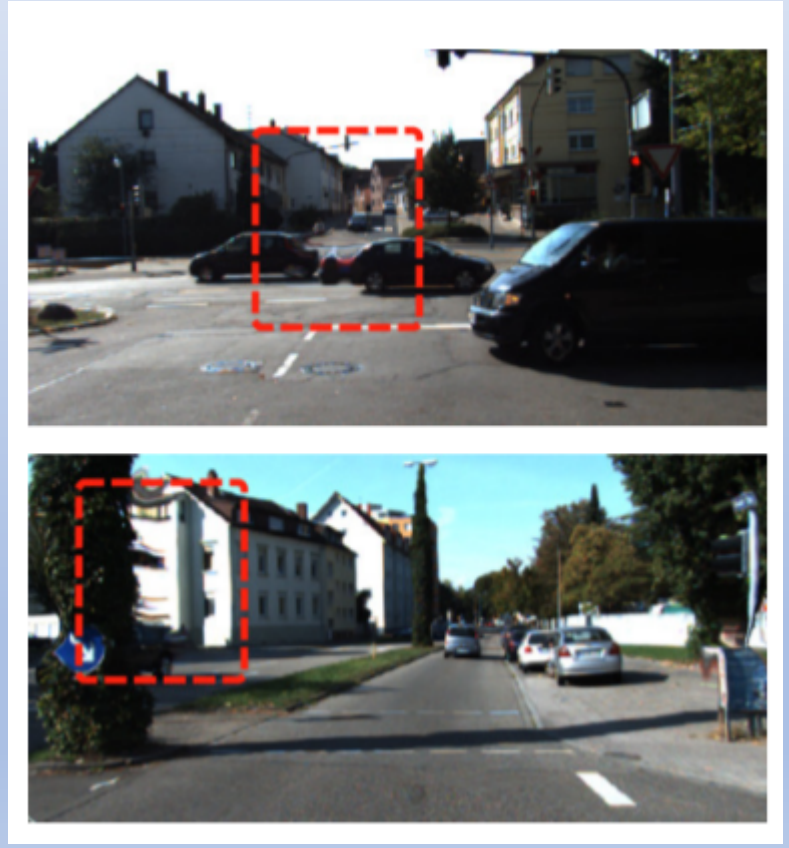
Motion Based Approach

- Result
- Produces sharp images



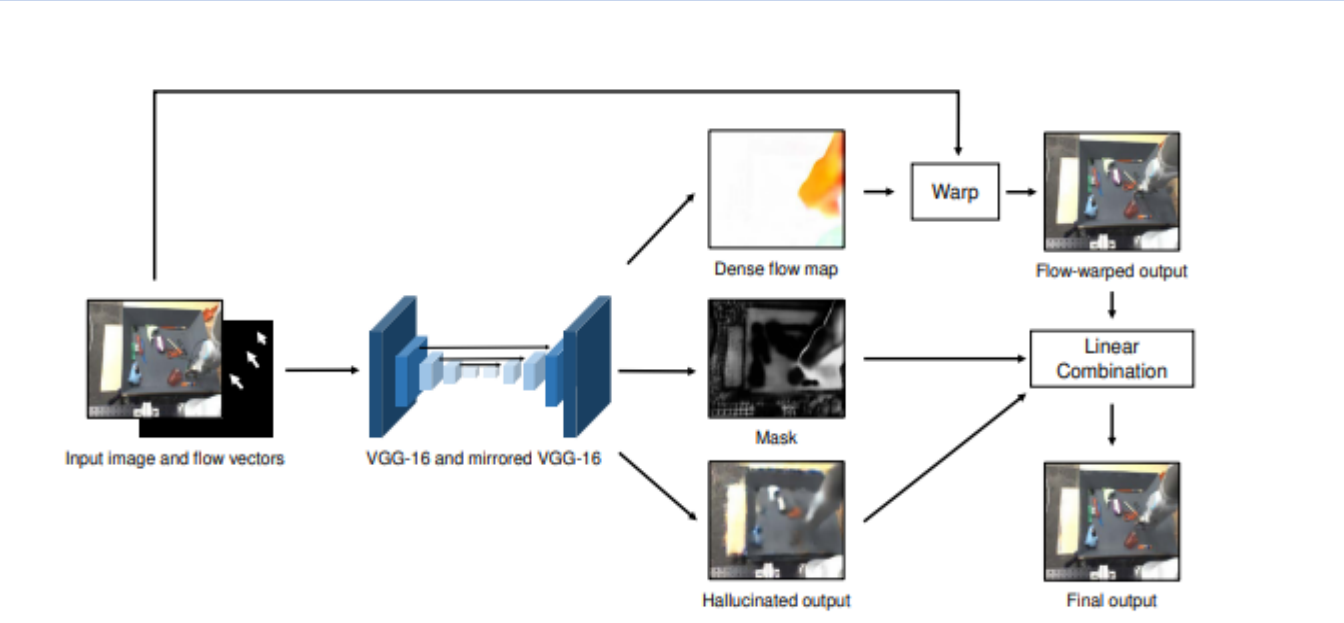
Motion Based Approach

- Drawback



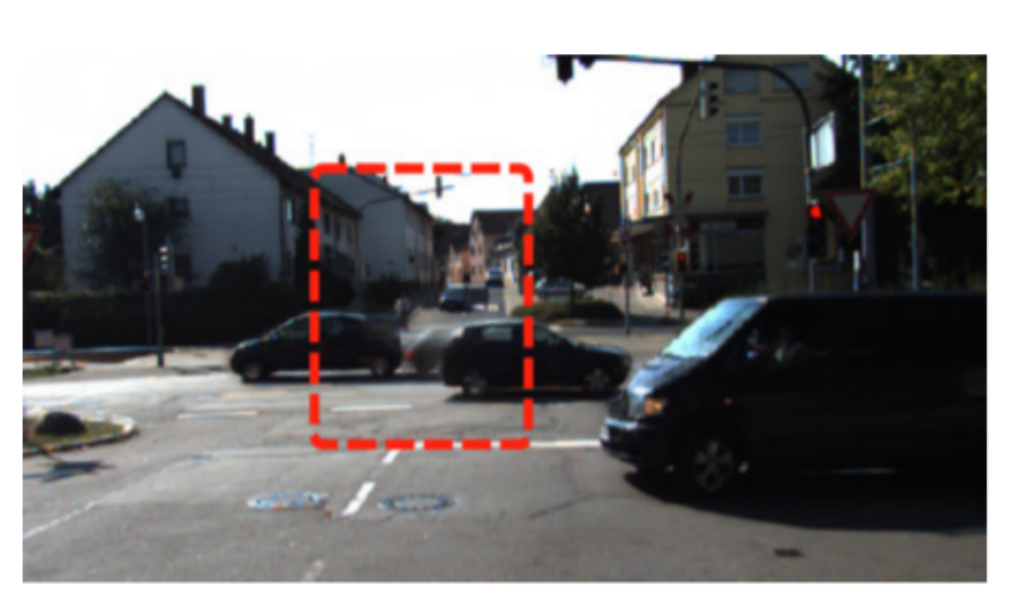
