Meta-Learning for Adaptation of Deep Optical Flow Networks

Chaerin Min , Tae Hyun Kim , and Jongwoo Lim*

Hanyang University, Republic of Korea

Introduction

Keywords: Unsupervised learning, meta-learning, instance-wise adaptation, optical flow, MAML

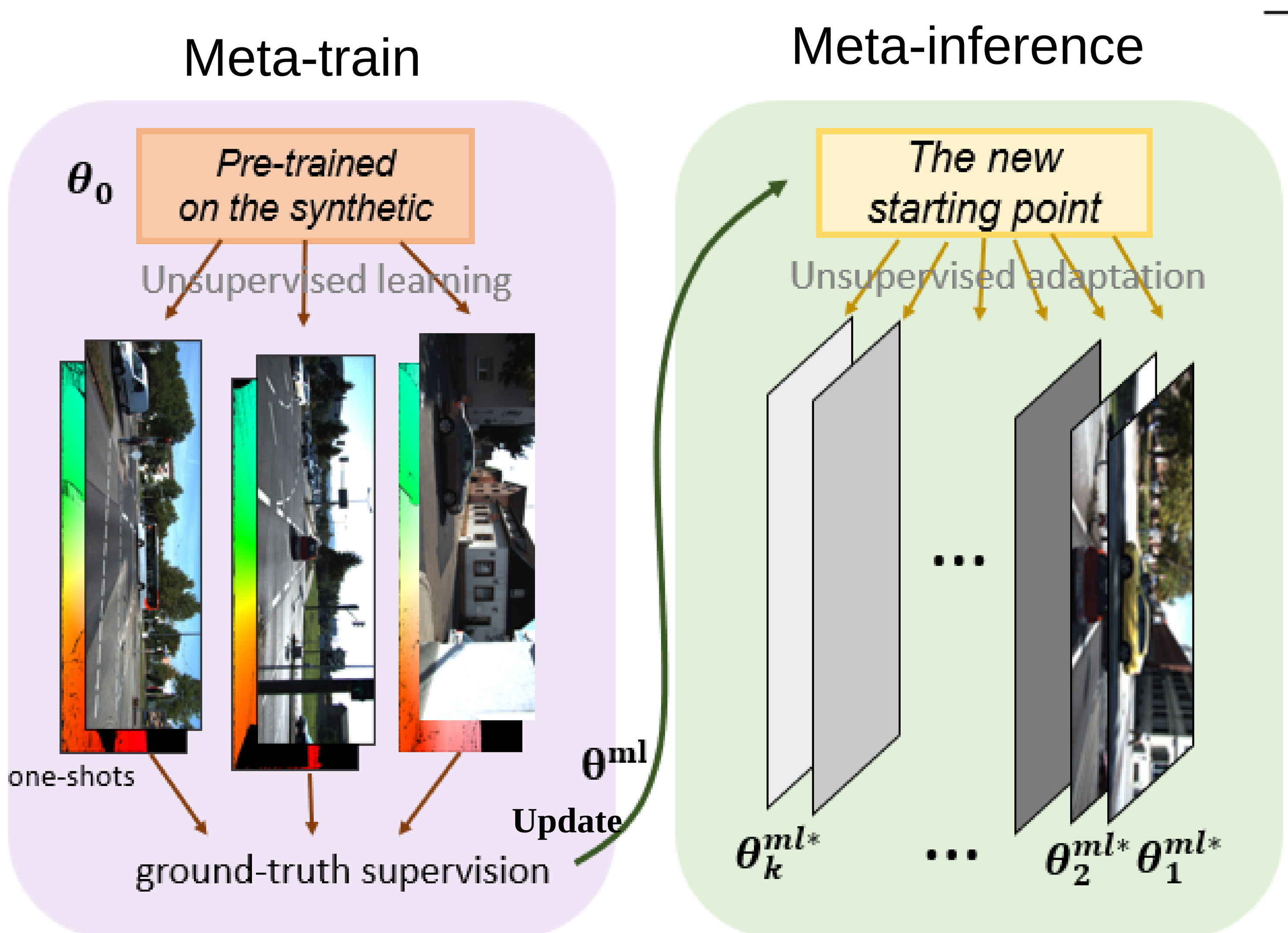
Problem statement

1. Optical flow cross-domain performance
2. The training largely relies on synthetic datasets.
3. The methods become less practical, but this problem has not been explored in-depth.

