VoteFlow: Enforcing Local Rigidity in Self-Supervised Scene Flow

Supplementary Material

1. Ablation studies on ground removal

We followed the official Argoverse 2 benchmark and removed the ground with rasterized maps in VoteFlow. Tab. I now includes performance after ground removal using Patchwork++ [1], an offline algorithm. Without maps, performance drops notably, particularly on pedestrians (Pd.) and wheeled VRUs (W.V). Compared to rasterized maps, Patchwork++ achieves 95.5% precision and 82.3% recall in ground classification. Cases without flat ground are not well studied in current benchmarks, which remains an open challenge.

Ground removal	Mean	Car	O. V.	Pd.	W. V
Patchwork++	0.369	0.246	0.364	0.477	0.389
Rasterized maps	**0.335**	**0.222**	**0.347**	**0.424**	**0.347**

Table I. **Impact of ground removal on Argoverse 2 val split.**

2. Ablation studies on the pillar size

Tab. II shows decreasing pillar size improves accuracy but increases latency. Our default size $0.2m$ balances performance and latency (on an A100 GPU).

Pillar (m)	Latency (ms)	Mean	Car	O.V.	Pd.	W. V.
0.1	58.6±6.5	**0.327**	**0.216**	0.360	**0.403**	**0.331**
0.2	25.6±5.2	0.335	0.222	**0.347**	0.424	0.347
0.4	**17.1±4.6**	0.371	0.226	0.369	0.524	0.366

Table II. **Ablation study on pillar sizes on Argoverse 2 val split.**

3. Ablation studies on loss functions

SeFlow [2] employs multiple losses to enforce consistent flow predictions. For example, the cluster loss $\mathcal{L}_{cluster}$ encourages consistent flow prediction from the same cluster; $\mathcal{L}_{static}$ directly forces the static flows to be zeros; $\mathcal{L}_{dynamic}$ is explicitly imposed on points that are classified as dynamic in preprocessing. The usage of these losses shares a purpose similar to our Voting Module, i.e., to make flow prediction consistent. To evaluate the impact of our Voting Module in an isolated environment, we conduct ablation studies where no additional losses are adopted other than $\mathcal{L}_{chamfer}$. Tab. III compares the performance of SeFlow and VoteFlow with $\mathcal{L}_{total}$ and $\mathcal{L}_{chamfer}$ alone (excluding $\mathcal{L}_{cluster}$, $\mathcal{L}_{dynamic}$, and $\mathcal{L}_{static}$). Both models show a significant performance drop when trained only with

Method	Loss	Bucket Normalized EPE (↓)				
		Dynamic (normalized EPE)				
		Mean	Car	O. V.	Pd.	W. V
SeFlow [2]	$\mathcal{L}_{total}$	0.369	0.234	**0.342**	0.541	0.358
VoteFlow (Ours)	$\mathcal{L}_{total}$	**0.335**	**0.222**	0.347	**0.424**	**0.347**
SeFlow [2]	$\mathcal{L}_{chamfer}$	0.463	0.347	0.579	0.541	0.386
VoteFlow (Ours)	$\mathcal{L}_{chamfer}$	**0.444**	**0.320**	**0.563**	**0.511**	**0.381**

Table III. **Impact of loss functions.** All results are from Argoverse 2 val split. We test the performance of our model and the baseline SeFlow [2], which have been trained by only $\mathcal{L}_{chamfer}$. This makes sure the model has not been regularized by any other loss functions related to motion rigidity. The performance improvement over SeFlow on Dynamic Mean indicates the benefit of the Voting Module in our design.

$\mathcal{L}_{chamfer}$, indicating the importance of the explicit loss as regularization. However, VoteFlow still outperforms SeFlow across all categories, with a notable 2.1%pt improvement in Dynamic Mean, demonstrating the effectiveness of the Voting Module.

4. Ablation studies on the Voting Module and voting features

Voting	Voting Feats	Mean	Car	O. V.	Pd.	W. V
✗	✗	0.373	0.222	0.397	0.512	0.362
✓	Fused feats $\mathbf{G}$	0.348	**0.220**	0.383	0.445	**0.344**
✓	Pillar feats $\mathbf{I}$	**0.335**	0.222	**0.347**	**0.424**	0.347

Table IV. Ablation study on the Voting Module and voting features. Taking (separated) features from the Pillar Feature Net further enhances performance over taking fused features from the U-Net.

