# VOLNet: An Aligned Approach for Visual-LiDAR Odometry

**Ming-Feng Li*** **Pujith Kachana*** **Shashwat Chawla*** **Yatharth Ahuja ***
{mingfenl, pkachana, shashwac, yahuja}@andrew.cmu.edu

## Abstract

Visual Odometry (VO), or the estimation of camera motions from sequential RGB frames, is essential in autonomous systems and visual SLAM. Traditional geometry-based methods often face challenges in environments with dynamic obstacles and rapid movements. To enhance performance, learning-based monocular VO approaches like TartanVO have been developed, demonstrating significant real-world effectiveness. However, these methods struggle with scale ambiguity due to a lack of inherent depth information. To address this, we propose **VOLNet**, a Visual-LiDAR Odometry method that integrates RGB images with LiDAR data, harnessing the semantic richness of RGB images and the precise geometric details from 3D LiDAR point clouds. This multimodal approach underscores the potential for improved odometry tasks. Typically trained on limited datasets such as KITTI, existing methods do not adequately adapt to varied and dynamic real-world scenarios. We train on TartanAir-v2, a large-scale synthetic dataset that includes diverse indoor and outdoor environments. Our method, free from time-sync and camera calibration issues, leverages high-quality synthetic data to demonstrate enhanced estimation accuracy and superior cross-dataset generalizability, pushing the boundaries of existing Visual-LiDAR Odometry baselines.

## 1 Introduction and Problem Definition

Accurate 3D localization is fundamental for autonomous systems, facilitating precise movement tracking in autonomous vehicles, robots, and drones. Visual odometry (VO), crucial for estimating camera motions from sequential RGB frames, plays an essential role in integrating RGB information with LiDAR sensors for various navigation and mapping tasks. For instance, Visual Simultaneous Localization and Mapping (SLAM) is increasingly vital due to the widespread availability of cameras and relies heavily on visual odometry. Traditional geometry-based methods are foundational but often struggle in complex environments with dynamic obstacles and rapid movements. In contrast, recent learning-based methods, while promising, lack the intrinsic geometric grounding of traditional approaches. Current visual odometry predominantly employs monocular camera systems, which, although cost-effective and broadly applicable, are limited by their inability to reliably determine depth, leading to potential inaccuracies in scale and overall pose estimation. These systems also typically underperform in challenging environmental conditions, such as poor lighting or the presence of dynamic objects, which can disrupt feature tracking continuity and reliability.

Moreover, most existing VO methods are trained and validated on relatively constrained datasets like KITTI, which primarily feature well-controlled urban driving scenarios. These datasets fail to capture the complexity and diversity of real-world environments, thus limiting the generalizability of these methods to scenarios beyond typical urban conditions, such as off-road environments or areas with highly dynamic and unpredictable elements.

To overcome these challenges, we propose leveraging the large-scale synthetic dataset, TartanAir-v2, which features a variety of environments and realistic instances, to address the limitations of limited real-world training data. Additionally, we propose a hybrid Visual-LiDAR Odometry system that integrates the semantic richness of RGB images with the geometric precision of 3D LiDAR point clouds. Our approach involves estimating the optical flow map between two neighboring images to ascertain their semantic correspondences. Concurrently, we project LiDAR points into 2D depth maps aligned with the RGB images. The RGB

*Everyone Contributed Equally – Alphabetical order

images, flow maps, and projected depth maps of LiDAR points are then concatenated and fed into a transformer-based network to estimate the final relative camera pose between the images.

By benefiting from the synthetic dataset, our approach not only improves the reliability of localization under various operational conditions but also extends the applicability of VO systems to more dynamic and unstructured environments. Leveraging high-quality images and accurate LiDAR data, our system demonstrates the complementary strengths of each sensor type, enhancing the robustness and accuracy of odometry across diverse environments.

