Meta-Learning for Adaptation of Deep Optical Flow Networks

Chaerin Min[1], Taehyun Kim[1,2], and Jongwoo Lim[1,2*]
[1]Department of Computer Science, Hanyang University, Seoul, Korea
[2]Department of Artificial Intelligence, Hanyang University, Seoul, Korea
{chaerinmin, taehyunkim, jlim}@hanyang.ac.kr

(a) Ground-truth (b) GMA[15] (c) Ours

Figure 1. Test-time domain adaptation via meta-learning helps exploit types of motion and context that are only available from test inputs. The second image shows the result by a pre-trained network, which can be substantially enhanced by our method.

Abstract

In this paper, we propose an instance-wise meta-learning algorithm for optical flow domain adaptation. Typical optical flow algorithms with deep learning suffer from weak cross-domain performance since their trainings largely rely on synthetic datasets in specific domains. This prevents optical flow performance on different scenes from carrying similar performance in practice. Meanwhile, test-time domain adaptation approaches for optical flow estimation are yet to be studied. Our proposed method, with some training data, learns to adapt more sensitively to incoming inputs in the target domain. During the inference process, our method readily exploits the information only accessible in the test-time. Since our algorithm adapts to each input image, we incorporate traditional unsupervised losses for optical flow estimation. Moreover, with the observation that optical flows in a single domain typically contain many similar motions, we show that our method demonstrates high performance with only a small number of training data. This allows to save labeling efforts. Through the experiments on KITTI and MPI-Sintel datasets, our algorithm significantly outperforms the results without adaptation and shows consistently better performance in comparison to typical fine-tuning with the same amount of data. Also qualitatively our proposed method demonstrates more accurate results for the images with high errors in the original networks.

1. Introduction

Many technologies have recently been introduced in meta-learning area [10], which consider learning to learn. Among various meta-learning approaches [6] presented an impressive result in quick adaptation to mutually different tasks.

Optical flow defines the apparent 2D motion field between a pair of images. In other words, it indicates pixel correspondences between neighboring frames in videos. Optical flow estimation is challenging due to fast moving objects and typical visibility problems such as occlusion. Highly accurate optical flow enables successful prediction of pixel correspondences in videos, therefore, it possesses high potential values and can be utilized for a wide range of applications such as motion estimation, object tracking, video super resolution, and motion segmentation.

Unfortunately, there is a lack of research on whether optical flow estimation can demonstrate high generalization abilities for real test datasets apart from the primarily synthetic training datasets [23, 5, 21, 4, 18]. This is because it is challenging to acquire optical flow ground-truth in real scenes. Tab. 1 shows that even for a synthetic dataset, the performance significantly decreases when the domain of the test differs from the training domain. Concerns can be raised that the performance of existing studies may not be fully applicable to the real data used in the field.

One might argue that fine-tuning on the test domain can solve this problem. However, note that it was the lack of optical flow ground-truth that prevented researchers from training a general network for most of the real environ-

Training Data	Method	Chairs (val)	Sintel(train)		KITTI-15(train)	
			Clean	Final	AEPE	F1-all(%)
C+T	PWC-Net[27]	2.30	2.55	3.93	10.35	33.7
	VCN[33]		2.21	3.68	8.36	25.1
	MaskFlowNet[35]		2.25	3.61	-	23.1
	FlowNet2[13]		2.02	3.54	10.08	30.0
	RAFT[28]		1.43	2.71	5.04	17.4
	GMA[15]	0.79	1.30	2.74	4.69	16.6
	RAFT+OCTC[14]		1.31	2.67	4.72	16.3
C	PWC-Net[27]	2.00	3.33	4.59	13.20	41.79
	DDflow[20]	2.97	4.83	4.85	17.26	-
	Uflow[16]	2.55	3.43	4.17	11.27	30.31

Table 1. Cross domain performance of existing methods. The networks are trained on FlyingChairs[5] dataset denoted as C, or FlyingChairs and FlyingThings[21] (C+T). These average end-point errors are from the published papers. The results show that they tend to struggle from inherent discrepancy among datasets.

ments. It follows that it is even not adequate to assume abundant labeled data in a test domain. Fine-tuning with only a small part of a dataset is not likely to yield decent performance on the rest of the unseen data. On the other hand, unsupervised training on the entire test dataset also does not guarantee good performance and can be prohibitively slow. Therefore we need to design a new approach, since typical fine-tuning for optical flow estimation via deep learning is unlikely to be successful for the reasons mentioned above. At this point, we introduce meta-learning into this problem. We propose to enable test-time adaptation with a limited number of labeled data in the test domain and a strictly restricted number of gradient descent iterations. The following summarizes our technical contributions.

- Our approach utilizes unique characteristics of individual test inputs in a new domain. To this end, we employ an unsupervised loss for our adaptation stage. Moreover, existing optical flow methods do not perform comparably on domains other than in-distribution domain. To best of our knowledge, we are the first approach to successfully handle this shortcoming by adopting meta-learning.
- Labeling ground truth optical flow in real scenes is a laborious task. Since our approach helps a network to become more sensitive to inputs in the target domain, our method can readily generalize to the target domain and does not require GT in the test-time.
- Experiments show that our model successfully handles the challenging condition of GT scarcity. Our method significantly outperforms the pretrained networks and demonstrates higher performance than naïve fine-tuning.

