Unifying Flow, Stereo and Depth Estimation

Haofei Xu[1,2] Jing Zhang[3] Jianfei Cai[4] Hamid Rezatofighi[4] Fisher Yu[1]

Dacheng Tao[5] Andreas Geiger[2,6]

[1]ETH Zurich [2]University of Tübingen [3]The University of Sydney [4]Monash University

[5]JD Explore Academy [6]MPI for Intelligent Systems, Tübingen

https://haofeixu.github.io/unimatch/

Motion & 3D Perception

- Our world is *dynamic & 3D*

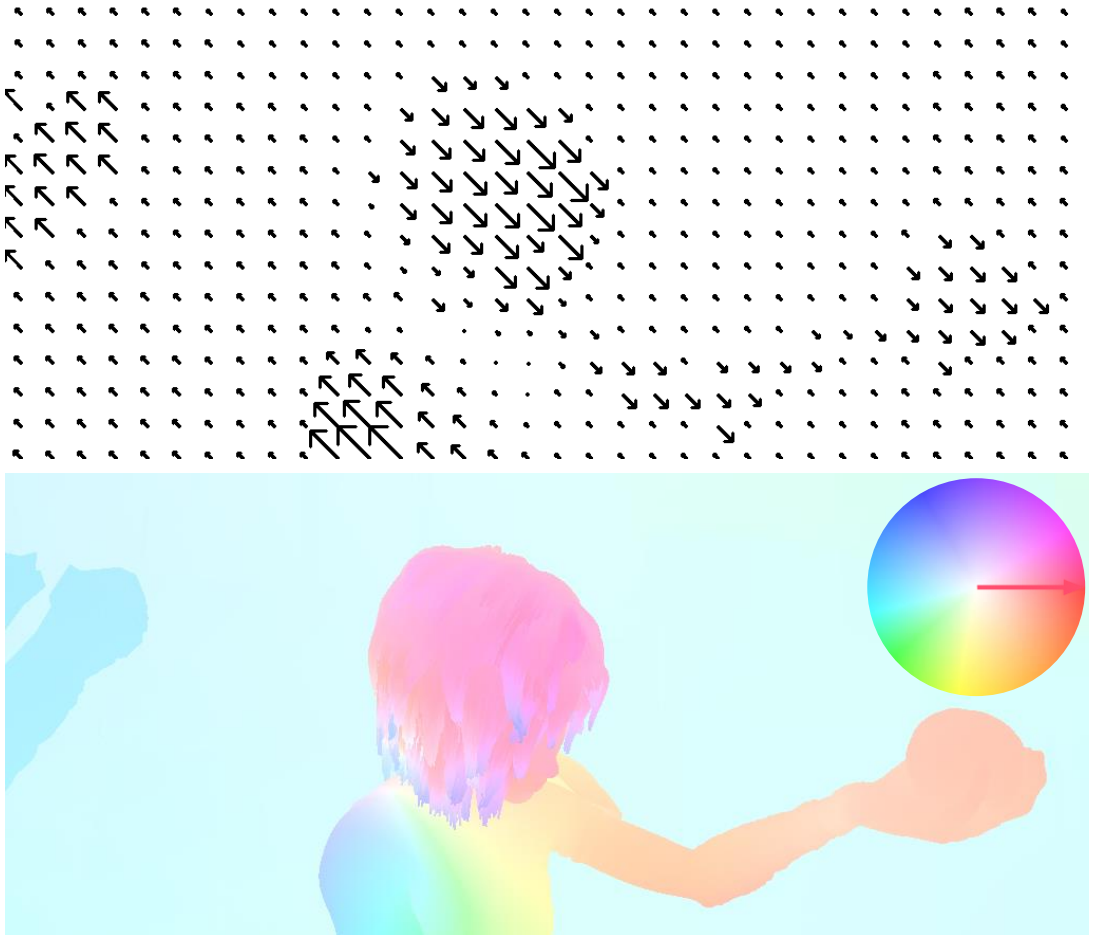
YouTube-8M

World Cup

Optical Flow

- Apparent motion between two video frames

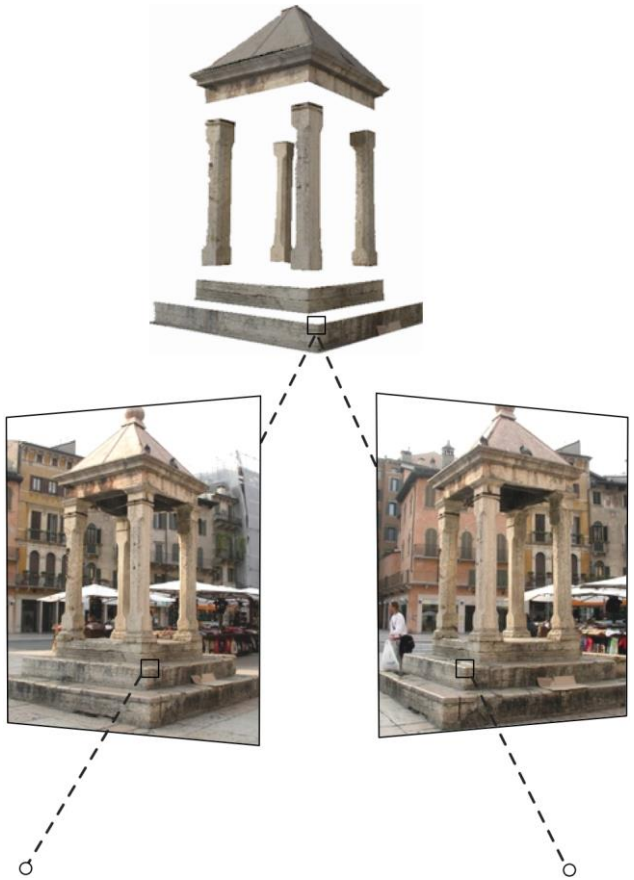
frame 1 & 2



optical flow

Depth

- Distance to the camera

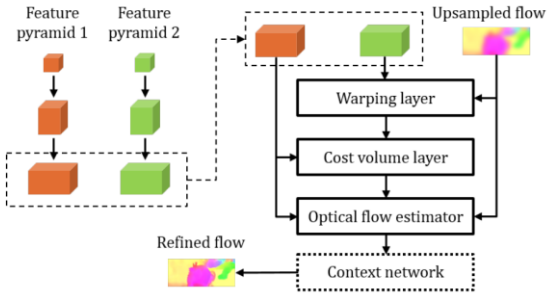


Problem settings:

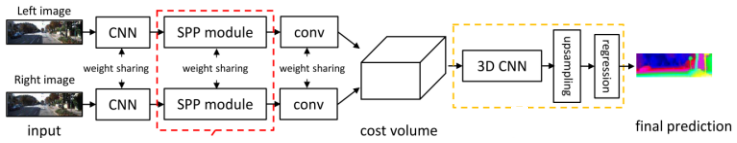
- Rectified stereo matching
- Unrectified depth estimation from posed images

Previous Methods

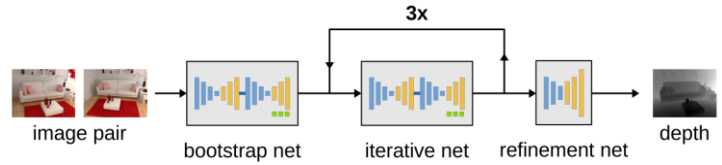
- Design *specialized* architectures for each *specific* task



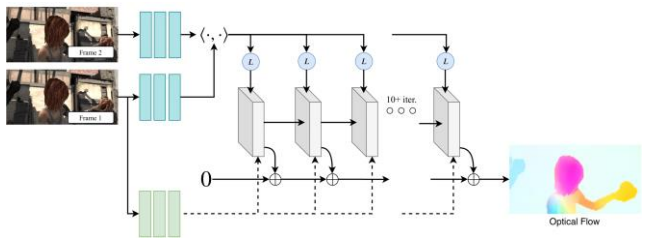
PWC-Net, CVPR 2018



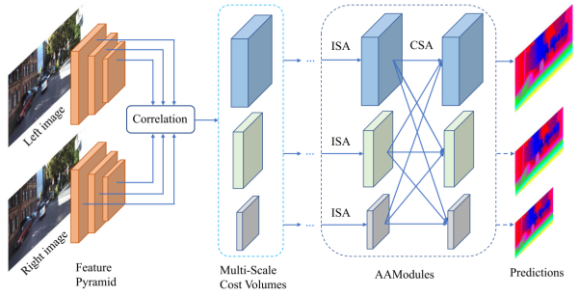
PSMNet, CVPR 2018



DeMoN, CVPR 2017



RAFT, ECCV 2020



AANet, CVPR 2020

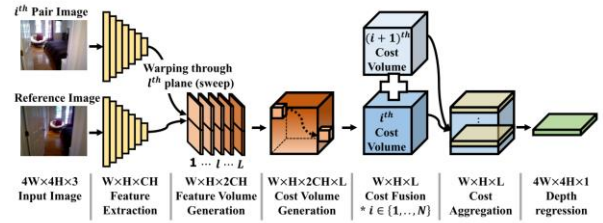


Figure 2: Overview of the DPSNet pipeline.

