Label-Efficient LiDAR Scene Understanding with 2D-3D Vision Transformer Adapters

Julia Hindel*, Rohit Mohan*, Jelena Bratulić, Daniele Cattaneo, Thomas Brox, and Abhinav Valada

***Abstract*— LiDAR semantic segmentation pre-training is hindered by the lack of large, diverse datasets. Moreover, most point cloud segmentation architectures incorporate custom network layers, limiting the transferability of advances from vision-based architectures. Inspired by recent advances in universal foundation models, we propose BALViT, a novel approach that leverages frozen vision foundation models as amodal feature encoders for learning strong LiDAR encoders. Specifically, BALViT incorporates both range-view and bird's-eye-view LiDAR encoding mechanisms, which we combine through 3D positional encoding. While the range-view features are processed through a frozen image backbone, our bird's-eye-view branch enhances them through multiple cross-attention interactions. Thereby, we continuously improve the vision network with domain-dependent knowledge, resulting in a strong LiDAR encoding mechanism with minimal parameter updates. Extensive evaluations of BALViT on the SemanticKITTI and nuScenes benchmarks demonstrate that it outperforms state-of-the-art methods on small data regimes. We make the code and models publicly available at http://balvit.cs.uni-freiburg.de.**

I. Introduction

Self-driving vehicles often rely on LiDAR sensors to semantically perceive their surroundings in various lighting conditions [1]. Recently, self-supervised representation learning has been introduced to pretrain perception models on unlabeled data, boosting performance with minimal labeled fine-tuning. While these techniques excel with images [2], [3] and natural language, the performance improvements are limited for LiDAR data. This is primarily due to the absence of large and diverse pre-training datasets that cover the significant domain shifts between different LiDAR sensors [4], [5]. Consequently, label-efficient techniques that leverage foundation models pre-trained on other modalities are required. Prior work in this direction focuses on extending vision and language foundation models [6], [7] for 3D perception, however these methods still train a randomly initialized LiDAR sub-network and require synchronized camera and LiDAR streams as shown in Fig. 1. We motivate the paradigm of a universal foundation model where only the patch embedding and the decoder are tailored to 3D point clouds [8]–[12]. Consequently, these methods can benefit from strong pre-trained feature extractors and can easily leverage advances in the field of foundation models [8], as the structured 2D grid representation of range view images preserves strong spatial priors that enable effective transfer of visual backbones despite differences from natural images.

* Authors contributed equally to this work.
Department of Computer Science, University of Freiburg, Germany.
This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – SFB 1597 – 499552394.

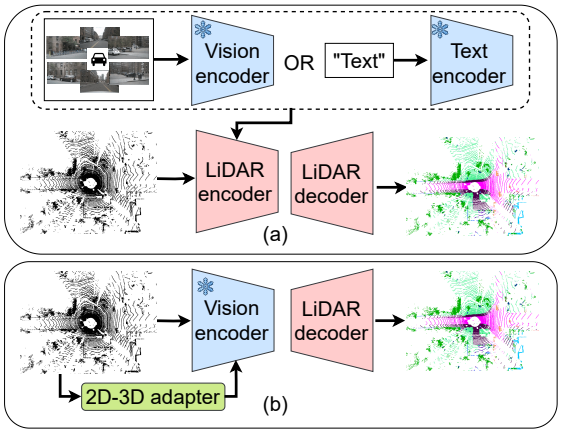

Fig. 1. Framework of different LiDAR semantic segmentation models. Learned modules are colored in red and frozen components in blue. (a) Vision or language models are employed to distill knowledge into tailored LiDAR architectures. (b) Transferring a pre-trained vision model into the LiDAR domain using a 2D-3D adapter (ours).

In this paper, we introduce BALViT (**B**ird-Eye-View **A**dapted **L**iDAR **ViT**), which seamlessly transforms vision foundation models with a novel 2D-3D adapter for label-efficient LiDAR semantic segmentation. In particular, we extend the vision model-based network RangeViT [8] with a lightweight, polar Bird's-Eye View (BEV)-based adapter. We reason that vision models excel at detecting object shapes, whereas our adapter incorporates geometric reasoning. Finally, we incorporate a separate BEV decoder branch to address and correct misclassifications, ensuring the model generates accurate 3D semantic segmentation by refining the output through complementary branches.