2. Related Works

2.1. Supervised optical flow networks

Traditionally, optical flow estimation has been relied on variational approaches [3, 9, 32]. Many recent studies take advantage of deep neural networks to improve performance, as they are good at exploiting spatial information of the scene and inferring the optical flows of occluded areas. FlowNet [5] first succeeded in applying deep learning to optical flow estimation and added correlation information between pixels. PWC-Net [27] effectively predicted both large and small flows using the coarse-to-fine technique. RAFT designed a neural net capable of iterative refinement, increasing model accuracy.

2.2. Unsupervised optical flow losses

Unsupervised learning allows neural networks to be learned even with unlabeled data. For optical flows, unsupervised learning can conventionally be performed using data terms called photometric consistency and prior knowledge such as smoothness [26]. Recently, to improve the smoothness regulation, edge-aware loss [31] or bi-directional Census loss [22] have been proposed. In addition, OAFlow [31] and DDFlow [20] enhanced performance by applying occlusion estimation using photometric and smoothness losses.

2.3. Domain adaptation for optical flow estimation

Domain adaptation by definition aims to apply a high-performance model to a target domain. For instance, in another field of image processing, medical imaging, some studies proposed to maximize performance in test-time through domain adaptation, such as [29]. In the optical flow field, [12] and [11] suggested the student-teacher model

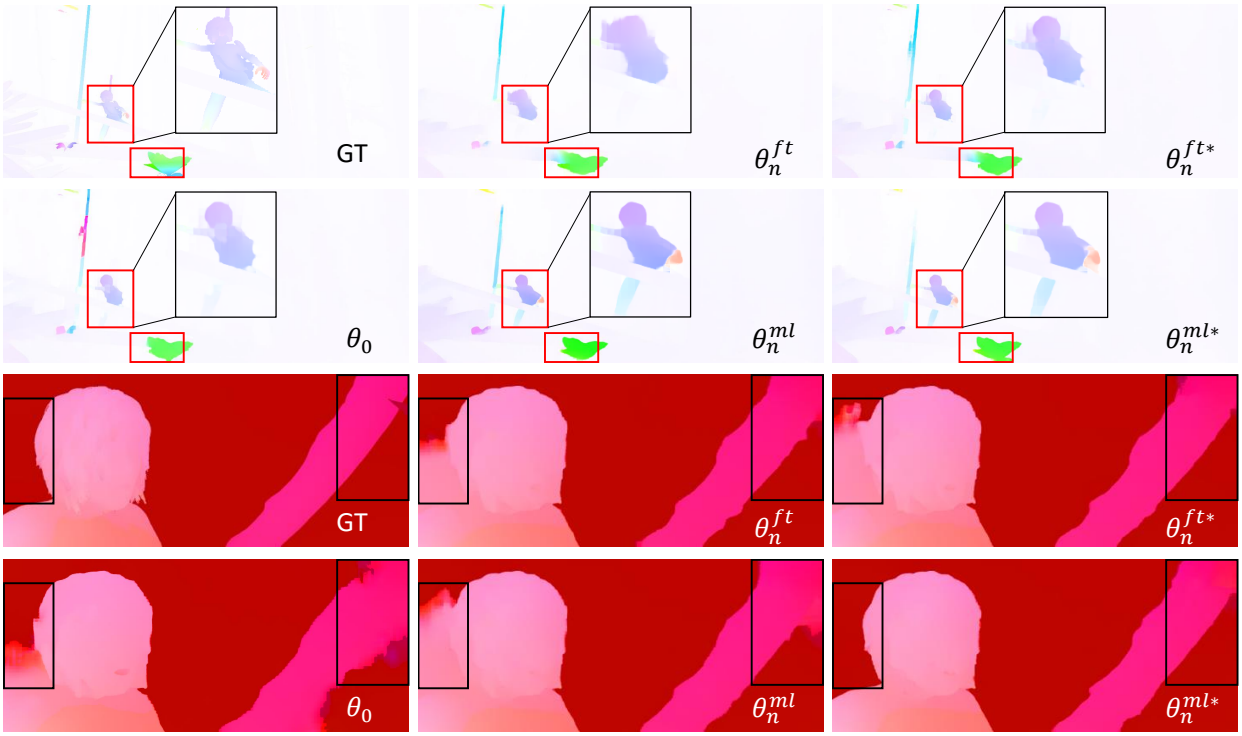


Figure 2. Qualitative Results on MPI Sintel[4] final pass. FlyingChairs[5]+FlyingThings[21] and FlyingChairs serve as the pre-training domains for the first two rows and the bottom two rows, respectively. θ_0, θ_n^{ft}, θ_n^{ml}, θ_n^{ft*}, and θ_n^{ml*} denote pretrained, fine-tuned, meta-trained, adapted from fine-tuned, and proposed, respectively.

for medical image analysis, and [30] attempted sim2world transfer by using the coarse-to-fine strategy. [7] mimics [28]'s optical flow predictor using meta-learning. In contrast to [7], our method does not require any additional parameters or changes to model structure. This implies the flexibility of our method to be applied to any other state-of-the-art networks.

