T-MAE: Temporal Masked Autoencoders for Point Cloud Representation Learning (Supplementary Material)

Weijie Wei, Fatemeh Karimi Nejadasl, Theo Gevers, and Martin R. Oswald

University of Amsterdam, the Netherlands

Abstract. This supplementary provides more details and analysis of our method. The implementation details are illustrated in Sec. S1. Additional ablation results are provided in Sec. S2. Section S3 discusses the selection of temporal gap during inference. More quantitative comparisons with other methods on the Waymo dataset [9] are depicted in Sec. S4. The transferability of T-MAE is verified in Sec. S5 and a comparison with multi-frame non-SSL methods is provided in Sec. S6. We analyze what the WCA module learns from the T-MAE pre-training in Sec. S7. Sec. S8 provides details about the comparison of finetuning iterations. Eventually, more qualitative results and limitations are presented in Sec. S9 and Sec. S10.

S1 Implementation Details

We follow the training settings of GD-MAE [13]. Some important configurations are listed in Tab. S1 and Tab. S2. We use the same masking ratio, the per-pillar number of predicted points K^O and target reconstructed points K^{GT} as GD-MAE because these parameters have negligible effects according to the ablation studies in GD-MAE [13].

Table S1: Training details.

Config	Pre-training	Finetuning
optimizer	AdamW [6]	
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.99$	
weight decay	0.01	
max learning rate	0.003	
learning rate scheduler	a cyclic learning rate with cosine annealing	
batch size (Waymo [9])	4	3
batch size (ONCE [7])	8	6
epoch	12	30
masking ratio	0.75	-
# predicted points K^O	16	-
# target reconstructed points K^{GT}	64	-

Table S2: Dataset-specific details. Finetuning time indicates the duration of finetuning using the entire training set.

Config	Waymo [9]	ONCE [7]
window size	$(8, 8, 1)$	
detection range - x-axis (m)	$(-74.88, 74.88)$	
detection range - y-axis (m)	$(-74.88, 74.88)$	
detection range - z-axis (m)	$(-2, 4)$	$(-3, 5)$
pillar size (m)	$(0.32, 0.32, 6)$	$(0.32, 0.32, 8)$
temporal batch	6	3
GPUs	8 × Tesla V100	4 × A100 (40GB)
Pre-training time (GPU day)	8 × 4	4 × 4.5
Finetuning time (GPU day)	8 × 5	4 × 0.25

Table S3: Ablation experiments on the Waymo dataset [9].

Ablation Target	Setting	L2 Overall	
		mAP	mAPH
Two-frame alignment	baseline	41.42	37.56
	w/ alignment	44.05$^{\uparrow 2.63}$	41.28$^{\uparrow 3.72}$
Data augmentation	w/ copy-n-paste [12]	46.76$^{\uparrow 5.34}$	43.93$^{\uparrow 6.37}$
Length of temporal batch	6→3	45.88$^{\uparrow 4.46}$	41.96$^{\uparrow 4.40}$
	6	46.76$^{\uparrow 5.34}$	43.93$^{\uparrow 6.37}$
	6→12	44.96$^{\uparrow 3.54}$	41.92$^{\uparrow 4.36}$

S2 Additional Ablation Study

Additional results on the Waymo dataset [9] are provided to justify design choices and hyperparameter values. Note that, in this section, we use 20% of the training set to pre-train our model and use 5% of the training set to fine-tune it for cost-effective experiments.

Two-frame Alignment. We start with using SiamWCA as a baseline and investigate whether it is necessary to align the previous frame to the coordinate system of the current frame. As shown in Tab. S3, aligning two frames significantly improves the metrics.

Data Augmentation. Based on the baseline with two-frame alignment, we further investigate the effect of a data augmentation technique, namely copy-n-paste [12]. Specifically, it first generates a ground truth instance database during dataset pre-processing. Then, during finetuning, several ground truth instances are randomly placed into the scene. This augmentation boosts the detection rates. Note that only ground truths in the 5% split are included in the database, which avoids ground truth leakage.

Length of Temporal Batch. This hyperparameter is set as 6 for the previous experiments in Tab. S3. In this ablation, we pre-train and fine-tune the model with different lengths of the temporal batch. As shown in Tab. S3, setting the length of the temporal batch as 6 achieves the optimal performance.

