Multimodal Alignment model for LiDAR and Image Data

CMSC848K - Final Project

Abubakar Siddiq
M.Eng Robotics
UID: 120403422
UMD College Park
absiddiq@umd.edu

Gayatri Davuluri
M.Eng Robotics
UID: 120304866
UMD College Park
gayatrid@umd.edu

Vijay Chevireddi
M.Eng Robotics
UID: 119491485
UMD College Park
creddy@umd.edu

Abstract

The integration of multimodal data, such as LiDAR point clouds and RGB images, is critical in enhancing perception for autonomous systems. This project proposes a novel multimodal alignment model to address the challenge of aligning spatial relationships between these two modalities. By leveraging pretrained encoders for LiDAR and RGB image data, the model generates scene-level embeddings that represent both modalities in a unified feature space. Inspired by the Q-Former architecture, the model incorporates a query-based transformer module to align embeddings and ensure robust multimodal feature fusion. Using the KITTI dataset for evaluation, this project aims to improve sensor fusion techniques, contributing to more accurate and robust perception systems. Experiments on the KITTI dataset have yielded promising results, highlighting the effectiveness of this approach in aligning multimodal data and extracting shared scenic features or embeddings from both modalities that represent the same scene.

1 Introduction

Sensor fusion is a cornerstone of perception systems in robotics and autonomous vehicles. It combines data from multiple sensors, such as cameras and LiDARs, to create a comprehensive understanding of the environment. Cameras provide rich visual data, including texture and color, while LiDAR sensors offer precise 3D spatial information. However, the stark differences in their data structures pose significant challenges for effective integration. This project aims to address these challenges by developing a novel multimodal alignment model that leverages advanced feature extraction and alignment techniques.

Our proposed model draws inspiration from the Q-Former architecture introduced in BLIP-2, which aligns text and image modalities. By adapting this approach to align image and LiDAR modalities, the project seeks to generate embeddings that effectively represent scene-level features from both data types. The KITTI dataset, known for its synchronized LiDAR and camera data, serves as the foundation for training and evaluation.

This report outlines the motivation, related work, proposed methodology, and expected outcomes of this project, offering insights into the significance of multimodal data alignment for perception systems.

2 Related work

Understanding and aligning multimodal data, such as LiDAR point clouds and RGB images, has been a topic of active research. Our work builds on advancements in multimodal learning, particularly in feature extraction and alignment, while addressing the limitations of prior approaches.

1. **Vision-LiDAR Sensor Fusion:**

   Vision-LiDAR fusion has evolved from early integration techniques to more sophisticated approaches leveraging intermediate feature fusion. For instance, the LiDAR Image Fusion Transformer (LIFT) combines sequential LiDAR point clouds and camera images to enhance 3D object detection performance (Zeng et al., 2022). However, many of these methods focus on task-specific applications like object detection and segmentation, limiting their generalizability to broader scene alignment tasks. Our project extends these concepts by aligning data at the embedding level, enabling shared scene understanding.

2. **Transformers in Multimodal Learning:**

   Transformers have revolutionized multimodal data processing, effectively modeling long-range dependencies across diverse data sources. The BLIP-2 architecture, with its Q-Former module, introduced an innovative framework for aligning textual and visual data through query embeddings (Li et al., 2023). Inspired by this, our work adapts Q-Former to bridge image and LiDAR modalities. Unlike BLIP-2, which focuses on language-vision fusion, we refine this architecture for independent and complementary feature extraction from image and LiDAR data.

3. **LiDAR Feature Extraction with Deep Learning:**

   Deep learning approaches like PointNet and PointNet++ have demonstrated success in extracting geometric structures from LiDAR data (Qi et al., 2017). Hierarchical models like PointNet++ improve upon their predecessor by incorporating local and global feature learning, crucial for tasks like segmentation and 3D detection. However, these methods lack mechanisms for alignment with other modalities, a limitation addressed in our work through embedding space alignment.

4. **Vision Transformers for Image Feature Extraction:**

   Vision Transformers (ViTs) provide state-of-the-art capabilities in image analysis by modeling global relationships within images (Dosovitskiy et al., 2020). While ViTs excel in visual feature extraction, integrating these features with LiDAR data remains underexplored. Our approach leverages the strengths of ViTs to process RGB data and align it with LiDAR embeddings.