Tab. IV firstly compares models of the same architecture without (✗) and with (✓) the Voting Module. Adding voting improves the mean by 3.8%pt. Additionally, we explored multiple configurations for the input features of our Voting Module. In the first configuration, we use the separate pillar indices $\mathbf{P}^t$ and $\mathbf{P}^{t+\Delta t}$ from the input point clouds to retrieve per-point features from fused feature $\mathbf{G}$. This design outperforms its counterpart without the Voting Module by a margin of 2.5%pt in Mean Dynamic, indicating $\mathbf{G}$ effectively encodes the fused semantics from both point clouds. An alternative design feeds the pseudo images $\mathbf{I^t}$ and $\mathbf{I^{t+\Delta t}}$ directly into the Voting Module. In contrast to $\mathbf{G}$, $\mathbf{I}$s are separate pillar features for source and target point clouds. As shown in Tab. IV, the second design achieves further perfor-

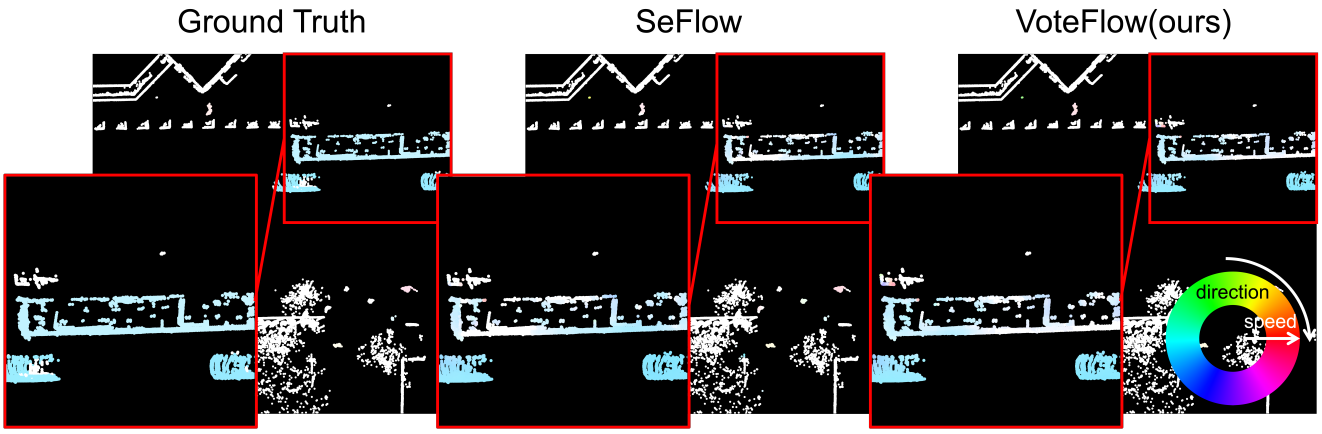

Figure 1. **Failure cases on Argoverse 2 validation set.** Colors indicate directions and saturation of the color indicates the scale of the flow estimation. Both SeFlow and VoteFlow struggle to predict consistent flows for large-size, rigidly moving objects.

mance improvement over the first one. We therefore argue that our proposed Voting Module is a universal method for exploiting the motion rigidity prior regardless of which type of features are used as input.

5. Additional Qualitative Results

We show a failure case of our model in Fig. 1 where it fails to produce consistent flow for larger objects that move rigidly. We suspect the failure originates from the fact that the predefined local neighborhood where local rigidity holds is not large enough to cover the entire object. Future work could explore adjusting the neighborhood range adaptively for such cases.

References

[1] Seungjae Lee, Hyungtae Lim, and Hyun Myung. Patchwork++: Fast and robust ground segmentation solving partial under-segmentation using 3d point cloud. In *IROS*, 2022. 1

[2] Qingwen Zhang, Yi Yang, Peizheng Li, Olov Andersson, and Patric Jensfelt. Seflow: A self-supervised scene flow method in autonomous driving. In *ECCV*, 2025. 1