2.4. Meta-learning

Meta-learning is to learn things that were not previously thought to be subject to learning. It is increasingly drawing attention, for it effectively adapts to new domains. Among meta-learning methods, recent optimization-based approaches inspire us, such as [1], [6], and [19]. We assume [6] is the most related approach to our method. MAML[6] is recognized for its capability to adapt to various domains with a limited number of steps. It achieves this by encoding shared prior knowledge across domains. In contrast to [6] which adapts to diverse domains, we focus on boosting the adaptation ability to respective inputs. To best of our knowledge, we are the first to seriously analyze the test-time domain adaptation of optical flow estimation and incorporate a meta-learning paradigm into the problem. In a nutshell, our proposed method is an algorithmic conversion, and it provides major benefits for individual inputs in a domain with insufficient GT. In this work, we present real optical flow estimation as a representative example.

3. Proposed method

Using two consecutive video frames (I^t, I^{t+1}), we can compute the optical flow V^t with conventional flow estimation networks as:

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

where f denotes a conventional flow estimation network with parameters θ, and each element of the flow field V^t is a two-dimensional vector which represents the motion displacement at a pixel location.

However, conventional flow estimation networks have difficulties in dealing with input frames when domain mismatch occurs, and thus require additional burden of fine-tuning. For instance, we may additionally fine-tune the networks pre-trained on FlyingChairs dataset [13] to the KITTI dataset [23] to calculate more accurate flow in the KITTI dataset. Unlike these typical fine-tuning approaches, we propose a new adaptation technique which allows test-time adaptation to a given specific input.

In this section, we first define the problem setting. Then, we provide justification for our motivation and build a background on unsupervised optical flow losses. Finally, we describe the algorithm and originality of our method in details.

3.1. Test-time adaptation of flow networks

Conventional flow estimation networks trained through motion distributions from a specific dataset have difficulties in handling input frames with a different motion distribution. To mitigate this problem, it is required to adapt the parameters of the pre-trained networks to the new test domain. In particular, we aim to adapt the flow networks to the given specific input at the test-phase by utilizing internal motion statistics. However, as ground-truth motion information is not available at the test-stage, we employ the conventional unsupervised losses which allow us to train the networks in an unsupervised manner for the test-time adaptation.

Our unsupervised loss function $\mathcal{L}_{un}$ is composed of data term $\mathcal{L}_{data}$ and regularization term $\mathcal{L}_{reg}$, and it yields,

$$\mathcal{L}_{un}(\theta) = \mathcal{L}_{data}(V^t[\theta]) + \lambda \cdot L_{reg}(V^t[\theta]), \tag{2}$$

where λ is a user parameter to adjust the regularization.

Specifically, our data term measures the data fidelity similar to [25, 8, 16], and the formulation is given by,

$$\begin{aligned}\mathcal{L}_{data} = \alpha \cdot (1 - SSIM(I^t(p), I^{t+1}(p + V^t(p)))) \\ +(1-\alpha) \cdot \left\| I^t(p) - I^{t+1}(p + V^t(p)) \right\|_1,\end{aligned} \tag{3}$$

where p denotes the pixel coordinates. The first and second terms compute the dataset fidelity based on SSIM [34] score and brightness constancy respectively, and α controls the balance between these two terms. Moreover, we adopt the edge-aware regularization [31] to preserve motion boundaries while enforcing smoothness on homogeneous regions, and our regularization term is as follows:

$$\mathcal{L}_{reg} = exp(-\frac{\nabla I^t}{\sigma}) \cdot \|\nabla V^t\|, \tag{4}$$

where ∇ denotes a linear operator to compute pixel-wise derivative and σ controls strength of the edge-awareness.

3.2. Meta-learning for test-time adaptation

By minimizing our unsupervised loss function in (2), we can naively update the pre-trained flow model separately on each test input from a new domain. In addition, we introduce a more fast adaptation technique which can further elevate the network performance and accelerate adaptation speed when there are a small number of annotated ground-truth flows on the new domain.

To be specific, we incorporate a meta-learning approach with our problem for the fast adaptation. In general, conventional meta-learning algorithm requires a large number of train-dataset to enable the network to sensitively respond to diverse task changes during the train-phase, and we additionally need few ground-truth dataset for the test-time adaptation at the test-phase.

In contrast, we meta-train the flow network with only a small number of annotated dataset since optical flows in a single domain typically contain many similar motions (*e.g.* forward motion in KITTI dataset), and we adapt the network parameters to an input at the test-time in an unsupervised manner. We embed our fast adaptation algorithm into the MAML algorithm [6] which is one of the representatives for its simplicity and flexibility. Through our meta-learning method, we provide the network a training of learning new types of motions and context in a specific domain, so adaptation becomes a much easier process than relying solely on the unsupervised loss in (2). Moreover, we achieve this without requiring a large amount of data from the target domain. Then, during the meta-inference stage, we post-process the meta-trained network to a test input without using the ground-truth dataset.