Our main contributions can be summarized as follows: 1) BALViT, a LiDAR semantic segmentation architecture that leverages pre-trained vision foundation models as a backbone. 2) a 2D-3D adapter for label-efficient refinement of vision models for sparse 3D segmentation. 3) Extensive evaluations on six label-efficient training settings. 4) Pre-trained models at http://balvit.cs.uni-freiburg.de.

II. Related Work

LiDAR Semantic Scene Segmentation networks leverage voxel [13], point [14] or range-view [8], [15] representations. However, voxel- and point-based approaches are resource-intensive and leverage custom architectures that are unaffected by advances in 2D vision research. Conversely, range-view networks convert point clouds into 2D projections, which facilitates use of standard ViT backbones [8]. We propose a 2D-3D adapter to enhance the geometric prediction

capabilities of pre-trained vision models for label-efficient LiDAR semantic segmentation.

Parameter-Efficient Fine-tuning adapts foundation models to a specific task with minimal parameter updates. A common practice is updating only bias terms [16]. Further, visual prompt tuning (VPT) [16] inserts learnable prompts while LoRA [17] focuses on attaching trainable rank decomposition matrices to frozen transformer blocks. For dense prediction tasks, ViT-Adapter [18] introduces a separate stem that interacts with the backbone via cross-attention. For dense 3D point clouds, various adapters target the transformation of 3D pre-trained models [9]–[12]. However, prior work shows that vision or language foundation models paired with 3D-adaptation mechanisms can surpass 3D pre-training for single-object segmentation [19]–[22]. Driven by this approach, we propose BALViT which is the first 2D-3D adapter for sparse, outdoor point cloud processing.

III. Technical Approach

Our proposed BALViT is tailored for adapting pre-trained vision transformer (ViT) backbones for LiDAR semantic segmentation. We argue that these architectures enable amodal feature encoding, which we enhance with our label-efficient 2D-3D adapter. Specifically, we first encode the point cloud in two separate branches, a range-view (RV) and a BEV encoder. Next, the RV features are processed by a frozen ViT backbone. During this backbone traversal, we continuously enhance the RV features with BEV features using our novel 2D-3D adapter. Finally, we decode each feature branch with separate 3D decoders to combine the strengths of both views, effectively reducing misclassifications. Fig. 2 provides an overview of our architecture.

A. Range-View Encoding

The network input is a LiDAR point cloud $P \in \mathbb{R}^{N\times 4}$ composed of N points, each with four values (x, y, z, i). The variables (x, y, z) are Cartesian coordinates, and i represents the returned LiDAR beam intensity. To convert the LiDAR point cloud into a range projection of size $H \times W$, we first compute the pixel position for each point $p_j \in P$ as follows:

$$\begin{bmatrix} h_j \\ w_j \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\left(1 - \arctan(y_j, x_j)\frac{1}{\pi}\right) W \\ \left(1 - \left(\arcsin(z_j, \frac{1}{r_j}) - \frac{f_{\text{down}}}{f_v}\right)\right) H \end{bmatrix}, \quad (1)$$

where $r_j = \sqrt{x_j^2 + y_j^2 + z_j^2}$ corresponds to the range of the point p_j and $f_v = f_{\text{up}} - f_{\text{down}}$ being the vertical view of the LiDAR sensor [8]. We then construct the RV image I^{RV} according to $I^{RV}_{h_j,w_j} = [r_j, z_j, i_j]$. We argue that range-views are best suited for direct processing with a pre-trained ViT backbone since this representation most closely resembles camera images. To encode images, a ViT backbone [23] commonly applies a single linear projection. To bridge the domain gap between camera images and I^{RV}, we replace this layer with our Conv PatchEmbed block that is composed of four residual blocks inspired by SalsaNext [24]. We later refer to features after this operation as RV_{stem}. We employ average pooling with a kernel size of (PH_{RV},PW_{RV}) and a final 1×1 convolution to convert RV_{stem} to D_{RV} channels. Finally, we add a classification token and 2D positional embedding $E_{2D} \in \mathbb{R}^{(M+1)\times D_{RV}}$ to transfer the flattened M features into a suitable format for further encoding by a ViT backbone [8].