Linear Fusion

- Combination of Flow map, Mask and Hallucinated output.
- Use of warping Function.



Linear Fusion

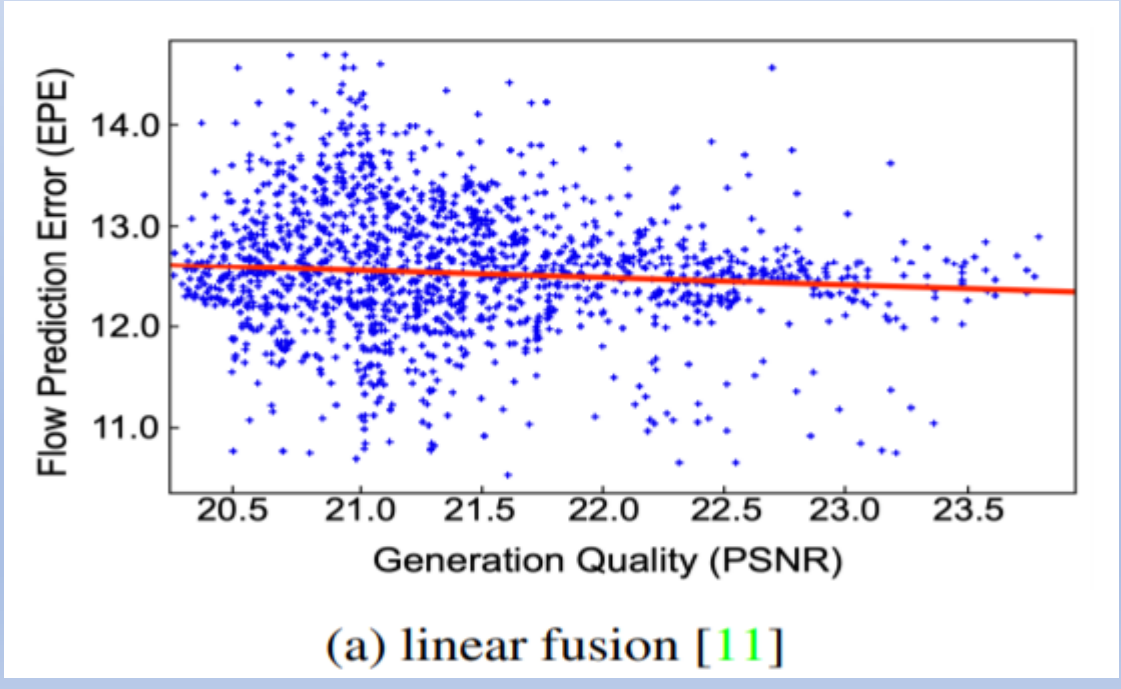
- Drawbacks
- Hallucinations leads to this kind of image.