DPSNet, ICLR 2019

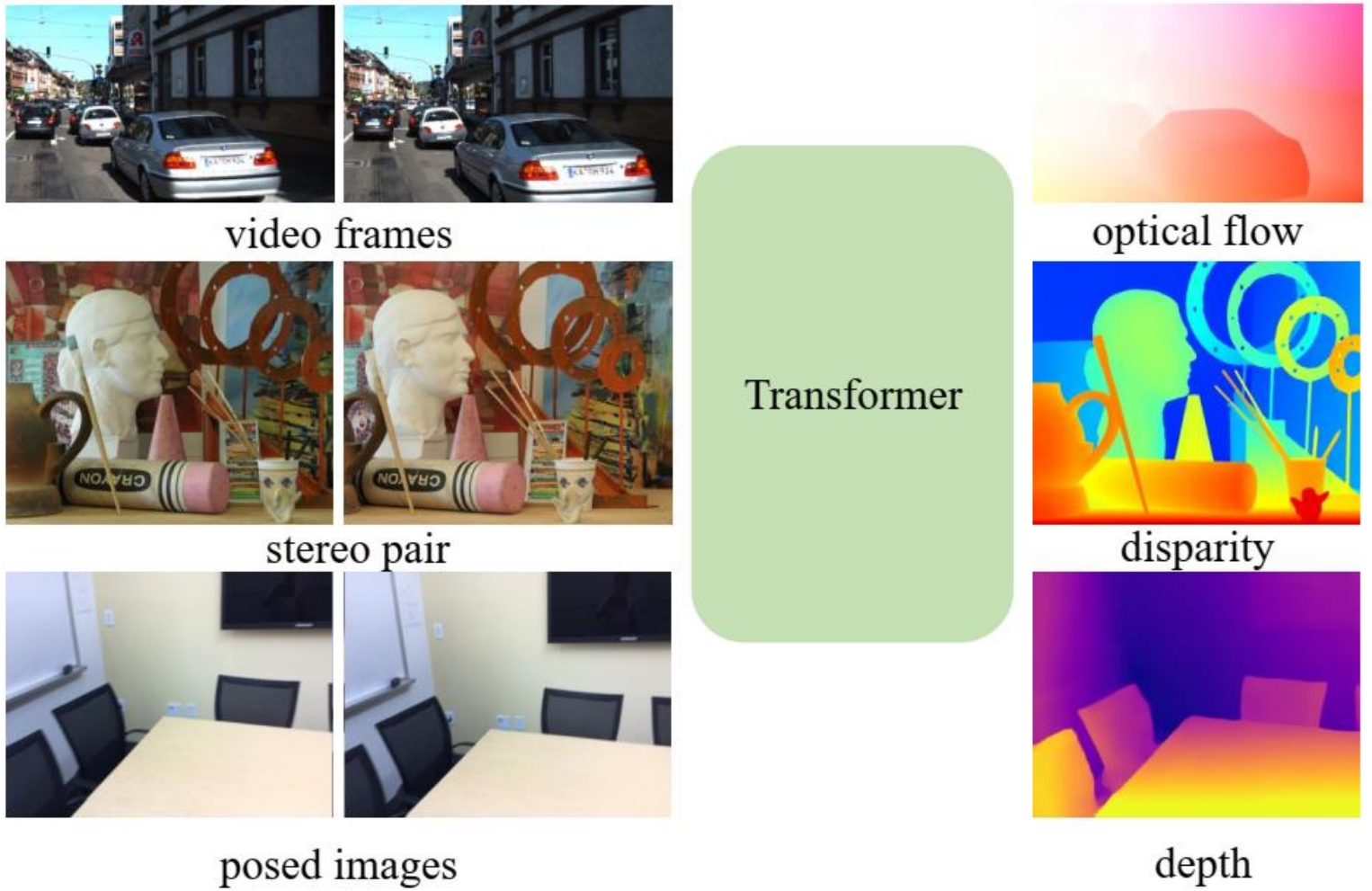
Optical Flow

Stereo Matching

Depth Estimation

Our Approach: UniMatch

- A *unified* model for flow, stereo and depth

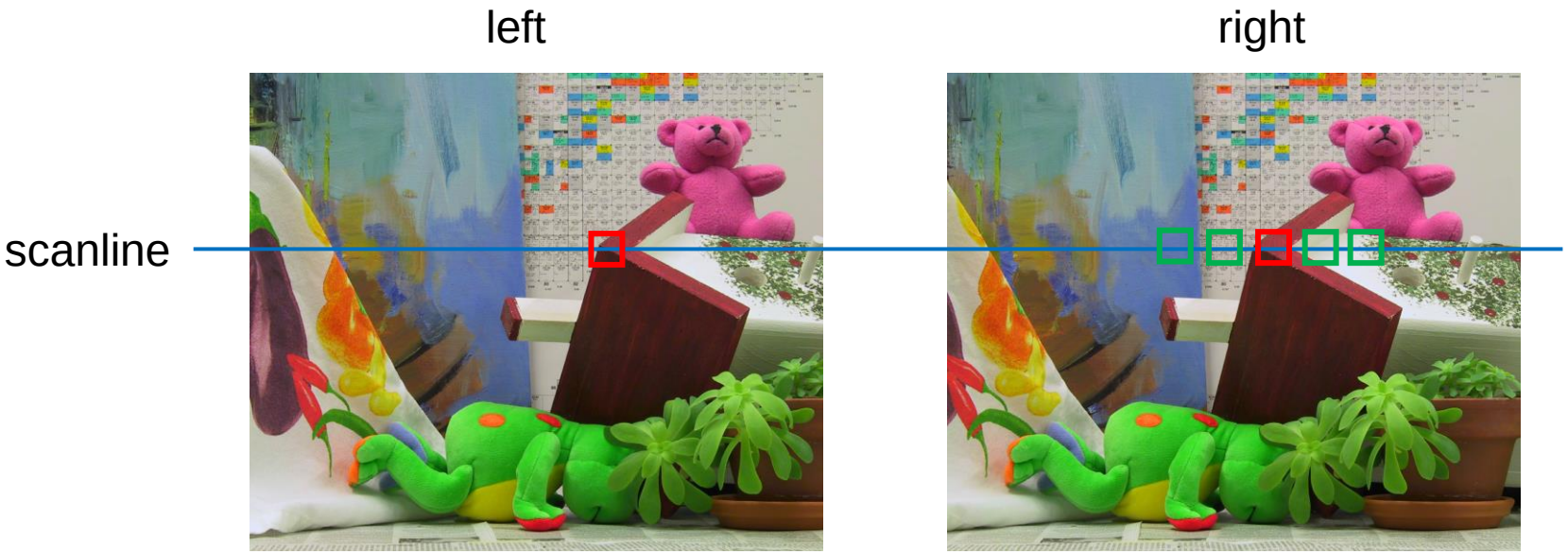


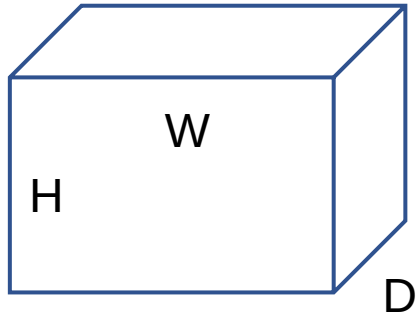
Why Unified Model?

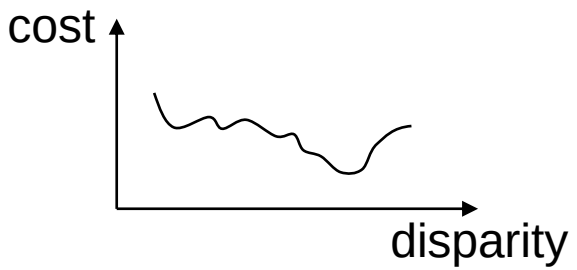
- Focus on the development of a single architecture
- Enable cross-task transfer: reuse pretrained models
- Towards general perception systems

Typical Stereo Pipeline

- Feature extraction → cost volume construction → cost aggregation → disparity computation → disparity refinement

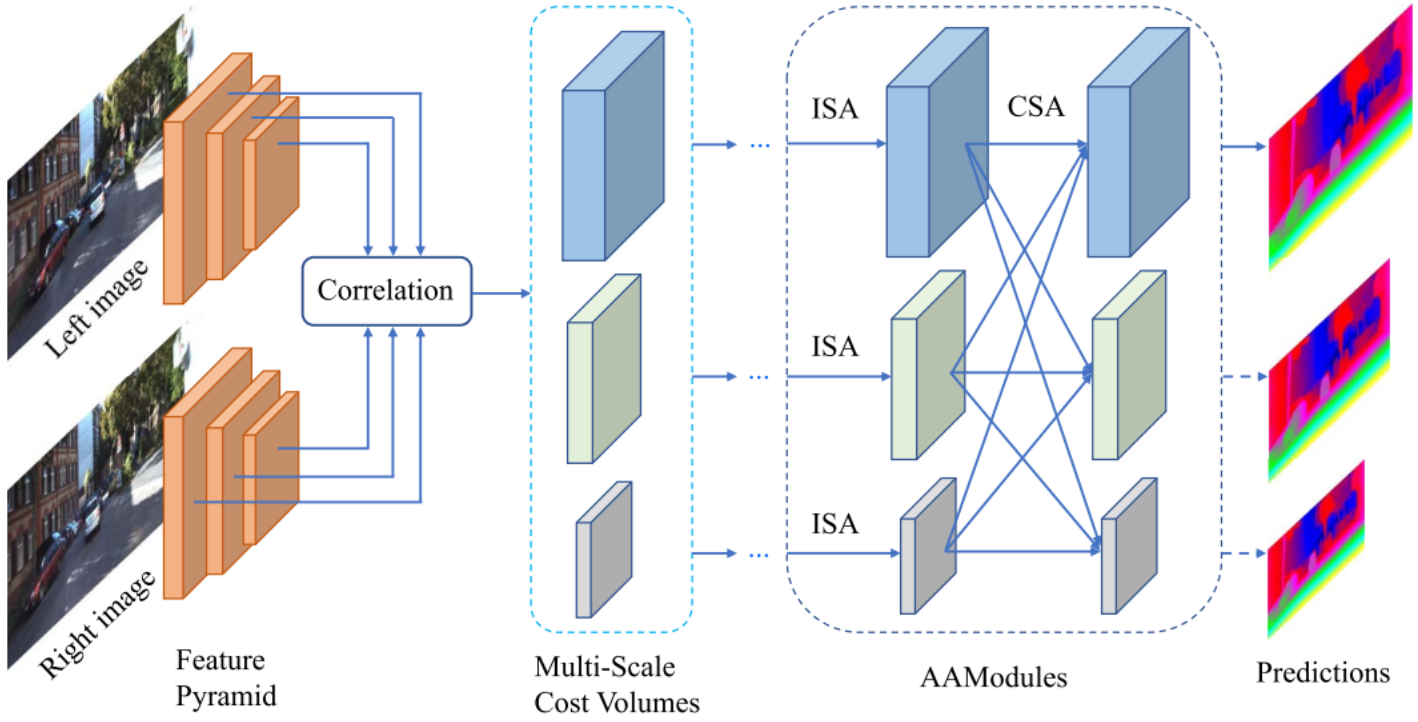






Leaning-based Stereo Pipeline

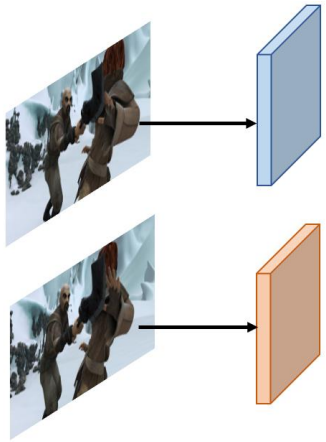
- Maintain traditional pipeline, replace handcrafted components with learnable networks



AANet

Xu and Zhang. AANet: Adaptive Aggregation Network for Efficient Stereo Matching. CVPR 2020