### 1.1 Problem Definition.

For visual-LiDAR odometry, we are provided with a sequence of $n$ RGB test images, $\mathcal{I} = \{I_0, I_1, \ldots, I_{n-1}\}$, and corresponding LiDAR point clouds $\mathcal{L} = \{L_0, L_1, \ldots, L_{n-1}\}$. Our method aims to estimate the relative camera pose $\xi i$ for each pair of consecutive test images $(I_i, Ii + 1)$ and their respective LiDAR scans $(L_i, L_{i+1})$. Each estimation $\xi i$ is represented as a rotation quaternion $q \in \mathbb{R}^4$ and a translation vector $t \in \mathbb{R}^3$, which describe the orientation and position changes between two consecutive frames. For each estimation, we fuse the inputs from $(I_i, Ii + 1)$ and $(L_i, L_{i+1})$, and then send them to a trained model that predicts $\xi_i$. This process leverages both visual and geometric data to enhance accuracy in pose estimation.

### 1.2 Contributions

Our proposed solution involves training a visual-LiDAR odometry model on the TartanAir-v2 dataset, a comprehensive synthetic dataset designed to simulate a wide range of environmental conditions. This training approach aims to overcome the limitations of existing models by including scenes with significant environmental variability and dynamic obstacles, thereby ensuring that our models are better prepared for real-world deployment. We contribute a novel learning-based visual-LiDAR framework that is robust to camera parameters, understands scale, and is data-driven and scalable. Through this approach, we seek to set new benchmarks in estimation accuracy and adaptability for Visual-LiDAR Odometry systems, pushing the boundaries of what is currently achievable in autonomous navigation.

## 2 Related Work and Background

Odometry serves as a fundamental component in robotic navigation, facilitating accurate estimation of a robot's position and motion over time. It has been an active research avenue with various landmark works over time - Mur-Artal et al. proposed ORB-SLAM (Mur-Artal and Tardós, 2017), a real-time monocular visual SLAM framework leveraging ORB features for robust odometry and loop closure detection, demonstrating state-of-the-art accuracy in both indoor and outdoor environments. Zhang and Scaramuzza introduced VIO (Visual-Inertial Odometry) (Zhang and Scaramuzza, 2018), combining visual and inertial measurements to address challenges posed by low-texture and dynamic lighting conditions, enabling robust navigation across diverse scenarios. Similarly, Lego-LOAM (Shan and Englot, 2018) advanced LiDAR odometry by incorporating both edge and surface features to enhance accuracy and robustness in complex 3D environments. For multi-modal systems, Kimera (Rosinol et al., 2020) integrated visual, inertial, and geometric data to achieve metric-semantic SLAM, enhancing odometry performance in dynamic and cluttered spaces. Further, Wang et al. employed deep learning in DeepVO (Teed et al., 2023) to learn odometry directly from sequential image data, showcasing the potential of data-driven methods for end-to-end trajectory estimation. These works collectively underscore the evolution of odometry from classical feature-based approaches to modern multi-modal and learning-based paradigms, addressing the increasing demands of autonomous systems. We explore the odometry estimation landscape in a more classified manner in this section.

### 2.1 Related Datasets

**KITTI Odometry.** The KITTI odometry dataset (Geiger et al., 2012) is a widely used benchmark for evaluating odometry and SLAM systems. This dataset includes 22 sequences of LiDAR point clouds and corresponding stereo images. For the purposes of this study, we utilize only the monocular left camera images for fusion with the LiDAR sensor.

**TartanAir.** The TartanAir dataset (Wang et al., 2020b) is a large-scale, synthetically generated vision and navigation dataset rendered using the Unreal Engine. The synthetic nature of the data guarantees precise, dense measurements, while the well-developed rendering process keeps the data reasonably photo-realistic.

**TartanAir-v2.** Building on the foundation of TartanAir, the TartanAir-v2 dataset expands the scale and richness of the data significantly. TartanAir-v2 introduces 100 unique environments, each featuring more sequences and an even greater variety of data modalities. Most notably, TartanAir-v2 includes a LiDAR modality which we make use of for this project.

### 2.2 Traditional Visual Odometry

For the RGB modality, the classical unimodal baseline **ORB-SLAM** (Mur-Artal and Tardós, 2017) extracts ORB features from images and uses them to establish correspondences between consecutive frames. By applying epipolar constraints and classical optimization techniques, it estimates camera poses, which are further refined through global optimization using a pose graph and loop closure. ORB-SLAM will establish a baseline for the accuracy achievable with monocular RGB images and geometric analysis, supporting our hypothesis that RGB images provide strong visual features for odometry.