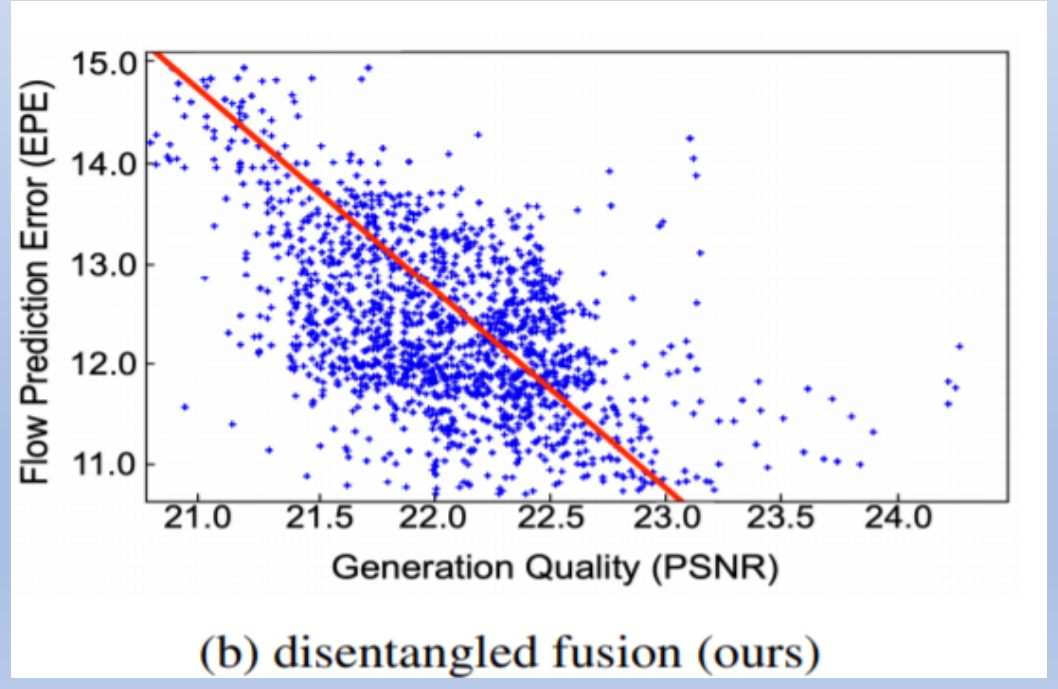


Motivation for Proposed Method

- Pixel and flow do not correlate significantly.
- Why not separate the two modules.

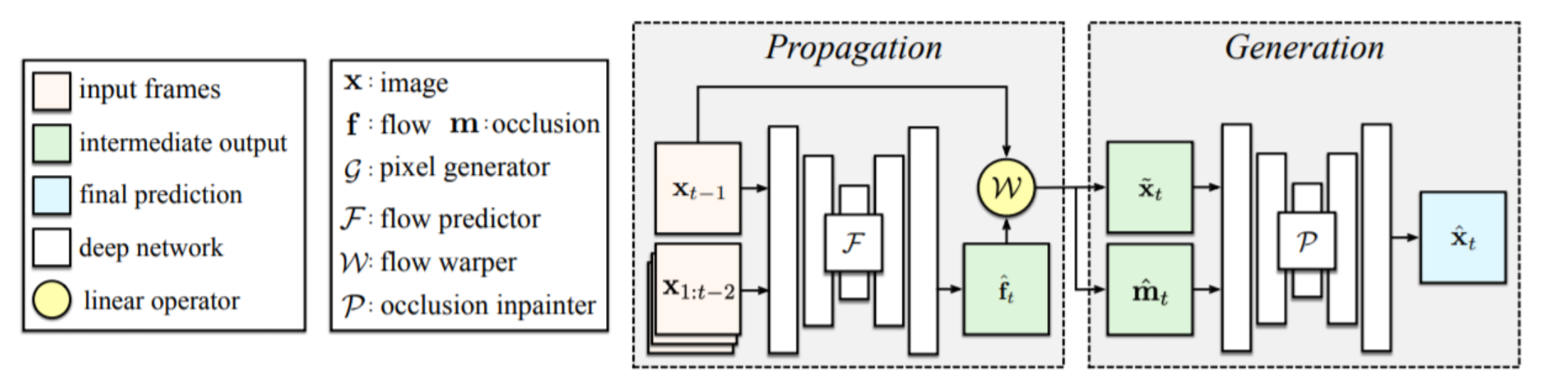


(a) linear fusion [11]



(b) disentangled fusion (ours)

DPG(proposed Method)