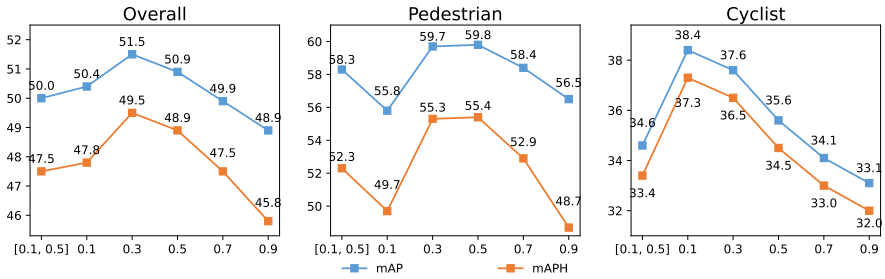


Fig. S1: Temporal interval for inference. Different temporal intervals were tested for inference. $[0.1, 0.5]$ indicates a random interval between 0.1 and 0.5 seconds. A fixed interval ranging from 0.1 to 0.9 seconds was also tested. Overall, a fixed interval of 0.3 seconds works best in our experiments.

S3 Temporal interval for inference.

This ablation study explores the impact of different temporal intervals on the model performance during inference. We establish a baseline with a random interval ranging from 0.1 to 0.5 seconds. In other words, $\mathcal{P}^{t_1}$ is selected randomly from the last five frame of $\mathcal{P}^{t_2}$. As a comparison, the temporal interval is fixed to values ranging from 0.1 to 0.9 seconds. If there is no satisfactory frame available, the earliest available frame is selected as $\mathcal{P}^{t_1}$. As shown in Fig. S1, The AP and APH for pedestrians achieve relatively high values when the interval is 0.3 and 0.5 seconds, while those for cyclists hit optimal when the interval is 0.1 seconds. The different optimal intervals could be attributed to the different speeds of the two categories. Pedestrians can barely move within 0.1 seconds, leading to historical information less useful. On the contrary, cyclists move relatively faster and thus the same bicycle of two frames is easy to be recognized as two instances if the temporal gap is large. Fig. S1 also illustrates that optimal overall performance is achieved when the interval is 0.3 seconds. Therefore, the previous third frame, which corresponds to a temporal interval of 0.3 seconds, is selected as $\mathcal{P}^{t_1}$ for the Waymo dataset.

S4 Additional Results on Waymo dataset

We explore two methods to obtain a subset of the training set for finetuning, but all models are evaluated on the same validation set. **Uniform Sampling** is commonly used [4, 13, 15]. Specifically, frames from all sequences are concatenated, from which the target frames are sampled with a fixed interval. **Data-efficient Benchmark** [11] selects a certain number of sequences as a training subset instead of uniform sampling frames, which aims to solve the data diversity issue existing in uniform sampling. More specifically, while finetuned on uniformly sampled frames, models always converge to a similar performance if they are finetuned for adequate iterations no matter how many percentages of labelled data are used. As a result, fewer data do not lead to a shorter finetuning time.

Table S4: Performance comparison on the Waymo validation set [9]. All methods are fine-tuned with 20% uniformly sampled frames, namely sampling one frame per five frames. Random initialization denotes training from scratch. Differences between T-MAE pre-training and random initialization are highlighted in red. ** indicates results from [13]. Other results are from AD-PT [17] or the survey [2]. Best results are highlighted as **first**, second, and third.

Data Amount	Initialization	Overall		Vehicle		Pedestrian		Cyclist	
		mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH
20% Uniform Sampling	Random	69.03	66.48	66.61	66.15	72.65	66.76	67.83	66.54
	ProposalContrast [15]	66.67	64.20	65.22	64.80	66.40	60.49	68.48	67.38
	BEV-MAE [5]	66.70	64.25	64.72	64.22	66.20	60.59	69.11	67.93
	Occupancy-MAE [8]	65.86	63.23	64.05	63.53	65.78	59.62	67.76	66.53
	MAELi-MAE [3]	65.60	63.00	64.22	63.70	65.93	59.79	66.66	65.52
	GD-MAE [13]**	70.24	67.14	67.67	67.22	73.18	65.50	69.87	68.71
	AD-PT [17]	67.17	64.65	65.33	64.83	67.16	61.20	69.39	68.25
	T-MAE (Ours)	**70.92**$^{\uparrow 1.89}$	**69.11**$^{\uparrow 2.63}$	**68.07**	**67.61**	**74.38**	**70.56**	**70.32**	**69.15**

Table S5: Performance on the Waymo validation set [9]. Results for other methods are taken from MV-JAR [11]. All methods are finetuned with Subset 1.