3.2.1 Meta-train for new domains

Similar to MAML [6], our meta-train stage is composed of two update steps as provided in Algorithm 1. During inner update step, we adapt the network parameters using the unsupervised loss in (2), and conduct meta-optimization with few labeled metaset through outer update step. To be specific, in the proposed meta-learning scenario, our task consists of two consecutive video frames (I^t, I^{t+1}) and corresponding ground-truth optical flow V^t_{gt}. At each outer update step, we randomly sample N_τ tasks from uniform distribution, and we adapt the network parameter for each task in an unsupervised fashion. Finally, we meta-optimize the flow parameter by minimizing the meta-objective as:

$$\mathcal{L}_{meta}(\theta) = \frac{1}{N_\tau} \sum_{i=1}^{N_\tau} \| f_{\theta_i}(I^t, I^{t+1}) - V^t_{gt} \|_1, \tag{5}$$

where θ_i denotes the network parameter adapted to a specific task using the unsupervised loss. Notably, we can use any conventional optimizers (*e.g.* SGD, ADAM) to minimize the two loss functions, $\mathcal{L}_{un}$ and $\mathcal{L}_{meta}$. We repeat this procedure until convergence and the proposed meta-train algorithm allows the meta-trained parameter θ^* to be generalized across similar tasks in a specific domain such as KITTI dataset.

3.2.2 Meta-inference for test images

We elaborate our final adaptation algorithm which is dubbed meta-inference during the test-phase in Algorithm 2. First of all, we use the meta-trained θ^{ml} as the starting point of this stage. Then, we perform the identical process as in the inner update step in Algorithm 1, and adapt the parameters by minimizing the unsupervised loss since the ground-truth dataset is not available at the test-time. At the end of N adaptations, we can render the final flow results V^t for a given test input frames I^t and I^{t+1} using the adapted flow parameter θ^{ml*}. We measure the accuracy of

Test Domain	Pre-train Data	pretrained		fine-tuned		ours	
		θ_0	θ_0^*	θ_n^{ft}	θ_n^{ft*}	θ_n^{ml}	θ_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)

Table 2. Quantitative results on KITTI 2015 and Sintel final datasets. We randomly split the test domain into metaset and testset three times and average performances. The standard deviations are enclosed in parenthesises. Asteriods indicate that the model is adapted on S_t by the unsupervised $\mathcal{L}_{un}$ individually on each input τ_i^t. Note that we achieve θ^{ml*} with as little as three gradient descent steps, framed by the inner-loop's 3 steps. On the other hand, we choose the best performance for θ^{ft*}. In case the epe of θ^{ft*} keeps increasing from the very first step, we select the third step for proper comparison to our method. We conduct experiments denoted as $\theta_0^*, \theta_n^{ft}, \theta_n^{ft*}$ ourselves. By doing this, we exhaustively verify our method's advantages over naive fine-tuning and unsupervised learning with the same S_m, S_t, and losses.

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 $i \leftarrow 1$ *to* N_τ **do**
    Sample a task $(I^t, I^{t+1}, V_{gt}^t) \sim U(T)$
    $\theta_i \leftarrow \theta$
    $V^t \leftarrow f_{\theta_i}(I^t, I^{t+1})$
    $j \leftarrow 0$
    while $j < N$ **do**
      $L_{un}(\theta_i) =$
       $L_{data}(V^t[\theta_i]) + \lambda \cdot L_{reg}(V^t[\theta_i])$
      $\theta_i = \theta_i - \alpha \nabla_{\theta_i} L_{un}(\theta_i)$
      $j \leftarrow j + 1$
    end
  end
  $L_{meta}(\theta) = \frac{1}{N_\tau} \sum_{i=1}^{N_\tau} \|f_{\theta_i}(I^t, I^{t+1}) - V_{gt}^t\|_1$
  $\theta \longleftarrow \theta - \beta \nabla_\theta L_{meta}(\theta)$
end
Return: $\theta^{ml} \leftarrow \theta$

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

$\theta^{ml*} \leftarrow \theta^{ml}$
$V^t \leftarrow f_{\theta^{ml*}}(I^t, I^{t+1})$
$j \leftarrow 0$
while $j < N$ **do**
  $L_{un}(\theta^{ml*}) =$
   $L_{data}(V^t[\theta^{ml*}]) + \lambda \cdot L_{reg}(V^t[\theta^{ml*}])$
  $\theta^{ml*} = \theta^{ml*} - \alpha \nabla_{\theta^{ml*}} L_{un}(\theta^{ml*})$
  $j \leftarrow j + 1$
end
Return: $V^t \leftarrow f_{\theta^{ml*}}(I^t, I^{t+1})$

the optical flow results in terms of end-point error (EPE), and we provide the EPE values over various test inputs in our experiments.

