Semi-Supervised Learning of Optical Flow by Flow Supervisor

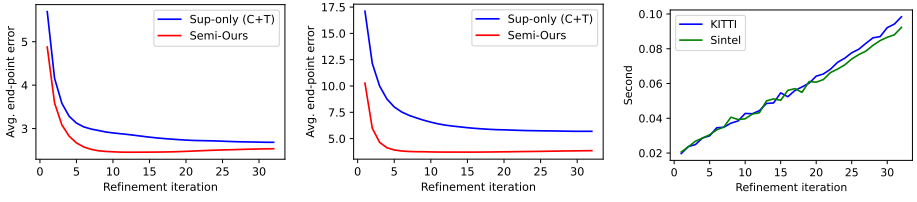
Supplementary Material

Woobin Im, Sebin Lee, and Sung-Eui Yoon

Korea Advanced Institute of Science and Technology (KAIST),
{iwbn,seb.lee,sungeui}@kaist.ac.kr

1 Additional Results

1.1 Refinement Steps and Faster Inference



(a) Sintel Final EPE (b) KITTI EPE (c) Time vs. Iteration

Fig. 1: **Results w.r.t. the refinement iteration**. **(a-b)** shows validation EPEs of the baseline (Sup-only) and ours (Semi-Ours). **(c)** shows the inference time per frame with different resolutions: KITTI (1242×375) and Sintel (1024×436). For the comparison of time, we use a single RTX 3090 GPU (24GB VRAM) with our TensorFlow implementation of RAFT.

In Fig. 1, we show EPEs with respect to refinement steps. Since we base our network on RAFT [3], setting an appropriate refinement step is crucial for the performance of estimation. By the experiments, we observe that training with our self-supervision method results in a faster convergence. In Fig. 1a-1b, we show that our semi-supervised learning method has a faster convergence iteration than the baseline (Sup-only) model. That is the reason we choose shorter refinement steps, i.e., 12, than the original paper, i.e., 32. Thanks to the fewer refinement steps, the inference time is reduced by about 50% (see Fig. 1c) on a target dataset, when we set the iteration to 12 instead of 32.

1.2 Stopping Gradients

Our design choice for `stop_grad` is to make the supervisor module an isolated module, which makes gradient computation step for the flow supervisor more

compact and efficient. Furthermore, stop-gradient is a widely adopted technique in self-supervised learning to prevent a degenerated solution [1]. We discovered that disabling `stop_grad` for each module shows worse results on Sintel Final:

Teacher encoder	$\mathbf{h}_s$	$\hat{\mathbf{y}}_{FS}$	**Ours** (all enabled)
2.58	2.52	diverged	**2.46**

Note that `stop_grad` for $\hat{\mathbf{y}}_s$ belongs to original RAFT.

1.3 More Qualitative Results on Sintel and KITTI

We show more qualitative comparisons in Fig. 4-5. Even though we do not exploit a ground truth of the target dataset, our method clearly improves challenging regions. In the KITTI examples, our method is especially effective on objects near image frames and shadows. For Sintel – which includes many motion blurs and lens effects – the network becomes robust to those challenging effects, as shown in the examples.

2 Experimental Details

2.1 Architecture

We show the implementation of our method in Fig. 3. Our architecture is based on RAFT [3], where the iterative refinement plays an important role. The self-supervision process is similar to the supervised learning, where the intermediate flows $\hat{\mathbf{y}}_s^i$ are supervised by a target flow $\mathbf{y}$. In the RAFT paper, the decaying parameter $0 < \gamma \leq 1$ is used in the loss function:

$$\ell_{\text{sup}} = \sum_{i=1}^{n} \gamma^{n-i} \|\hat{\mathbf{y}}_s^i - \mathbf{y}\|_1. \tag{1}$$

Similarly, we apply the decaying strategy in our self-supervision by minimizing:

$$\ell_{\text{FS}} = \sum_{i=1}^{n} \gamma^{n-i} \rho(\hat{\mathbf{y}}_s^i - \hat{\mathbf{y}}_{\text{FS}}^{n+m}), \tag{2}$$

where $\hat{\mathbf{y}}_{\text{FS}}^{n+m}$ is the pseudo label predicted by the flow supervisor. Specifically, we use $\gamma = 0.8$ (ℓ_{sup}), $\gamma = 0.8$ ($\mathcal{L}_{\text{FS}}$), and $\gamma = 1.0$ ($\mathcal{L}_{\text{TS}}$). For KITTI, we use $\gamma = 0.8$ ($\mathcal{L}_{\text{TS}}$, $\mathcal{L}_{\text{TU}}$) for the supervisor model.

