MASKEDFUSION360: RECONSTRUCT LIDAR DATA BY QUERYING CAMERA FEATURES

Royden Wagner, Marvin Klemp, and Carlos Fernandez Lopez
Karlsruhe Institute of Technology (KIT)

ABSTRACT

In self-driving applications, LiDAR data provides accurate information about distances in 3D, but lacks the semantic richness of camera data. Therefore, state-of-the-art methods for perception in urban scenes fuse data from both sensor types. In this work, we introduce a novel self-supervised method to fuse LiDAR and camera data for self-driving applications. We build up-on masked autoencoders (MAEs) and train deep learning models to reconstruct masked LiDAR data from fused LiDAR and camera features. In contrast to related methods that use birds-eye-view representations, we fuse features from dense spherical LiDAR projections and features from fish-eye camera crops with a similar field of view. Therefore, we reduce the learned transformations to moderate perspective transformation and do not require additional modules to generate dense LiDAR representations. Code is available at: `https://github.com/KIT-MRT/masked-fusion-360`

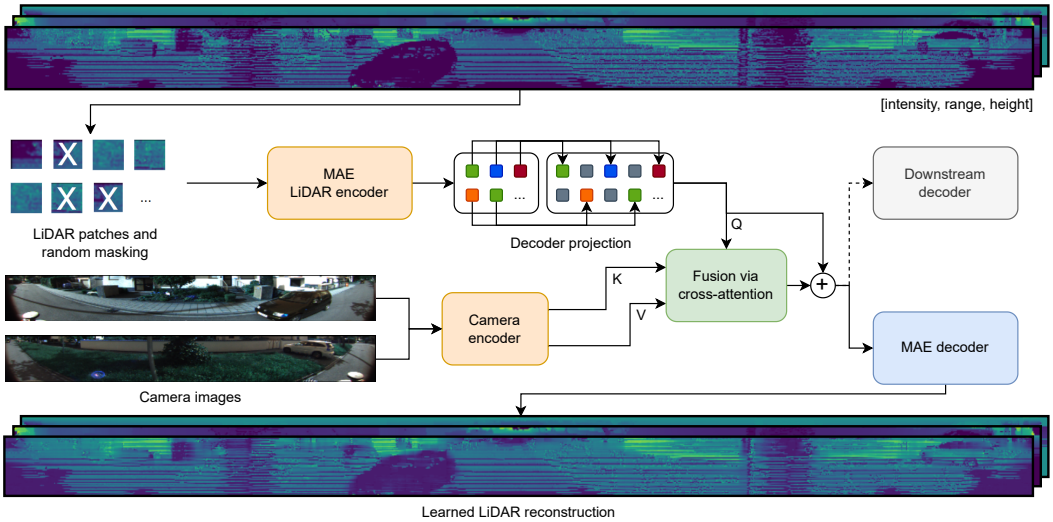

Figure 1: **Reconstruct LiDAR data by querying camera features.** Spherical projections of LiDAR data are transformed into patches, afterwards, randomly selected patches are removed and a MAE encoder is applied to the unmasked patches. The encoder output tokens are fused with camera features via cross-attention. Finally, a MAE decoder reconstructs the spherical LiDAR projections.

1 INTRODUCTION

Simple algorithms that scale well (He et al., 2016; Vaswani et al., 2017; Brown et al., 2020) drive deep learning research forward. A recent addition to this trend are self-supervised pre-training methods that leverage un-labeled data to improve downstream tasks in computer vision (He et al., 2020; Chen et al., 2020; Caron et al., 2021). For self-driving applications, self-supervised pre-training of perception models (Ma et al., 2019) or algorithms for domain adaptation (Wang et al., 2020) are examples. LiDAR data provides accurate information about distances in 3D, but lacks the semantic