B. Polar Bird-Eye View Encoding

We complement the RV features with an orthogonal lightweight BEV branch. Specifically, we use a polar BEV representation I^{BEV} with grid cell dimension H_{BEV}, W_{BEV}, and Z_{BEV}. We first convert the (x_j, y_j, z_j) coordinates of each point p_j into polar coordinates according to:

$$\begin{bmatrix} \rho_j \\ \phi_j \\ \theta_j \end{bmatrix} = \begin{bmatrix} \sqrt{x_j^2 + y_j^2} \\ -\arctan 2(y_j, -x_j) \\ \frac{(z_j - f_{down})Z_{BEV}}{f_v} \end{bmatrix}. \quad (2)$$

Then, we randomly select N_p points per BEV grid cell and leverage a PointNet-inspired encoder [25] consisting of 4 blocks of fully-connected layers to compute a pointwise embedding of dimension D_{BEV}. Next, we perform max pooling over the Z_{BEV}-dimension to obtain BEV features of size (D_{BEV}, H_{BEV}, W_{BEV}). We create multi-scale BEV features with our spatial prior module to enhance context capture. This module consists of a ResNet-inspired stem and three convolutional blocks to transform the acquired BEV features into multi-scale representations $c1$, $c2$, $c3$, $c4$ with a channel dimension of D_{RV}. Subsequently, we add a 2D learnable positional embedding to maintain the scale correlations when flattening the features.

C. Adapter Module

Before combining the two encoder branches, we add 3D positional embeddings $E_{3D} \in \mathbb{R}^{(M+1)\times D_{RV}}$ to both feature maps. Our sinusoidal positional embedding is computed separately in x, y, and z dimensions following the original transformer's positional encoding [26]. Then, we concatenate the component-wise embeddings and upscale the resulting vector with two 1×1 convolutions to obtain a positional embedding of dimension D_{RV}. We leverage the transformations in Eq. (1) and Eq. (2) to extract the Cartesian coordinates of each feature in the two views, RV and BEV. We then use these geometric positions to infer their positional embeddings and add them to the respective feature maps. Our 3D positional embedding ensures that the feature map interactions account for the spatial geometries of the 3D scene.

The RV features are further encoded through subsequent blocks of the frozen ViT backbone. We perform feature integration of our two branches (RV and BEV) at different layers of the ViT backbone using our tailored interaction modules. Each interaction entails two parallel injector modules where one injector module (INJ) updates the features of one network branch (X) by performing sparse cross-attention with the other branch (Y) following Eq. (3). The second injector performs the same operation but with reversed branches.

$$F_i^{\text{X}} = F_i^{\text{X}} + \gamma_i \text{Attention}(\text{norm}(F_i^{\text{X}}), \text{norm}(F_i^{\text{Y}})). \quad (3)$$

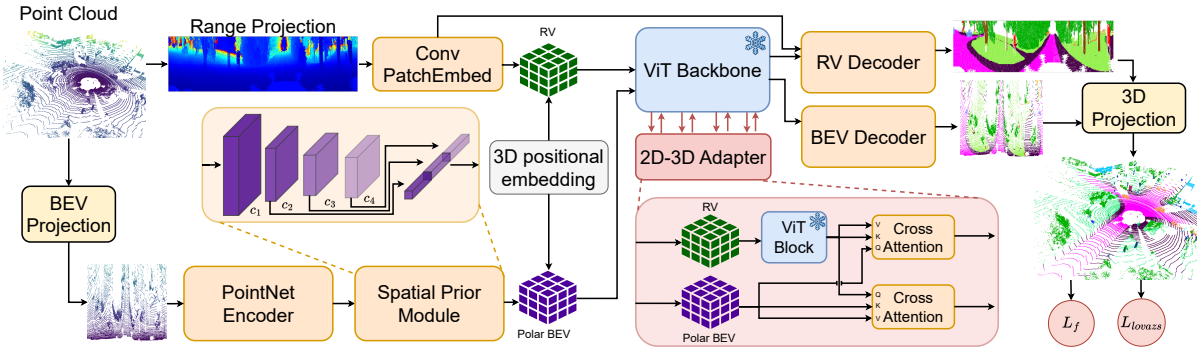

Fig. 2. Our network BALViT encodes a point cloud in orthogonal range-view (RV) and bird-eye-view (BEV) branches. Our spatial prior module converts the BEV branch into multi-scale features, which interact with the RV branch during its traversal of the frozen ViT backbone. Last, our two decoders independently obtain pointwise class labels from the respective feature maps.

