1. Expected Number of Correspondences

In Sec. 3 of the main paper, we point out that randomly subsampling stereoKITTI's input point clouds does not fix the one-to-one correspondence issue because there are still an expected 745 points with correspondences. Here, we explain how we arrived at that number. Each example in stereoKITTI comprises approximately 90,000 pairs of points (p_i, q_i). Many methods uniformly sample 8192 points from the p_i's, and the q_i's separately. For each chosen p_i there is some probability that its corresponding q_i is also chosen as input, and there is a random variable X, which is the number of times this happens. We are interested in computing the expectation of this random variable. Let Φ be the indices of the 8192 points chosen for the first frame. Now let x_i be an indicator random variable that is 1 if q_i is chosen as input and 0 otherwise. Observe that $X = \sum_{i \in \Phi} x_i$ and $\mathbb{E}[x_i] = p(q_i \text{ is chosen}) = \frac{8192}{90000}$. By the linearity of expectation: $\mathbb{E}[X] = \sum_{i \in \Phi} \mathbb{E}[x_i] = 8192 \cdot \frac{8192}{90000} \approx 745$.

2. Label Creation

Multiple authors have used ground truth multi-object tracks to scene-flow labels [1, 2]. We follow the same procedure to create labels for Argoverse 2.0.

Problem Statement: We assume as input two point clouds $\mathbf{P}^t \in \mathbb{R}^{N \times 3}, \mathbf{P}^{t+\Delta} \in \mathbb{R}^{M \times 3}$. The superscript indicates that these point clouds are separated by a small time delta (usually Δ = 0.1s), and the variables N, M indicate that the two point clouds have different numbers of points. The goal is to predict a set of flow vectors $\{\mathbf{f}_i \in \mathbb{R}^3\}_{i=1}^N$ which describe the motion of each point from time t to time $t + \Delta$. Some datasets also give access to the ego-motion of the sensor since in the autonomous vehicle setting this information is available from odometry or GPS.

Flow Label Creation: For each point cloud in a MOT dataset we have a set of oriented bounding boxes $\{B_i^t\}_{i=1}^K$. For each B_i^t, if the second frame at time $t + \Delta$ contains a corresponding bounding box $B_j^{t+\Delta}$, we can extract the rigid transformation $\mathbf{R}_i^t, \mathbf{t}_i^t$ that transforms points in the first box to the second. For each point $\mathbf{p}_j$ inside the bounding box we assign it the flow $\mathbf{f}_j = \mathbf{R}_i \mathbf{p}_j + \mathbf{t}_i$. Points not belonging to any bounding box are assigned the ego-motion as flow. For objects which only appear in one frame, we cannot compute the ground truth flow and so they are ignored for evaluation purposes but included in the input.

Limitations: This procedure for producing flow labels has two drawbacks. First is that all the points inside a bounding box may not be moving rigidly. While it is important to be aware of this we believe the errors introduced are small since most objects are rigid vehicles and the sampling rate of 10Hz prevents large deviations from the model. Second, there can exist objects that are not captured by the tracking labels but are nonetheless dynamic. This second limitation is the main reason for our interest in self- and un-supervised methods as existing work has demonstrated degradation of supervised performance on objects not included in the training labels [2]. However, it is supervised training that creates the correlation between performance on an object class and its presence in the labels. For self-supervised methods we expect performance on tracked objects to correlate with those that may be missing.

3. Local Dynamic Segmentation Model Details

In Sec. 3 of the main paper, we use a local classification model to demonstrate how the sampling pattern of dynamic objects in stereoKITTI makes it trivial to identify moving vs. static objects. Here we give details of the architecture of that model. The architecture is shown in Fig. 1 and is essentially a very simple version of PointNet [5] with the transform modules removed. The key component of this model is that the input contains only the relative positions of the query point's neighbors, so no global information is available to the network.

4. Optimization Details

Our baseline uses 6 parameters, which we detail here. In general, the same parameters are used across all datasets (Arogverse, Waymo, NuScenes, lidarKITTI). However, due to the large variations in sparsity in both Waymo and NuScenes two parameters were adjusted: the early stopping criterion, and the DBSCAN epsilon parameter. These adjustments were not found through a parameter search, but by visually inspecting the flow results and clusters respectively.