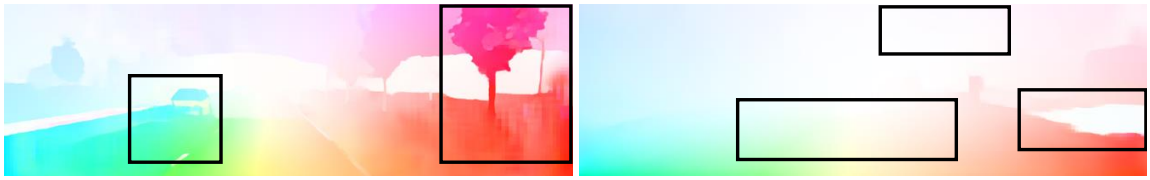
3.3. Differences from previous arts

Previous arts [1, 6, 19] present their meta-learning methods either in a supervised manner or in an unsupervised manner. On the contrary, our proposed algorithm is a hybrid variant of meta-learning. In Algorithm 1, we meta-train the network in a supervised fashion. Then, in Algorithm 2, we perform a quick adaptation with an unsupervised loss in the test-time. In summary, our method is a mixed approach that allows fast adaptation in the test-phase.

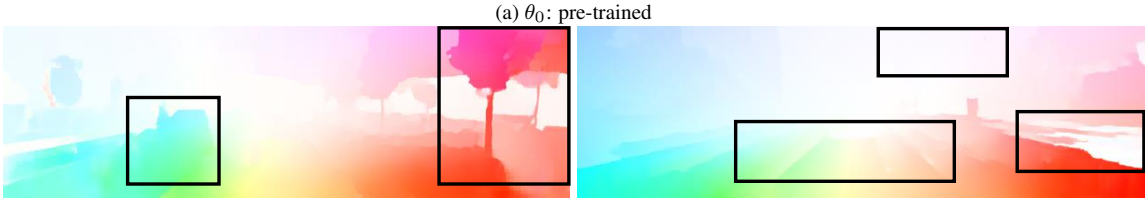
4. Experiments

4.1. Implementational details

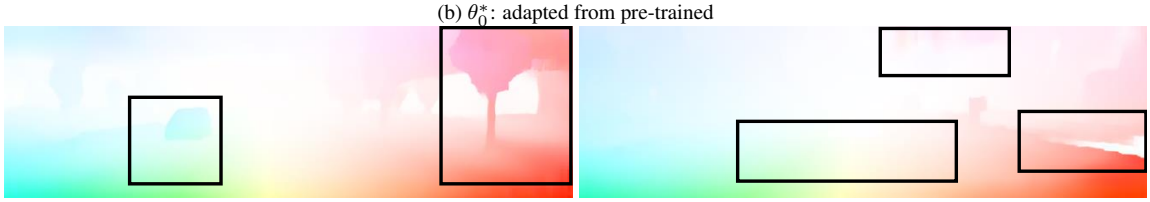
We use pytorch [24] and learn-to-learn [2] libraries to implement our adaptation algorithms. Learning rate was set to be 1.25×10^{-4} for fine-tuning, following the learning rate of the baseline model [15]. We set the learning rate to be 5×10^{-6} for β and 1.0×10^{-5} for our α and all



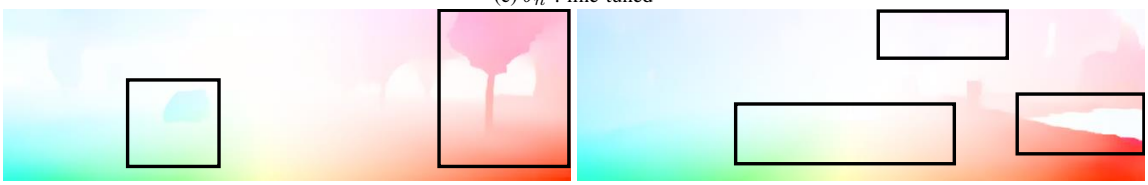
(a) θ_0: pre-trained



(b) θ_0^*: adapted from pre-trained

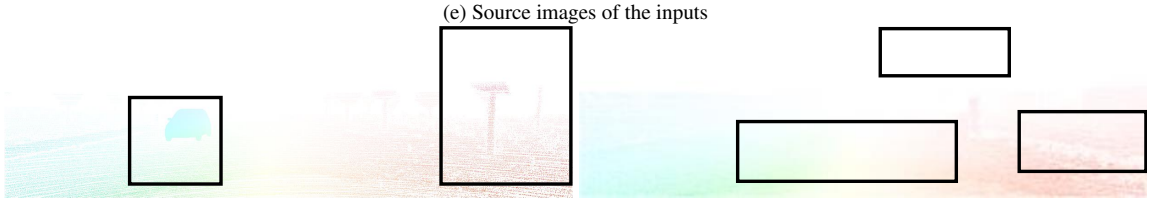


(c) θ_n^{ft}: fine-tuned



(d) θ_n^{ml*}: proposed

(e) Source images of the inputs



(f) Ground-truth

Figure 3. A couple of qualitative examples of evaluations on a real-world dataset, KITTI-15's S_t.

the other adaptations as similar approaches [5, 14, 28] have adopted less learning rates. We used standard Adam [17] optimizer to minimize the proposed losses in Algorithm 1 and 2. We set our adaptation gradient descent steps i.e., N 3. For flow estimation, we adopt GMA [15] as our baseline model where the number of iterations of convGRU cells in GMA is 12. For evaluation, we measure the performance of the flow networks in terms of EPE.

