LetsMap: Unsupervised Representation Learning for Label-Efficient Semantic BEV Mapping

Supplementary Material

In this supplementary material, we present additional experimental results to analyze the performance of LetsMap, our unsupervised representation learning framework for semantic BEV mapping. To this end, we present further ablation experiments in Sec. S.1 and additional qualitative results in Sec. S.2.

S.1 Additional Ablative Experiments

In this section, we present additional ablative experiments to further study the impact of various parameters on the overall performance of the model. Specifically, we study the influence of (1) different DINOv2 variants (Sec. S.1.1) and (2) masking patch size in our novel T-MAE module (Sec. S.1.2) on the overall performance of the model. Further, we also present the results obtained when the native backbones of the best baselines are replaced with our backbone (Sec. S.1.3) and when the BEV percentage split defined in SkyEye [9] is used for model finetuning (Sec. S.1.4).

S.1.1 DINOv2 Backbone Variants

In this section, we study the influence of different variants of the DINOv2 backbone on the overall performance of the model. Specifically, we first pretrain the model using four variants, namely, *vit-b*, *vit-s*, *vit-l*, and *vit-g*, and finetune each of them using 1% of semantic BEV labels. Tab. S.1 presents the results of this ablation study. We observe that *vit-s* yields the lowest performance among all variants, achieving 3.41 pp lower than *vit-b*. Being the smallest of all variants, *vit-s* does not generate features that are as representative as its larger counterparts, thus resulting in its reduced overall performance.

The three larger variants, i.e., *vit-b*, *vit-l*, and *vit-g*, however, yield very similar mIoU scores with the difference between the highest and lowest performance being only 0.80 pp. In other words, the performance of semantic BEV mapping saturates after *vit-b* and does not improve upon using a larger backbone. We infer that this behavior can be attributed to one of the following reasons: (1) larger DINOv2 backbones provide better features for FV tasks, but these features do not easily transfer to the task of BEV mapping, or (2) 1% of BEV labels are insufficient to leverage the full potential of larger backbones. Given the similar performance of *vit-b* as compared to *vit-l* and *vit-g* while being more efficient in terms of number of parameters, we use the *vit-b* variant of the DINOv2 backbone in this work.

Table S.1: Ablation study on the impact of different DINOv2 backbones on the overall model performance. All models in this experiment are finetuned using only 1% of BEV labels. All metrics are reported in [%] on the KITTI-360 dataset.

Backbone	**vit-s**	**vit-b**	**vit-l**	**vit-g**
mIoU	25.55	**28.96**	28.40	28.16

Table S.2: Ablation study on the impact of masking patch size in T-MAE on the overall performance of the model. All models are finetuned using only 1% of labels in BEV. All metrics are reported in [%] on the KITTI-360 dataset.

Patch Size	**Road**	**Side.**	**Build.**	**Terr.**	**Per.**	**2-Wh.**	**Car**	**Truck**	**mIoU**
14	**71.45**	33.41	36.89	37.48	0.75	3.69	30.05	9.23	27.87
28	70.58	**34.26**	**40.68**	**38.53**	1.35	**4.74**	**30.94**	**10.58**	**28.96**
56	70.02	34.10	38.87	37.88	**1.37**	4.71	30.91	9.66	28.44

S.1.2 Masking Patch Size

In this section, we analyze the influence of masking patch sizes used for masking the input image in our novel temporal MAE (T-MAE) module on the overall performance of the model. To this end, we first pretrain the model using masking patches of size 14, 28, and 56, and then finetune the resultant model on 1% of BEV labels. Tab. S.2 presents the results of this ablation study.

We observe that a masking patch size of 28 gives the highest mIoU score across all the evaluated patch sizes. A smaller patch size does not mask out enough of an object and consequently does not present a challenging reconstruction task during the unsupervised pretraining phase. In contrast, a larger patch size masks out significant distinguishing regions in the image which hinders the representation learning ability of the network during the pretraining phase. The effect of patch sizes is noticeable across all classes while being significant for dynamic objects which experience a substantial reduction in the IoU scores when too little of the object is masked out. Given these observations, we use a patch size of 28 in our LetsMap framework.