2.2 Padding Operation

As mentioned in the main text, we use a cropping operation to give supervision from the supervisor network to the student network. Passing student outputs requires the student outputs to be aligned with the uncropped images (i.e., teacher inputs). Since RAFT use 1/8 processing resolution, we use a padding operation performed at 1/8 resolution, and the random offset coordinates for cropping operation is constrained to multiples of 8; this results in sizes of all inputs – augmented, privileged, and crop offsets – to be multiples of 8.

2.3 Optimization

In the fine tuining stage, we use batch size 1 each from a labeled dataset and an unlabeled dataset, which requires a single RTX3090 GPU, and it takes about one day to converge. We use Adam optimizer [2], and we decay the learning rate from 10^{-5} by 1/2 every 25,000 steps. Different from supervised training, we use the generalized Charbonnier loss $\rho(\cdot)$ in ℓ_{sup}, ℓ_{self}, and ℓ_{FS} for semi-supervised training. Detailed hyper-parameters and reproducible experimental settings are provided in our code.

Unsupervised Loss. In $\mathcal{L}_{\text{TU}}$ and ablation results, we use the unsupervised loss $\ell_{\text{unsup}}(\cdot)$ for comparison. As mentioned in the main paper, we use the photometric loss, occlusion handling, and the smoothness loss, which are brought from SMURF implementation [1]. Following the code, we use `census=1`, `smooth1=2.5`, `smooth2=0.0`, and `occlusion='wang'` for Sintel; `census=1.0`, `smooth1=0.0`, `smooth2=2.0`, and `occlusion='brox'` for KITTI. We do not use the self-supervision loss (`selfsup=0.0`) since our method includes the similar self-supervision strategy.

3 Self-Supervision Example

Our self-supervision is performed by the flow supervisor which is conditioned on the student outputs and clean inputs. The process is summarized in Fig. 2. In the example, we show consecutive frames from a driving scene, where the observer is moving forward, so that objects near the image frame are hidden by the frame. For example, the tree on the right side is not visible in the second cropped image, while the original second image contains the tree. Thus, the teacher prediction can correct the student prediction by utilizing the privileged information, as shown in the figure. Compared to the ground truth, we can observe the correcting direction $\hat{\mathbf{y}}_s - \hat{\mathbf{y}}_{\text{FS}}$ is close to $\hat{\mathbf{y}}_s - \mathbf{y}$ computed by GT. Thus, our flow supervisor can generate a desirable supervision signal to guide the student network by the privileged inputs.

References

1. Chen, X., He, K.: Exploring simple siamese representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15750–15758 (2021) 2
2. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 3
3. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: European Conference on Computer Vision. pp. 402–419. Springer (2020) 1, 2, 4

[1] https://github.com/google-research/google-research/tree/master/smurf

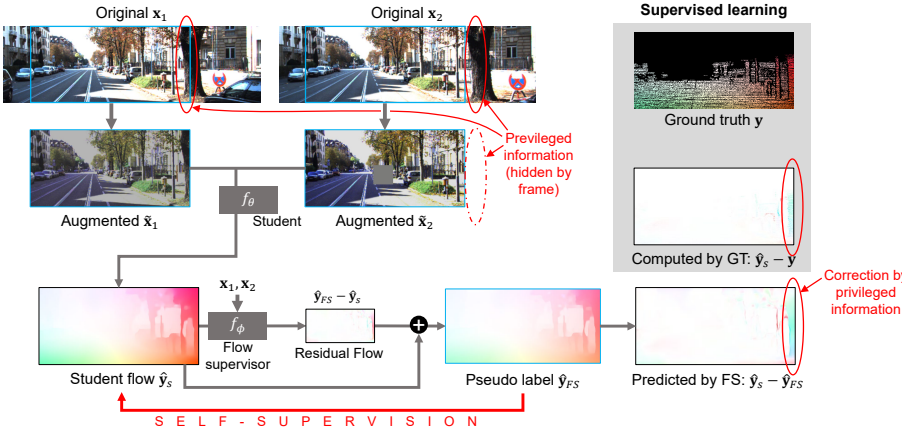
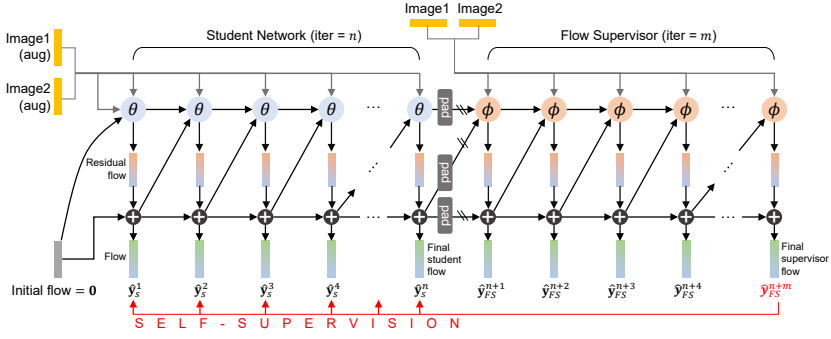
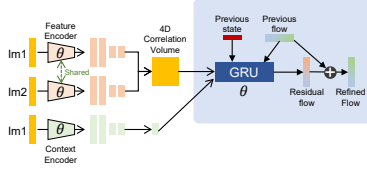