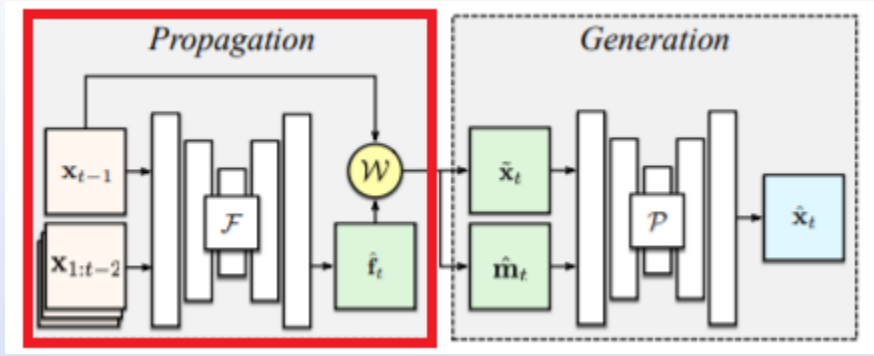


Proposed Method

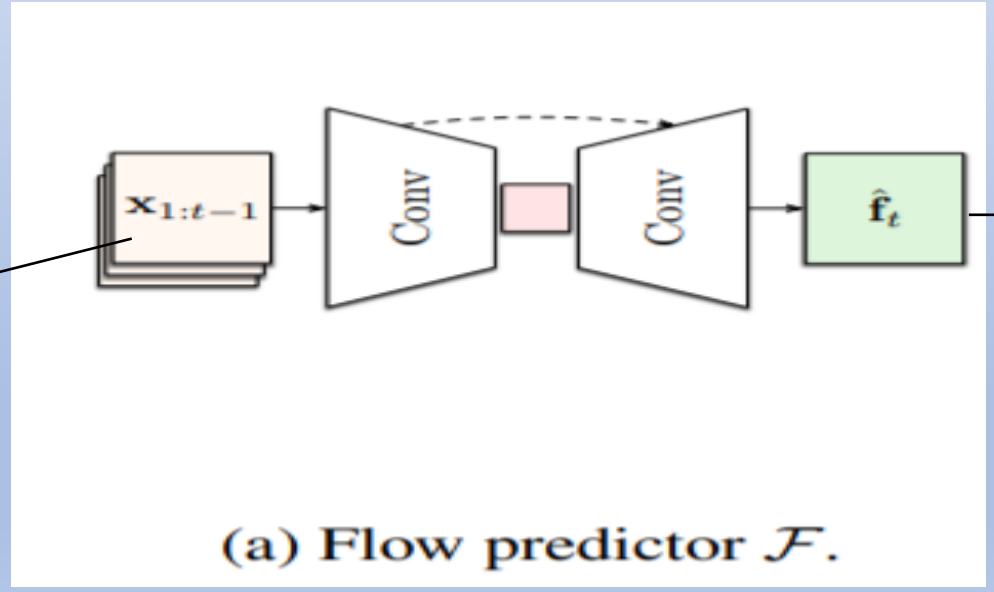
- Similar to Linear Fusion Method, but different in few ways:

- Train pixel and flow predictors separately.
- Use of occlusion Inpainter for pixel generation.
- Use flow predictions as a proxy for post warping confidence.

Dpg(Flow Predictor)



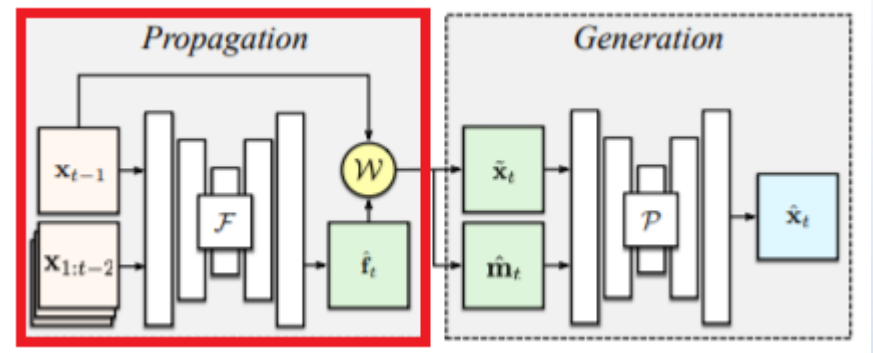
- Predict Future Motion Dynamics



Input Frames

2 channel Flow Field

Dpg(Flow Predictor)



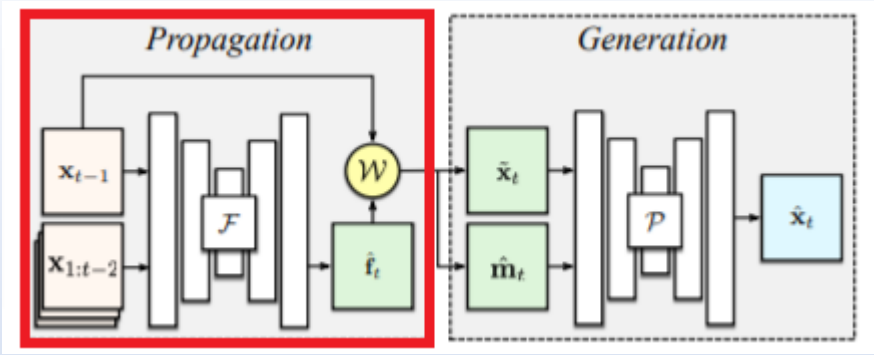
- The flow field is calculated using the equation(1)

$$\hat{\mathbf{f}}_t = \{(\Delta_i, \Delta_j)\} = \mathcal{F}(\mathbf{x}_{1:t-1}). \qquad (1)$$

- The future frame is sampled from the last given frame using the equation(2).

$$\tilde{\mathbf{x}}_t = \mathcal{S}(\mathbf{x}_{t-1}; \hat{\mathbf{f}}_t), \qquad (2)$$

Dpg(Flow Predictor)



- To train the Flow predictor module two different loss functions are used.

1- Masked Pixel Loss.