Finetuning split	Initialization	Overall		Vehicle		Pedestrian		Cyclist	
		mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH
5% S1	Random	49.59	46.15	54.42	53.87	52.59	44.17	41.76	40.41
	PointContrast [10]	48.97	44.91	52.35	51.85	52.49	41.95	42.07	40.91
	ProposalContrast [15]	49.87	45.83	52.79	52.31	53.30	43.00	43.51	42.18
	MV-JAR [11]	52.73	48.99	**56.66**	**56.21**	57.52	47.61	44.02	43.15
	T-MAE (Ours)	**53.44**$^{\uparrow 3.85}$	**51.58**$^{\uparrow 5.43}$	56.18	55.65	**57.92**	**54.17**	**46.22**	**44.90**
10% S1	Random	57.44	54.48	59.63	59.10	60.38	53.25	52.31	51.09
	PointContrast [10]	55.22	51.31	55.62	55.15	59.25	49.17	50.81	49.60
	ProposalContrast [15]	55.59	51.67	55.57	55.12	60.02	49.98	51.18	49.90
	MV-JAR [11]	58.61	55.12	58.92	58.49	63.44	54.40	53.48	**52.47**
	T-MAE (Ours)	**59.24**$^{\uparrow 1.80}$	**57.26**$^{\uparrow 2.78}$	**59.71**	**59.19**	**64.44**	**60.29**	**53.58**	52.31

The data-efficient benchmark enables models to converge with much fewer iterations. Therefore, to evaluate pre-trained representation more efficiently, we conduct experiments on the data-efficient benchmark, as presented in Tab. 1.

Results Analysis. Under the uniform sampling setting, we finetuned the T-MAE pre-trained model with 20% uniformly sampled frames. As shown in Tab. S4, T-MAE outperforms other methods, which aligns with the statement made in the main paper. Under the data-efficient setting, the 5% and 10% splits contain limited samples, which may lead to performance variance. Therefore, except for the split 0 used for Tab. 1, we also compare models in Tab. S5 and Tab. S6 when they are finetuned with split 1 and split 2, both of which are provided by MV-JAR [11] as well. Furthermore, the average results are presented in Tab. S7. In conclusion, it can be observed that the model performance exhibits variation when finetuned with different subsets. However, T-MAE pre-training consistently improves model performance compared to random initialization. T-MAE pre-training surpasses other SSL methods in terms of most class-specific metrics and all overall metrics. Notably, T-MAE outperforms other methods in terms of all metrics while the results are averaged (see Tab. S7).

Table S6: Performance on the Waymo validation set [9]. Results for other methods are taken from MV-JAR [11]. All methods are finetuned with Subset 2.

Finetuning split	Initialization	Overall		Vehicle		Pedestrian		Cyclist	
		mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH
5% S2	Random	44.25	41.48	53.04	52.53	55.16	49.42	24.56	22.48
	PointContrast [10]	44.48	40.55	51.87	51.37	55.36	45.03	26.22	25.24
	ProposalContrast [15]	45.21	41.45	52.29	51.82	56.23	46.28	27.10	26.24
	MV-JAR [11]	47.93	44.50	56.22	55.78	58.80	49.77	28.75	27.95
	T-MAE (Ours)	**49.01**$^{\uparrow 4.75}$	**46.21**$^{\uparrow 4.74}$	**56.50**	**55.95**	**60.70**	**54.70**	**29.82**	**27.98**
10% S2	Random	56.81	53.97	59.49	58.98	62.44	55.54	48.50	47.40
	PointContrast [10]	54.80	51.02	55.41	54.95	60.56	50.86	48.44	47.24
	ProposalContrast [15]	54.77	51.09	55.64	55.20	60.54	51.16	48.14	46.92
	MV-JAR [11]	58.29	54.99	59.17	58.74	64.58	56.02	**51.12**	**50.20**
	T-MAE (Ours)	**58.64**$^{\uparrow 1.83}$	**56.71**$^{\uparrow 2.73}$	**60.20**	**59.70**	**66.10**	**61.79**	49.62	48.63

Table S7: Average performance on the Waymo validation set [9], averaged across the models finetuned with Subset 0∼2. Results for other methods are taken from MV-JAR [11]. T-MAE consistently enhances model performance with a notable margin compared to random initialization.

Finetuning split	Initialization	Overall		Vehicle		Pedestrian		Cyclist	
		mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH
5% S0∼S2	Random	45.84	42.64	53.84	53.30	53.73	46.12	29.95	28.50
	PointContrast [10]	46.26	42.25	52.11	51.61	53.84	43.40	32.82	31.74
	ProposalContrast [15]	47.23	43.28	52.58	52.10	54.61	44.37	34.50	33.38
	MV-JAR [11]	50.39	46.72	56.45	56.00	57.99	48.36	36.74	35.81
	T-MAE (Ours)	**51.31**$^{\uparrow 5.47}$	**49.08**$^{\uparrow 6.44}$	**56.60**	**56.08**	**59.44**	**54.72**	**37.88**	**36.46**
10% S0∼S2	Random	56.77	53.86	59.63	59.12	60.97	53.94	49.70	48.52
	PointContrast [10]	54.57	50.75	55.26	54.80	59.85	50.05	48.61	47.41
	ProposalContrast [15]	54.75	50.96	55.47	55.01	60.19	50.51	48.60	47.37
	MV-JAR [11]	58.12	54.72	58.84	58.41	63.77	55.03	51.74	50.73
	T-MAE (Ours)	**59.27**$^{\uparrow 2.50}$	**57.32**$^{\uparrow 3.46}$	**60.06**	**59.55**	**65.26**	**61.06**	**52.50**	**51.34**

S5 Transferring performance.