richness of camera data. Therefore, state-of-the-art methods for perception in urban scenes (Li et al., 2022; Piergiovanni et al., 2021) fuse data from both sensor types. In this work, we introduce a novel self-supervised method to fuse LiDAR and camera features for self-driving applications. We build up-on masked autoencoders (MAEs) (He et al., 2022) and train vision transformers (ViTs) (Dosovitskiy et al., 2020) to reconstruct masked LiDAR data from fused LiDAR and camera features. Related methods (Ku et al., 2018; Li et al., 2022; Bai et al., 2022) fuse LiDAR and camera data in birds-eye-view (BEV) representations, which involves learning complex transformations from camera coordinates to birds-eye-view coordinates. Hess et al. (2023) and Min et al. (2022) apply masked autoencoding to voxel representations of LiDAR data, which requires additional modules for learning dense voxel embeddings from sparse LiDAR voxels. In contrast, we fuse features from dense spherical LiDAR projections with features from fish-eye camera crops with a similar field of view. Therefore, we reduce the learned spatial transformations to moderate perspective transformations. Moreover, our method does not require additional modules to generate dense LiDAR representations. Overall, our contributions are twofold, we introduce a novel self-supervised training regime to fuse LiDAR and camera data and perform initial evaluation on a self-driving dataset.

2 METHOD

Our proposed fusion method builds up-on masked autoencoding (He et al., 2022), which is a recent form of denoising autoencoding. Masked autoencoders (MAEs) consist of a ViT-based encoder and a ViT-based decoder. Accordingly, input images are divided into patches and processed as a sequence of tokens. During training, 50% of these patches are randomly masked and the training target is the reconstruction of the masked patches. We use a MAE encoder as LiDAR encoder (Figure 1). After transforming the LiDAR data into patches, randomly selected patches are removed and the MAE encoder is only applied to the remaining patches. Therefore, our LiDAR encoder achieves the same computational efficiency as a vanilla MAE encoder. The encoder output tokens are projected to the decoder tokens by re-introducing tokens for masked patches. The decoder tokens are fused with camera features via cross-attention. For this attention mechanism, the LiDAR tokens serve as queries Q and the camera tokens as keys K and values V (Figure 1, Appendix A). Finally, a MAE decoder reconstructs the spherical LiDAR projections from the fused tokens.

Experiment. We perform an initial evaluation of our method with the KITTI-360 dataset (Liao et al., 2022) (Appendix B). Figure 2 shows crops of a masked LiDAR intensity channel and the corresponding reconstruction and target. The importance of camera features is demonstrated by the superior reconstruction quality (higher MSSIM scores (Wang et al., 2003)) when LiDAR and camera features are used. We simulate missing camera features by using zero matrices as camera input.

Figure 2: **Left:** Masked spherical LiDAR projection and corresponding reconstruction and target. **Right:** Comparison of reconstruction quality with camera features vs. without camera features.

3 CONCLUSION

In contrast to related methods that use birds-eye-view representations, we fuse features from dense spherical LiDAR projections and features from fish-eye camera crops. Therefore, we reduce the learned spatial transformations to moderate perspective transformation and do not require additional modules to generate dense LiDAR representations. Future steps include evaluating the performance of our fusion method as pre-training for semantic scene understanding in urban scenarios.

ACKNOWLEDGEMENTS

This work was accomplished within the project HAIBrid (FKZ 01IS21096A). We acknowledge the financial support for the project by the Federal Ministry of Education and Research of Germany (BMBF). Furthermore, this work was supported by the Helmholtz Association's Initiative and Networking Fund on the HAICORE@FZJ partition.

URM STATEMENT

The authors acknowledge that at least one key author of this work meets the URM criteria of ICLR 2023 Tiny Papers Track.

REFERENCES

Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1090–1099, 2022.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 357–366, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

Georg Hess, Johan Jaxing, Elias Svensson, David Hagerman, Christoffer Petersson, and Lennart Svensson. Masked autoencoder for self-supervised pre-training on lidar point clouds. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 350–359, 2023.

J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3): 90–95, 2007. doi: 10.1109/MCSE.2007.55.

Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3d proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–8. IEEE, 2018.

Yingwei Li, Adams Wei Yu, Tianjian Meng, Ben Caine, Jiquan Ngiam, Daiyi Peng, Junyang Shen, Yifeng Lu, Denny Zhou, Quoc V Le, et al. Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17182–17191, 2022.

Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 3288–3295. IEEE, 2019.

Gregory P Meyer, Ankit Laddha, Eric Kee, Carlos Vallespi-Gonzalez, and Carl K Wellington. Lasernet: An efficient probabilistic 3d object detector for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12677–12686, 2019.

Chen Min, Dawei Zhao, Liang Xiao, Yiming Nie, and Bin Dai. Voxel-mae: Masked autoencoders for pre-training large-scale point clouds. *arXiv preprint arXiv:2206.09900*, 2022.