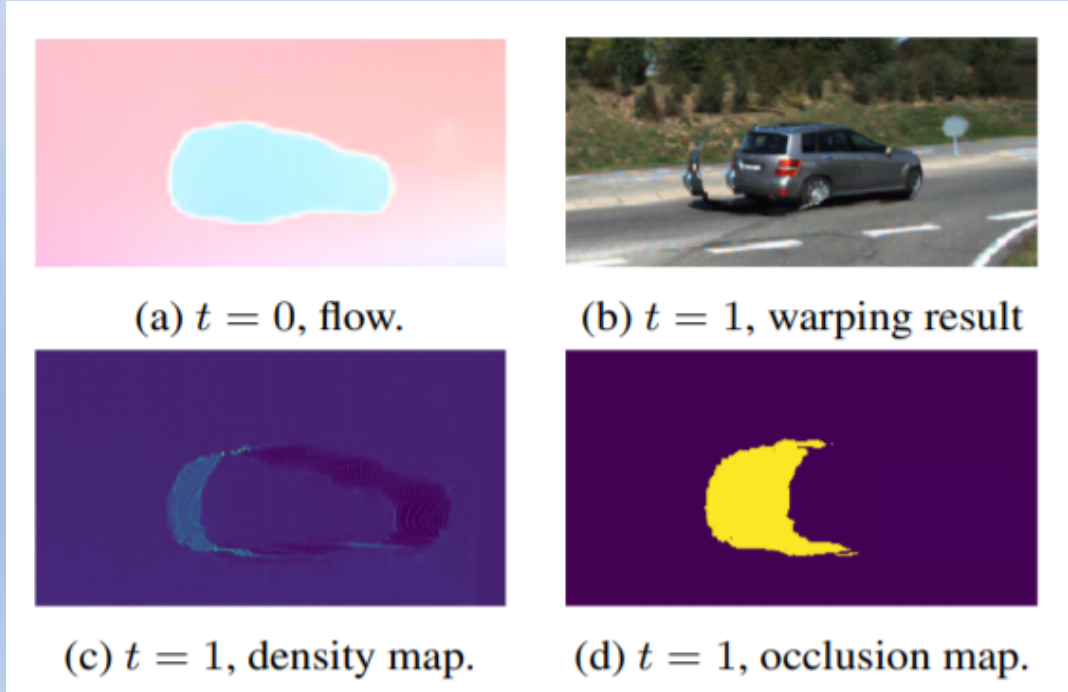
$$\mathcal{L}_p(\tilde{\mathbf{x}}_t, \mathbf{x}_t; \hat{\mathbf{m}}_t) = \alpha \frac{1 - SSIM(\tilde{\mathbf{x}}_t \odot \hat{\mathbf{m}}_t, \mathbf{x}_t \odot \hat{\mathbf{m}}_t)}{2} + (1-\alpha)\|\tilde{\mathbf{x}}_t \odot \hat{\mathbf{m}}_t - \mathbf{x}_t \odot \hat{\mathbf{m}}_t\|_1 \quad (3)$$

2-Smooth Loss

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

Dpg(Occlusion Map)

- Ghosting Effect in dis-occluded Regions.
- These regions should excluded.



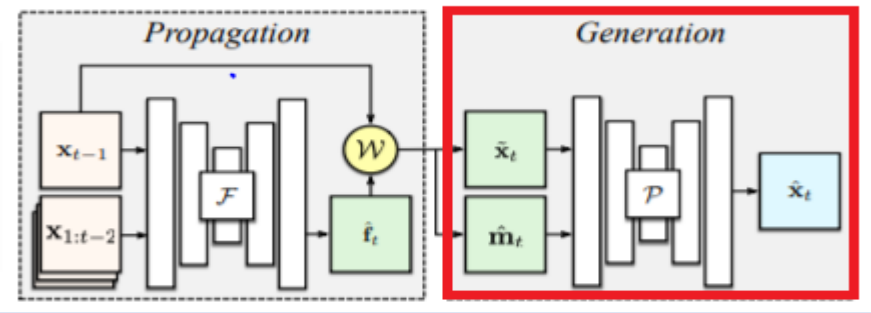
(a) $t = 0$, flow.

(b) $t = 1$, warping result

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

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

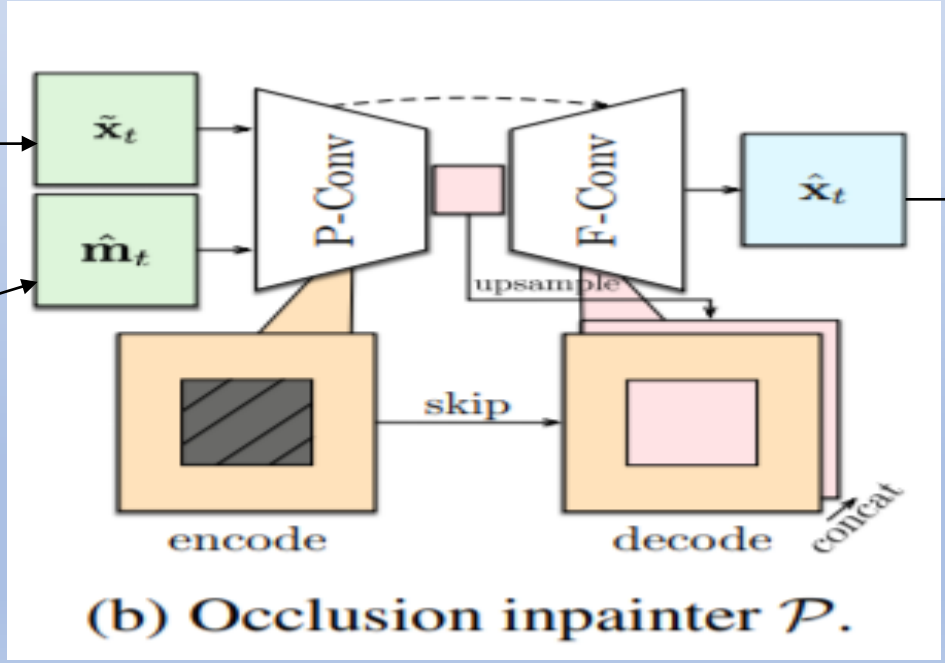
Dpg(occlusion Inpainting)



- Fills the missing pixels in the propagated frame.