Optical Flow Pipeline



feature extraction

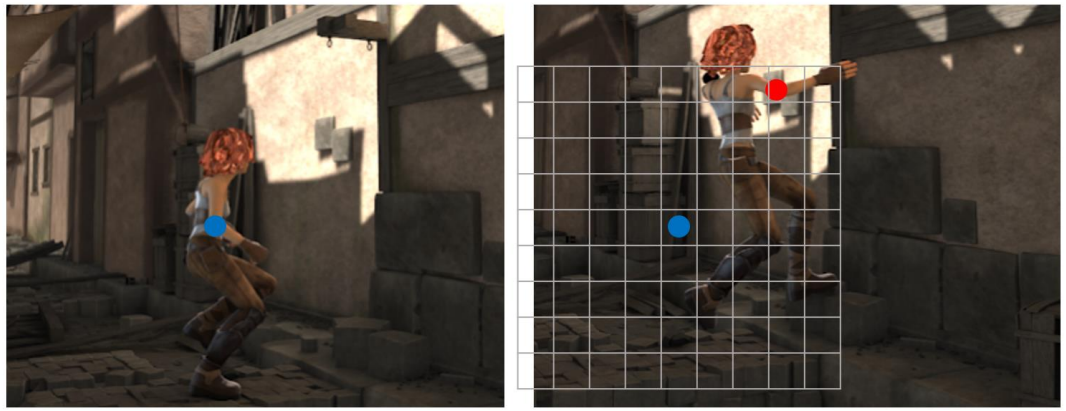
feature1

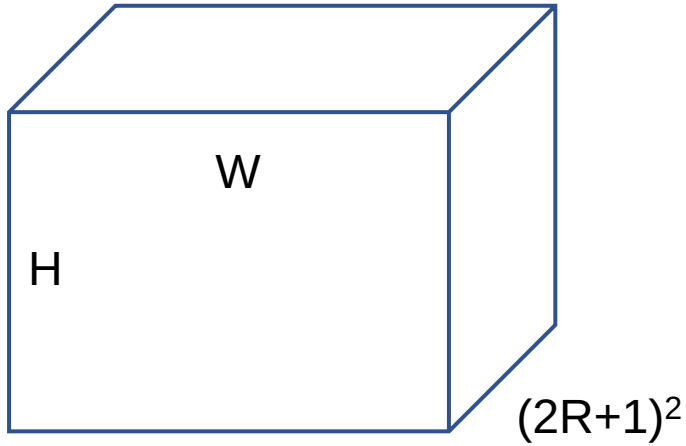
feature2

local correlation

convolutions

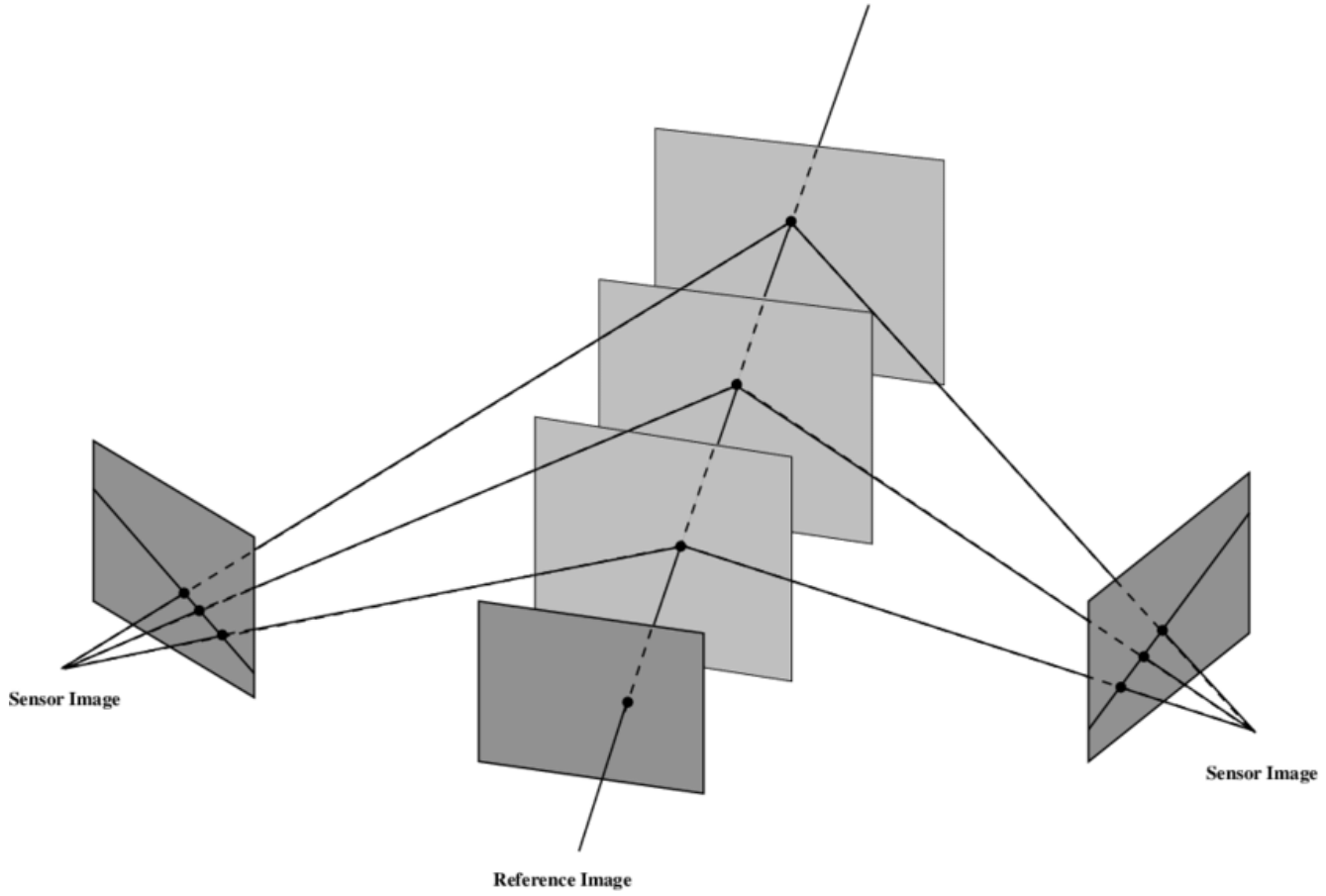
flow





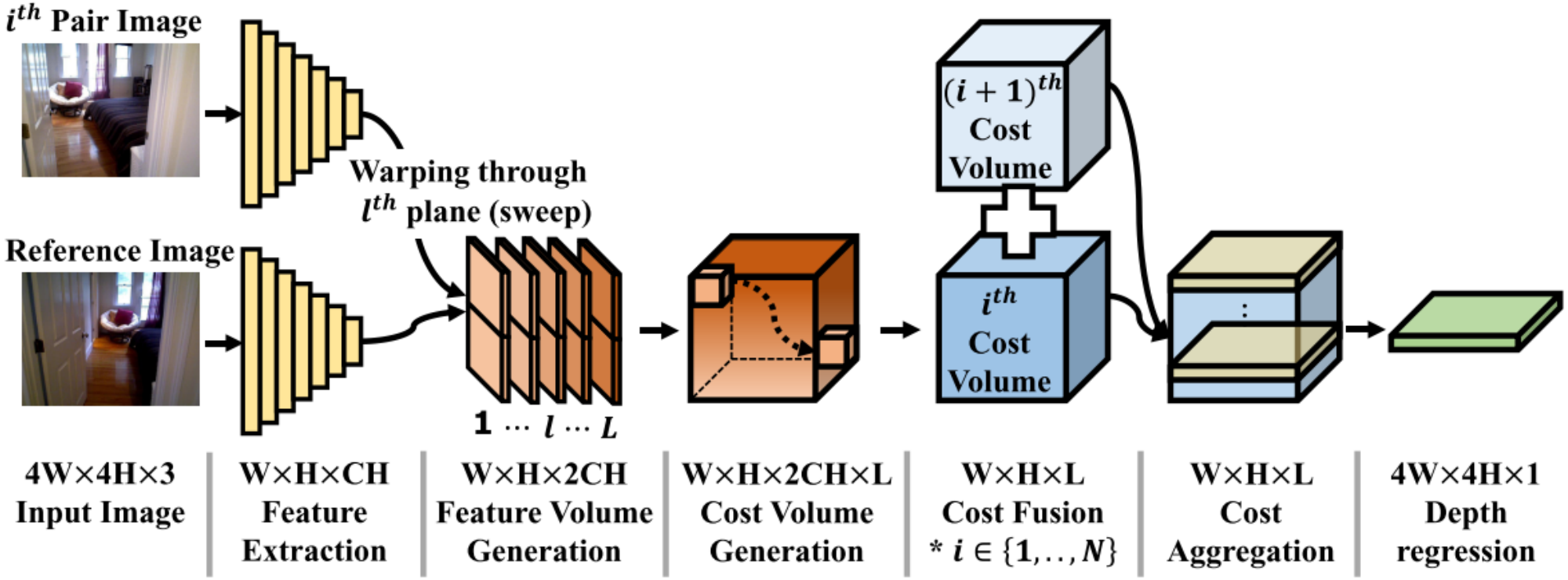
local cost volume

Depth Estimation



Plane-sweep stereo

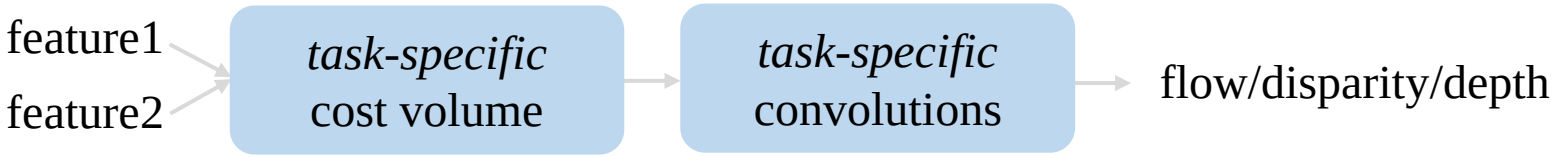
Depth Estimation Pipeline



DPSNet

Im el al. DPSNet: End-to-end Deep Plane Sweep Stereo. ICLR 2019

Summary of Previous Pipelines



- *Task-specific* cost volume size:

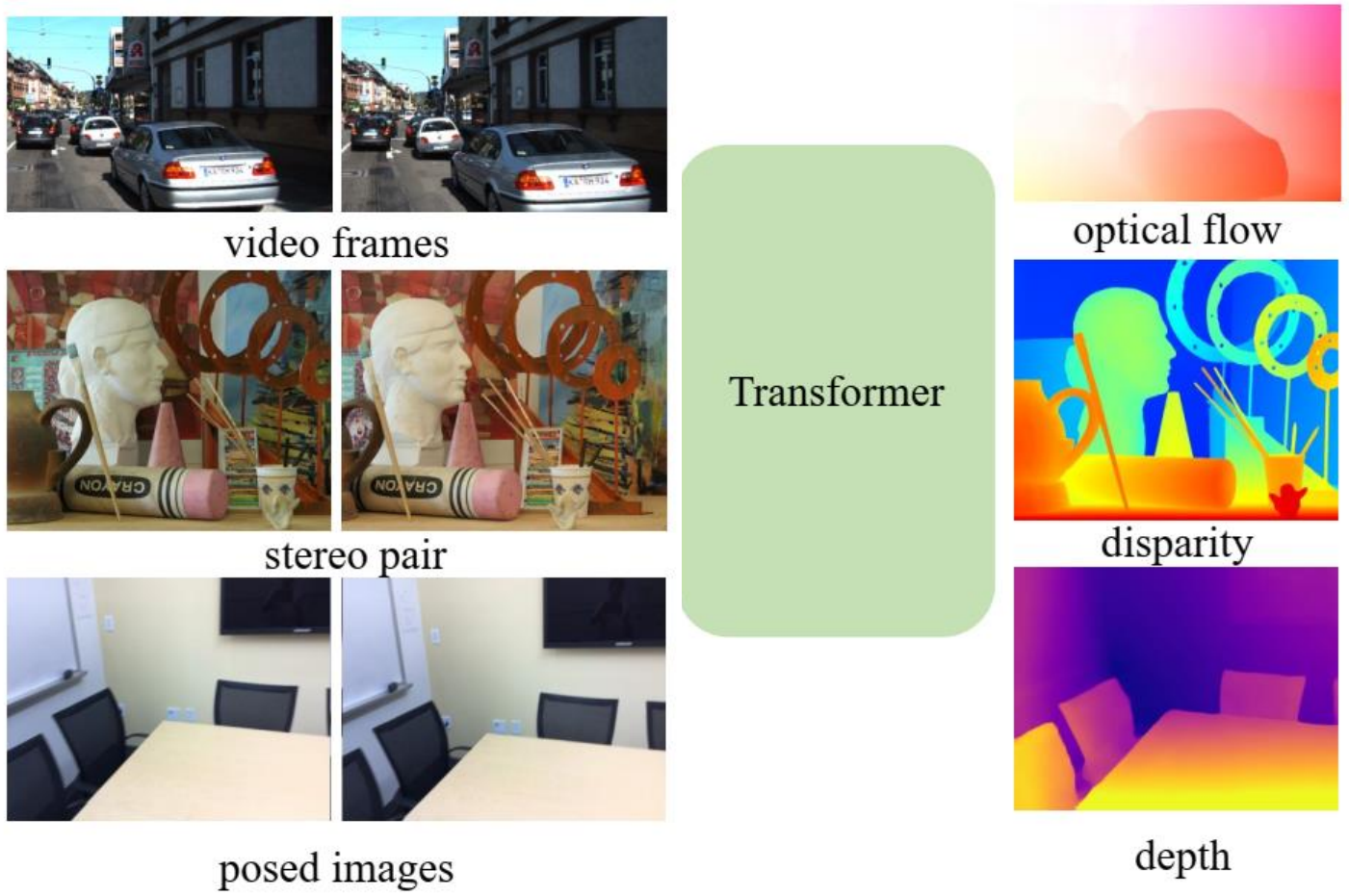
  flow: [H, W, $(2R+1)^2$], stereo: [H, W, D, {C}], depth: [H, W, K, {C}]

- *Task-specific* convolutions:

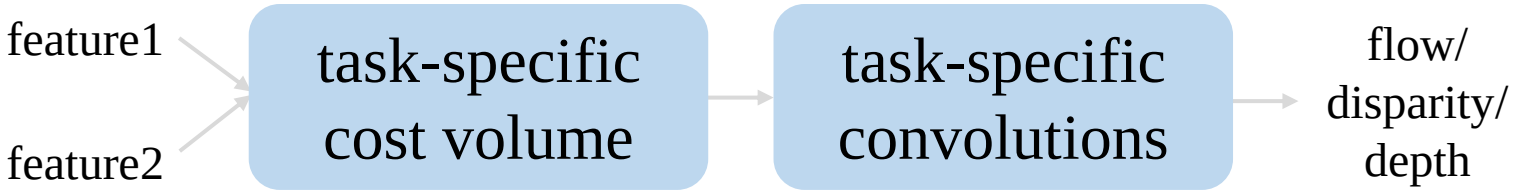
  - Convolutions are dependent on the cost volume size
  - Different types of convolutions (2D, ConvGRU, or 3D)

Our Insight

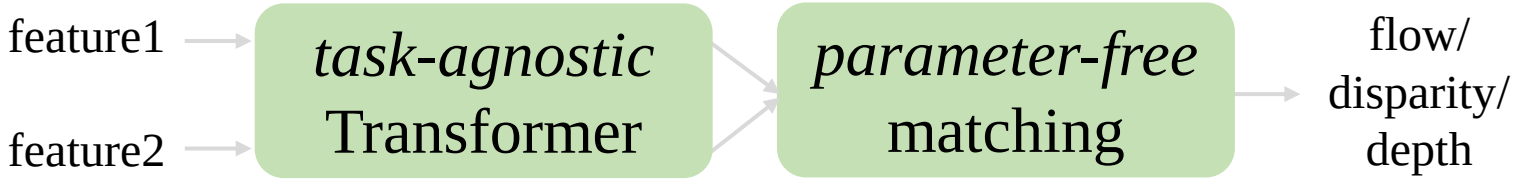
- Unified dense correspondence matching (UniMatch)
- Learn strong features with a Transformer (in particular cross-attention)