Our parallel stacked cross-attentions enhance the RV features from the frozen ViT backbone branch with domain-specific knowledge from the BEV branch (X=RV, Y=BEV), while the BEV features are refined with novel context from the RV features (X=BEV, Y=RV). We repeat these interactions after layers 5, 11, 17, and 23 of the ViT backbone. Since the backbone is frozen, only the weights of the interaction modules are updated, making our method parameter-efficient.

D. Decoder

We attach separate LiDAR decoders to the RV and BEV network branches to predict semantic segmentation on each view independently. For the RV branch, we use the decoder proposed by RangeViT [8], which is composed of a convolutional decoder, a PixelShuffle layer, and a 3D refiner. For the BEV branch, we propose a progressive decoder to combine the multi-scale BEV feature maps. First, we aggregate features maps c_1, c_2, c_3, and c_4 to obtain combined features $c_{1:4}$ using UpConv blocks and skip connections. Last, we upsample the feature map to the original BEV input resolution using PixelShuffle [27]. We then use three blocks of convolutions to recover the Z_{BEV}-dimension and predict a semantic class for every BEV cell grid. Finally, we employ a grid sampler to reproject our polar BEV predictions from polar (S, W_{BEV}, H_{BEV}, Z_{BEV}) to 3D coordinates (x, y, z) as detailed in Sec. III-B, where S is the number of classes.

We train each decoder branch separately using using multi-class Focal [28] and Lovász-Softmax [29] losses. During inference, we merge pointwise RV and BEV predictions based on their highest logits.

IV. Experimental Evaluation

In this section, we quantitatively and qualitatively evaluate the performance of BALViT on LiDAR semantic segmentation and highlight our contribution in an ablation study. We evaluated our method on two benchmark autonomous driving datasets, SemanticKITTI and nuScenes. SemanticKITTI [30] comprises 19,130 training and 4,071 validation scans captured using a 64-beam LiDAR, annotated point-wise for 19 semantic classes. On the other hand, nuScenes [31] includes 28,130 training and 6,019 validation scans recorded with a 32-beam LiDAR, with point-wise annotations for 16 semantic classes.

TABLE I

Label-efficient training results on 0.1%, 1%, 10% of the SemanticKITTI and nuScenes datasets.

	Method	mIoU [%]					
		SemanticKITTI			**nuScenes**		
		0.1%	1 %	10%	0.1%	1 %	10%
FS	SR-Unet18 [32]	-	39.50	-	-	30.30	56.15
FS	FRNet [15]	30.09	40.78	61.55	28.03	48.98	69.99
FS	SphereFormer [13]	29.21	42.81	58.81	30.42	50.06	69.25
FS	RangeViT [8]	28.74	43.53	58.53	27.79	52.88	**71.84**
VD	SLidR [33]	-	44.60	-	-	38.30	59.84
VD	ST-SLidR [34]	-	44.72	-	-	40.75	60.75
VD	SEAL [35]	-	46.63	-	-	45.84	62.97
VD	CLIP2Scene [36]	-	42.60	-	-	56.30	-
PEFT	Frozen ViT backbone	29.97	45.91	58.10	28.72	54.70	63.28
PEFT	Bias tuning	30.86	45.63	58.14	28.15	56.05	65.08
PEFT	LoRA [17]	31.65	46.27	59.53	28.27	57.57	66.38
PEFT	VPT [16]	31.07	46.08	58.38	29.68	55.67	65.52
PEFT	Vit Adapter [18]	29.55	45.01	57.43	27.50	56.06	67.71
PEFT	BALViT (Ours)	**32.85**	**51.80**	**61.91**	**31.86**	**59.27**	70.13

The results are reported on the val set, and all metrics are in [%]. **FS**: fully-supervised methods. **VD**: vision distillation. **PEFT**: parameter-efficient fine-tuning.

We use a Cityscapes pre-trained ViT-S backbone. Refer supplementary material for additional details.

A. Quantitative Results

We compare BALViT with four fully supervised methods and four vision model distillation schemes that leverage self-supervised pretraining on the entire nuScenes dataset. We select the fully supervised models based on comparable size and top performance on SemanticKITTI 100%, using published code and the augmentations described in Sec. S.0.A. For the vision model distillation schemes, we report their published performance. We also compare against five parameter-efficient fine-tuning methods that we integrate into the RangeViT architecture [8], using the same training configurations as our approach. All models are evaluated with the mean intersection-over-union (mIoU) metric.