### 2.3 Traditional LiDAR Odometry

For the LiDAR modality, the unimodal classical baseline **ICP-SLAM** (Vizzo et al., 2023) leverages ICP (Iterative Closest Points) to compute correspondences between two point sets by identifying the nearest neighbors and then calculating the transformation that minimizes the distance between these approximated correspondences. This process is repeated until the transformation converges below a threshold or reaches a maximum number of iterations. ICP can be effective with consecutive frames, although its accuracy diminishes in the presence of noise or lower sampling rates. This method will serve as a baseline to gauge the accuracy achievable with LiDAR alone and assess how well LiDAR's inherent 3D structure can be utilized.

### 2.4 Traditional Visual-LiDAR Odometry

To enhance the accuracy and robustness of odometry and mapping tasks, advanced baselines such as **DV-LOAM** (Wang et al., 2021) and **SDV-LOAM** (Yuan et al., 2023) effectively integrate visual and LiDAR information. **DV-LOAM** utilizes frame-to-frame tracking and sliding window optimization for efficient pose estimation, refining keyframe poses with LiDAR data, which is particularly beneficial in

environments with sparse visual features. Extending these capabilities, **SDV-LOAM** incorporates semi-direct approaches like LiDAR-Assisted Visual Odometry to improve depth estimation and adapt to dynamic environments. Its cascaded Vision-LiDAR architecture harmonizes direct visual odometry with precise LiDAR measurements for more accurate and robust pose estimates. Traditional methods, such as **V-LOAM** (Zhang and Singh, 2015) and **LIMO** (Graeter et al., 2018), also leverage high-frequency visual odometry estimates as motion priors for LiDAR, with **LIMO** utilizing depth information to mitigate scale uncertainty. Additionally, **PL-LOAM** (Huang et al., 2020) provides novel scale correction algorithms alongside pure visual tracking methods, further bridging the gap between 3D and 2D data integration challenges.

### 2.5 Learning-Based Visual Odometry

Recent works in learning-based visual odometry have explored new architectures and methodologies for improving efficiency, accuracy, and robustness.

**TartanVO** (Wang et al., 2020a) will serve as our learning-based unimodal baseline using RGB images. As one of the top-performing odometry models for monocular images, TartanVO is trained exclusively on the TartanAir dataset (Wang et al., 2020b), which we will also employ. This baseline will demonstrate the advantages of large-scale data and learning-based approaches while validating the effectiveness of the TartanAir dataset.

**Deep Patch VO (DPVO)** (Teed et al., 2023) proposes a novel recurrent network architecture designed for tracking image patches across time. While dense flow has traditionally been assumed essential for redundancy against incorrect matches, DPVO demonstrates that sparse patch-based matching can achieve better accuracy and efficiency. On standard benchmarks, DPVO outperforms all prior work, including the state-of-the-art VO system **DROID**, using a third of the memory and running 3x faster on average.

**DytanVO** (Shen et al., 2023) is a supervised learning-based VO method designed to address dynamic environments. It processes two consecutive monocular frames in real-time and predicts camera ego-motion iteratively. DytanVO achieves an average improvement of 27.7% in absolute trajectory error (ATE) over state-of-the-art VO solutions in dynamic scenarios, performing competitively among dynamic visual SLAM systems that optimize trajectories in the backend.

**Salient Sparse VO** (Chen et al., 2024) introduces a hybrid VO framework with pose-only supervision, reducing the dependency on dense flow labels. The framework includes two cost-effective innovations: self-supervised homographic pre-training for improving optical flow learning and a random patch-based salient point detection strategy for more accurate optical flow patch extraction. These designs significantly enhance generalization capabilities in diverse and challenging environments, making it a robust solution for VO tasks.

These recent advancements in visual odometry have shifted the focus toward lightweight, efficient architectures and methods that address specific challenges like sparsity, dynamics, and generalization, setting new benchmarks for performance and applicability in real-world scenarios.