Methodology Comparison



Previous task-specific methods



Our unified model

Xu et al. GMFlow: Learning Optical Flow via Global Matching. CVPR 2022, Oral

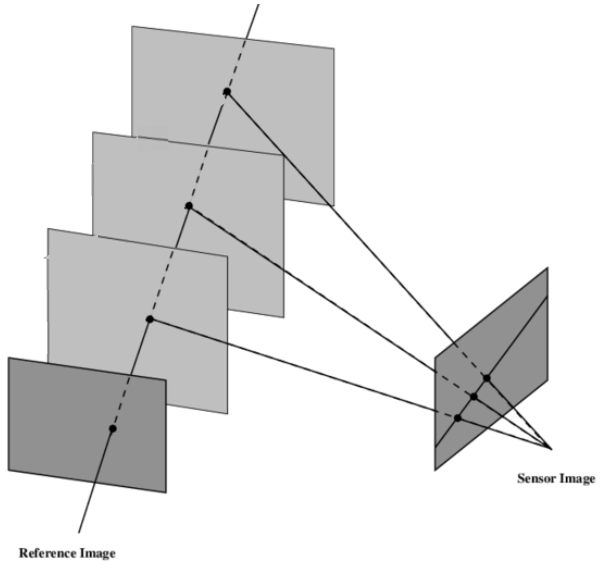
Flow Matching


- Inputs: $\boldsymbol{I}_1$ $\boldsymbol{I}_2$
- Feature extraction: $\boldsymbol{F}_1, \boldsymbol{F}_2 \in \mathbb{R}^{H \times W \times D}$
- Global correlation: $\boldsymbol{C} = \frac{\boldsymbol{F}_1 \boldsymbol{F}_2^T}{\sqrt{D}} \in \mathbb{R}^{H \times W \times H \times W}$
- Softmax normalization: $\boldsymbol{M} = \mathrm{softmax}(\boldsymbol{C}) \in \mathbb{R}^{H \times W \times H \times W}$
- Correspondence: $\hat{\boldsymbol{G}} = \boldsymbol{M}\boldsymbol{G} \in \mathbb{R}^{H \times W \times 2}$ $\quad$ $\boldsymbol{G} \in \mathbb{R}^{H \times W \times 2}$
- Optical flow: $\boldsymbol{V} = \hat{\boldsymbol{G}} - \boldsymbol{G} \in \mathbb{R}^{H \times W \times 2}$

Stereo Matching

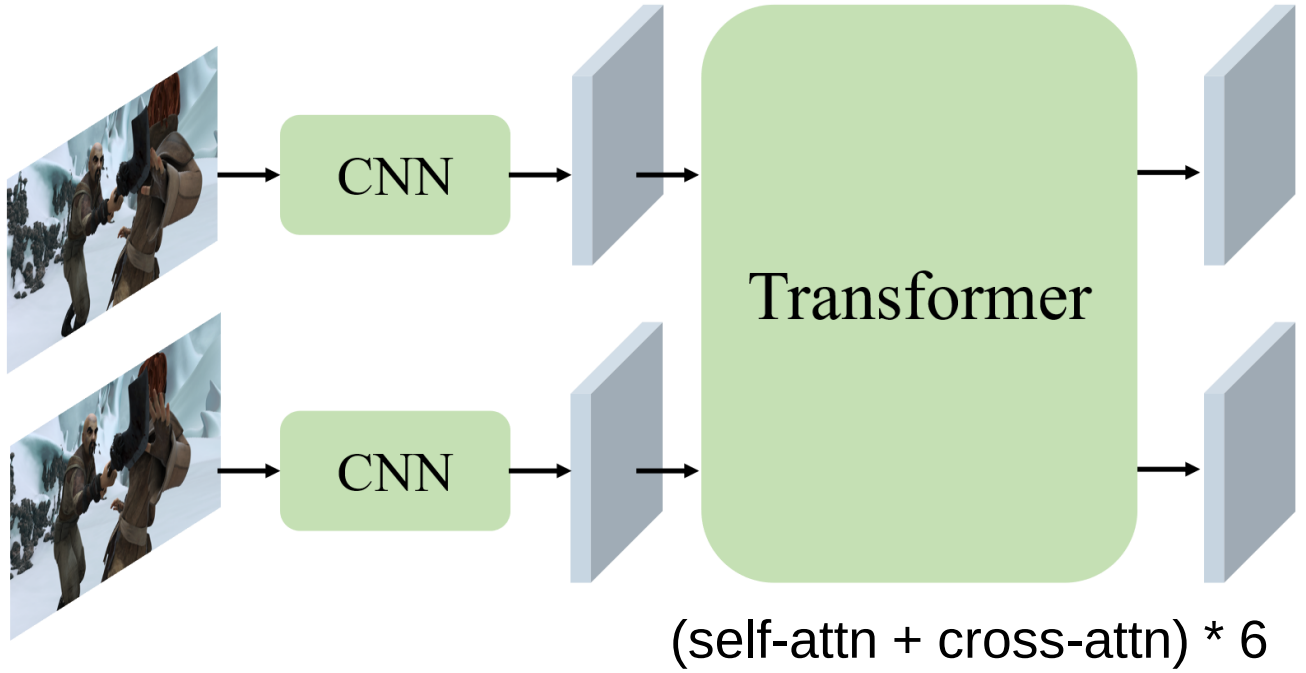
- Inputs: $\boldsymbol{I}_1$ $\boldsymbol{I}_2$
- Feature extraction: $\boldsymbol{F}_1, \boldsymbol{F}_2 \in \mathbb{R}^{H \times W \times D}$
- Horizontal global correlation: $\boldsymbol{C}_{\text{disp}} \in \mathbb{R}^{H \times W \times W}$
- Softmax normalization: $\boldsymbol{M}_{\text{disp}} = \text{softmax}(\boldsymbol{C}_{\text{disp}}) \in \mathbb{R}^{H \times W \times W}$
- Correspondence: $\hat{\boldsymbol{G}}_{\text{1D}} = \boldsymbol{M}_{\text{disp}} \boldsymbol{P} \in \mathbb{R}^{H \times W}$ $\boldsymbol{P} = [0, 1, 2, \cdots, W-1] \in \mathbb{R}^{W}$
- Disparity (positive): $\boldsymbol{V}_{\text{disp}} = \boldsymbol{G}_{\text{1D}} - \hat{\boldsymbol{G}}_{\text{1D}} \in \mathbb{R}^{H \times W}$ $\boldsymbol{G}_{\text{1D}} \in \mathbb{R}^{H \times W}$

Depth Matching



- Inputs: $\boldsymbol{I}_1$ $\boldsymbol{I}_2$
- Feature extraction: $\boldsymbol{F}_1, \boldsymbol{F}_2 \in \mathbb{R}^{H \times W \times D}$
- Discretize depth range $[d_{\min}, d_{\max}]$: $[d_1, d_2, \cdots, d_N]$
- Warping: $\mathcal{H}(\hat{\boldsymbol{G}}_{2\mathrm{D}}) = \boldsymbol{K}_2 \boldsymbol{E}_2 \boldsymbol{E}_1^{-1} d_i \boldsymbol{K}_1^{-1} \mathcal{H}(\boldsymbol{G}_{2\mathrm{D}}) \in \mathbb{R}^{H \times W \times 3}$ $\boldsymbol{F}_2^i \in \mathbb{R}^{H \times W \times D}$
- Correlation: $\boldsymbol{C}^i = \frac{\boldsymbol{F}_1 \cdot \boldsymbol{F}_2^i}{\sqrt{D}} \in \mathbb{R}^{H \times W}$ $\boldsymbol{C}_{\mathrm{depth}} = [\boldsymbol{C}^1, \boldsymbol{C}^2, \cdots, \boldsymbol{C}^N] \in \mathbb{R}^{H \times W \times N}$
- Softmax normalization: $\boldsymbol{M}_{\mathrm{depth}} = \mathrm{softmax}(\boldsymbol{C}_{\mathrm{depth}}) \in \mathbb{R}^{H \times W \times N}$
- Depth: $\boldsymbol{V}_{\mathrm{depth}} = \boldsymbol{M}_{\mathrm{depth}} \boldsymbol{G}_{\mathrm{depth}} \in \mathbb{R}^{H \times W}$ $\boldsymbol{G}_{\mathrm{depth}} = [d_1, d_2, \cdots, d_N] \in \mathbb{R}^N$

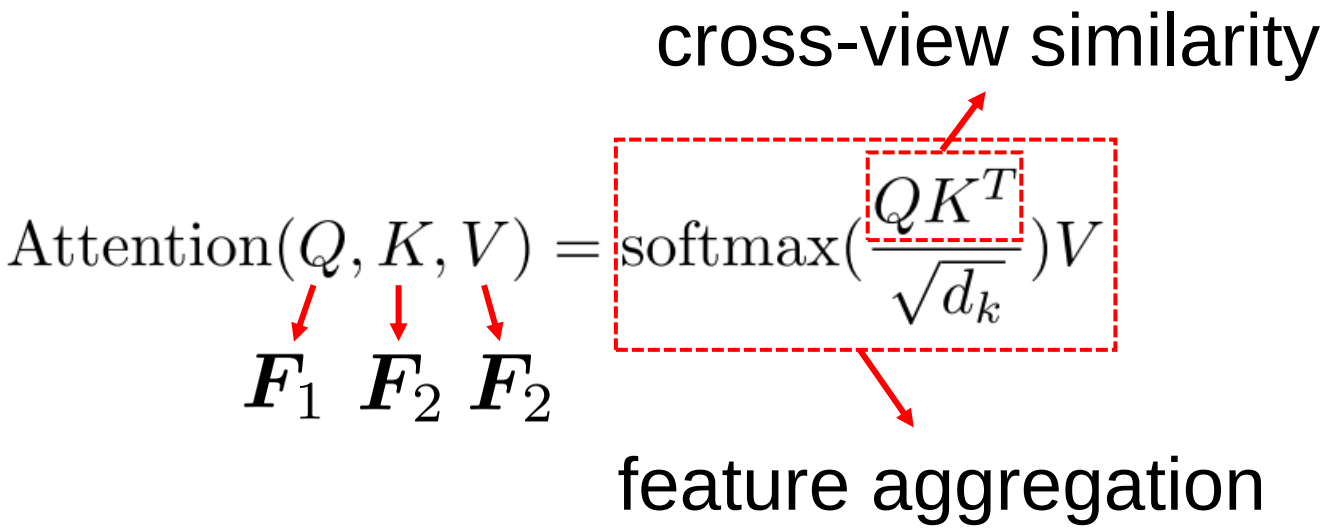
Feature Extraction



- Key: model cross-view interactions with cross-attention
- Efficient implementation: shifted local window (Swin) attention

Liu et al. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. ICCV 2021

Why Cross-Attention?



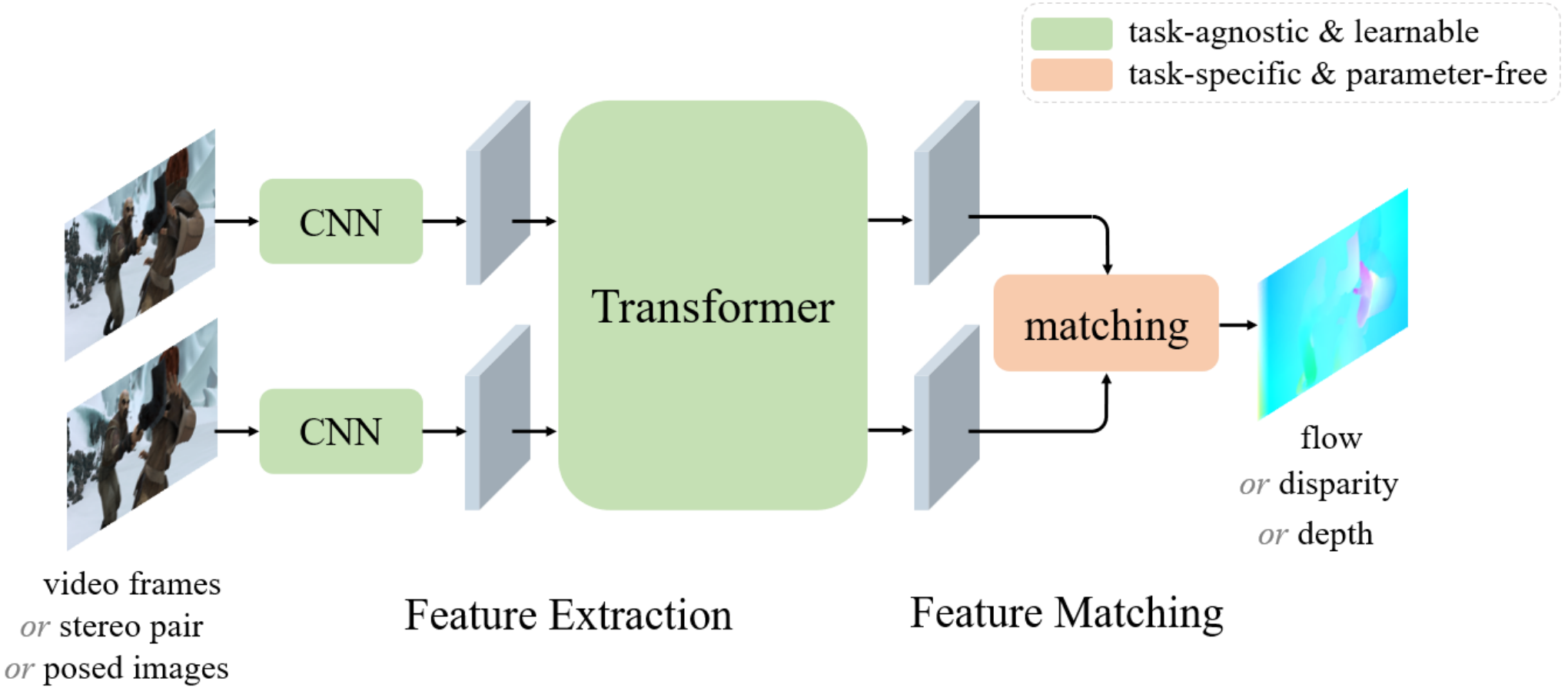
- Feature aggregation via cross-view similarity
- Similar features will be enhanced! (matching becomes easier😊)