S.1.3 Impact of DINOv2 on Baseline Approaches

In this section, we analyze the impact of the DINOv2 backbone on the overall performance of the baseline models. Specifically, we replace the native backbones of the two best baselines, PanopticBEV [10] and SkyEye [9], with a pretrained DINOv2 backbone as used in our model. We follow the setting defined in Sec. 4.4 of the main paper and report the results when finetuning with varying percentages of BEV labels in Tab. S.3. We observe that PoBEV reports slightly better performance across all percentage splits when using the DINOv2 backbone with the highest improvement of 1.41 pp observed when using 100% of BEV labels. In contrast, we observe that the performance of SkyEye deteriorates when the native encoder is replaced with the DINOv2 backbone. At lower percentage splits of

1%, 5%, and 10%, the BEV segmentation performance drops by 2.85 pp, 3.76 pp, and 2.62 pp which indicates that the SkyEye framework is unable to adapt the DINOv2 features to this task. The performance drop is observed across all classes and is especially large for *car* and *two-wheeler* which we believe is a consequence of not having an explicit scene geometry estimation module to estimate the extent of objects in the scene. We infer that the native backbone of SkyEye absorbs a significant chunk of scene geometry, but when replaced with a frozen backbone as in our model, SkyEye fails to learn sufficient geometric information. We thus conclude that using our backbone in the baseline approaches results in only a slight improvement in PoBEV and deteriorates the BEV segmentation performance in SkyEye.

Table S.3: Performance of baseline approaches when using the DINOv2 backbone as used in LetsMap. All experiments are on the KITTI-360 dataset.

BEV	Model	FV	PT	Backbone	Road	Side.	Build.	Terr.	Pers.	2-Wh.	Car	Truck	mIoU
1%	PoBEV	✗	-	Native	60.41	20.97	24.65	23.38	0.15	0.23	21.71	1.23	19.09
	SkyEye	✓	✓		69.26	33.48	32.79	**39.46**	0.00	0.34	**32.36**	7.93	26.94
	PoBEV	✗	-	DINOv2	62.36	21.02	27.18	24.22	0.04	0.12	17.50	0.95	19.17
	SkyEye	✓	✓		65.13	29.56	29.02	34.22	0.78	2.87	26.04	5.12	24.09
	LetsMap	✗	✓		**70.58**	**34.26**	**40.68**	38.53	**1.35**	**4.74**	30.94	**10.58**	**28.96**
5%	PoBEV	✗	-	Native	64.45	27.36	30.15	31.66	0.69	0.98	29.75	6.06	23.89
	SkyEye	✓	✓		72.16	37.20	34.89	**42.97**	**4.77**	**9.16**	**40.74**	9.88	31.47
	PoBEV	✗	-	DINOv2	67.61	30.73	30.97	32.80	0.42	0.47	25.48	5.58	24.26
	SkyEye	✓	✓		69.84	34.19	32.80	37.13	2.54	4.74	32.49	7.93	27.71
	LetsMap	✗	✓		**73.74**	**39.56**	**42.07**	41.49	2.46	6.32	34.68	**14.88**	**31.90**
10%	PoBEV	✗	-	Native	66.58	30.28	31.76	34.50	1.22	3.28	33.43	7.56	26.08
	SkyEye	✓	✓		73.36	38.30	37.54	**44.62**	**4.80**	**9.67**	**42.84**	10.06	32.65
	PoBEV	✗	-	DINOv2	68.99	33.17	35.81	34.15	0.70	1.58	29.74	10.06	26.77
	SkyEye	✓	✓		72.19	36.18	35.26	39.84	3.78	5.61	36.95	10.44	30.03
	LetsMap	✗	✓		**74.74**	**39.40**	**43.63**	43.33	2.91	6.95	37.62	**18.09**	**33.33**
50%	PoBEV	✗	-	Native	69.88	33.81	33.40	40.48	2.47	4.63	38.81	9.84	29.16
	SkyEye	✓	✓		73.10	**39.23**	38.08	**45.72**	**4.05**	**10.44**	**44.72**	12.10	**33.43**
	PoBEV	✗	-	DINOv2	73.04	37.38	37.86	41.31	1.82	3.83	37.13	14.85	30.90
	SkyEye	✓	✓		73.66	38.85	41.49	41.73	2.90	6.99	38.43	12.42	32.06
	LetsMap	✗	✓		**74.29**	38.48	**43.87**	42.77	2.80	5.22	37.68	**15.20**	32.54
100%	PoBEV	✗	-	Native	70.14	35.23	34.68	40.72	2.85	5.63	39.77	14.38	30.42
	SkyEye	✓	✓		73.57	**39.45**	38.74	**46.06**	**3.95**	**9.66**	**45.21**	10.92	**33.44**
	PoBEV	✗	-	DINOv2	73.29	37.81	40.23	42.11	1.78	3.32	38.66	**17.42**	31.83
	SkyEye	✓	✓		73.51	39.13	40.04	42.08	3.17	5.90	39.29	12.72	31.98
	LetsMap	✗	✓		**74.81**	38.59	**42.58**	43.67	3.52	6.21	38.47	15.24	32.88