Results on SemanticKITTI and nuScenes are presented in Tab. I. Notably, BALViT outperforms all supervised and self-supervised baselines by at least 1.44pp on the 0.1% and 1% settings. This can be attributed to the strong vision priors from the pretrained ViT, which are effectively enhanced by

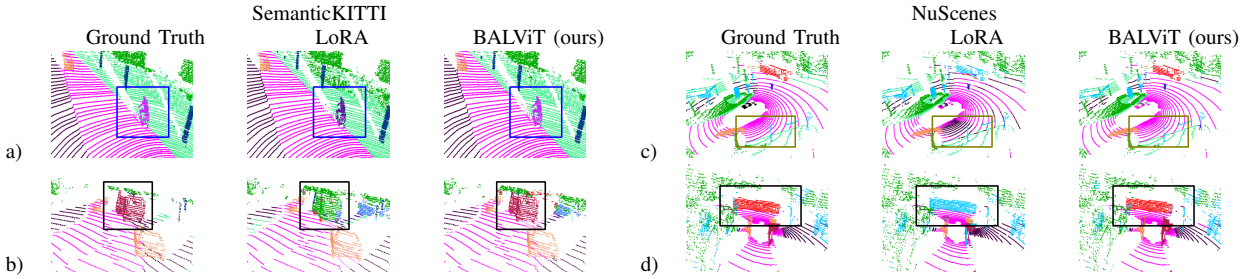

Fig. 3. Qualitative results of BALViT on LiDAR semantic segmentation on SemanticKITTI and nuScenes.

TABLE II

ABLATION STUDY ON VARIOUS COMPONENTS OF BALVIT.

SemanticKITTI mIoU (%)				
2D-3D Adapter	3D PE	SPM	BEV decoder	1%
				45.91
✓				48.53
✓	✓			49.06
✓	✓	✓		49.70
✓	✓	✓	✓	**51.80**

3D PE: 3D positional embedding, SPM: Spatial Prior Module.

TABLE III

COMPARISON WITH DIFFERENT VISION BACKBONE INTITIALIZATION.

SemanticKITTI mIoU (%)	
Backbone initialization	1 %
Supervised random init.†	44.58
Depth Anything [38]	45.11
Dino [39]	46.94
Dino v2 [6]	47.65
MoCo v3 [37]	48.47
Supervised ImageNet [40]	48.68
Cityscapes [41]	**51.80**

†: vision backbone is updated during training.

our novel 2D-3D adapter. However, at the 10% settings, the performance gap narrows, indicating that LiDAR-specific transformers have strong encoding capabilities that require more labeled data. In the last block of Tab. I, we compare parameter-efficient methods on the same base architecture as our method. We find that parameter-efficient fine-tuning is preferred to full-finetuning (RangeViT) in all scenarios. We conclude that 2D pre-trained networks provide strong priors for 3D segmentation, benefiting more from enhancement than from retraining. BALViT outperforms other parameter-efficient fine-tuning methods by at least 1.80pp on the 1% and 10% training settings. Our method of combining RV and BEV features within a frozen ViT model adds spatial knowledge in the feature encoding process, which outperforms standard parameter-efficient vision approaches.

B. Ablation Study and Qualitative Results

In this section, we study the impact of various network components of our approach. We perform all ablation experiments on the 1% SemanticKITTI setting.

1) Influence Network Components: In Tab. II, we incrementally integrate our network components into the Frozen ViT backbone configuration, observing cumulative performance gains at each step. First, adding the 2D-3D adapter with unaligned single-scale improves performance by 2.62pp, enhancing RV features via the BEV feature branch. Aligning RV and BEV features with our 3D positional embedding adds 0.53pp, likely due to better cross-attention between coordinated features. Mutli-scale BEV features further improve performance by 0.63pp, and lastly incorporating the BEV decoder yields an additional gain of 2.1pp.

2) Influence of Pre-trained Vision Backbone: We assess the impact of different pretrained vision backbones on our network's performance in Tab. III. A randomly initialized ViT yields the lowest performance, emphasizing the need for strong vision models in our network. While Dino v2 [6] and MoCo v3 [37] pretrainings result in superior performance on 2D vision benchmarks, we observe the highest score when leveraging a network pre-trained on a similar domain (i.e. Cityscapes dataset). Consequently, we reason that popular foundation models are biased towards their training modality and lack amodal feature representations.