To assess the transferability of T-MAE pre-training, we pre-train the model on the ONCE dataset [7] and then fine-tune it on the Waymo dataset [9]. Table S8 demonstrates that T-MAE offers robust generalizability and transferability.

S6 Multi-frame comparison with non-SSL methods.

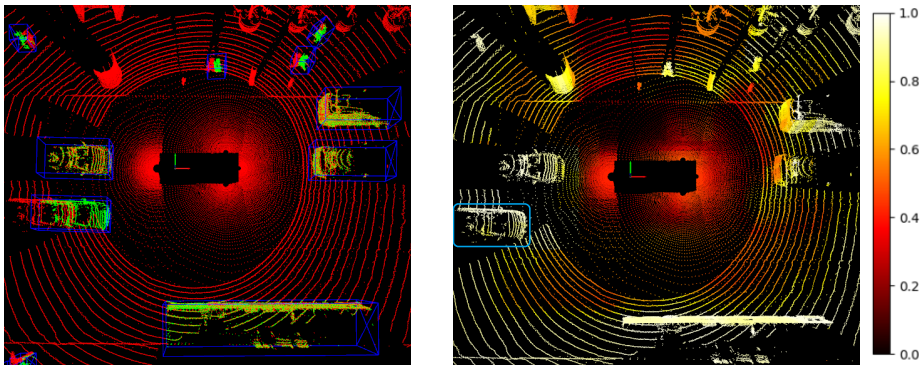
Since there are no existing multi-frame SSL methods, we compare our approach with robust non-SSL baselines that utilize multi-frame as inputs. Table S9 demonstrates our method surpasses other supervised methods while requiring only two frames, highlighting the effectiveness of our method.

Table S8: Transferability comparison.

Initialization	Data	Overall	Vehicle	Pedestrian	Cyclist
Random	-	66.48	66.15	66.76	66.54
GD-MAE [13]	ONCE	66.61	67.18	64.82	67.83
T-MAE (Ours)	ONCE	**67.86**	**68.06**	**67.25**	**68.26**

Table S9: Multi-frame comparison with non-SSL methods. "-" indicates not available.

Method	Frames	L2 Overall		Vehicle		Pedestrian		Cyclist	
		mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH
3D-MAN [14]	16	-	-	67.6	67.1	62.6	59.0	-	-
CenterPoint [16]	4	70.8	69.4	69.1	68.6	71.7	68.6	71.6	**70.9**
SST [1]	3	-	-	68.5	68.1	75.1	70.9	-	-
T-MAE (Ours)	2	**72.3**	**70.5**	**69.4**	**68.9**	**75.8**	**72.0**	**71.8**	70.7



(a) Input (b) Attention scores for prior frame

Fig. S2: Visualization of attention scores. (a) The input consists of two point clouds: the entire previous frame (red points) and the current frame (green points) that contains only points within the ground-truth bounding boxes. Note that, the blue bounding boxes in (a) serve solely for visualization purposes and do not function as input. **(b)** The pillar-wise attention scores are visualized. The attention scores are derived from the WCA module and mapped to a colormap ranging from black to white. The primary attention is placed on the target objects from the previous frame. This implies that the WCA is able to locate corresponding objects. Notably, WCA is capable of accurately trace to the source object, which is manually indicated by a skyblue box, even in cases where the vehicle is moving.

S7 Attention Learned by T-MAE Pre-training

In the main paper, T-MAE pre-trained weights are loaded to both the Siamese encoder and WCA module. We conduct an ablation study where only the Siamese encoder is initialized by the pre-trained weights and the WCA module is randomly intialized. As shown in Tab. S10, initializing WCA with T-MAE pre-trained weights significantly improves mAPH for pedestrians, which indicates a way better direction detection for pedestrians. It also boosts metrics for cyclists with a big margin.

To further understand the knowledge acquired by the WCA module during pre-training, we employ attention visualization to identify critical regions of the previous frame. In particular, the entire previous frame and the target objects of the current frame are input into the SiamWCA that loads T-MAE pre-trained weights. More precisely, the input to the model for the current frame solely

Table S10: Ablation on WCA initialization. SE stands for the Siamese Encoder.