### 2.6 Learning-Based LiDAR Odometry

**LO-Net** (Li et al., 2020a) is a popular learning-based unimodal baseline using LiDAR data. It leverages a novel weighted geometric constraint loss to efficiently extract features from point clouds. This baseline will help establish a benchmark for widely deployed LiDAR-based odometry solutions across various datasets, including TartanAir.

### 2.7 Learning-Based Visual-LiDAR Odometry

In recent years, innovative learning-based methods such as **H-VLO** (Liu et al., 2024a) and **DVLO** (Liu et al., 2024b) have been developed to integrate visual and LiDAR odometry by training models to predict camera motions.

**H-VLO** has demonstrated robust performance by effectively combining the semantic information from camera data with the depth and structural detail from LiDAR data. This fusion approach leverages the strengths of both modalities, leading to improved trajectory accuracy and enhanced robustness in complex environments characterized by varying levels of occlusion, lighting, and scene intricacies.

Despite these advancements, methods like **H-VLO** primarily focus on feature-level fusion but often struggle to capture the fine-grained pixel-to-point correspondences necessary for precise pose estimation. This challenge is exacerbated by the inherent structural differences between sparse LiDAR points and dense camera pixels, which can

lead to data misalignment and consequently limit the effectiveness of multi-modal fusion.

To address these challenges, **DVLO** introduces a novel local-to-global fusion network with bi-directional structure alignment, specifically designed to enhance the integration of LiDAR and visual features for more accurate pose estimation. **DVLO** improves spatial consistency by clustering neighboring visual information in 2D images and projecting LiDAR depth data onto these images. This method ensures a more effective alignment of LiDAR points with visually similar pixels, thereby enhancing overall model performance. Additionally, MVL-SLAM (An et al., 2022) employs the RCNN network architecture to fuse RGB images with multi-channel depth images from 3D LiDAR points, offering another layer of depth and texture integration. Furthermore, LIP-Loc (Shubodh et al., 2024) proposes a pre-training strategy for cross-modal localization, utilizing contrastive learning to jointly train image and point encoders, thereby fostering better coherence between the data modalities.

However, while learning-based approaches like **H-VLO** and **DVLO** demonstrate promising results in visual-LiDAR odometry, they predominantly rely on training and validation on constrained datasets such as KITTI. These datasets primarily feature well-controlled urban driving scenarios and fail to reflect the complexity and diversity of real-world environments adequately. This limitation restricts the generalizability of these methods to more varied and dynamic settings.

Overall, despite the successes of these various unimodal, multimodal, traditional, and learning-based methods, it is clear that the problem remains far from solved. Issues such as dynamics and harsh conditions such as lighting and textures lead to failure for most of these methods. We explore how a more aligned fusion of LiDAR and RGB data can help alleviate these issues.

### 2.8 Relavent Techniques

A key technique we utilize is **point unprojection**, which enables explicit correspondences between LiDAR points and camera pixels. By leveraging camera parameters and the calibration extrinsics from LiDAR to the camera, LiDAR points can be unprojected onto the camera frame, aligning spatially across the modalities. This technique allows the creation of ”pointmaps,” as employed in DVLO, where pointmaps are used as a critical training representation. Furthermore, works such as **DUSt3R** (Wang et al., 2023) and **MASt3R**(Leroy et al., 2024), which train models for pointmap prediction and 3D feature learning, demonstrate that pointmaps inherently carry strong 3D inductive biases. These biases make pointmaps an excellent representation for learning odometry, as evidenced by DUSt3R and MASt3R's success in achieving state-of-the-art performance in relative pose estimation. This underscores their potential for bridging the gap between 2D and 3D feature spaces in odometry tasks.