Fig. 2: **Self-supervision example.**

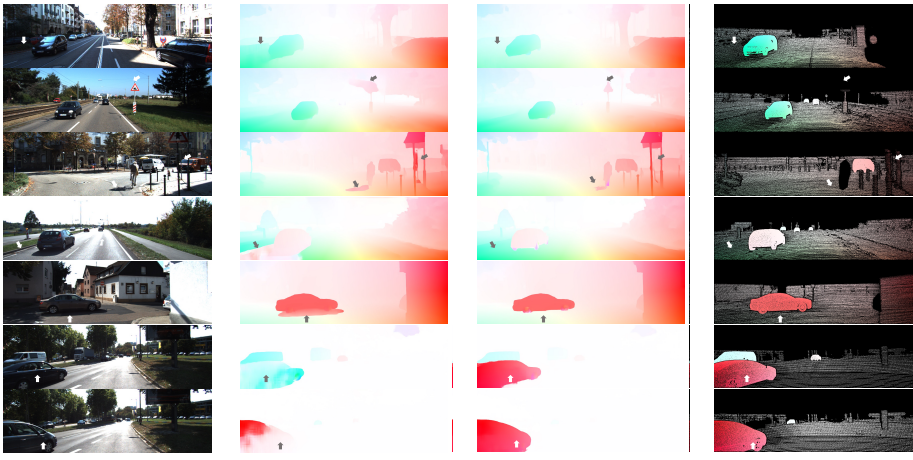


(a) Overall structure for self-supervision



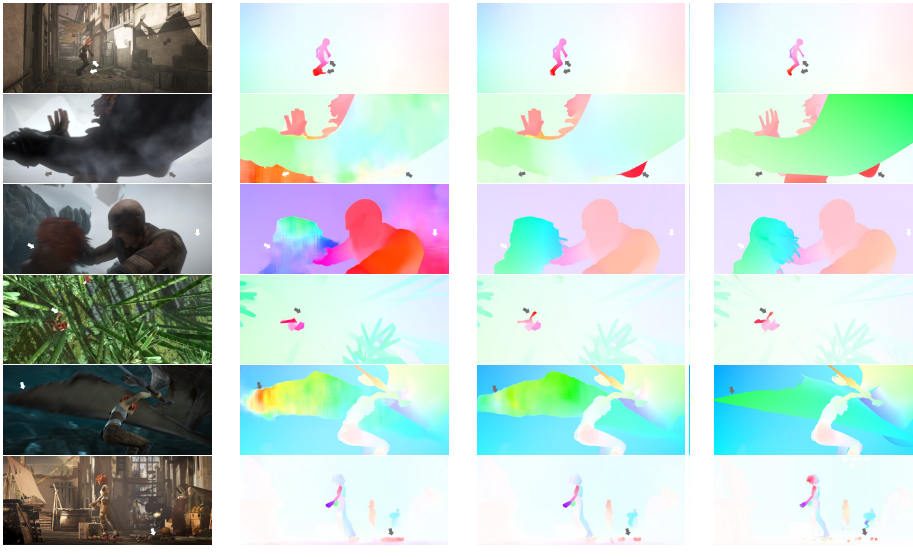
(b) A refinement step of RAFT [3]

Fig. 3: **Detailed architecture. (a)** summarizes the detailed structure of our flow supervisor. Our flow supervisor shares the design of iterative refinement RNN module of RAFT. Since we feed full images to the flow supervisor, we pad the outputs of student network to feed them to the supervisor. **(b)** depicts one refinement step of RAFT with the feature encoder and context encoder. For technical details of each layer, please refer to [3].



(a) Input (b) Sup-only (c) Semi-Ours (d) Ground truth

Fig. 4: **Qualitative results on KITTI.** We compare results of RAFT trained on VKITTI. **(b)** shows optical flows predicted by RAFT pretrained on VKITTI. **(c)** shows flows prediected by our semi-supervised method, which utilizes an additional KITTI dataset without ground-truth. All results are obtained on unseen samples.



(a) Input (b) C+T (Sup-only) (c) C+T (Semi-Ours) (d) Ground truth

Fig. 5: **Qualitative results on Sintel Final.** We compare results of RAFT trained on C+T. **(b)** shows optical flows predicted by RAFT pretrained on FlyingChairs and FlyingThings. **(c)** shows flows prediected by our semi-supervised method, which utilizes an additional Sintel dataset without ground-truth. All results are obtained on unseen samples.