Initialization	Pre-trained		Overall		Vehicle		Pedestrian		Cyclist	
	SE	WCA	mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH
Random	✗	✗	43.68	40.29	54.05	53.50	53.45	44.76	23.54	22.61
Partially random	✓	✗	48.19$^{\uparrow 4.51}$	45.16$^{\uparrow 4.87}$	55.91	55.38	56.29	48.74	32.38	31.36
T-MAE (ours)	✓	✓	**51.47**$^{\uparrow 7.79}$	**49.46**$^{\uparrow 9.17}$	**57.13**	**56.63**	**59.69**	**55.28**	**37.61**	**36.48**

Table S11: Quantitative details on comparison of finetuning iterations. Our scheme requires much less iterations for finetuning compared to MV-JAR [11].

Method	Data amount	Epochs	Number of input scans ($\times 10^3$ per epoch)	Total iterations ($\times 10^5$)	Number of GT used ($\times 10^3$ per epoch)	L2 mAPH (Overall)	L2 mAPH (Pedestrian)
MV-JAR [11]	5%	72	7.9	5.70	7.9	46.68	47.69
	10%	60	15.8	9.50	15.8	54.06	54.66
	20%	48	31.6	15.19	31.6	59.15	59.02
Random/T-MAE	5%	30	7.9	2.37	4.0	40.29/49.46	44.76/55.28
	10%	30	15.8	4.75	7.9	53.13/57.99	53.04/61.10
	20%	30	31.6	9.50	15.8	57.61/61.80	58.41/64.66

consists of points contained within the 3D ground-truth bounding boxes. The purpose of removing other points is to identify the specific areas of emphasis within the WCA module when the queries are solely target objects, *e.g.* vehicles, pedestrians, and cyclists. As illustrated in Fig. S2 (b), the attention is mainly attached to target objects, indicating that the WCA successfully detects and localizes these entities in the previous frame. Furthermore, the presence of vehicular action is perceptible even if the WCA module is only trained with unlabeled data, indicating the effectiveness of our T-MAE pre-training strategy.

S8 Comparison of training iterations

MV-JAR [11] applies varying numbers of epochs during finetuning, depending on the data amount. Rather than varying numbers, we employ a predetermined number of epochs, leading to significantly fewer iterations for finetuning without a performance drop. Detailed information is presented in Tab. S11. The *total number of iterations* is calculated by multiplying *the number of input scans* by *the number of epochs*. As depicted in Fig. 1 and Tab. S11, T-MAE outperforms MV-JAR [11] while requiring a smaller number of finetuning iterations.

S9 Qualitative Results

Fig. S3 shows the qualitative results of our method on the Waymo dataset [9]. Fig. S4 and Fig. S5 illustrates qualitative results on the ONCE datset [7]. Our method generates accurate bounding boxes even if the scene is complex.

S10 Limitations

While our work has achieved encouraging results, there is space for further improvements. For instance, the alignment of two frames relies on their transforma-

tion matrices; without proper alignment, there is a drop in performance. If this could be eliminated, the proposal would be more efficient. Another shortcoming is that the transformer blocks are computationally heavy, which restricts the window size. This leads to a smaller receptive field and thus makes the network easy to lose track of fast-moving objects. The transformer-based encoder also increases the inference time, which was tested on a single NVIDIA RTX 3090 and measured as 429 ms per frame. Consequently, the network is not real-time.