S.1.4 BEV Finetuning using SkyEye Split

In this section, we report the results obtained upon finetuning both the baselines as well as our model with the single random set generated for each BEV percentage split as defined in SkyEye [9]. Please note that all networks are finetuned using the corresponding percent of BEV ground truth labels. Tab. S.4 presents the results

Table S.4: Ablation study on the impact of our unsupervised pretraining on the overall network performance using the finetuning split defined in SkyEye. All experiments are on the KITTI-360 dataset.

BEV	Model	FV	PT	Epochs	Road	Side.	Build.	Terr.	Pers.	2-Wh.	Car	Truck	mIoU
1%	PoBEV	✗	-	100	61.70	17.10	27.81	26.72	0.07	0.36	21.51	0.84	19.51
	SkyEye	✓	✓		72.56	34.33	36.70	**41.66**	0.00	0.16	**33.85**	10.29	28.71
	LetsMap	✗	✗		70.89	33.88	37.71	37.41	0.80	**2.87**	31.59	6.59	27.72
	LetsMap	✗	✓		**72.94**	**37.79**	**43.70**	38.29	**0.87**	2.57	30.62	**10.86**	**29.70**
10%	PoBEV	✗	-	50	70.00	32.75	38.07	34.43	0.80	3.33	34.46	9.25	27.89
	SkyEye	✓	✓		76.07	40.30	40.30	**45.33**	3.75	8.15	**42.64**	10.73	33.41
	LetsMap	✗	✗		**76.69**	40.41	42.55	42.17	1.33	6.57	40.46	18.06	33.53
	LetsMap	✗	✓		74.47	**41.16**	**46.31**	43.31	**5.48**	**8.80**	41.55	**21.24**	**35.29**
50%	PoBEV	✗	-	30	72.09	35.64	36.64	42.41	1.61	3.92	41.41	9.77	30.44
	SkyEye	✓	✓		76.43	39.89	45.22	**46.64**	**5.10**	7.93	**42.43**	12.30	34.49
	LetsMap	✗	✗		75.46	39.45	42.71	39.69	3.85	5.70	41.88	17.82	33.32
	LetsMap	✗	✓		**76.54**	**42.65**	**49.23**	41.47	3.36	**8.61**	38.76	**19.42**	**35.01**

of this study. We observe that our pretraining strategy significantly improves the performance of our model across all three percentage splits with the largest improvement of 1.98 pp observed when using 1% of BEV labels. We also note that LetsMap outperforms SkyEye by 0.99 pp and 1.88 pp when using 1% and 10% which highlights the impact of our approach in low label regimes. Thus, in line with Sec. 4.4 and Tab. 3, we conclude that our novel pretraining strategy positively influences network performance on the BEV segmentation task and results in competitive segmentation performance even in extremely low label regimes.

S.2 Additional Qualitative Results

In this section, we qualitatively evaluate the performance of our model by comparing the semantic BEV maps obtained when the amount of BEV supervision is gradually increased from 1% to 100%. Fig. S.1 presents the results of this evaluation. Fig. S.1(a, b, c, d) present the results on the KITTI-360 dataset and Fig. S.1(e, f, g, h) present the results on the nuScenes dataset.