Propagated Frame

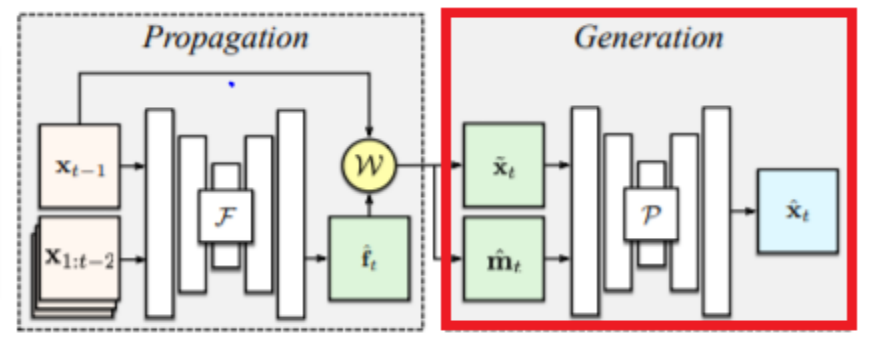
Occlusion Map



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

Final Prediction

Dpg(Occlusion Inpainting)



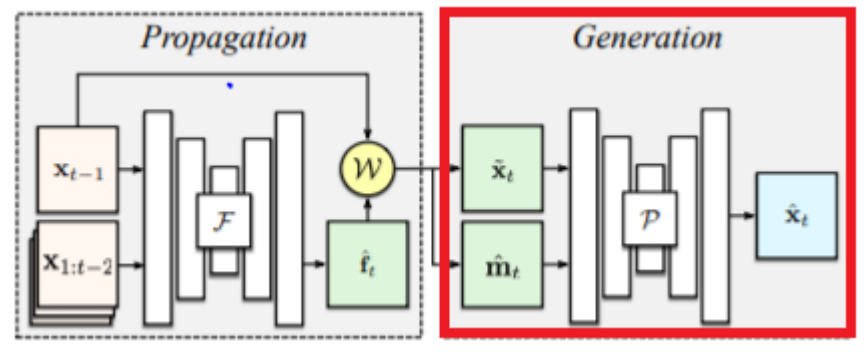
- The training of this module is done with all temporal independent losses.
- Total 4 loss are used to train the inpainter module.

1-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.

Dpg(Occlusion Inpainting



- 2- Perceptual and Style Loss

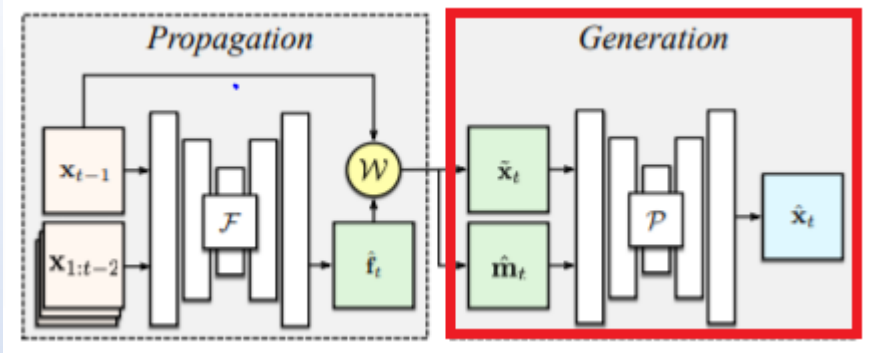
$$\mathcal{L}_{prc} = \frac{1}{n}\sum_{n} \| \left[\psi(\tilde{\mathbf{x}}) - \psi(\mathbf{x})\right] \odot \hat{\mathbf{m}}_t \|_1 + \beta \| \left[\psi(\tilde{\mathbf{x}}) - \psi(\mathbf{x})\right] \odot (1 - \hat{\mathbf{m}}_t) \|_1,$$

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

- Total variance Loss(encourage similar texture)

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

Dpg(Occlusion Inpainting)



- Semantic Loss

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

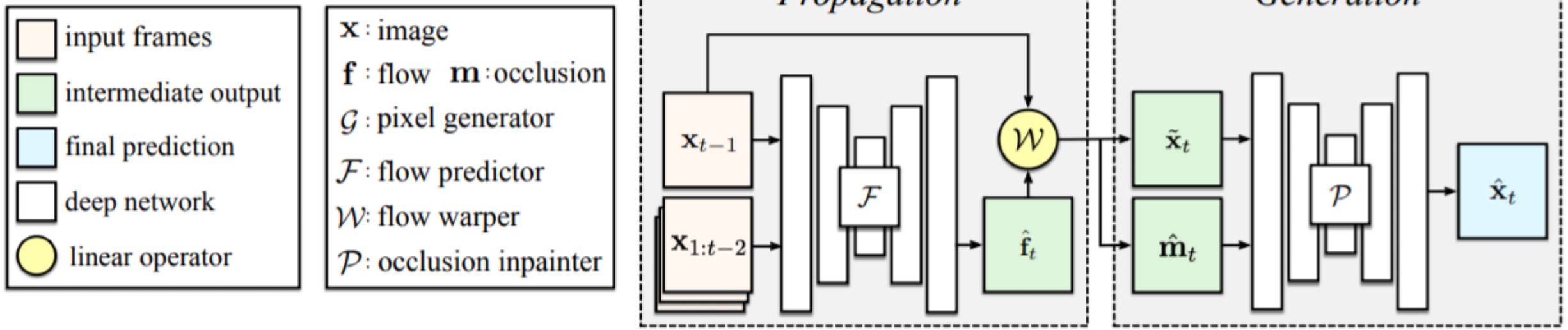
- Finally total loss for inpainting module is