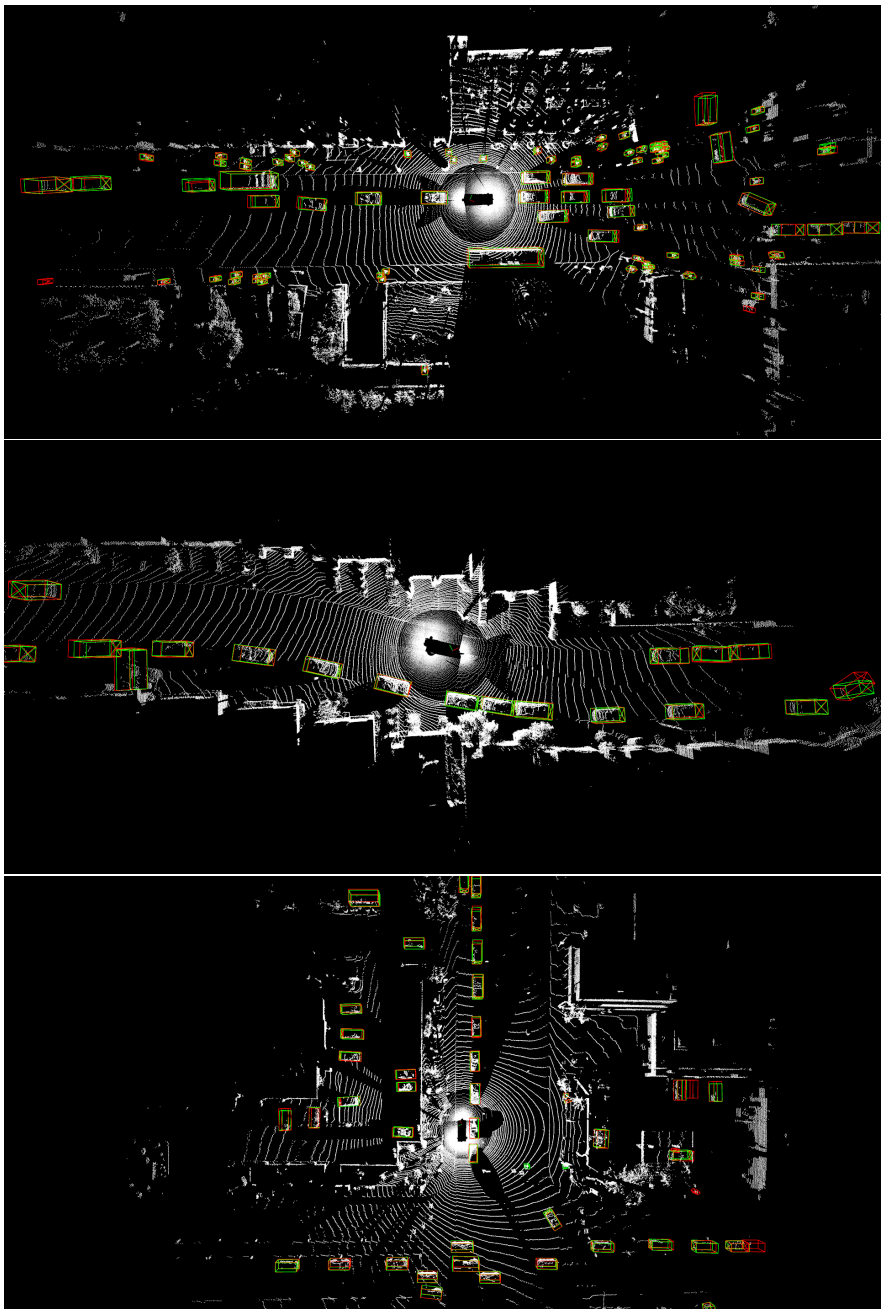


Fig. S3: Qualitative results on the Waymo dataset [9]. We depict ground truth and predictions as boxes colored in red and green for several exemplary scenes.