3) Qualitative Results: We qualitatively compare BALViT with the best-performing parameter-efficient method LoRA [17] in Fig. 3, confirming our BALViT's superior segmentation performance. Our method produces more coherent predictions with fewer misclassifications and patchified regions. Unlike LoRA, which misclassifies entire objects in a, BALViT correctly labels them using independent decoder branches. It also better segments larger objects in b and d due to improved localization. On nuScenes in c, BALViT refines class boundaries (sidewalk, road) by leveraging multi-scale BEV features for finer boundary delineation. These results highlight BALViT's stronger semantic understanding in low-label settings.

V. CONCLUSION

We present BALViT, a novel method for LiDAR semantic segmentation tailored for small data regimes. Our approach encodes a RV projection with a frozen ViT backbone which we enhance with our 2D-3D adapter. Subsequently, we merge the predictions of our RV and BEV decoders for improved performance. We observe that our approach outperforms existing state-of-the-art supervised and self-supervised baselines on label-efficient training settings of 0.1% and 1% on the SemanticKITTI and nuScenes datasets. The proposed method is one of the early works to show that 2D vision foundation models provide valuable priors for LiDAR semantic scene segmentation. Consequently, we motivate future work to focus on efficient 2D-3D adaption mechanism and enhancing foundation models with amodal learning capabilities.

REFERENCES

[1] R. Mohan, D. Cattaneo, F. Drews, and A. Valada, "Progressive multi-modal fusion for robust 3d object detection," in *Annual Conf. on Robot Learning*, 2024.

[2] C. Lang, A. Braun, L. Schillingmann, K. Haug, and A. Valada, "Self-supervised representation learning from temporal ordering of automated driving sequences," *IEEE Robotics and Automation Letters*, vol. 9, no. 3, pp. 2582–2589, 2024.

[3] J. Hindel, N. Gosala, K. Bregler, and A. Valada, "Inod: Injected noise discriminator for self-supervised representation learning in agricultural fields," *IEEE Robotics and Automation Letters (RA-L)*, 2023.

[4] J. Sanchez, J.-E. Deschaud, and F. Goulette, "3dlabelprop: Geometric-driven domain generalization for lidar semantic segmentation in autonomous driving," *arXiv preprint arXiv:2501.14605*, 2025.

[5] A. Xiao, X. Zhang, L. Shao, and S. Lu, "A survey of label-efficient deep learning for 3d point clouds," in *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 46, no. 12, 2024, pp. 9139–9160.

[6] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. HAZIZA, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "DINOv2: Learning robust visual features without supervision," *Transactions on Machine Learning Research*, 2024.

[7] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*, 2021.

[8] A. Ando, S. Gidaris, A. Bursuc, G. Puy, A. Boulch, and R. Marlet, "Rangevit: Towards vision transformers for 3d semantic segmentation in autonomous driving," *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 5240–5250, 2023.

[9] D. Liang, T. Feng, X. Zhou, Y. Zhang, Z. Zou, and X. Bai, "Parameter-efficient fine-tuning in spectral domain for point cloud learning," *arXiv preprint arXiv:2410.08114*, 2024.

[10] X. Zhou, D. Liang, W. Xu, X. Zhu, Y. Xu, Z. Zou, and X. Bai, "Dynamic adapter meets prompt tuning: Parameter-efficient transfer learning for point cloud analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2024, pp. 14 707–14 717.

[11] Y. Zha, J. Wang, T. Dai, B. Chen, Z. Wang, and S.-T. Xia, "Instance-aware dynamic prompt tuning for pre-trained point cloud models," in *Proc. Int. Conf. Comput. Vis.*, 2023, pp. 14 115–14 124.

[12] B. Fei, L. Liu, W. Yang, Z. Li, W.-M. Chen, and L. Ma, "Parameter efficient point cloud prompt tuning for unified point cloud understanding," *IEEE Transactions on Intelligent Vehicles*, pp. 1–16, 2024.

[13] X. Lai, Y. Chen, F. Lu, J. Liu, and J. Jia, "Spherical transformer for lidar-based 3d recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2023.

[14] X. Wu, L. Jiang, P.-S. Wang, Z. Liu, X. Liu, Y. Qiao, W. Ouyang, T. He, and H. Zhao, "Point transformer v3: Simpler faster stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2024, pp. 4840–4851.

[15] X. Xu, L. Kong, H. Shuai, and Q. Liu, "Frnet: Frustum-range networks for scalable lidar segmentation," *arXiv preprint arXiv:2312.04484*, 2023.