Ablation: Transformer Components

setup	Things (val)	Sintel (train)		Param (M)
	clean	clean	final	
full	**6.67**	**2.28**	**3.44**	4.2
w/o cross attn.	10.84	4.48	6.32	3.8
w/o position	8.38	2.85	4.28	4.2
w/o FFN	8.71	3.10	4.43	1.8
w/o self attn.	7.04	2.49	3.69	3.8

Cross-attention contributes most

Architecture So Far



Experimental Comparison

Method	#blocks	Things (val, clean)				Param (M)
		EPE	s_{0-10}	s_{10-40}	s_{40+}	
cost volume + conv	0	18.83	3.42	6.49	49.65	1.8
	4	10.99	1.70	3.41	29.78	4.6
	8	9.59	1.44	2.96	26.04	8.0
	12	9.04	1.37	2.84	24.46	11.5
	18	8.67	1.33	2.74	23.43	15.7
Transformer + softmax	0	22.93	8.57	11.13	52.07	1.0
	1	11.45	2.98	4.68	28.35	1.6
	2	8.59	1.80	3.28	21.99	2.1
	4	7.19	1.40	2.62	18.66	3.1
	6	**6.67**	**1.26**	**2.40**	**17.37**	4.2
conv + softmax	6	17.06	5.79	7.74	40.03	5.1

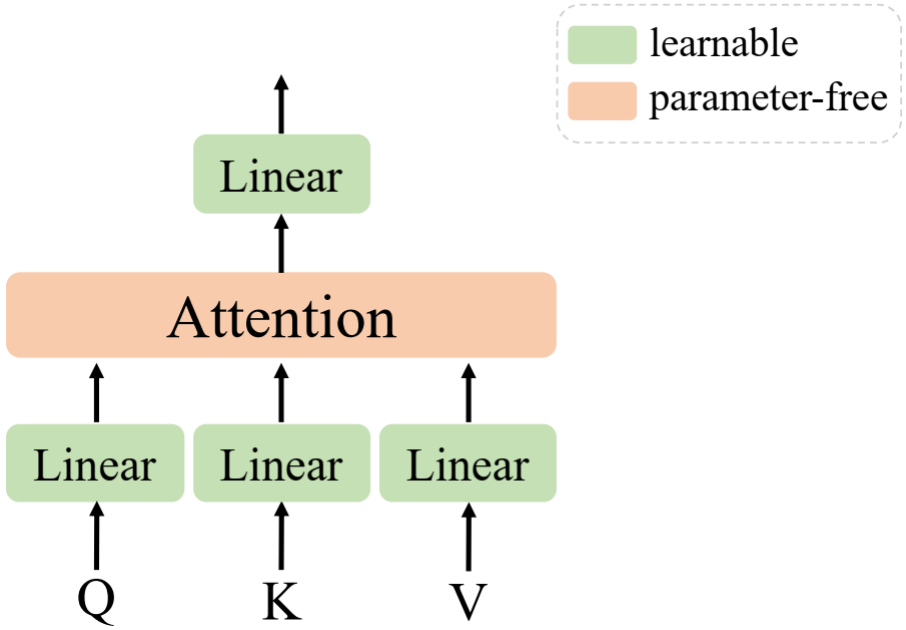
Our method is significantly better, especially for large motion (s40+)

Ablation: Global vs. Local Matching

matching space	Things (val, clean)			
	EPE	s_{0-10}	s_{10-40}	s_{40+}
global	**6.67**	1.26	**2.40**	**17.37**
local 3×3	31.78	1.19	12.40	85.39
local 5×5	26.51	**0.89**	6.67	76.76
local 9×9	19.88	1.01	2.44	61.06

Global matching is significantly better for large motion

1D Cross-Attention for Stereo



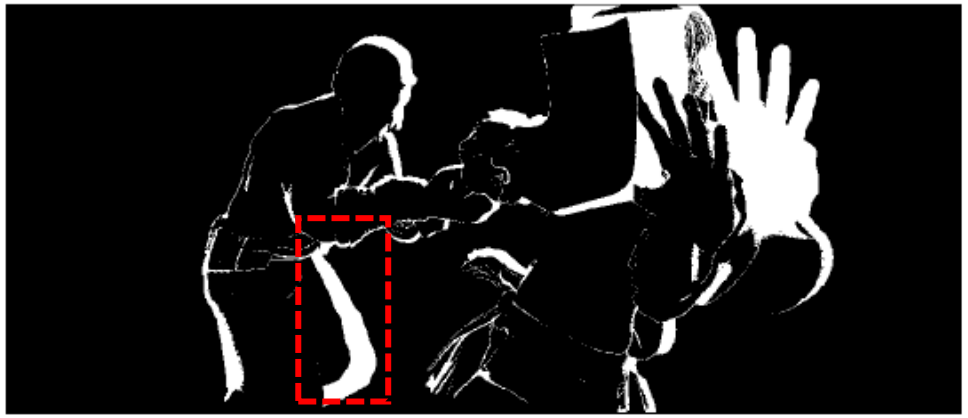
Attention	Things		KITTI		Time (ms)
	EPE	D1	EPE	D1	
2D	1.25	3.97	1.80	13.66	61
1D	**1.22**	**3.70**	**1.61**	**10.53**	**50**

1D cross-attention for stereo is faster and better, while the learnable parameters remain exactly the same for all tasks

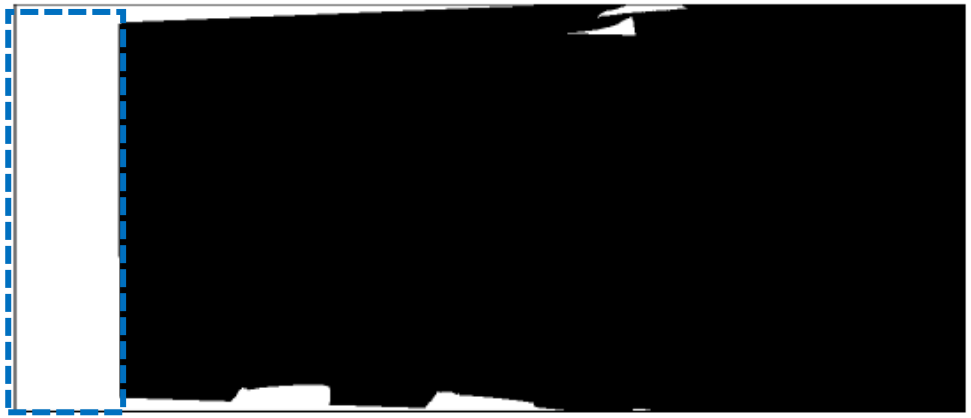
When Matching Fails?

img0

img1



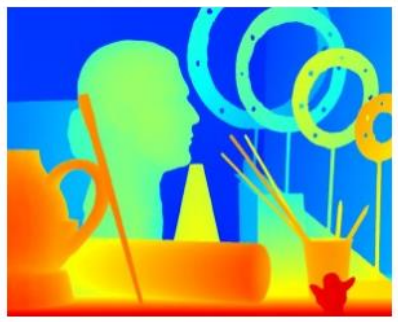
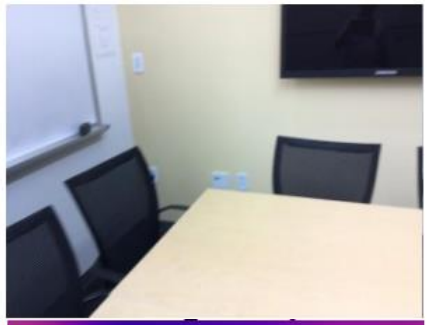
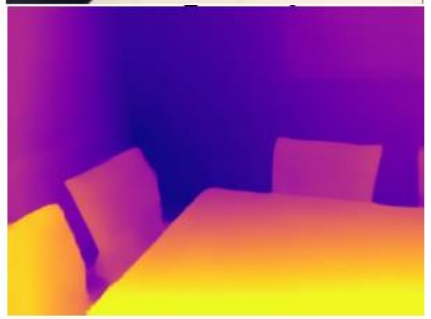
occlusion



out-of-boundary

Propagation



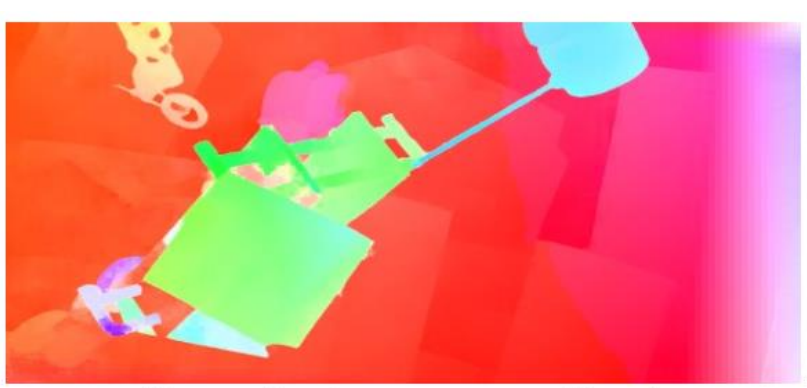
flow disparity depth

- Observation: image and flow/disparity/depth share structure similarity
- Self-attention for propagation:

$$\hat{\boldsymbol{V}} = \text{softmax}\left(\frac{\boldsymbol{F}_1 \boldsymbol{F}_1{}^T}{\sqrt{D}}\right)\boldsymbol{V}$$

Propagation

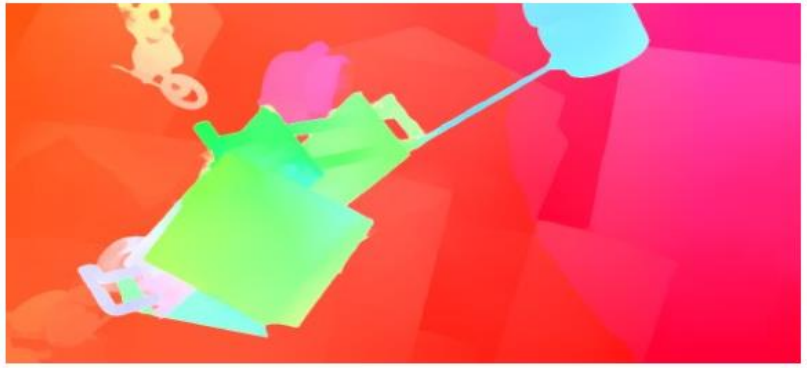
frame 1



flow (w/o propagation)

error (w/o propagation)

frame 2

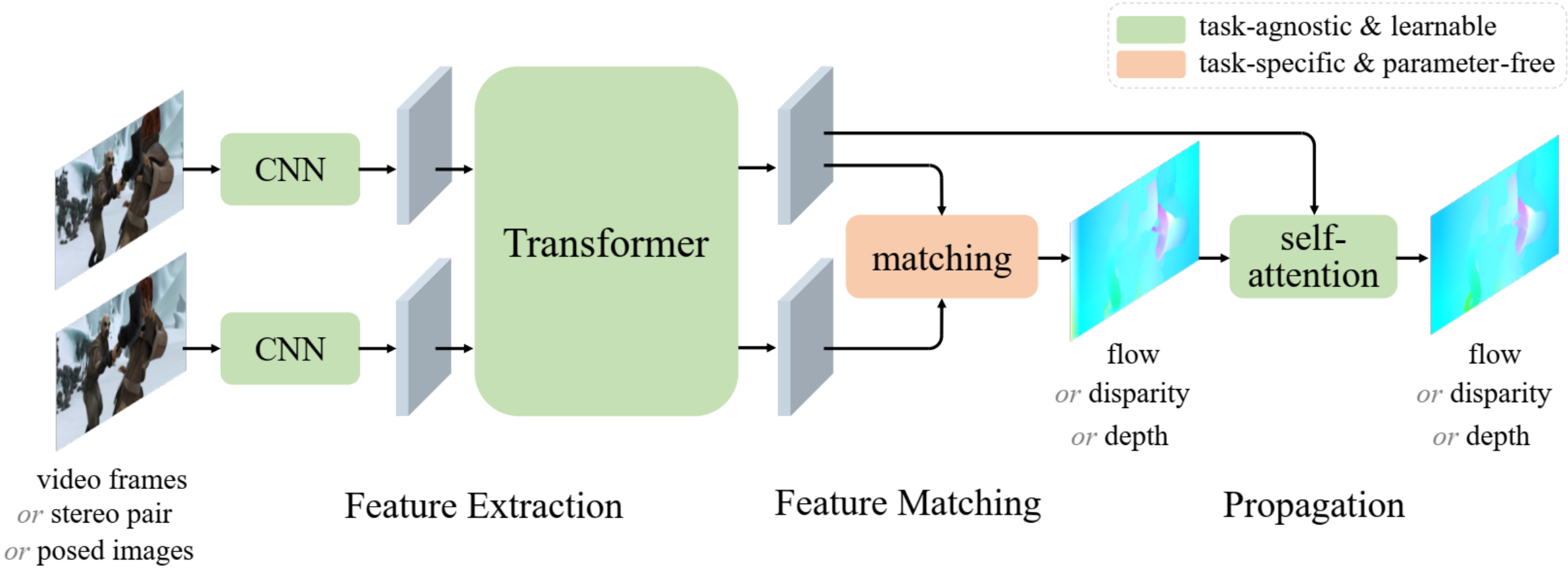


flow (w/ propagation)