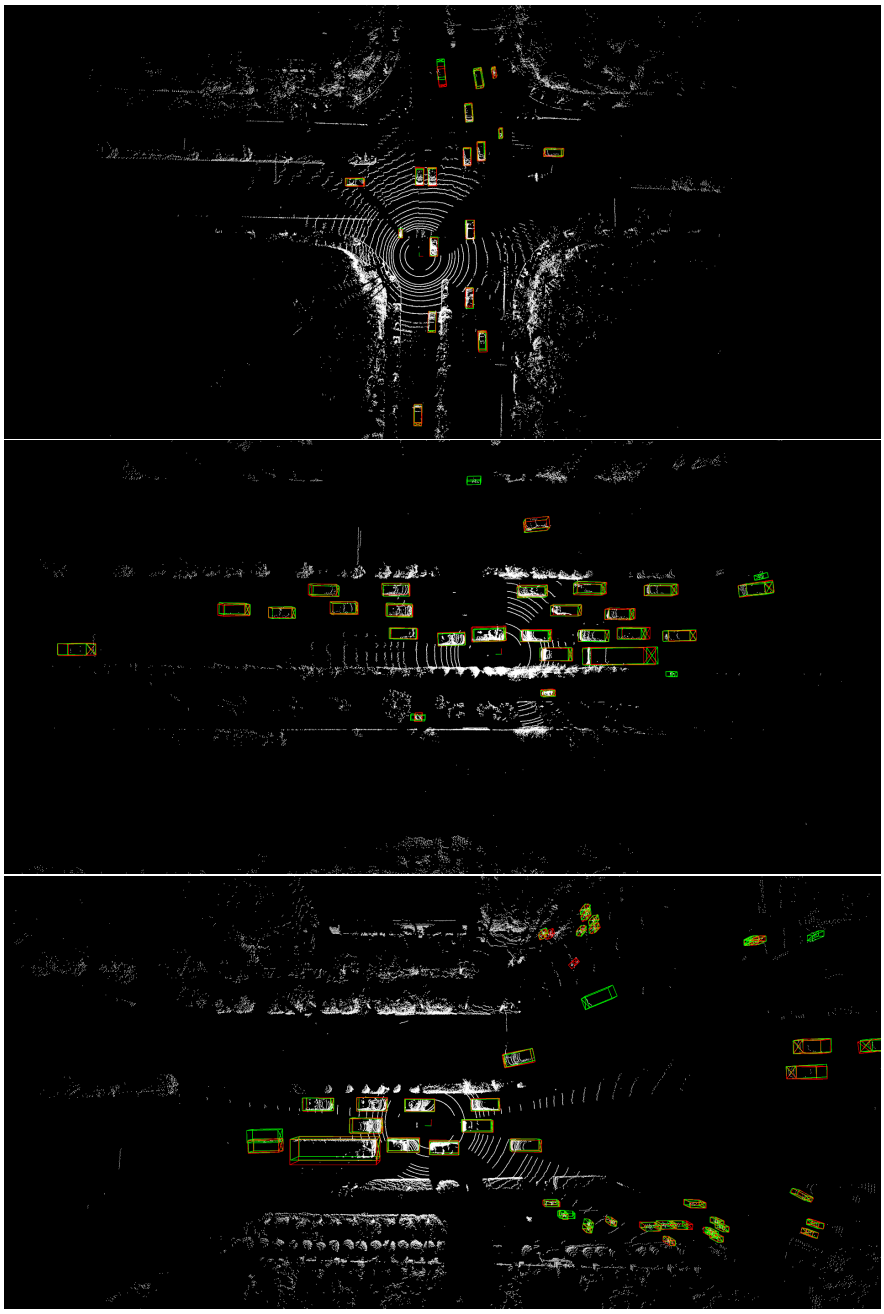


Fig. S4: Qualitative results on the ONCE dataset [7]. We depict ground truth and predictions as boxes colored in red and green for several exemplary scenes.