5. **Independent vs. Joint Feature Learning:**

   Prior studies debate the merits of independent versus joint feature learning for multimodal data. Models like the Multimodal Transformer Network (MTNet) advocate for joint learning to capture inter-modal relationships (Ma et al., 2023). However, joint learning often requires extensive retraining, reducing computational efficiency. Our method balances independent feature extraction with effective alignment through a trainable intermediate module, optimizing performance without retraining pretrained encoders.

6. **KITTI Dataset for Benchmarking:**

   The KITTI dataset provides synchronized camera and LiDAR data, serving as a benchmark for evaluating perception models in autonomous driving scenarios. Existing methods like PointPillars (Lang et al., 2019) and PV-RCNN (Shi et al., 2020) achieve high accuracy on KITTI tasks but often emphasize task-specific performance. Our work leverages KITTI to validate a more generalizable approach to multimodal alignment, focusing on embedding similarity as the evaluation metric.

By building on these advancements, our project introduces a novel embedding alignment framework that addresses gaps in prior research, particularly in achieving general-purpose multimodal alignment with pretrained encoders.

3 Methodology

3.1 Model Architecture

Our Multimodal Alignment Model is designed to generate compact and comparable scene-level embeddings for image and LiDAR point-cloud data, enabling effective multimodal alignment. As shown in Figure 1, the architecture integrates three main components: the Image Encoder, the LiDAR Encoder, and the Query-Based Transformer (Q-Former). These components collaboratively process multimodal inputs to produce related feature embeddings in such a way that they are similar if the image and LIDAR data represents the same scene and vice-versa.

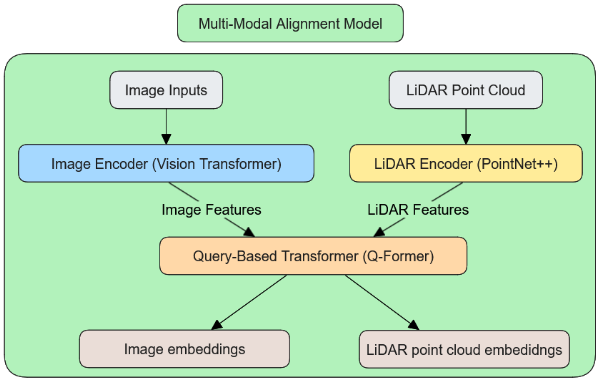

Multi-Modal Alignment Model
Image Inputs
LiDAR Point Cloud
Image Encoder (Vision Transformer)
LiDAR Encoder (PointNet++)
Image Features
LiDAR Features
Query-Based Transformer (Q-Former)
Image embeddings
LiDAR point cloud embedidngs

Figure 1: Model Architecture

1. **Image Encoder:** The image encoder leverages a pretrained Vision Transformer (ViT) to process 2D visual inputs. ViT divides input images into fixed-size patches, each treated as a token, and passes them through multiple transformer layers to extract high-dimensional feature representations. To preserve computational efficiency, the pretrained ViT model is frozen during training, ensuring robust feature extraction while avoiding the need for retraining. The encoder outputs a sequence of features that effectively represent global spatial relationships in the image, serving as an essential component for scene-level understanding.
2. **LiDAR Encoder:** The LiDAR encoder utilizes a pretrained PointNet++ model, which is specifically designed to process raw 3D point cloud data. PointNet++ incorporates hierarchical feature learning to capture both local and global geometric structures within the point cloud. By employing a multi-resolution grouping strategy, it ensures the extraction of fine-grained spatial patterns critical for downstream tasks. The weights of the PointNet++ model are frozen during training. The output of the LiDAR encoder is projected into a higher-dimensional space, ensuring compatibility with the image features and enabling seamless integration into the shared embedding space.
3. **Query-Based Transformer (Q-Former):** As shown in Figure 2, the Q-Former serves as the bridge between the image and LiDAR encoders, refining the extracted features and aligning them into a shared embedding space. This module is inspired by the Q-Former architecture in BLIP-2 but has been adapted for the unique challenges of image-LiDAR data alignment. The Q-Former introduces learnable query embeddings that interact dynamically with input features through a transformer encoder.

   These query embeddings act as task-relevant extractors, isolating high-level information from both image and LiDAR features. The Q-Former processes the two modalities independently, ensuring that their unique characteristics are retained while aligning their embeddings for compatibility. The output is a set of compact, high-dimensional embeddings that summarize the essential information from each modality.