Assumptions

We have an access to some data in the target domain, prior to the test. The deep flow network can be trained by gradient descent.

Methodology



- Problem set-up

$$V^t = f_\theta(I^t, I^{t+1})$$

- Data term and Regularization term

$$\mathcal{L}_{un}(\theta) = \mathcal{L}_{data}(V^t[\theta]) + \lambda \cdot \mathcal{L}_{reg}(V^t[\theta])$$

Assure photometric consistency

$$\mathcal{L}_{data} = \alpha \cdot (1 - SSIM(I^t(p), I^{t+1}(p + V^t(p)))) + (1-\alpha) \cdot \left\| I^t(p) - I^{t+1}(p + V^t(p)) \right\|_1 ,$$

Encourage spatial coherence

$$\mathcal{L}_{reg} = exp(-\frac{\nabla I^t}{\sigma}) \cdot \|\nabla V^t\|$$

Algorithm 1: Meta-train algorithm.

Require:
$U(T)$: uniform distribution over tasks
θ: pre-trained flow network parameter
N_τ: number of tasks
N: adaptation number, α, β: update steps
Output:
meta-trained flow parameter θ^{ml}

```
while until convergence do
    for i ← 1 to N_τ do
        Sample a task (I^t, I^{t+1}, V^t_gt) ~ U(T)
        θ_i ← θ
        V^t ← f_{θ_i}(I^t, I^{t+1})
        j ← 0
        while j < N do
            L_un(θ_i) =
              L_data(V^t[θ_i]) + λ · L_reg(V^t[θ_i])
            θ_i = θ_i − α∇_{θ_i} L_un(θ_i)
            j ← j + 1
        end
    end
    L_meta(θ) = 1/N_τ Σ_{i=1}^{N_τ} ||f_{θ_i}(I^t, I^{t+1}) − V^t_gt||_1
    θ ⟵ θ − β∇_θ L_meta(θ)
end
Return: θ^ml ← θ
```

Algorithm 2: Meta-inference algorithm.

Input:
I^t, I^{t+1}: two adjacent tst input frames
N: adaptation number, α: update step
Require:
θ^{ml}: meta-trained flow network parameter
N: adaptation number, α: update step
Output: adapted flow result V^t

```
θ^{ml*} ← θ^{ml}
V^t ← f_{θ^{ml*}}(I^t, I^{t+1})
j ← 0
while j < N do
    L_un(θ^{ml*}) =
      L_data(V^t[θ^{ml*}]) + λ · L_reg(V^t[θ^{ml*}])
    θ^{ml*} = θ^{ml*} − α∇_{θ^{ml*}} L_un(θ^{ml*})
    j ← j + 1
end
Return: V^t ← f_{θ^{ml*}}(I^t, I^{t+1})
```

1. Through the meta-train phase, the model learns a more general rule of how to adapt.
2. In the meta-inference time, the model promptly leverages the information that lies in the test inputs.