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

Dpg-Training Phase

- Overall Training objective is formulated as

$$\min_{\mathcal{F},\mathcal{P}} \left(\mathcal{L}_{\mathcal{F}} + \mathcal{L}_{\mathcal{P}}\right).$$



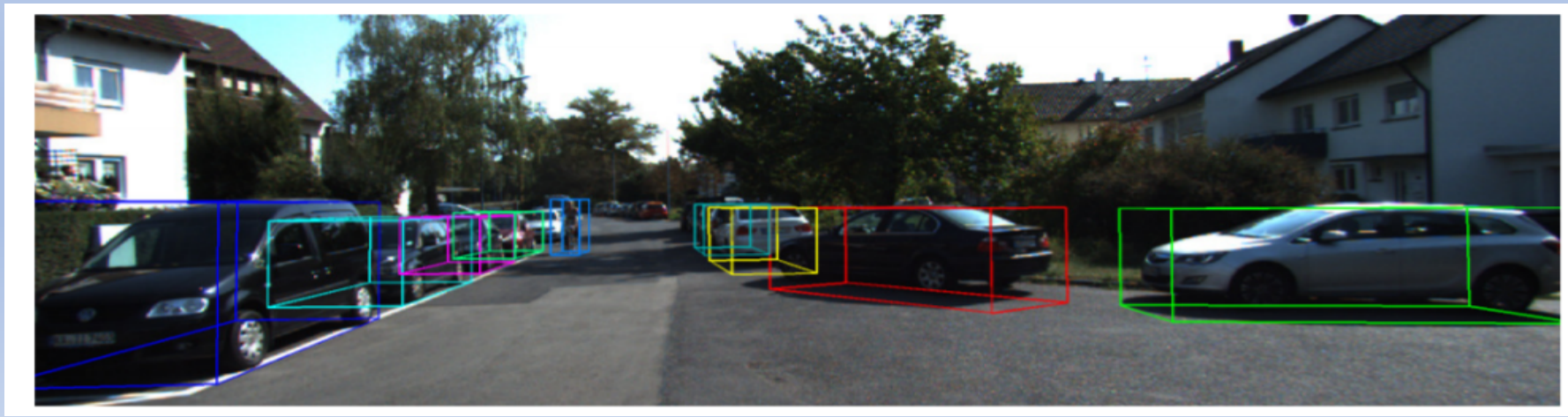
Datasets

- Caltech Pedestrian Detection Benchmark
- 10 hours of 640x480 30Hz videos.



Dataset

- KITTI Dataset
- Two versions Flow and Raw version.
- Raw version has 6 hours of traffic scenarios at 10-100 Hz having bounding box annotations.
- Flow version has 3823 examples of training, 378 for validation and 4167 for testing.



Results

- Metrics used are PSNR, SSIM and LPIPS
- Next Frame Prediction on Caltech Pedestrian.

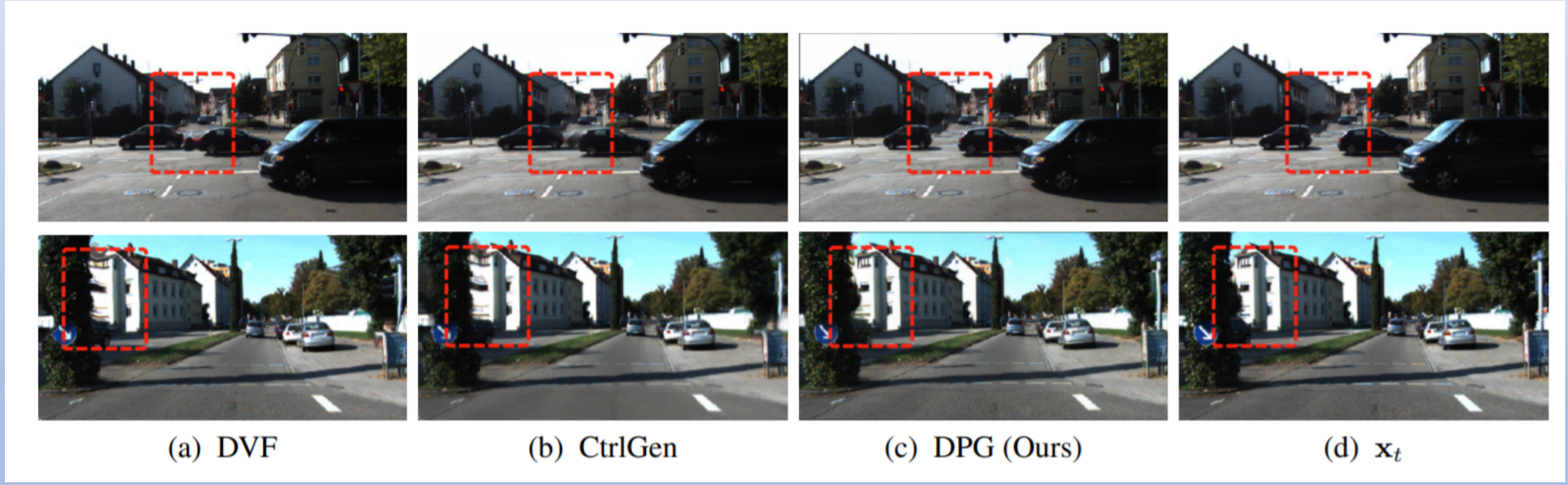
Method	Family	PSNR↑	SSIM↑	LPIPS↓ ($\times 10^{-2}$)
Repeat	–	23.3	0.779	5.11
BeyondMSE [21]	*P*	–	0.881	–
PredNet [20]	*P*	27.6	0.905	7.47
ContextVP [4]	*P*	**28.7**	0.921	6.03
DVF [19]	*M*	26.2	0.897	5.57
Dual Motion GAN [17]	*F*	–	0.899	–
CtrlGen [11]	*F*	26.5	0.900	6.38
DPG (Ours)	*F*	28.2	**0.923**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