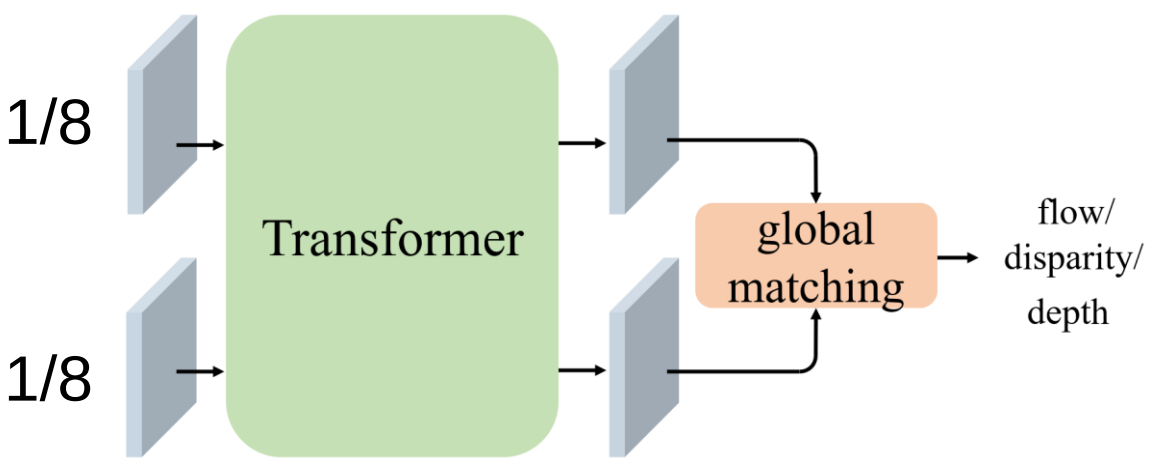
error (w/ propagation)

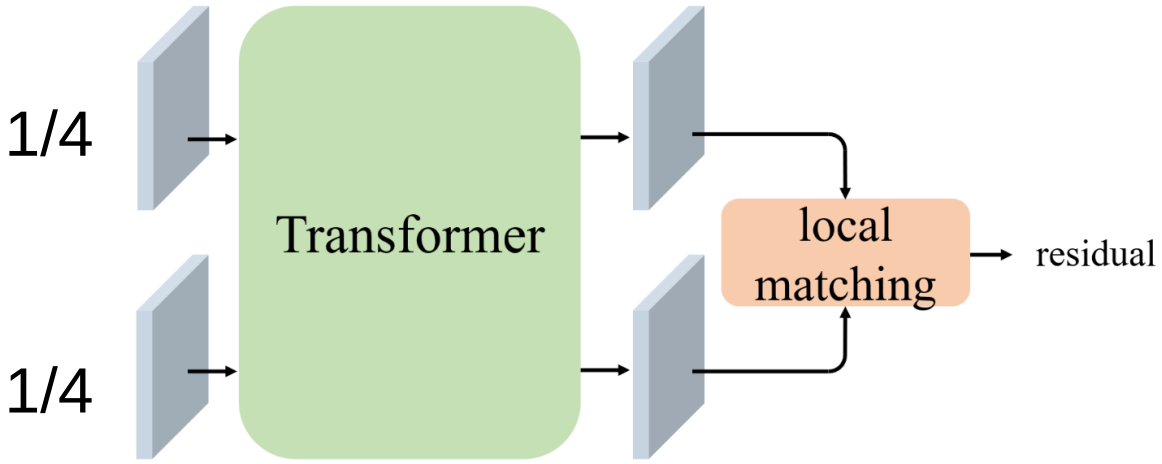
Propagation greatly improves occluded and out-of-boundary pixels

Unified Model



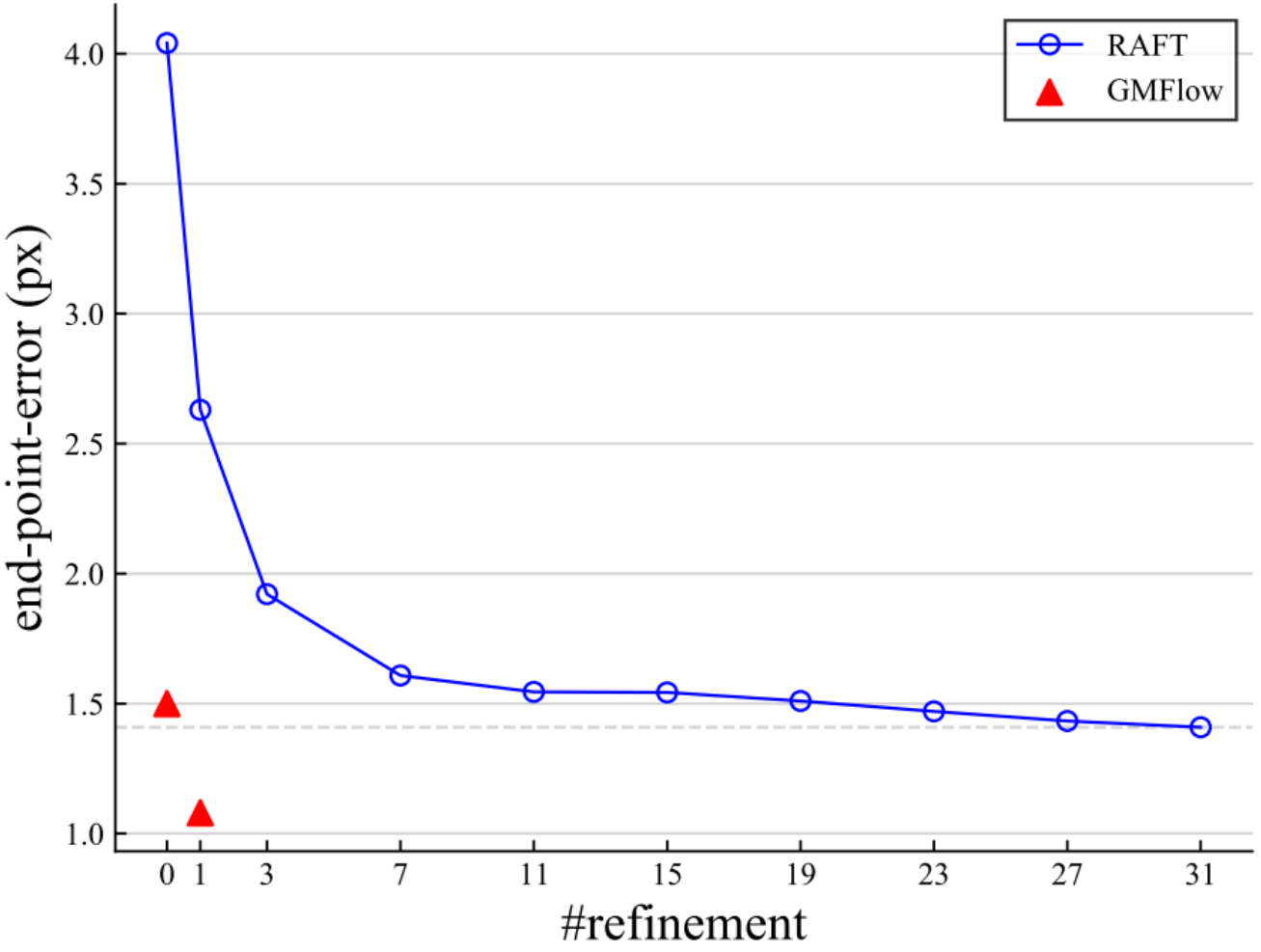
Hierarchical Matching





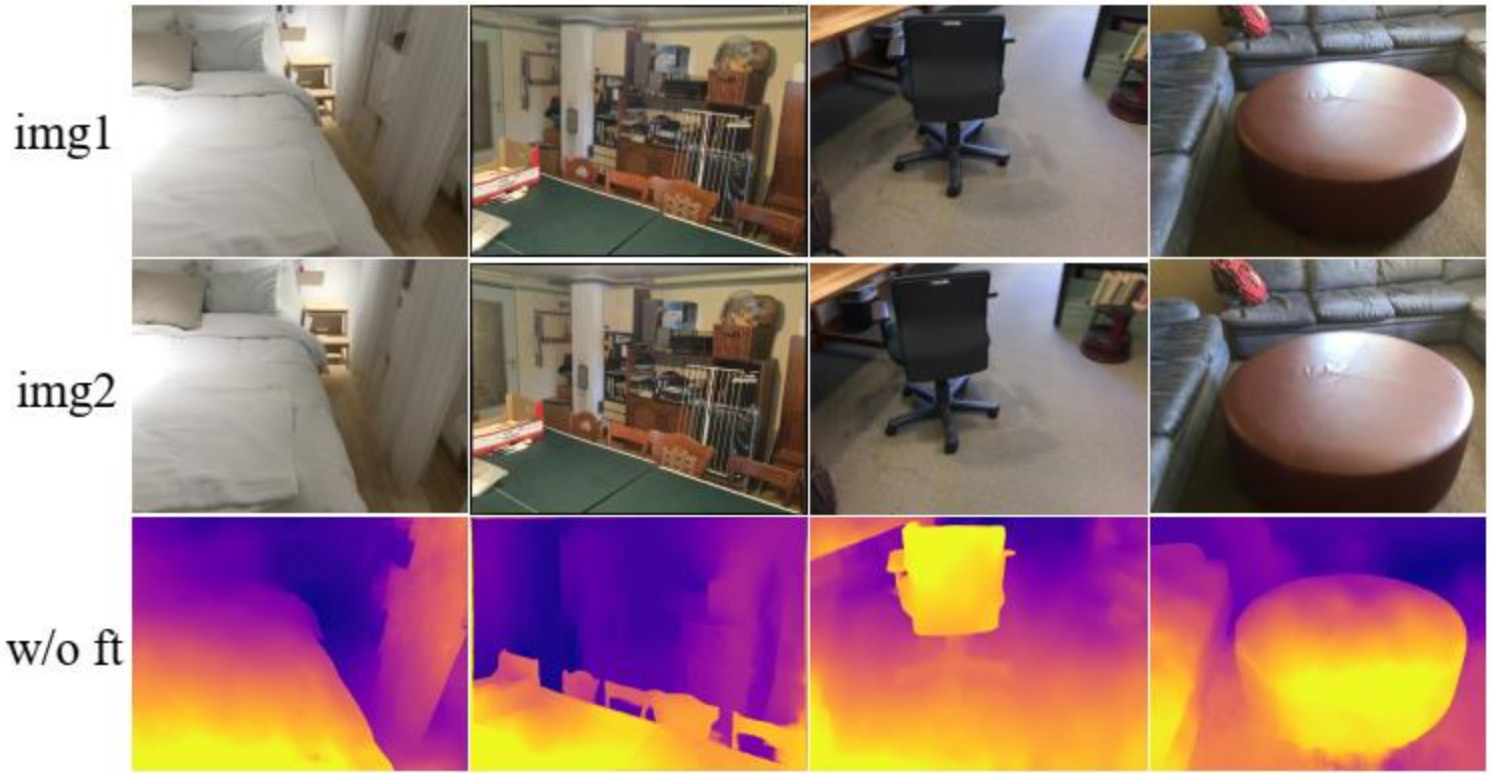
- An optional hierarchical matching refinement at 1/4 feature resolution
- All the learned parameters still remain exactly the same for all tasks