4.2. Dataset settings for comparison

We use conventional benchmark datasets as in [5, 14, 28] and selected FlyingChairs[5] and FlyingThings[21] for pre-training. The abbreviations of these two datasets are C and T, respectively. We use officially provided pretrained networks of GMA [15] as our baseline and evaluated networks adapted by the proposed algorithms on two independent datasets, KITTI-15 and MPI-Sintel. We let K and S denote these datasets as in previous works [5, 28, 14]. Specifically, KITTI-15 training split was created from real-world images on roads and consists of 200 frames. On the other hand, Sintel train split was made from synthetic scenes, and it comprises 23 scenes and 1041 frames in total. Notably, we adopted the final pass on Sintel dataset as a representative. This is because final pass is made up of fully rendered

images including atmospheric effect, motion blur, and camera depth-of-field blur. Therefore, the final pass is more likely to meet our purpose of testing with more challenging images.

Previous works [5, 14, 28] conventionally examine their models on S and K after training on C and T. Then, they fine-tune their models on entire S and K trainsets respectively. The next procedure is evaluating each fine-tuned model on the same entire S or K trainsets. This is obviously not a fair evaluation - that is why those results are often put inside parenthesises -, so we avoid reporting such results. Instead, [5, 14, 28] additionally provide outcomes from KITTI and MPI-Sintel benchmarks testsets.

4.3. Quantitative and qualitative flow results

We further highlight our generalization ability by repeated random samplings of metaset since the KITTI trainset, for example, comprises no more than 200 image pairs. In Tab. 2, we present the average of three random splittings of S_m for meta-train and S_t for meta-inference. We note that S_m and S_t are two disjoint sets. In Tab. 2, we let θ_0, θ_n^{ft}, θ_n^{ml} denote the pre-trained, the fine-tuned for n iterations, and the meta-trained with n outer-loop iterations, respectively, Then, we use * for denoting the adaptation or the meta-inference. The pretrained AEPEs on KITTI and Sintel datasets are 10.2, 4.65, 4.11, 2.75 while AEPE of our final meta-inference results are 3.32, 2.69, 3.47, 2.74, which shows superiority of the proposed algorithm. During our final fast adaptation process, the error decreases from 5.74 to 3.32 in case of pretraining on C and testing on KITTI. Similar tendency appears in the rest of the rows. This performance gain is achieved by only three gradient descent steps.

For the qualitative results, the bottom two rows in Fig. 2 effectively illustrates our method's benefits. In this particular frame, left and right sides are disoccluded and occluded in the next frame. θ_n^{ml} denotes meta-trained parameters by n iterations. Our method's final result from θ_n^{ml*} outperforms other methods in such challenging cases. We analyze that this is because the parameter θ_n^{ml} has been trained to minimize the EPE after few adaptation process with the same Eq. 2. This enables our θ_n^{ml} to easily adapt to θ_n^{ml*}. Although fine-tuned parameters θ_n^{ft} is similar in performance with θ_n^{ml} as an intermediate state, improvements are hardly seen in θ_n^{ft*}. Adaptation after naive fine-tuning still poorly handles frame boundary areas.

Note that to the best of our knowledge, we are the first to analyze optical flow test-time learning with a small amount of training data from a target domain. Since most of the optical flow estimation papers utilize Sintel and KITTI for the evaluation, we are performing an uncommon analysis on those datasets. Thus, although a direct comparison to the state-of-the-art methods is nearly unfeasible, we open up a new possibility of improving the estimation results in a test-time manner. For instance, in Tab. 1, our baseline GMA[15] reports 4.69 AEPE on KITTI-15 when pretrained on C+T. However, our method's performance in Tab. 3 begins with 4.68, which is in effect 4.69, as θ_0 and reduces it to 3.11 with only five training inputs from 200 KITTI-2015 target data. Moreover, the additional test-time learning only takes 3 gradient descent steps.

4.4. Ablation study: comparison to fine-tuning

In order to further verify our method's advantage, we conducted fine-tuning with the same S_m for ourselves. If we assume that metaset is available, one may question whether meta-learning is necessary rather than fine-tuning. This questioning motivates us to present the results of traditional fine-tuning on S_m. The performances of fine-tuning are 3.59, 2.73, 3.84, 2.81, respectively for each row in Tab. 2. Furthermore, we perform Algorithm. 2 on top of θ_0 and θ_n^{ft} respectively. We denote them as θ_0^* and θ_n^{ft*} as counterparts of our θ_n^{ml*}. For θ_0^* and θ_n^{ft*}, the end point error increases for some cases on KITTI and partially descreases for the other cases. This limited improvement is due to the fact that the general fine-tuning is a plain training process, and this training does not take into account the subsequent adaptation. As a result, unsupervised learning becomes highly susceptible to D_{new}'s satisfaction of optical flow priors and eventually malfunctions for difficult dataset such as KITTI. The performance gain from 10.3 to 9.6 in the first row results from the initial high error. High error allows larger space for error drop. However, for θ_n^{ml*}, additional performance gain from θ_n^{ml}can be obtained at almost any case, resulting in better outcomes than fine-tuning. We argue that this performance is the effect of transporting the parameter set to a position that is suitable for test-time adaptation scheme. Therefore, a specialized network for each incoming input is made more possible in our method.