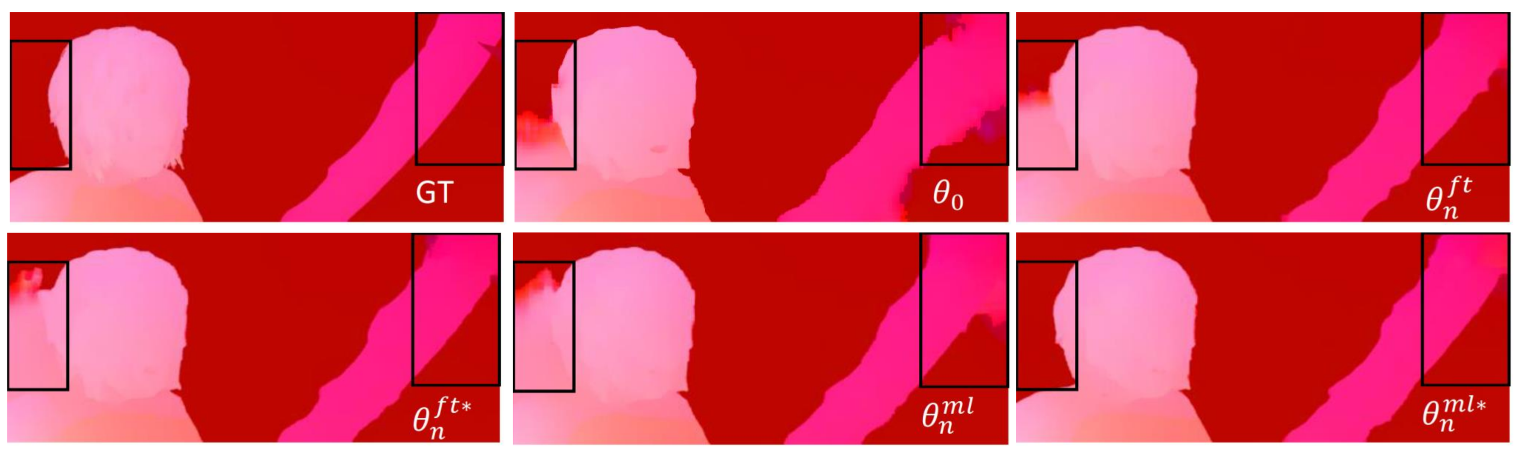
- Meta loss $\mathcal{L}_{meta}$

$$\mathcal{L}_{meta}(\theta) = \frac{1}{N_\tau} \sum_{i=1}^{N_\tau} \|f_{\theta_i}(I^t, I^{t+1}) - V^t_{gt}\|_1$$

Experimental Results

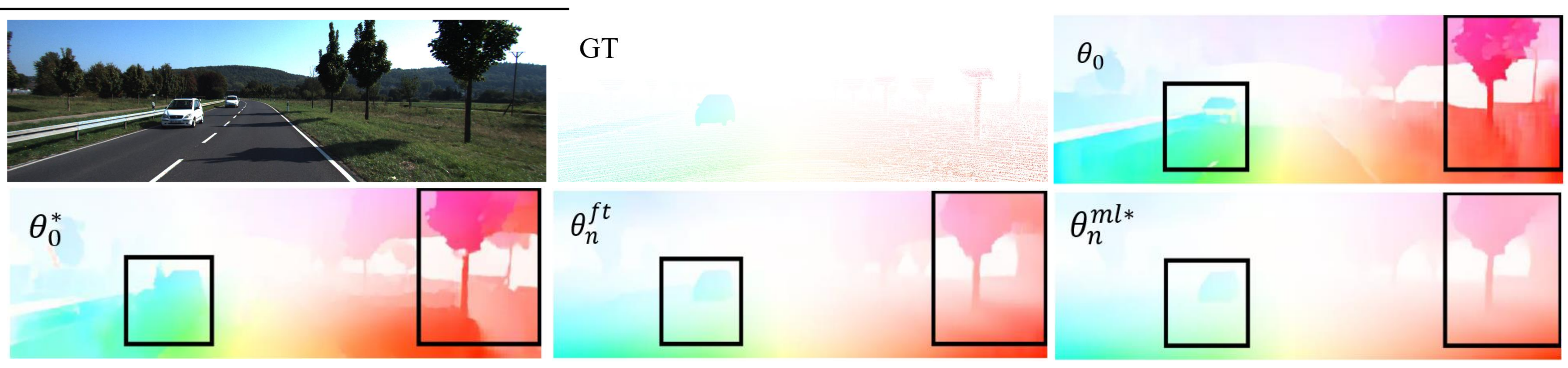
Comparison to the naïve fine-tuning

Test Domain	Pre-train Data	pretrained		fine-tuned		ours	
		θ_0	θ_0^*	θ_n^{ft} →	$\theta_n^{ft^*}$	θ_n^{ml} →	$\theta_n^{ml^*}$
KITTI 2015	C	10.25(0.09)	9.58(0.18)	3.59(0.46)	4.31(0.47)	5.74(1.23)	3.32(0.19)
	C+T	4.65(0.03)	5.17(0.07)	2.73(0.54)	3.40(0.70)	2.81(0.65)	2.69(0.68)
Sintel final	C	4.11(0.02)	3.93(0.02)	3.84(0.02)	3.77(0.03)	3.58(0.30)	3.47(0.28)
	C+T	2.75(0.01)	2.74(0.10)	2.75(0.01)	2.74(0.10)	2.75(0.01)	2.74(0.10)



(C: FlyingChairs,
T: FlyingThings,
*: Unsupervised adaptation)

Through learning to learn via meta-learning, ours is successful in adaptation to test inputs, in contrast to the naïve approaches.