## 3 Task Setup and Data

Visual odometry is a complex task that requires a precise understanding of the relative ego-motion and object dynamics. Currently, there is no single modality that can effectively capture enough data to enable this reasoning. Monocular images lack the necessary depth information to understand scale, while depth and LiDAR data are often very noisy and do not have salient enough features to compute correspondences between frames. Given that there is no satisfactory unimodal solution, we propose that visual odometry should therefore be seen as a fundamentally multimodal task. Monocular RGB input will help us get rich visual features in the contextual scene, whereas the LiDAR modality will help us consider and construct semantic sense in the same. We can expect some overlap in the information between these modalities due to possible correlations between features and their depth - but they also complement each other by providing separate information.

### 3.1 Problem Setup

Accurate 3D localization is essential for autonomous robotic systems to precisely track the movement of autonomous driving cars, robots, or drones. By integrating RGB or LiDAR sensors, estimated camera poses from 3D localization can further facilitate the dense map reconstruction of 3D scenes. Visual Simultaneous Localization and Mapping (SLAM) is becoming increasingly important due to the widespread availability of cameras and the rich information provided by images. Visual odometry (VO), a key component of visual SLAM, has seen notable progress through geometry-based methods (Engel et al., 2014), (Mur-Artal and Tardós, 2017), (Engel et al., 2017). However, these methods are often unreliable in challenging conditions such as varying illumination, dynamic environments, and rapid movements. Deep neural network-based methods have demonstrated the ability to learn more robust feature extractors than traditional, engineered ones, leading to more capable VO models (Wang et al., 2018), (Vijayanarasimhan et al., 2017).

### 3.2 Modality Analysis

Some of the key insights from evaluating these baselines are as follows:

1. **ORB-SLAM**: This method exhibited relatively poor performance in featureless areas and dynamic environments. However, as shown in Table 5.6, ORB-SLAM performed well in terms of rotational error.

2. **TartanVO**: This monocular, learning-based visual odometry method outperformed other similar methods. Its superior performance is attributed to a diverse training dataset and the incorporation of camera intrinsic parameters into its architecture.

3. **DVLO**: It utilizes a local-to-global feature fusion approach, integrating features from both narrow receptive fields and broader global contexts. This strategy enhances its performance by effectively combining local details and global information from visual and LiDAR data.

### 3.3 Dataset

The TartanAir-v2 dataset is a large-scale, synthetically generated vision and navigation dataset rendered using the Unreal Engine. The synthetic nature of the data guarantees precise, dense measurements, while the well-developed rendering process keeps the data reasonably photo-realistic.

| **Dataset** | **TartanAir-v2** |
|---|---|
| **Data Size** | 40TB |
| **Environments** | 100 environments |
| **Data Modalities** | RGB, LiDAR, Flow |
| **Rig Config** | 6 pinhole cameras |

Table 1: Data Statistics of TartanAir-v2.

### 3.4 Metrics

The metrics we are considering for our use-case and application are $t_{rel}$ and $r_{rel}$, which report the relative translation and rotational error, respectively. These metrics are elaborated as follows.

**Metric 1, $t_{rel}$:** This metric represents the average relative translation error in the trajectory across the sequence. It is computed as:

$$t_{rel} = \frac{1}{N} \sum_{i=1}^{N} \frac{\|\mathbf{t}_i - \mathbf{t}_i^{gt}\|}{\|\mathbf{t}_i^{gt}\|}$$

where $\mathbf{t}_i$ is the estimated translation for frame $i$, $\mathbf{t}_i^{gt}$ is the ground truth translation, and $N$ is the number of frames.

**Metric 2, $r_{rel}$:** This metric reports the average relative rotation error in the trajectory across the sequence. It is computed as:

$$r_{rel} = \frac{1}{N} \sum_{i=1}^{N} \arccos \left( \frac{\text{trace}(\mathbf{R}_i^{gt} \mathbf{R}_i^{\top}) - 1}{2} \right)$$

where $\mathbf{R}_i$ is the estimated rotation matrix for frame $i$, $\mathbf{R}_i^{gt}$ is the ground truth rotation matrix, and $N$ is the number of frames.