4.5. Analysis: the amount of labeled data

In this study, we analyze the possibility of restricting the number of training data in the target domain. Such exploration of small S_m has not been heavily investigated by other optical flow studies. Nevertheless, we empirically demonstrate that exploiting knowledge from a smaller portion of a new domain is achievable. In Tab. 3, in case at most 50% of the test domain should be labeled for training, fine-tuning finds a similar point at the loss function's hyper-plane, reaching our method's results. However, such assumption of exhaustive labeling often becomes expensive due to the fact that obtaining accurate optical flow GT for real scenes is difficult to be automated. On the other hand, our θ_n^{ml*}'s performance significantly improves upon θ_n^{ft} when there are only a limited number of meta-training data available. As our method yields larger gains in Tab. 2

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}	θ_n^{ft*}	θ_n^{ml}	θ_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)

Table 3. Analaysis on the effect of the amount of labeled data. S_m is splitted from the target domain. The results are computed on KITTI-2015 while θ_0 is pretrained on C+T. In case of lower ratio of metaset, our method significantly improves upon naive fine-tuning. All results are in pixel unit.

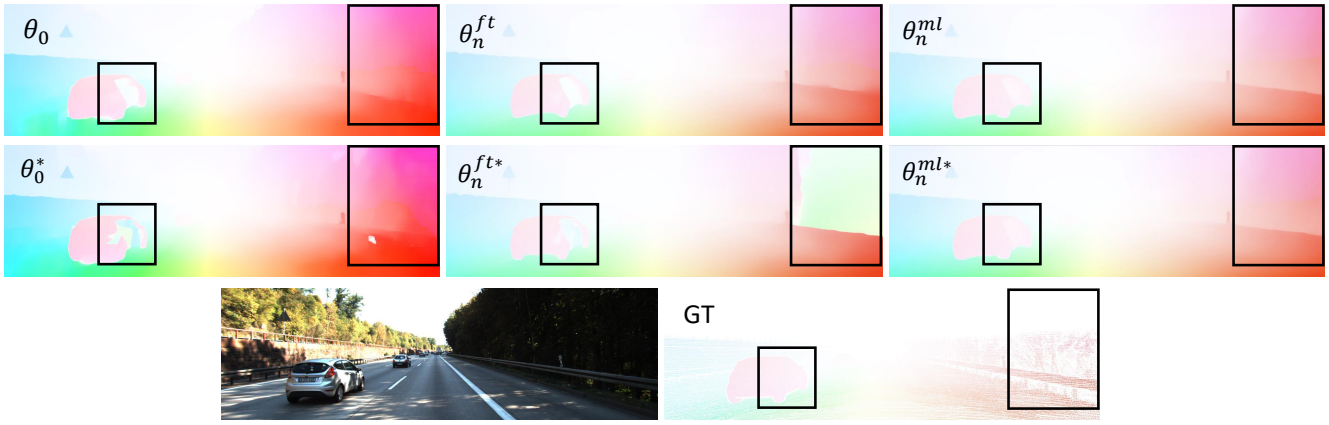


Figure 4. Qualitative results on KITTI-2015. Note that only 2.5% of the target domain data was used for test-time adaptation with and without meta-learning. Both methods have drastic improvements over the baseline parameter (i.e., θ_0), but our result from meta-inference is a better outcome for the moving car and the out-of-frame occlusion in the rightmost areas. In addition, θ_n^{ml*} benefits from the adaptation while the others obviously do not with the same unsupervised loss.

when starting from pre-training on C, we can arguably assume even larger gains for Tab. 3 in that case. Standard training including fine-tuning typically requires abundant training data and is otherwise susceptible to over-fitting. Moreover, the gain's standard deviation remains about the half of the average, which further implies our method's consistent advantages over the counterpart.

5. Conclusion

We present a test domain adaptation method that enables a neural network to have separate sets of network coefficients for different scenarios. Then, we provide the optical flow problem as a prime example. Current optical flow estimation methods have significant performance gaps between the training domains and test domains. It follows that they struggle to be generalized to unseen real scenes. Our approach formulates a revised meta-learning framework so that a pretrained network can learn to adapt on a novel domain. This is made possible by exploiting internal motion statistics in different motion distributions. For reliability, we demonstrate several comparative analysis with naive methods such as fine-tuning and test-time unsupervised learning. The proposed method exhibits significant improvements in most of the performances on widely used datasets. This achievement can be implemented in a simple and effective way and is independent of the base optical flow network.