3.2 Embedding Alignment

Embedding alignment is a critical component of this multimodal alignment model, ensuring that the final embeddings from the Q-former are represented within a shared feature space. This align-

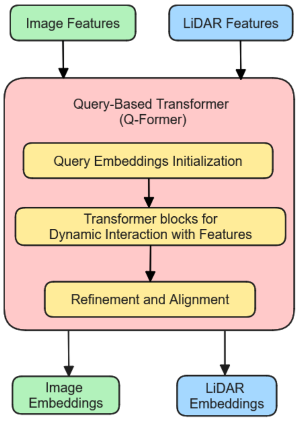


Figure 2: Query-Based Transformer (Q-Former)

ment process utilizes separate linear projection layers, applied to the outputs of the Query-Based Transformer (Q-Former), to refine and transform the modality-specific embeddings.

The alignment mechanism is designed to achieve the following objectives:

- *Similarity:* Project the individual embeddings into a unified space where the corresponding embeddings from the same scene exhibit a high similarity.
- *Distinctiveness:* Ensure that embeddings from different scenes remain distinct, minimizing overlap in the shared space.

The alignment process leverages:

- *Linear Projections:* Each modality's embeddings are processed through separate linear layers to match dimensionality and enable comparability.
- *Cosine Similarity:* Corresponding embeddings are evaluated and optimized using cosine similarity loss, encouraging the model to learn relationships between modalities effectively.

This embedding alignment ensures that multimodal data integration is both robust and semantically meaningful, addressing challenges in feature compatibility across modalities.

3.3 Training Strategy

1. **Data Preprocessing:**
   - Image Data: Images were resized to match the input dimensions of the Vision Transformer and normalized using standard ImageNet statistics. This ensured compatibility with the pretrained image encoder and preserved global spatial information.
   - LiDAR Data: Point clouds were processed to retain spatial coordinates (x, y, z) and padded or truncated to maintain consistent input dimensions for batch processing. This facilitated efficient input handling without compromising the quality of 3D spatial features.
2. **Loss Function:** A cosine similarity loss function was employed to align the embeddings generated by the image and LiDAR modalities. This loss measured the similarity between corresponding embeddings, encouraging alignment for matched inputs while ensuring distinction for non-matching pairs. This approach robustly captured the relationships between modalities.
3. **Optimization:** The model was optimized using the AdamW optimizer, which balanced fast convergence and regularization. Only the Q-Former and projection layers were updated during training, while the pretrained encoders remained frozen. This strategy leveraged the pretrained features effectively, reducing computational overhead and preserving the robustness of the pretrained encoders.

4. **Training Process:** The training process included both training and validation steps, with an equal subset of the KITTI dataset used for each. During training, the model learned to align the multimodal embeddings through iterative updates. Validation batches were periodically evaluated to monitor model performance, ensuring early detection of overfitting or convergence issues. Model checkpoints were saved based on validation loss to retain the best-performing model.

4 Results

The experiments conducted were aimed to analyze the effect of dropout, regularization, and batch size on the performance of the multimodal model trained to align image embeddings and point cloud embeddings.

4.1 Graph 1: Training with Dropout and Regularization Effects

This experiment incorporated dropout and regularization techniques to prevent overfitting and improve generalization. The graph shows a steady convergence of the validation loss over epochs. Regularization helped maintain a smoother learning curve, indicating that the model avoided excessive reliance on training data. Dropout contributed by randomly dropping connections during training, forcing the model to learn more robust features. The noisy gradient updates is attributed to the low batch size that was used, the maximum batch size that was available to us was 16 and going higher require a larger GPU with more VRAM, we used NVIDIA L40s with 48GB of RAM.

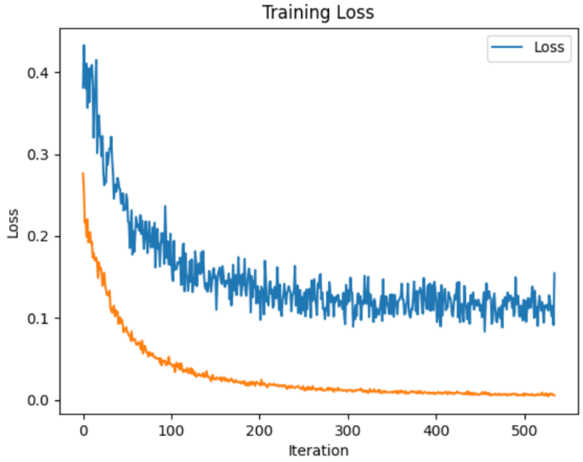


Figure 3: *Training loss with dropout and regularization*

Training Details:

- Dropout rate and regularization parameters were tuned for optimal performance.
- Batch size: 16 (due to resource constraints).
- Training was performed on *Lightning AI Studio* GPU instances using NVIDIA L40s GPUs with 48GB VRAM.