# 4 Baselines

## 4.1 Visual Odometry Methods

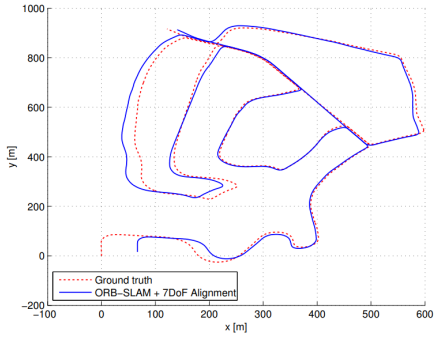
We assessed two visual odometry methods, one based on classical algorithms and the other on learning-based approaches. Their evaluations are presented in the following sub-sections.

### 4.1.1 ORB-SLAM

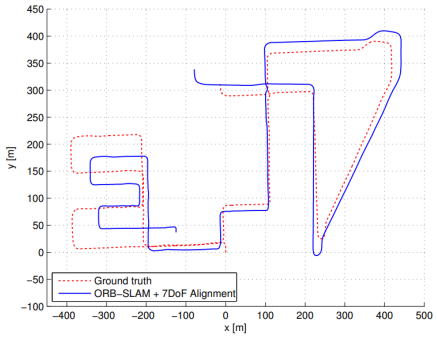
ORB-SLAM2 (Mur-Artal and Tardós, 2017) is a powerful and versatile visual odometry system known for its robust performance across various camera setups, including monocular, stereo, and RGB-D cameras. It was evaluated as a classical unimodal (visual) odometry baseline due to its wide adoption and success.

The algorithm operates with three main components: tracking, local mapping, and loop closure. Tracking estimates camera pose in real-time, local mapping updates the map using keyframes and landmarks, and loop closure detects previously visited locations, reducing positional drift over long distances. Designed for efficiency, ORB-SLAM2 runs in real-time on standard CPUs, making it practical for real-world applications.

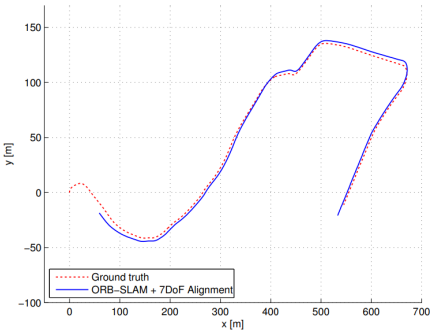
Key features of ORB-SLAM2 include robust loop closure capabilities that enhance long-term accuracy and adaptability to different environments. Its performance on the KITTI (Geiger et al., 2012) odometry benchmark—a widely used dataset for outdoor visual odometry evaluation—demonstrates ORB-SLAM2's scalability and effectiveness, with an average translational error of 1.15% and a rotational error of 0.0027 degrees per meter for stereo input. The system consistently performs well across sequences, showcasing its resilience to various environmental challenges and confirming its reliability for large-scale environments. Some interesting results of the interesting inference runs are shown in figure 1.




(a) Sequence 02




(b) Sequence 08




(c) Sequence 10

Figure 1: Results of ORB-SLAM2 on KITTI Sequences 02, 08, 10. It is interesting to note the performances around loop crossings and long-tail trackings.

### 4.1.2 TartanVO

TartanVO (Wang et al., 2020a) is the closest unimodal baseline to our work, as it uses the same two-frame formulation for odometry and is trained on the predecessor to the dataset we plan to use, TartanAir (Wang et al., 2020b). It also uses optical flow to establish correspondences between frames for pose estimation. Our goal is to match or exceed TartanVO's performance when trained on only image data, and with the addition of LiDAR data, the performance should improve significantly as LiDAR provides crucial information, enabling 3D grounding and more accurate pose estimation.

The intrinsic metric we want to test against for TartanVO is their flow loss. Since we also plan on using flow as a method or correspondence computation, we analyze the effects of learning from a flow pre trained flow model, GMFlow, and how this affects the motion prediction.

As shown in Figure 3, we see that flow has an interesting relationship with the overall motion loss.