AJ Piergiovanni, Vincent Casser, Michael S Ryoo, and Anelia Angelova. 4d-net for learned multi-modal alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15435–15445, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Yan Wang, Xiangyu Chen, Yurong You, Li Erran Li, Bharath Hariharan, Mark Campbell, Kilian Q Weinberger, and Wei-Lun Chao. Train in germany, test in the usa: Making 3d object detectors generalize. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11713–11723, 2020.

Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pp. 1398–1402. IEEE, 2003.

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021.

A FUSION VIA CROSS-ATTENTION

The ViT-based encoders for LiDAR and camera data process their inputs as sequences of patch tokens ([Patch]) and a learnable class token ([CLS]). We use the class token of the LiDAR encoder and the patch tokens of the camera encoder to fuse information via cross-attention (Chen et al., 2021). As shown in Figure 3, the LiDAR class token is used as queries vector and concatenated with the camera patch tokens to generate keys and values matrices for a standard attention module. Afterwards, the LiDAR class token is added to the attention output to compute a fused class token. In this way, the additional LiDAR class token can learn where LiDAR data is sparse or masked and query the camera tokens accordingly. Furthermore, this cross-attention mechanism is computed in a token-to-sequence manner. This reduces the computational complexity compared to the vanilla sequence-to-sequence manner from $O(n^2)$ to $O(n)$, where n is the sequence length.

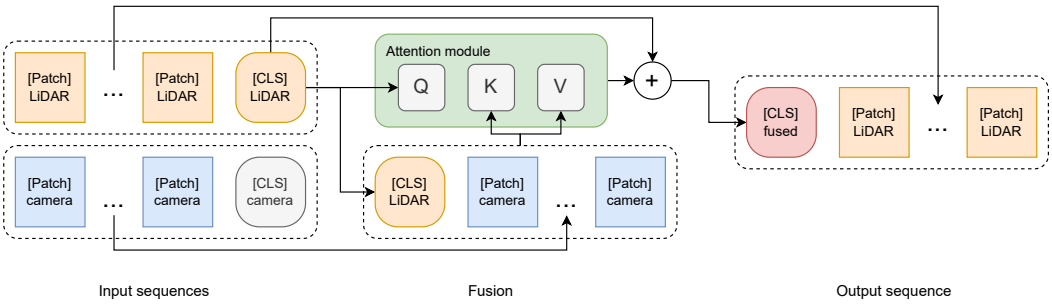

Figure 3: **Fuse LiDAR and camera tokens via cross-attention.** The attention module computes: $\text{softmax}(\frac{QK^T}{\sqrt{d_K}}) \cdot V$, where Q, K, and V are query, key, and value vectors, and d_K is the size of key vectors.

B EXPERIMENTAL SETUP

Dataset. KITTI-360 contains recordings from driving in urban and sub-urban areas in Germany. The dataset contains 76k sets of fish-eye camera images and LiDAR scans as training data and 13k sets as testing data. We use randomly selected 70k and 6k samples from the training data as training and validation splits and the testing data as test split. Following Meyer et al. (2019), we transform all LiDAR scans into spherical projections and store range, height, and intensity data as 3 channels (Figure 4).

Model and training. We use a patch size of 8×8 pixels for our model. The smaller patch size than in a vanilla ViT (16×16) is chosen to better handle fine granular semantic correlations (Xie et al., 2021). Both our ViT-based encoders have an embedding dimension of 2048 and a depth of 8. The cross-attention block for fusion has an embedding dimensions of 1024 and a depth of 2. The reconstruction decoder has an embedding dimensions of 1024 and a depth of 8. As loss, we compute the mean squared error between the target and the reconstructed patches. We choose AdamW (Loshchilov & Hutter, 2019) as optimizer with an initial learning rate of 10^{-4} and use cosine annealing (Loshchilov & Hutter, 2017) to reduce the learning rate while training for 57 epochs. During training, we track the multiscale structural similarity index measure (MSSIM) between the learned reconstruction and the training target to asses the reconstruction quality.

Results. Table 1 shows the achieved MSSIM scores after training for 57 epochs.