4.2 Graph 2: Training without Dropout

In this experiment, dropout was not applied, allowing the model to use all connections during training. The graph demonstrates gradual convergence and the loss plateaued with improvement in validation accuracy. The training and validation loss performed similarly but still maintained the requirement of lower validation loss for generalizability.

Key Observations:

- The training and validation losses decrease at a similar rate.

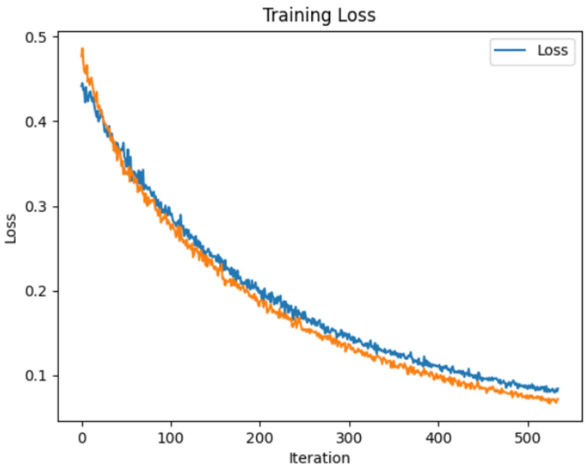


Figure 4: *Training loss without dropout and regularization*

- The absence of dropout made the model overly dependent on specific training features.

4.3 Graph 3: Training on subset of dataset (2000 datapoints) without Dropout and Tuned Hyperparameters

This experiment involved training the model on a reduced dataset (25%) with tuned hyperparameters. Dropout was again excluded. The graph shows a slower training loss reduction compared to the full dataset scenarios. However, by carefully fine tuning the hyperparameters, including the learning rate and weight decay, the model achieved the best training performance overall. The curve demonstrates smooth and consistent convergence with minimal signs of overfitting.

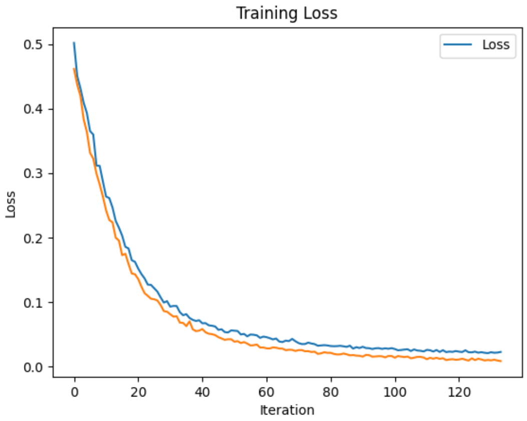


Figure 5: *Training on subset of dataset without dropout and tuned hyperparametrs*

Key Observations:

- Despite the reduced dataset size, tuning hyperparameters improved training stability.
- Figure 5 represents the best training results, showcasing a balance between loss convergence and model generalization.

Training Details:

- Hyperparameters such as learning rate (3e-5), weight decay (5e-6), and optimizer (AdamW) settings were adjusted to compensate for the reduced dataset size and also utilized a learning rate scheduler (StepLR).

- Batch size remained at 16 due to hardware limitations.

The training experiments utilized limited computational resources (maximum batch size of 16) but successfully demonstrated the effect of regularization techniques and data availability on model performance. Figure 5 stands out as the *best-performing result*, proving that with proper hyperparameter tuning, the model can achieve excellent performance even on smaller datasets.

Downstream task experimentation - Object detection

To extend the multimodal alignment model, a detection head was added for object detection. As shown in Figure 6, it uses the Q-former embeddings as inputs and utilizes shared feature extraction layers and three branches to predict the bounding boxes of the objects detected in the scene both in image and Lidar point cloud:

- *Classification Branch:* Predicts object categories (e.g., car, pedestrian, cyclist).
- *2D Bounding Box Branch:* Outputs 2D bounding boxes in image space.
- *3D Bounding Box Branch:* Outputs 3D bounding boxes in Point cloud space.