[16] M. Jia, L. Tang, B.-C. Chen, C. Cardie, S. Belongie, B. Hariharan, and S.-N. Lim, "Visual prompt tuning," in *European Conference on Computer Vision*, 2022, pp. 709–727.

[17] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *Int. Conf. Learn. Represent.*, 2022.

[18] Z. Chen, Y. Duan, W. Wang, J. He, T. Lu, J. Dai, and Y. Qiao, "Vision transformer adapter for dense predictions," in *Int. Conf. Learn. Represent.*, 2023.

[19] Y. Tang, R. Zhang, J. Liu, Z. Guo, B. Zhao, Z. Wang, P. Gao, H. Li, D. Wang, and X. Li, "Any2point: Empowering any-modality large models for efficient 3d understanding," in *Proc. Springer Eur. Conf. Comput. Vis.*, 2022, pp. 456–473.

[20] M. Li, D. Li, G. Yang, Y.-m. Cheung, and H. Huang, "Adapt pointformer: 3d point cloud analysis via adapting 2d visual transformers," *arXiv preprint arXiv:2407.13200*, 2024.

[21] D. Hegde, J. M. J. Valanarasu, and V. M. Patel, "Clip goes 3d: Leveraging prompt tuning for language grounded 3d recognition," *arXiv preprint arXiv:2303.11313*, 2023.

[22] X. Huang, Z. Huang, S. Li, W. Qu, T. He, Y. Hou, Y. Zuo, and W. Ouyang, "Frozen clip transformer is an efficient point cloud encoder," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 3, 2024, pp. 2382–2390.

[23] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *Int. Conf. Learn. Represent.*, 2021.

[24] T. Cortinhal, G. Tzelepis, and E. Erdal Aksoy, "Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds," in *Int. Symp. on Advances in Visual Computing*, 2020, pp. 207–222.

[25] C. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 77–85, 2016.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Adv. Neural Inform. Process. Syst.*, vol. 30, 2017.

[27] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 1874–1883.

[28] T.-Y. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *Proc. Int. Conf. Comput. Vis.*, 2017.

[29] M. Berman, A. Rannen Triki, and M. B. Blaschko, "The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 4413–4421.

[30] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences," in *Proc. Int. Conf. Comput. Vis.*, 2019.

[31] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 11 618–11 628, 2019.

[32] C. B. Choy, J. Gwak, and S. Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks," *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 3070–3079, 2019.

[33] C. Sautier, G. Puy, S. Gidaris, A. Boulch, A. Bursuc, and R. Marlet, "Image-to-lidar self-supervised distillation for autonomous driving data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

[34] A. Mahmoud, J. S. K. Hu, T. Kuai, A. Harakeh, L. Paull, and S. L. Waslander, "Self-supervised image-to-point distillation via semantically tolerant contrastive loss," *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 7102–7110, 2023.

[35] Y. Liu, L. Kong, J. CEN, R. Chen, W. Zhang, L. Pan, K. Chen, and Z. Liu, "Segment any point cloud sequences by distilling vision foundation models," in *Adv. Neural Inform. Process. Syst.*, 2023.

[36] R. Chen, Y. Liu, L. Kong, X. Zhu, Y. Ma, Y. Li, Y. Hou, Y. Qiao, and W. Wang, "Clip2scene: Towards label-efficient 3d scene understanding by clip," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2023.

[37] X. Chen*, S. Xie*, and K. He, "An empirical study of training self-supervised vision transformers," *arXiv preprint arXiv:2104.02057*, 2021.

[38] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao, "Depth anything: Unleashing the power of large-scale unlabeled data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2024.

[39] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proc. Int. Conf. Comput. Vis.*, 2021.

[40] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.* Ieee, 2009, pp. 248–255.