Results

- Next Frame Prediction on KITTI Flow Dataset

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**

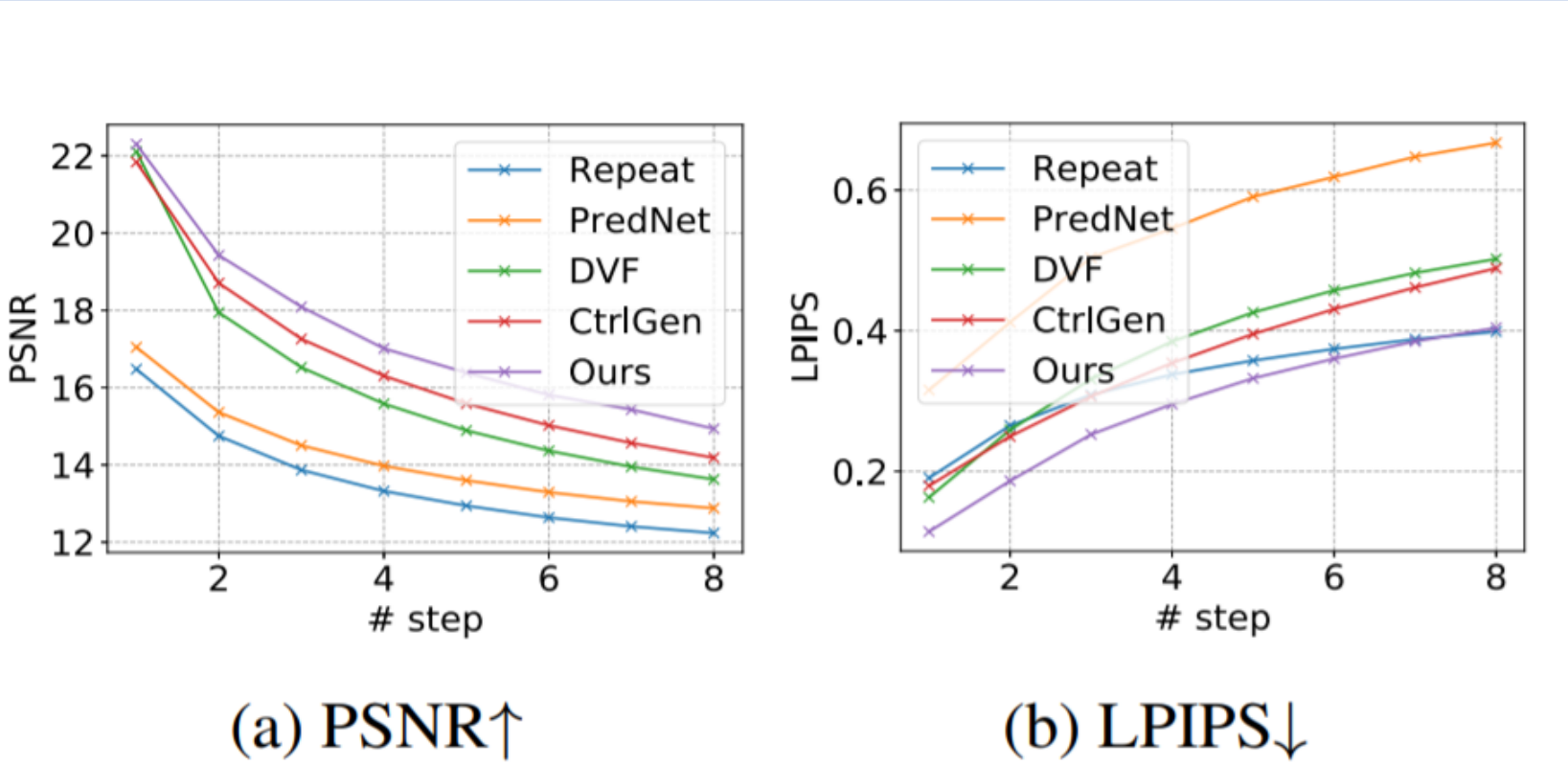
Results



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

Results

- Multi Frame Accuracy



Conclusion

- The authors present a method for video prediction by disentangling motion-specific propagation and motion agnostic generation.
- The authors propose a disentangled fusion pipeline which gates two tasks into two separate modules and directly evaluate pixel density map after warping for sparse occlusion maps.