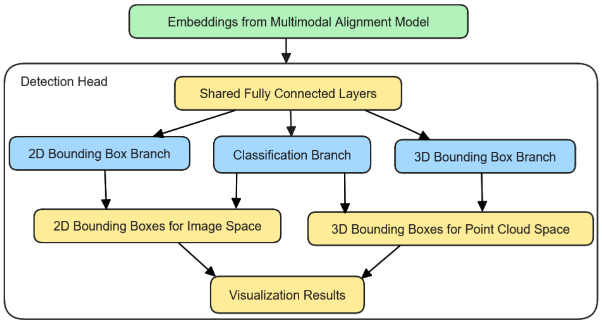
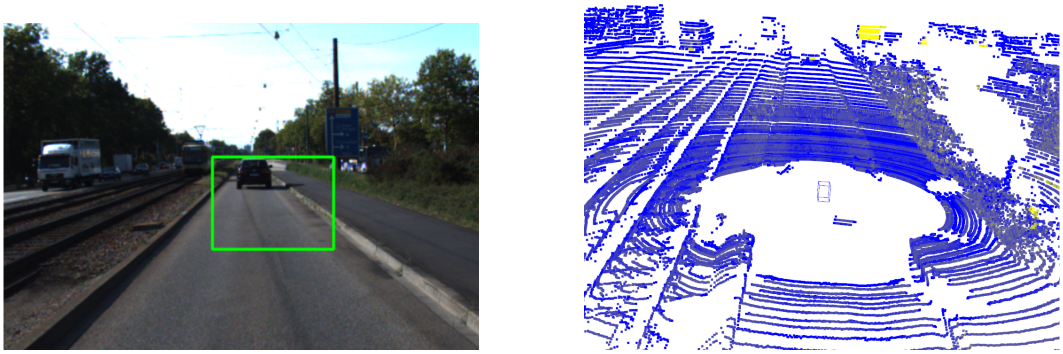

Embeddings from Multimodal Alignment Model
Detection Head
Shared Fully Connected Layers
2D Bounding Box Branch
Classification Branch
3D Bounding Box Branch
2D Bounding Boxes for Image Space
3D Bounding Boxes for Point Cloud Space
Visualization Results

Figure 6: *Detection head*

The model performed object detection, classifying and localizing objects as shown in Figure 7 for an example datapoint.

- 2D Results: Bounding boxes overlaid on images
- 3D Results: Bounding boxes visualized in LiDAR point clouds



(a) *2D Results: Bounding boxes overlaid on images.*

(b) *3D Results: Bounding boxes visualized in LiDAR point clouds.*

Figure 7: Comparison of 2D and 3D bounding box results.

The performance of the detection head was suboptimal, as the primary focus of this project was to develop the Q-Former and evaluate its output for embedding similarity. Consequently, the detection head was not fully optimized for high performance due to the exhaustion of GPU instance credits.

Conclusion

This project successfully developed and evaluated a multimodal alignment model for LiDAR and RGB image data using the KITTI dataset. By leveraging pretrained encoders and a query-based transformer module inspired by Q-Former, the model effectively aligned embeddings from both modalities into a shared feature space. Experimental results demonstrated the robustness of the proposed architecture, with key techniques like dropout, regularization, and hyperparameter tuning enhancing the model's performance and generalization capabilities.

Additionally, the project also explored the implementation of a detection head using Q-Former embeddings to classify and localize objects in both 2D and 3D spaces. With further training and optimization, the detection head has the potential to deliver promising results. Future work could involve scaling the detection head and adjusting it accordingly for further enhancement in real-time applications.

References

[1] Li, J., et al. (2023). *BLIP-2: Bootstrapping Language-Image Pretraining with Frozen Image Encoders and Large Language Models* arXiv:2301.12597.

[2] Zeng, Y., et al. (2022). *LIFT: Learning 4D LiDAR-Image Fusion Transformer for 3D Object Detection.* CVPR 2022.

[3] Qi, C. R., et al. (2017). *PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space.* NeurIPS 2017.

[4] Dosovitskiy, A., et al. (2020). *A Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.* arXiv:2010.11929.

[5] Geiger, A., et al. (2012). *Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite.* CVPR 2012.

[6] Ma, X., et al. (2023). *MTNet: Multimodal Transformer Network for Hyperspectral and LiDAR Data Fusion.* IEEE Transactions on Geoscience and Remote Sensing.

[7] Lang, A. H., et al. (2019). *PointPillars: Fast Encoders for Object Detection From Point Clouds.* CVPR 2019.

[8] Shi, S., et al. (2020). *PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection. CVPR 2020.*