	$\text{MSSIM}_{\text{train}}$	$\text{MSSIM}_{\text{val}}$	$\text{MSSIM}_{\text{test}}$
With camera features	0.9694	0.9691	0.9612
Without camera features	0.6772	0.6771	0.6410

Comparison to masked autoencoding without camera features. We evaluate the reconstruction performance of our model with and without camera features. The evaluation mode without camera

features is similar to vanilla masked autoencoding but with spherical projections of LiDAR data instead of natural images as inputs and targets. In this mode, the camera features are zero matrices and the corresponding camera patch tokens are zero vectors. Therefore, in the cross-attention mechanism for information fusion (Figure 3), the LiDAR class token only attends to itself and is forwarded via the skip connection. Hence, our model becomes a masked autoencoder with only one encoder. As shown in Figure 2 right, on average 30% higher MSSIM scores are achieved with camera features vs. without camera features. Furthermore, the reconstruction quality without camera features decreases during training (0.72 MSSIM at epoch 2 vs. 0.68 MSSIM at epoch 57). This shows that our model learns to leverage camera features during training.

Qualitative results. Figure 4 shows additional qualitative results of our method. Overall, after training, our model is able to reconstruct the spherical LiDAR projections using our fusion algorithm.

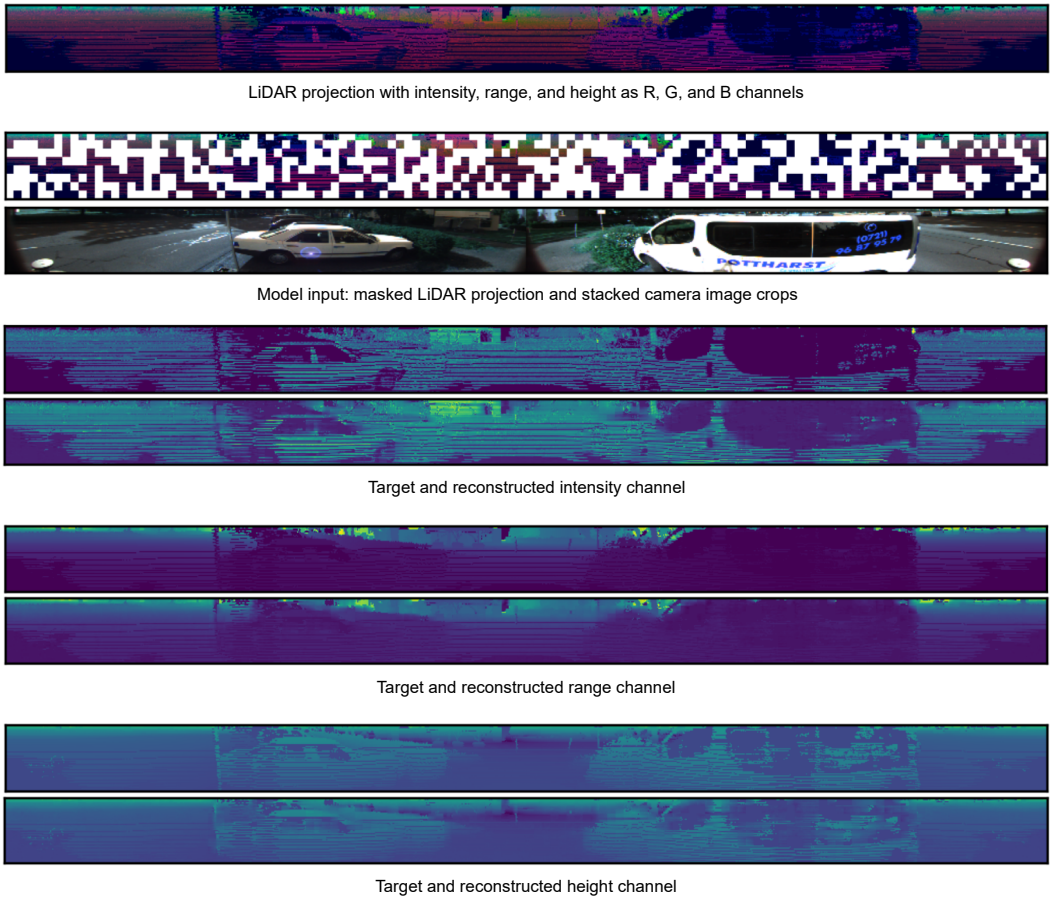

Figure 4: **Validation sample and corresponding LiDAR reconstructions.** To improve the visibility, we pseudo-colorized all channels using the viridis colormap (Hunter, 2007).