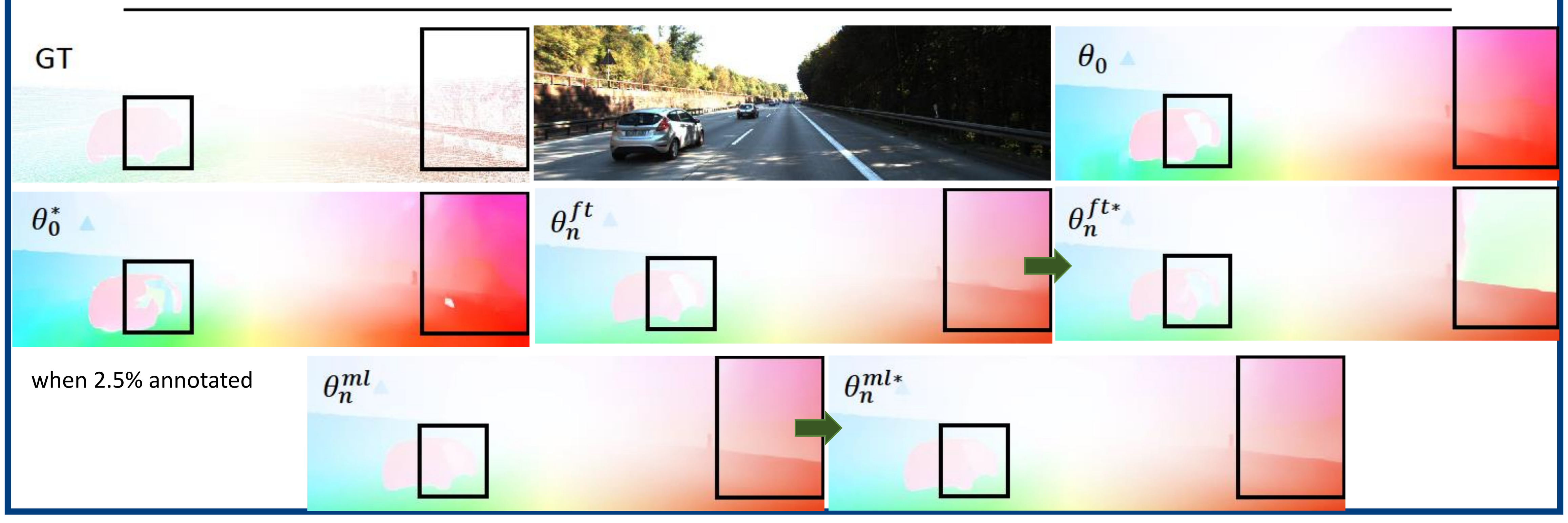


Extra Advantages

Less annotated data in test domain

1. Dense optical flow is difficult to be annotated.
2. Our method allows less ground-truths while not requiring them in test-time.

Size of S_m	Ratio of S_m	pretrained		fine-tuned		ours		Our gain over θ_n^{ft}
		θ_0	θ_0^*	θ_n^{ft}	$\theta_n^{ft^*}$	θ_n^{ml}	$\theta_n^{ml^*}$	
5	2.5%	4.68(0.04)	5.14(0.05)	3.79(0.23)	3.93(0.28)	3.12(0.15)	3.11(0.12)	+0.65(0.29)
10	5%	4.69(0.07)	5.18(0.04)	3.05(0.28)	3.05(0.33)	2.86(0.19)	2.83(0.18)	+0.22(0.11)
20	10%	4.65(0.03)	5.17(0.07)	2.73(0.54)	3.40(0.70)	2.81(0.65)	2.69(0.68)	+0.04(0.03)
100	50%	4.59(0.05)	4.95(0.05)	1.41(0.11)	1.42(0.11)	1.45(0.11)	1.45(0.11)	-0.04(0.06)



Conclusion

Contribution

- In this paper, the first approach to handle different motion distributions is proposed.

Conclusion

- We exploited extra information from each test input.
- We proposed the successful formulation of the unsupervised adaptation of flow networks, using meta-learning.