Comparison with RAFT



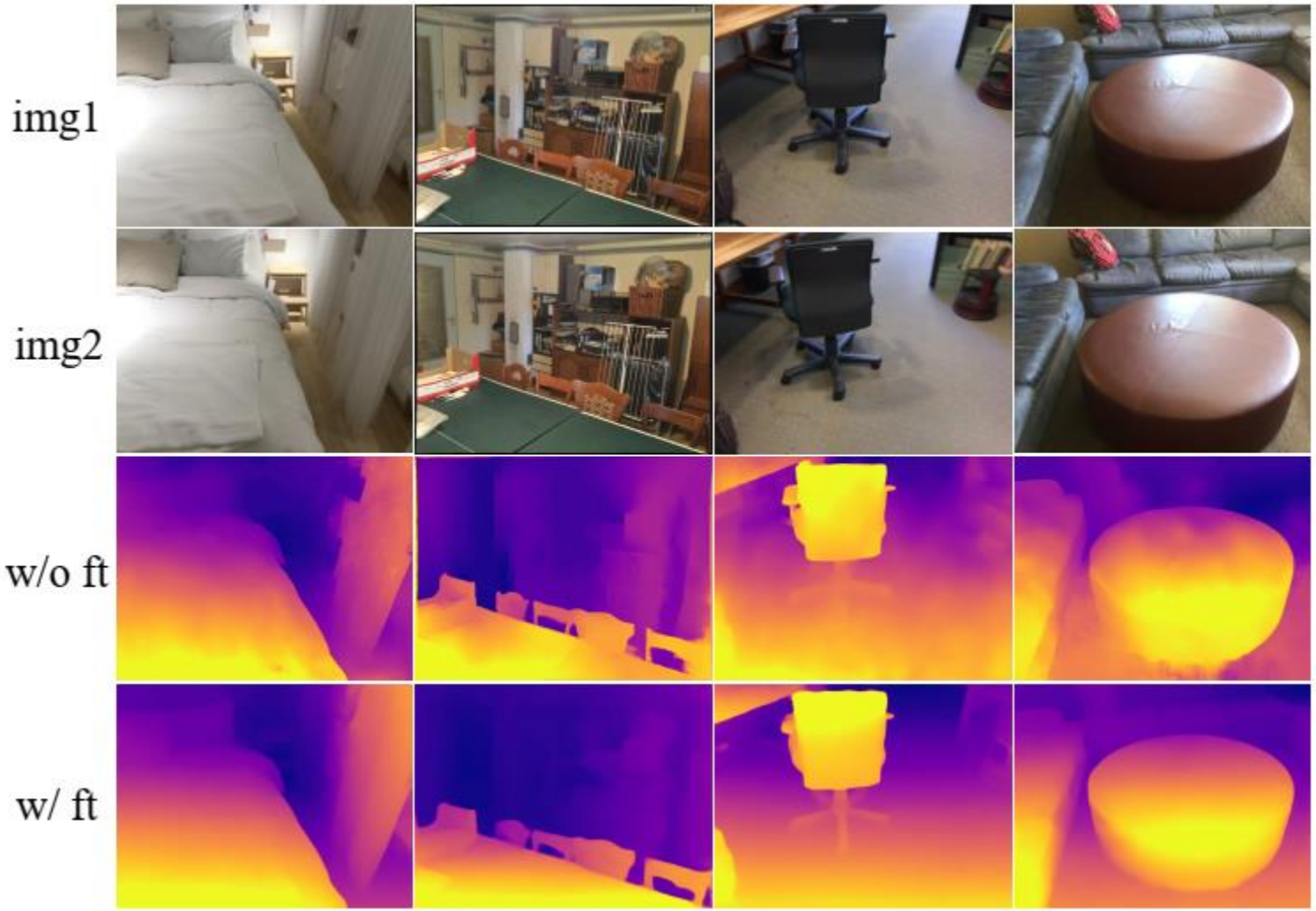
Teed and Deng. RAFT: Recurrent All Pairs Field Transforms for Optical Flow. ECCV 2020

Cross-Task Transfer



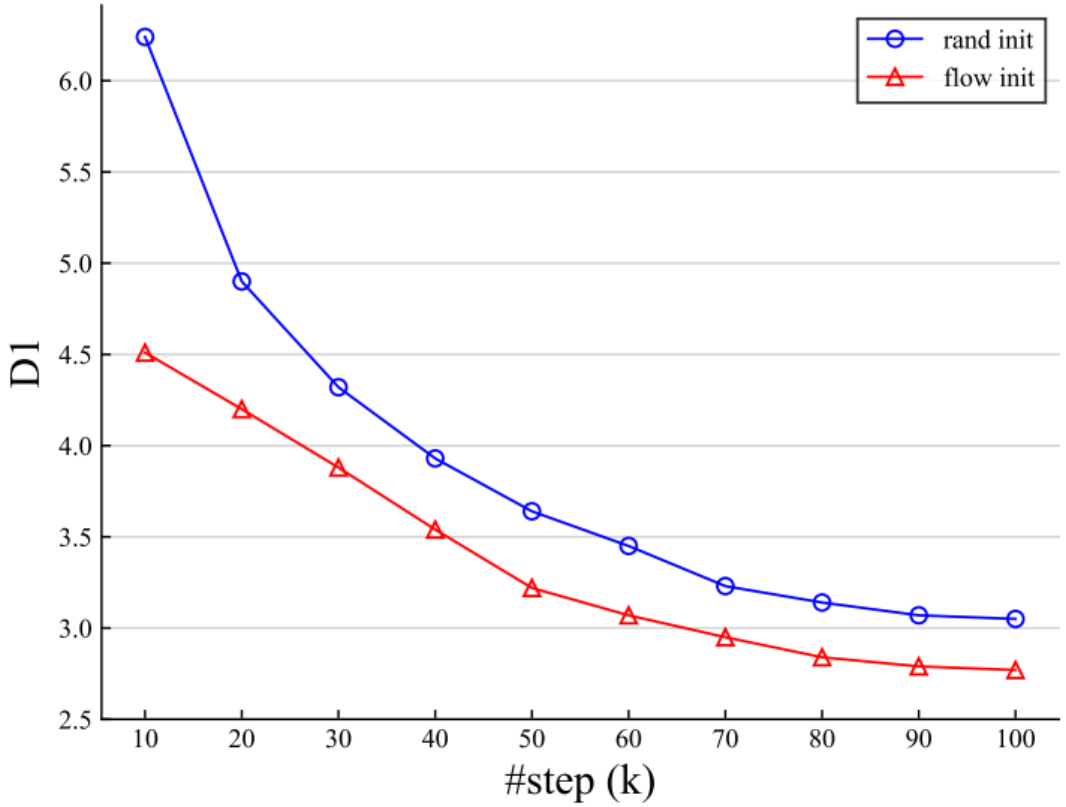
Flow to Depth Transfer

Cross-Task Transfer

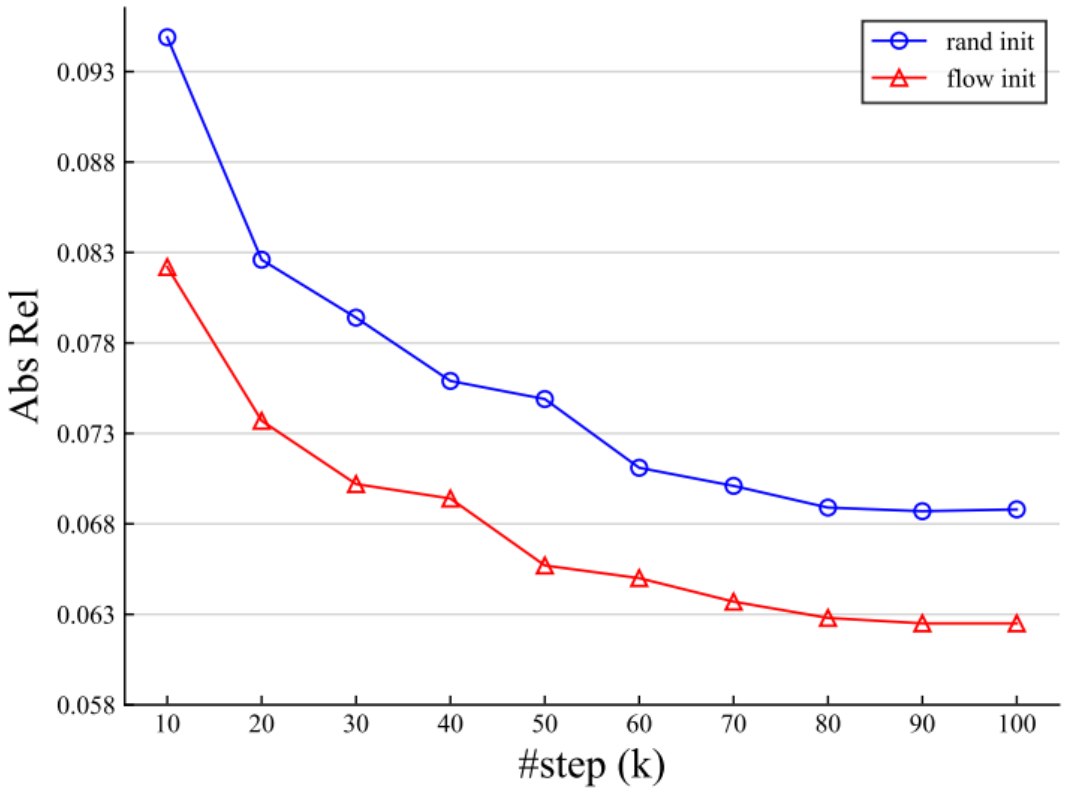


Flow to Depth Transfer

Cross-Task Transfer



(a) **Flow to stereo transfer: error curves of disparity prediction error *vs*. numbers of training steps.**

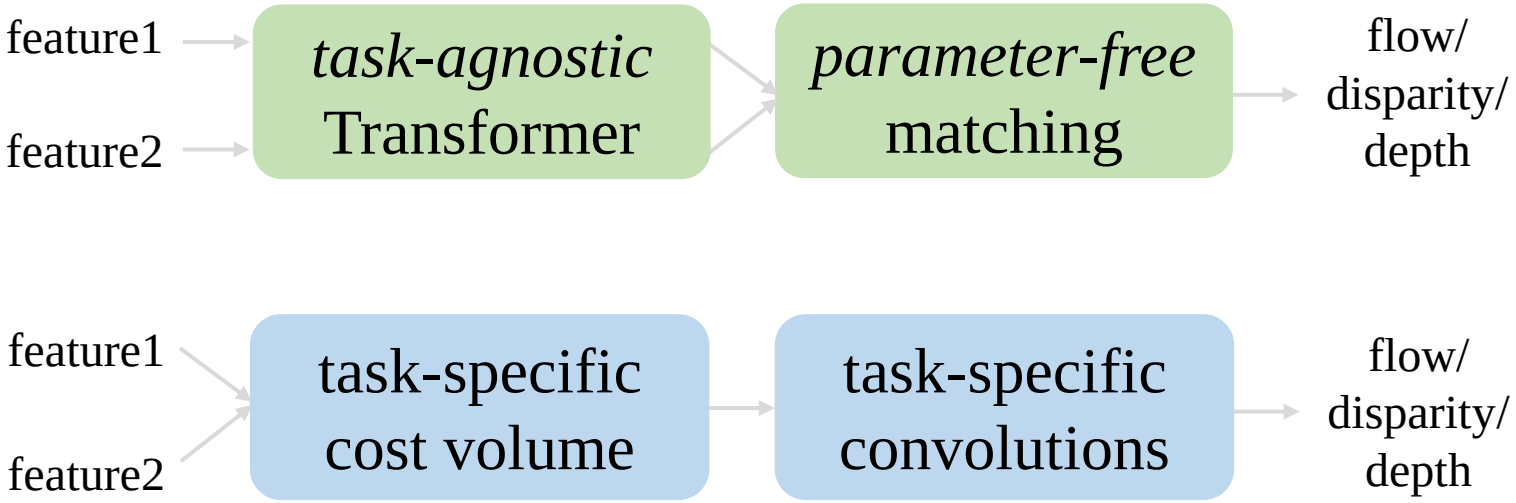


(b) **Flow to depth transfer: error curves of depth prediction error *vs*. numbers of training steps.**