Acknowledgement

This work was supported in part by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (NRF-2021R1A2C2010245)

This work was partially supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2022-0-00156, Fundamental research on continual meta-learning for quality enhancement of casual videos and their 3D metaverse transformation.)

This work was supported in part by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2020-0-01373, Artificial Intelligence Graduate School Program(Hanyang University))

References

[1] Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your maml. *arXiv preprint arXiv:1810.09502*, 2018.

[2] Sébastien M R Arnold, Praateek Mahajan, Debajyoti Datta, Ian Bunner, and Konstantinos Saitas Zarkias. learn2learn: A library for Meta-Learning research. Aug. 2020.

[3] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 25–36. Springer, 2004.

[4] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 611–625. Springer, 2012.

[5] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2758–2766, 2015.

[6] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning (ICML)*, pages 1126–1135. PMLR, 2017.

[7] Zhiyi Gao, Yonghong Hou, Yan Liu, and Xiangyu Li. Metaflow: a meta-learning-based network for optical flow estimation. *Journal of Electronic Imaging*, 30(3):033029, 2021.

[8] Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 8977–8986, 2019.

[9] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.

[10] Mike Huisman, Jan N Van Rijn, and Aske Plaat. A survey of deep meta-learning. *Artificial Intelligence Review*, 54(6):4483–4541, 2021.

[11] Sontje Ihler, Felix Kuhnke, Max-Heinrich Laves, and Tobias Ortmaier. Self-supervised domain adaptation for patient-specific, real-time tissue tracking. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 54–64. Springer, 2020.

[12] Sontje Ihler, Max-Heinrich Laves, and Tobias Ortmaier. Patient-specific domain adaptation for fast optical flow based on teacher-student knowledge transfer. *arXiv preprint arXiv:2007.04928*, 2020.

[13] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2462–2470, 2017.

[14] Jisoo Jeong, Jamie Menjay Lin, Fatih Porikli, and Nojun Kwak. Imposing consistency for optical flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3181–3191, 2022.

[15] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning to estimate hidden motions with global motion aggregation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 9772–9781, 2021.

[16] Rico Jonschkowski, Austin Stone, Jonathan T Barron, Ariel Gordon, Kurt Konolige, and Anelia Angelova. What matters in unsupervised optical flow. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 557–572. Springer, 2020.

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[18] Daniel Kondermann, Rahul Nair, Katrin Honauer, Karsten Krispin, Jonas Andrulis, Alexander Brock, Burkhard Gussefeld, Mohsen Rahimimoghaddam, Sabine Hofmann, Claus Brenner, et al. The hci benchmark suite: Stereo and flow ground truth with uncertainties for urban autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19–28, 2016.

[19] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*, 2017.

[20] Pengpeng Liu, Irwin King, Michael R Lyu, and Jia Xu. Ddflow: Learning optical flow with unlabeled data distillation. In *Association for the Advancement of Artificial Intelligence (AAAI)*, volume 33, pages 8770–8777, 2019.

[21] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4040–4048, 2016.

[22] Simon Meister, Junhwa Hur, and Stefan Roth. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In *Association for the Advancement of Artificial Intelligence (AAAI)*, volume 32, 2018.

[23] Moritz Menze, Christian Heipke, and Andreas Geiger. Joint 3d estimation of vehicles and scene flow. *ISPRS annals of the photogrammetry, remote sensing and spatial information sciences*, 2:427, 2015.

[24] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[25] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12240–12249, 2019.

[26] Zhe Ren, Junchi Yan, Bingbing Ni, Bin Liu, Xiaokang Yang, and Hongyuan Zha. Unsupervised deep learning for optical flow estimation. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2017.

[27] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8934–8943, 2018.

[28] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 402–419. Springer, 2020.

[29] Thomas Varsavsky, Mauricio Orbes-Arteaga, Carole H Sudre, Mark S Graham, Parashkev Nachev, and M Jorge Cardoso. Test-time unsupervised domain adaptation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 428–436. Springer, 2020.

[30] Hengli Wang, Rui Fan, Peide Cai, Ming Liu, and Lujia Wang. Undaf: A general unsupervised domain adaptation framework for disparity or optical flow estimation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 01–07. IEEE, 2022.

[31] Yang Wang, Yi Yang, Zhenheng Yang, Liang Zhao, Peng Wang, and Wei Xu. Occlusion aware unsupervised learning of optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4884–4893, 2018.

[32] Andreas Wedel, Daniel Cremers, Thomas Pock, and Horst Bischof. Structure-and motion-adaptive regularization for high accuracy optic flow. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1663–1668. IEEE, 2009.

[33] Gengshan Yang and Deva Ramanan. Volumetric correspondence networks for optical flow. *Advances in neural information processing systems*, 32, 2019.

[34] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1983–1992, 2018.

[35] Shengyu Zhao, Yilun Sheng, Yue Dong, Eric I Chang, Yan Xu, et al. Maskflownet: Asymmetric feature matching with learnable occlusion mask. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6278–6287, 2020.