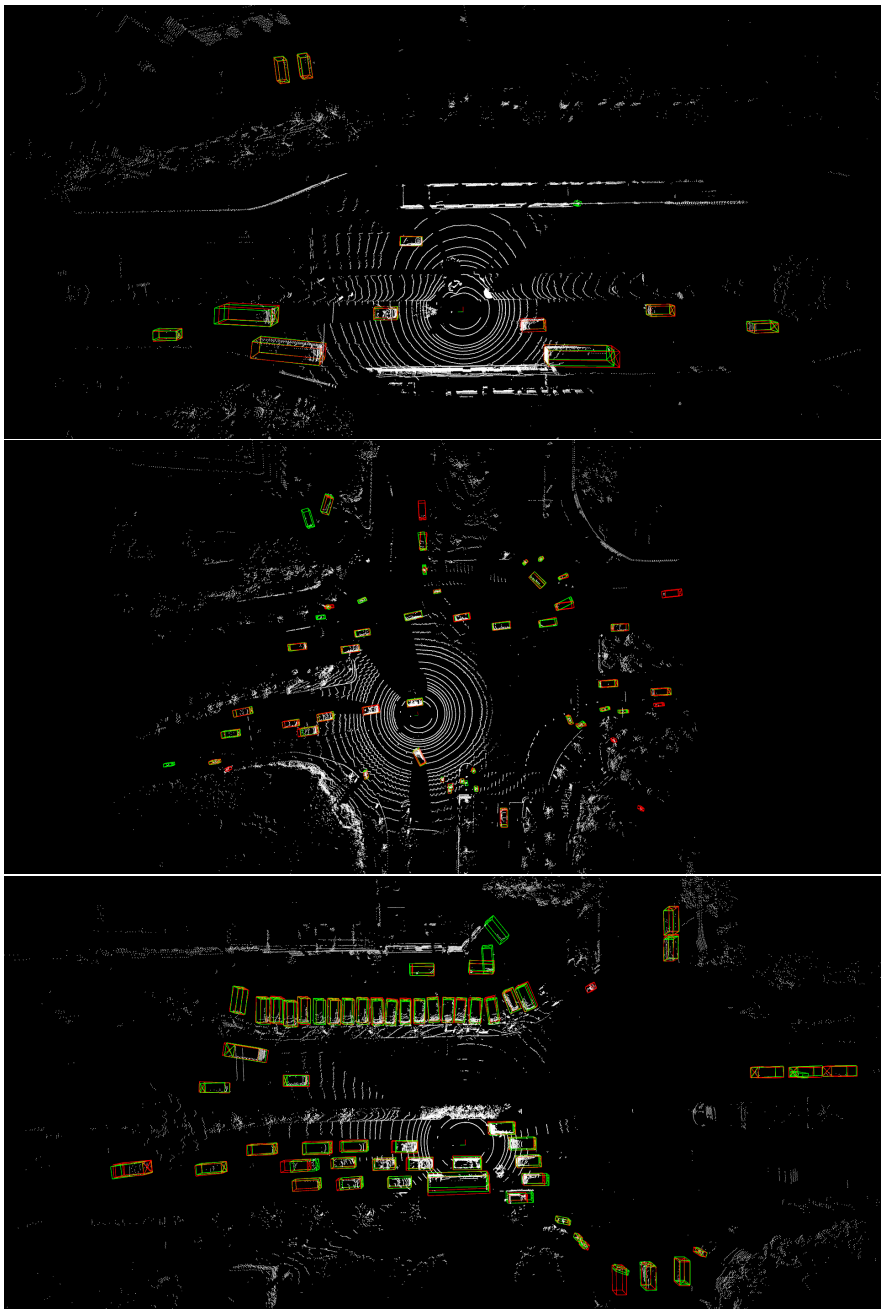


Fig. S5: Qualitative results on the ONCE dataset [7]. We depict ground truth and predictions as boxes colored in red and green for several exemplary scenes.

References

1. Fan, L., Pang, Z., Zhang, T., Wang, Y.X., Zhao, H., Wang, F., Wang, N., Zhang, Z.: Embracing single stride 3d object detector with sparse transformer. In: CVPR (2022)
2. Fei, B., Yang, W., Liu, L., Luo, T., Zhang, R., Li, Y., He, Y.: Self-supervised learning for pre-training 3d point clouds: A survey. arXiv:2305.04691 (2023)
3. Krispel, G., Schinagl, D., Fruhwirth-Reisinger, C., Possegger, H., Bischof, H.: MAELi: Masked autoencoder for large-scale LiDAR point clouds. arXiv preprint arXiv:2212.07207 (2023)
4. Liang, H., Jiang, C., Feng, D., Chen, X., Xu, H., Liang, X., Zhang, W., Li, Z., Van Gool, L.: Exploring geometry-aware contrast and clustering harmonization for self-supervised 3d object detection. In: ICCV. pp. 3273–3282 (2021)
5. Lin, Z., Wang, Y.: BEV-MAE: Bird's eye view masked autoencoders for outdoor point cloud pre-training. In: AAAI (2024)
6. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019)
7. Mao, J., Niu, M., Jiang, C., Liang, H., Chen, J., Liang, X., Li, Y., Ye, C., Zhang, W., Li, Z., Yu, J., Xu, H., Xu, C.: One million scenes for autonomous driving: Once dataset. In: NeurIPS (2021)
8. Min, C., Xu, X., Zhao, D., Xiao, L., Nie, Y., Dai, B.: Occupancy-MAE: Self-supervised pre-training large-scale lidar point clouds with masked occupancy autoencoders. IEEE Transaction on Intelligent Vehicles (2022)
9. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., Vasudevan, V., Han, W., Ngiam, J., Zhao, H., Timofeev, A., Ettinger, S., Krivokon, M., Gao, A., Joshi, A., Zhang, Y., Shlens, J., Chen, Z., Anguelov, D.: Scalability in perception for autonomous driving: Waymo open dataset. In: CVPR. pp. 2443–2451 (2020)
10. Xie, S., Gu, J., Guo, D., Qi, C.R., Guibas, L.J., Litany, O.: Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In: ECCV (2020)
11. Xu, R., Wang, T., Zhang, W., Chen, R., Cao, J., Pang, J., Lin, D.: MV-JAR: Masked voxel jigsaw and reconstruction for LiDAR-based self-supervised pre-training. In: CVPR (2023)
12. Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection. Sensors **18**(10), 3337 (2018)
13. Yang, H., He, T., Liu, J., Chen, H., Wu, B., Lin, B., He, X., Ouyang, W.: GD-MAE: Generative decoder for MAE pre-training on LiDAR point clouds. In: CVPR (2023)
14. Yang, Z., Zhou, Y., Chen, Z., Ngiam, J.: 3d-man: 3d multi-frame attention network for object detection. In: CVPR (2021)
15. Yin, J., Zhou, D., Zhang, L., Fang, J., Xu, C.Z., Shen, J., Wang, W.: Proposalcontrast: Unsupervised pre-training for lidar-based 3d object detection. In: ECCV (2022)
16. Yin, T., Zhou, X., Krähenbühl, P.: Center-based 3d object detection and tracking. In: CVPR (2021)
17. Yuan, J., Zhang, B., Yan, X., Chen, T., Shi, B., Li, Y., Qiao, Y.: Ad-pt: Autonomous driving pre-training with large-scale point cloud dataset. In: NeurIPS (2023)