Faster training speed & better performance

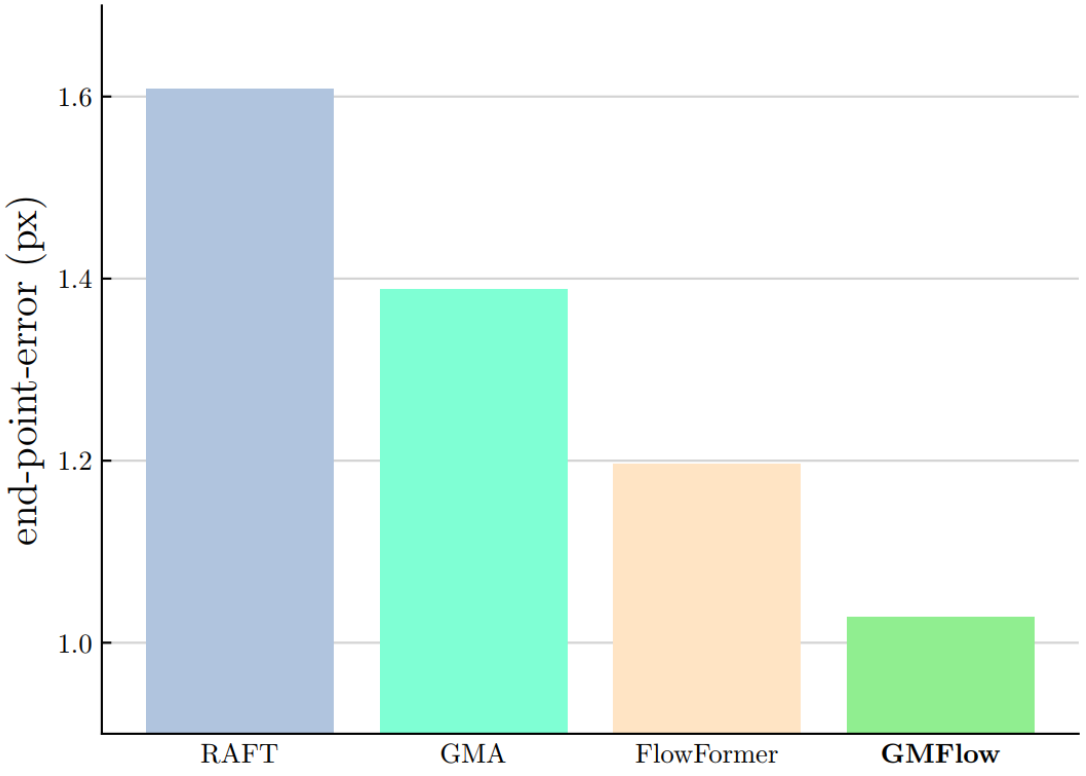
System-level Comparisons

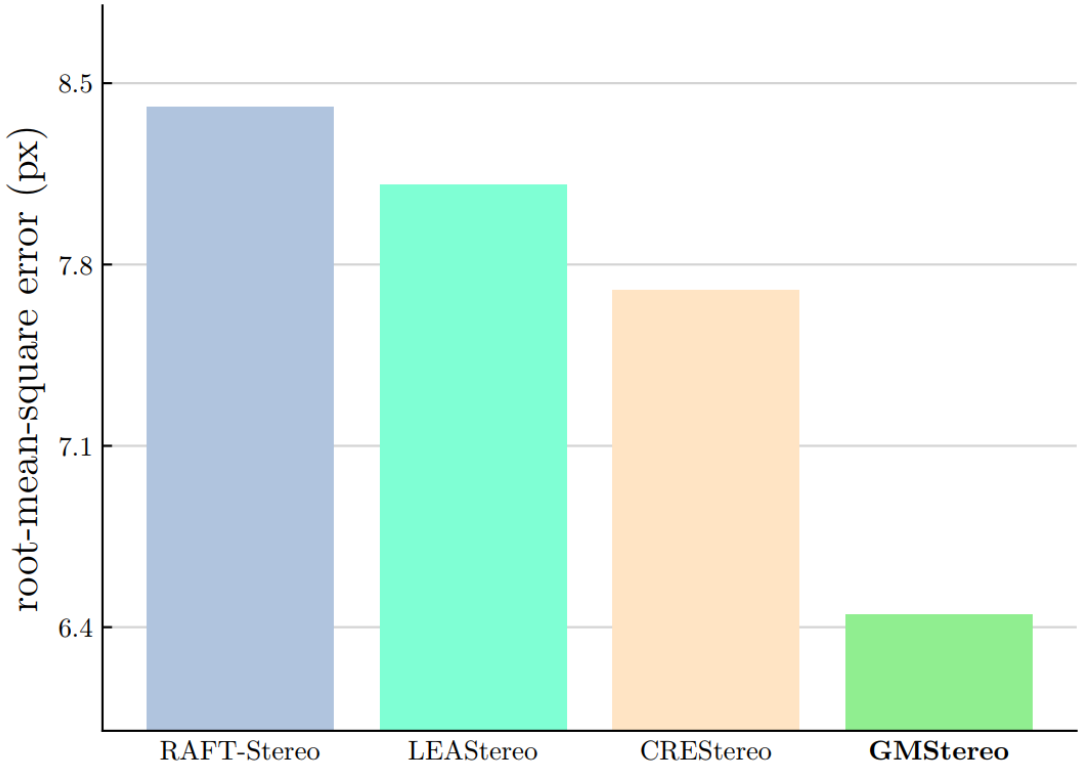
- Use a few additional local refinements (flow: 6, stereo: 3, depth: 1)



- Name our models for flow, stereo and depth as GMFlow, GMStereo and GMDepth (Global Matching)

Benchmark Comparison: Flow & Stereo



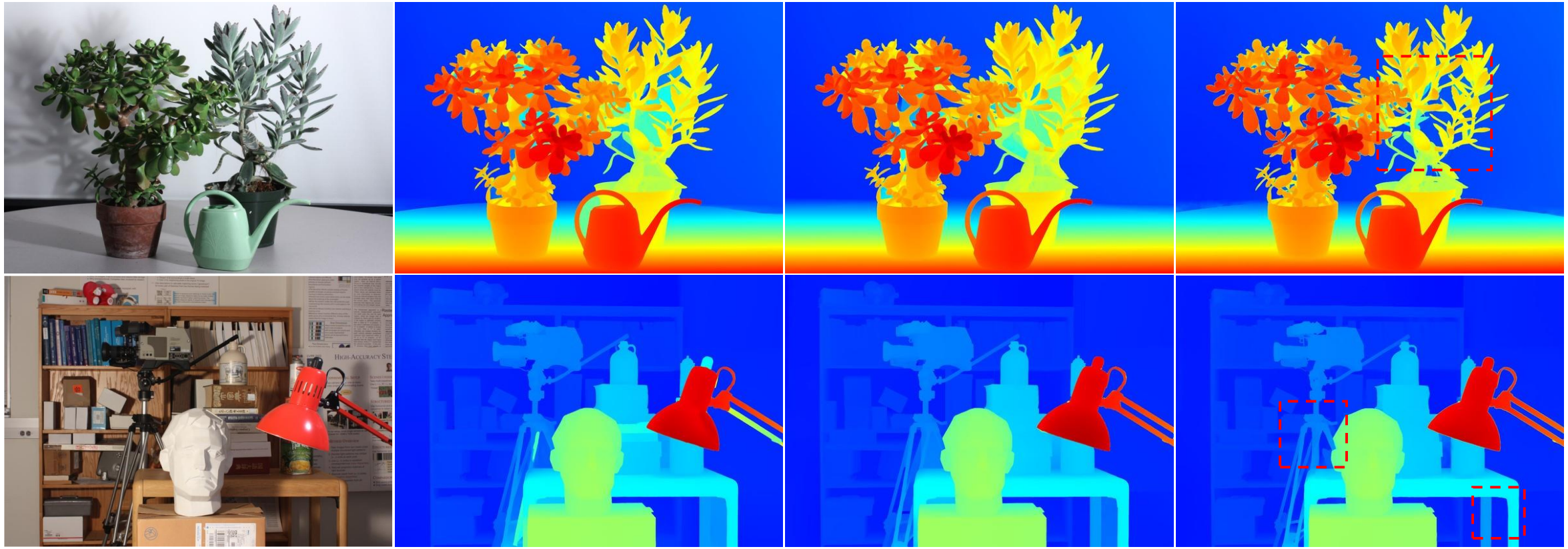


1st places on Sintel (clean) and Middlebury Stereo (RMS metric)

Visual Comparison: Flow

Our GMFlow better captures fast moving small object than RAFT

Visual Comparison: Stereo



Left image | RAFT-Stereo | CREStereo | GMStereo

Our GMStereo produces sharper object structures

Benchmark Comparison: Depth

Model	Abs Rel	Sq Rel	RMSE	RMSE log	Time (s)
DeMoN [37]	0.231	0.520	0.761	0.289	0.69
BA-Net [20]	0.161	0.092	0.346	0.214	0.38
DeepV2D [64]	**0.057**	**0.010**	**0.168**	**0.080**	0.69
GMDepth	0.059	0.019	0.179	0.082	0.04

ScanNet

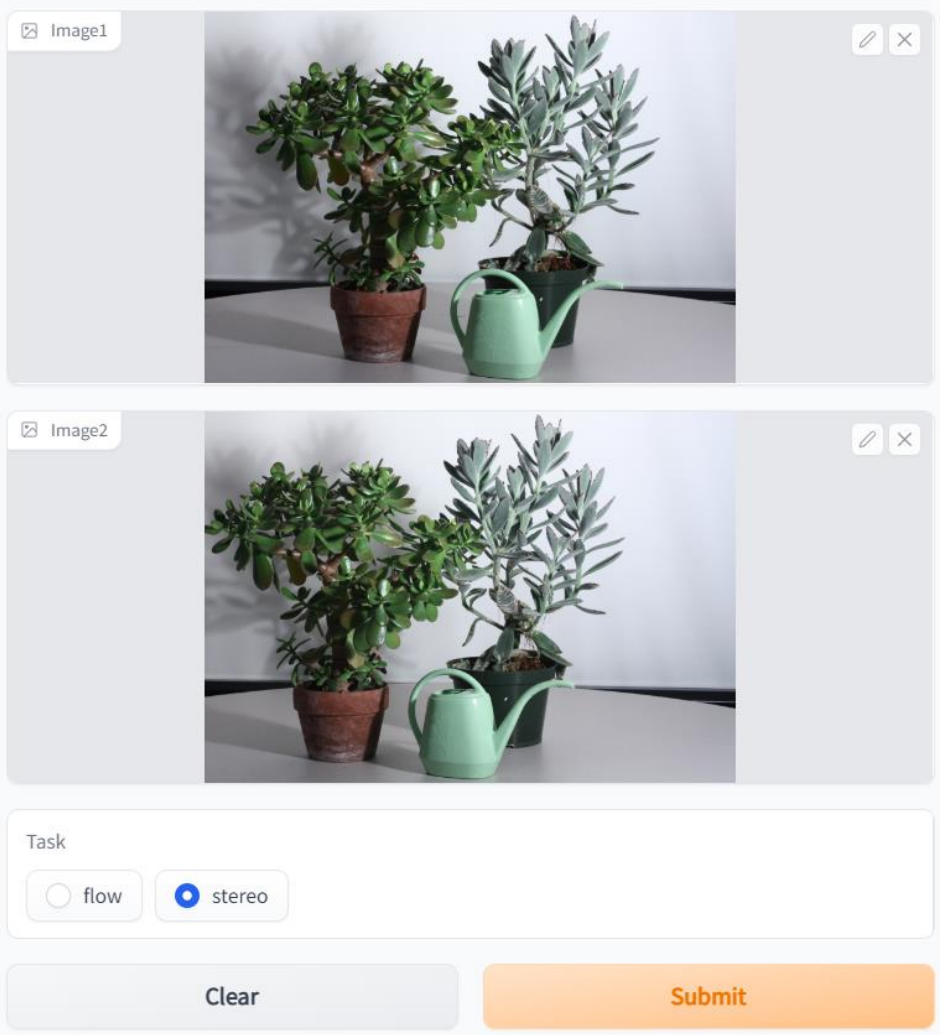
Dataset	Model	Abs Rel	Sq Rel	RMSE	RMSE log
RGBD-SLAM	DeMoN [37]	0.157	0.524	1.780	0.202
	DeepMVS [104]	0.294	0.430	0.868	0.351
	DPSNet [63]	0.154	0.215	0.723	0.226
	IIB [29]	**0.095**	-	**0.550**	-
	GMDepth	0.101	**0.177**	0.556	**0.167**
SUN3D	DeMoN [37]	0.214	1.120	2.421	0.206
	DeepMVS [104]	0.282	0.435	0.944	0.363
	DPSNet [63]	0.147	0.107	0.427	0.191
	IIB [29]	**0.099**	-	**0.293**	-
	GMDepth	0.112	**0.068**	0.336	**0.146**
Scenes11	DeMoN [37]	0.556	3.402	2.603	0.391
	DeepMVS [104]	0.210	0.373	0.891	0.270
	DPSNet [63]	0.056	0.144	0.714	0.140
	IIB [29]	0.056	-	0.523	-
	GMDepth	**0.050**	**0.069**	**0.491**	**0.106**

State-of-the-art or competitive performance while being much faster

More Visual Results

More Visual Results

Online Demo



https://huggingface.co/spaces/haofeixu/unimatch

Code & Model Available

https://github.com/autonomousvision/unimatch

Code

Model	Params (M)	Time (ms)	Download
GMFlow-scale1-things	4.7	26	download
GMFlow-scale1-mixdata	4.7	26	download
GMFlow-scale2-things	4.7	66	download
GMFlow-scale2-sintel	4.7	66	download
GMFlow-scale2-mixdata	4.7	66	download
GMFlow-scale2-regrefine6-things	7.4	122	download
GMFlow-scale2-regrefine6-sintelft	7.4	122	download
GMFlow-scale2-regrefine6-kitti	7.4	122	download
GMFlow-scale2-regrefine6-mixdata	7.4	122	download

https://github.com/autonomousvision/unimatch/blob/master/MODEL_ZOO.md

Model Zoo: 20+ models

Conclusion & Discussion

- Cross-view Transformer features + matching → unified model
- Cross-task transfer
- Real-time inference speed?
- Train all three tasks jointly?
- Unsupervised learning?