Neural Prior Parameters: The forwards and backward flow networks have the same structure as [4] and we optimize them with the Adam [3] optimizer using a learning rate of 0.004 and no weight decay. We stop the optimization when no progress is made for 100 iterations (200 when optimizing on Waymo).

RANSAC Parameters: For all clusters, we do 250 RANSAC iterations and we use an inlier threshold of 0.2. When thresholding on the translation component, we use the same threshold as the dataset's dynamic threshold: $0.5\,\mathrm{m/s}$.

DBSCAN Parameters: For Argoverse, Waymo and lidarKITTI we use an epsilon parameter of 0.4 but for NuScenes we increase this to 0.8 due to the sparsity. For all datasets, we use a minimum point threshold of 10.

5. Ablation

To test the components of our method, we performed an ablation study on Argoverse. The results are shown in Tab. 1 and show that both Motion Compensation and Rigid Refinement play key roles in improving the performance of the

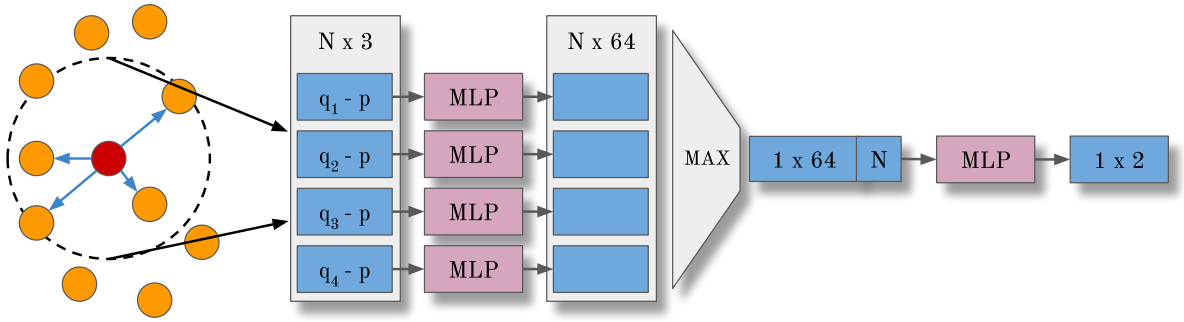

Figure 1. Our local segmentation model only uses information about the relative position of points in a local neighborhood around the predicted point. The local and global MLPs each have two hidden layers of 64 units.

	EPE				AccR	AccS
	Avg	Dynamic	Static		Dynamic	Dynamic
		FG	FG	BG	FG	FG
Backbone	0.088	0.193	0.033	0.039	0.542	0.327
w/ Motion Compensation	0.066	0.112	0.042	0.046	0.756	0.515
w/ Motion & w/ Rigid Refinement	0.055	0.105	0.033	0.028	0.777	0.537

Table 1. Ablation of the two main components of our model, motion compensation and rigid refinement.

backbone scene flow method. Motion Compensation has the largest impact on estimating dynamic motion since it allows the network to simply assign zero to all background points. The rigid refinement step improves dynamic motion estimates as well, but also has a large impact on static points. This is due to it fixing phantom motion estimates on walls caused by "swimming" artifacts.

References

[1] Stefan Andreas Baur, David Josef Emmerichs, Frank Moosmann, Peter Pinggera, Björn Ommer, and Andreas Geiger. Slim: Self-supervised lidar scene flow and motion segmentation. In *Int. Conf. Comput. Vis.*, pages 13126–13136, 2021. 1

[2] Philipp Jund, Chris Sweeney, Nichola Abdo, Zhifeng Chen, and Jonathon Shlens. Scalable scene flow from point clouds in the real world. *IEEE Rob. Aut. Letters*, 7(2):1589–1596, 2021. 1

[3] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[4] Xueqian Li, Jhony Kaesemodel Pontes, and Simon Lucey. Neural scene flow prior. *Adv. Neural Inform. Process. Syst.*, 34:7838–7851, 2021. 1

[5] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 652–660, 2017. 1