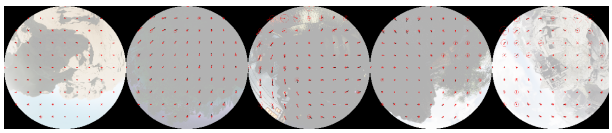


Figure 2: Example intermediate flow visualizations from TartanVO

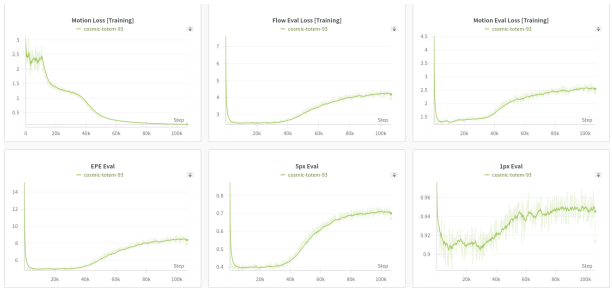



Figure 3: TartanVO motion and flow metrics during validation

The model is initialized with pre-trained flow and is trained using MSE on the motion. We see in the first plot that training motion loss starts off small and decreases, and in the second plot the flow validation loss starts off small, and in the third plot, the motion validation plot starts off small. This shows that flow can be a strong inductive bias to initialize odometry. More interestingly, as training progresses and the model overfits, we see that the training motion loss decreases while the validation motion and flow losses increase, showing that there is a tight correlation between the model's ability to perform optical flow and its ability to predict pose.

Another test we wanted to do with TartanVO is checking the scales of the rotation and translation components of the loss. The overall training objective in the sum of the rotation and translation losses and, as shown in Figure 4, the rotation loss seems to be lower than the translation. We will need to balance these two losses properly to achieve better performance, or the model will be biased toward learning rotations over translations.

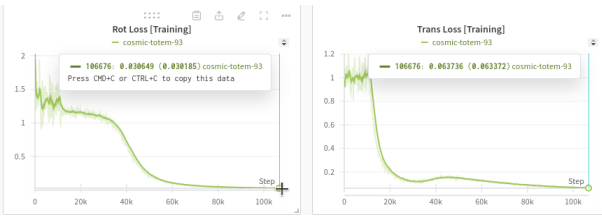



Figure 4: TartanVO motion losses for rotation and translation

## 4.2 LiDAR Odometry Methods

We evaluated two LiDAR odometry methods: a classical approach, which is a variant of ICP SLAM that utilizes ScanContext for loop detection and miniSAM for loop closure, and a learning-based method, EfficientLO-Net (Li et al., 2020b). The evaluations of both methods are presented in the following sub-sections.

### 4.2.1 ICP-SLAM

A SLAM pipeline, evaluated using only poses as states, was tested with ICP SLAM for odometry, Scan Context for loop detection, and miniSAM-based graph optimization on the KITTI dataset. The entire trajectory was empirically assessed to understand the pipeline's performance. For ICP, random downsampling with 7000 points was used, while the parameters for Scan Context were set to: Ring = 20, Sector = 60, and 30 ring key candidates.

Figure 5 illustrates the pipeline's performance on KITTI trajectory 2, with a loop closure threshold of 0.11. In this case, the loop was correctly detected, and the trajectory was optimized, aligning with the expected performance. Figures 6 show the performance on KITTI trajectory 8 with loop detection thresholds of 0.07 and 0.20, respectively. In this case, either no loop was detected or a false loop was identified. This suggests that Scan Context may struggle with loop detection in scenarios where there is a significant change in lane level, as observed in KITTI trajectory 8.

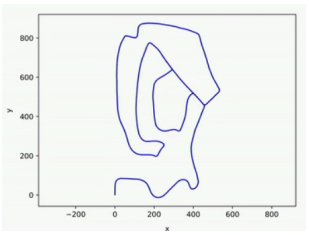


Figure 5: ICP SLAM on Trajectory 2: Loop Detected

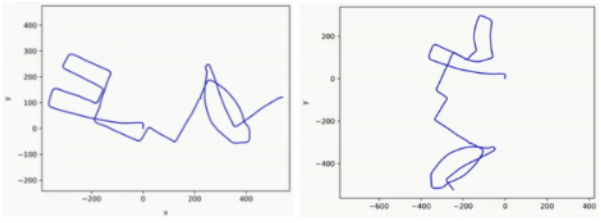


Figure 6: ICP SLAM on Trajectory 8: Left - Threshold 0.007, Right - Threshold 0.20