We observe that the semantic BEV map predictions are largely consistent across all the percentage splits of the two datasets with only minor differences pertaining to the predicted object extents. This behavior is evident in Fig. S.1(d, f) where the model finetuned with 1% of BEV data tends to stretch objects along the radial direction, while models finetuned with higher percentage splits are not significantly affected by this factor. Moreover, we note that the 1% model is able to both detect and localize all the objects in the BEV map to a high degree of accuracy, with only minor errors in the heading of the detected objects (Fig. S.1(c)). Further, we observe in Fig. S.1(a, f, h) that the model finetuned with 1% labels is able to accurately reason about occlusions in the scene, such as the road behind the truck in Fig. S.1(a) and the regions beyond the curve in the road in Fig. S.1(h). This occlusion handling ability stems from the use of an

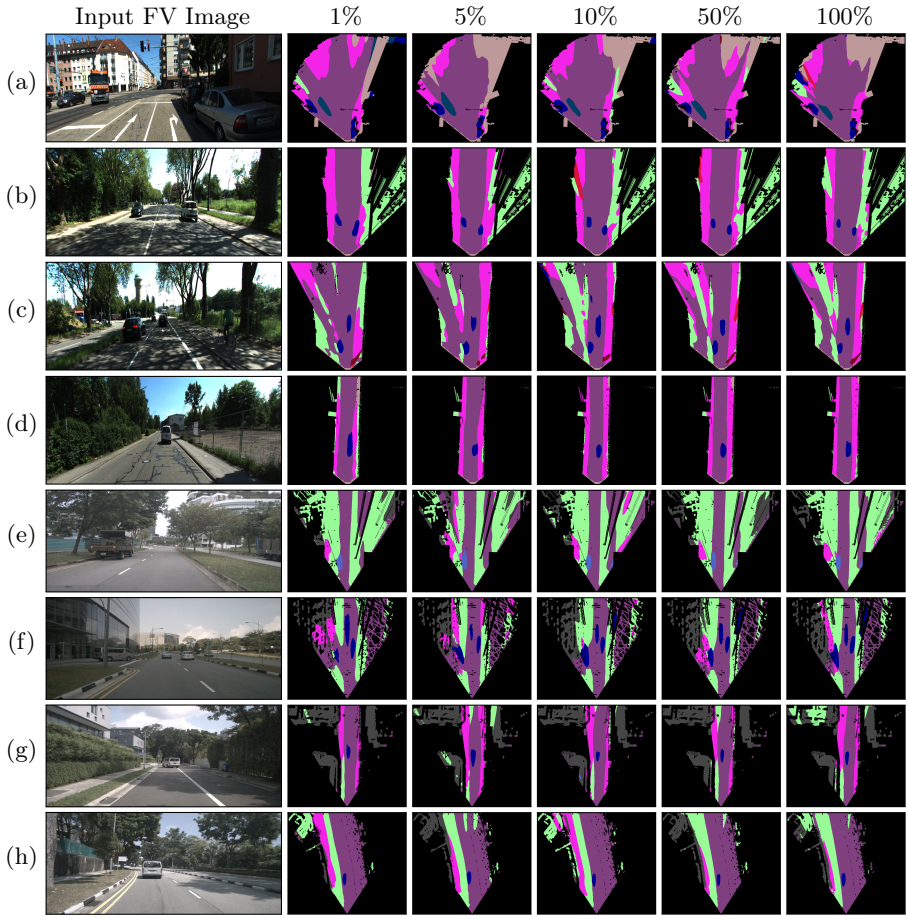

Fig. S.1: Qualitative results obtained when LetsMap is finetuned using 1%, 5%, 10%, 50% and 100% of labels in BEV. Figures (a-d) depict predictions on the KITTI-360 dataset, while figures (e-h) show the predictions on the nuScenes dataset.

independent implicit field-based geometry pathway to reason about the scene geometry in the unsupervised pretraining step. In some cases, however, the scene priors learned during the pretraining step do not generalize well to a given image input. For example, we observe in Fig. S.1(c) that the grass patch next to the vehicle in the adjacent lane is erroneously predicted as a road for the 1% model, while the models finetuned with more than 10% BEV data accurately capture this characteristic. Nonetheless, these observations reinforce the fact that our unsupervised pretraining step encourages the network to learn rich geometric and semantic representations of the scene which allows models finetuned with extremely small BEV percentage splits to generate accurate BEV maps.