[41] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

Label-Efficient LiDAR Scene Understanding with 2D-3D Vision Transformer Adapters

- Supplementary Material -

Julia Hindel*, Rohit Mohan*, Jelena Bratulić, Daniele Cattaneo, Thomas Brox, and Abhinav Valada

A. Implementation Details

We use RV projections of size $(64, 2048)$ with the patch size being defined as $PW_{rv} = 8$ and $PH_{rv} = 2$. We set the dimensionality of our RV features to $D_{rv} = 384$. Our polar BEV representation contains $480 \times 360 \times 32$ grid cells for the full point cloud, with each feature having $D_{bev} = 512$ dimensions. We train our networks and baselines for 100, 500, and 1000 epochs with dataset splits of 0.1%, 1%, and 10%, respectively. We use a batch size of 4, a learning rate of 0.0004, and a cosine learning rate scheduler with 20 warm-up epochs. Further, we take random RV crops of size (64, 384) for SemanticKITTI and (64, 768) for nuScenes. The wider crop for nuScenes accounts for its lower horizontal resolution and overall sparser lidar compared to SemanticKITTI. Consequently, we equally subsample our BEV features, resulting in polar grid sizes of $480 \times 68 \times 32$ and $480 \times 135 \times 32$, respectively. We augment the point clouds with random horizontal flips with a probability of 0.5 and scale the point cloud in ranges of [0.95-1.05]. Further, we randomly resample rare class points. For nuScenes, we additionally randomly rotate in all three axes in the range of -5 and 5 degrees with a 50% likelihood.

B. Inference

During inference, we merge pointwise RV and BEV predictions according to a predefined threshold s:

$$\text{Output} = \begin{cases} \hat{y}_{\text{RV}}, & \text{if } \hat{y}_{\text{RV}} > s \\ \hat{y}_{\text{RV}} & \text{if } \hat{y}_{\text{RV}} \leq s \text{ and } \hat{y}_{\text{BEV}} < s \\ \hat{y}_{\text{BEV}}, & \text{if } \hat{y}_{\text{RV}} \leq s \text{ and } \hat{y}_{\text{BEV}} > s \end{cases}, \quad (1)$$

where $\hat{y}_{\text{RV}}$ and $\hat{y}_{\text{BEV}}$ are the highest logits of the RV and BEV predictions, respectively. We set the inference threshold s to 0.9 in all experiments. The fusion leverages the complementary strengths of RV and BEV views. RV generally provides stronger fine-grained local details, while BEV offers a more globally consistent geometric context that is more effective for larger spatial reasoning. By selecting predictions with high confidence, the mechanism acts as a simple yet effective uncertainty-aware selector, allowing the model to favor the view with stronger evidence at each point. With this merging strategy, we can correct miss-predictions in the RV feature branch, which results in a further boost in performance as presented in Sec. IV-B.

Department of Computer Science, University of Freiburg, Germany.

TABLE S.1

ABLATION STUDY ON VARYING SEQUENCES OF INJECTOR (INJ) ON 1% SEMANTICKITTI TRAINING SETTING.

SemanticKITTI mIoU (%)	
Injector strategy	1 %
$\text{INJ}_{bev,rv}$-ViTBlock-$\text{INJ}_{rv,bev}$	46.17
$\text{INJ}_{rv,bev}$-ViTBlock-$\text{INJ}_{bev,rv}$	48.42
ViTBlock-$\text{INJ}_{rv,bev}$‖$\text{INJ}_{bev,rv}$	**51.80**

C. Ablation on Influence of Injector Modules

As described in Sec. III-C, we perform our two injector operations (INJ) in parallel to enhance the information flow between the two feature branches. In this section, we study the order of operations in our 2D-3D adapter module. Specifically, we analyze the impact of applying one injector before a ViT block and the second INJ after the same block in Tab. S.1. In the first row, we refine the BEV features before the block and the RV feature after, while in the second row we reverse the order. In the third row, we show that applying both injection operations in parallel results in the highest performance. These results show that parallel injectors enhance the learning of different aspects simultaneously, which allows the network to capture more diverse features.

D. Ablation Influence of Increased Training Data

We emphasize that our proposed method is tailored for small data regimes, whereas tailored LiDAR architectures benefit from training on more data, as shown in Tab. S.2. Consequently, we argue that when large quantities of data are available, LiDAR-specific models are preferred. Nevertheless, we show that our method also outperforms full fine-tuning of the RangeViT architecture when training on 100% of the dataset, which confirms the effectiveness of our 2D-3D adapter and independent decoder branches.

TABLE S.2

PERFORMANCE ON 100% SEMANTICKITTI TRAINING DATA.

SemanticKITTI mIoU (%)	
Method	100 %
SphereFormer	67.8†
FRNet	**68.7**†
RangeViT [8]	60.28
Frozen ViT	60.51
Bias tuning [16]	61.94
LoRA [17]	59.53
VPT [16]	61.46
Vit Adapter [18]	61.29
BALViT (Ours)	62.37

†: performance recorded in published work.