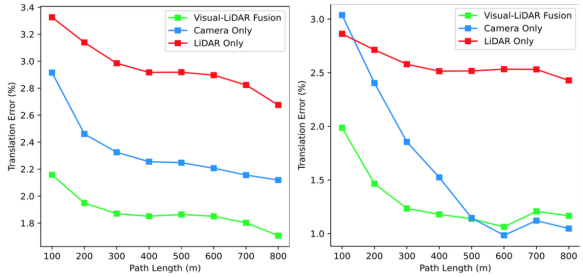



Figure 7: Translation errors averaged across sub-sequences of various lengths (100, 200, ..., 800 meters) within sequences 09 and 10 of the KITTI dataset. The performance results for the LiDAR-only method (red lines), camera-only method (blue lines), and LiDAR-camera fusion method (green lines) are displayed.

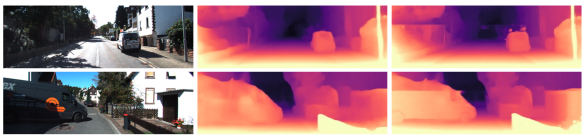


Figure 8: Effectiveness of visual-LiDAR fusion for better estimation of dense depth map. Each row contains one example of the improvement in RGB image frame, DepthNet estimation, fusion result, respectively. Top row: attachments to the roof of the vehicle exist in the fused map where DepthNet is not able to estimate it. Bottom row: Vehicle boundaries are more noticeable and DepthNet is fooled due to graphics at the side hood of the van.

#### 4.2.2 EfficientLO-Net

Efficient LO-Net was evaluated as a learning-based lidar odometry method. The average translational error(%) for sequences 00-05, over distances ranging from 100 to 800 meters, remained below 0.55%, with the rotational error (deg/m) not exceeding 0.38. In contrast, for Sequence 08, the translational error increased to 1.14%. Sequence 08 presents a particularly challenging urban environment, characterized by sharp turns, intersections, dynamic obstacles like moving vehicles, and varying road structures. These elements likely posed significant challenges for Efficient LO-Net, making it difficult to maintain accurate feature associations across frames. The presence of dynamic objects and the complex structure of the scene likely contributed to increased translational drift, as the model may have struggled to differentiate between the vehicle's motion and that of surrounding objects and to manage occlusions and variable feature distributions in the lidar data.

### 4.3 Multimodal Odometry Methods

#### 4.3.1 H-VLO

Since H-VLO (Liu et al., 2024a) demonstrates robust performance by effectively combining LiDAR and camera modalities, we leverage its structure to examine the benefits of integrating these modalities in a LiDAR-camera fusion method. This approach is contrasted with single-modality methods that rely solely on either LiDAR or camera data, allowing us to isolate and understand the unique advantages provided by each data source as well as their combined effect. Figure 7 illustrates the translational errors averaged across sub-sequences of various lengths (100, 200, ..., 800 meters) within sequences 09 and 10 of the KITTI dataset. The performance results for the LiDAR-only method (red lines), camera-only method (blue lines), and LiDAR-camera fusion method (green lines) are displayed.

The LiDAR-only method, represented by the red lines, yields the least accurate trajectory estimates, showing that depth information alone is insufficient for precise motion estimation and mapping, particularly in scenarios where detailed semantic cues are essential for distinguishing objects and understanding scene context. Without these semantic details, the LiDAR-only approach struggles, resulting in higher translational errors, particularly over longer sub-sequence lengths.

On the other hand, the camera-only method (blue lines) performs notably better, benefiting from the rich semantic information inherent in RGB images. Camera data captures texture, color, and object detail, which enhances scene interpretation and aids in predicting object movements and scene transitions. As a result, the camera-only method achieves significantly lower translational errors compared to the LiDAR-only approach, especially in sub-sequences where scene complexity demands higher contextual understanding. However, despite its advantages, the camera-only method still encounters limitations in environments where depth perception is crucial but cannot be accurately derived from monocular RGB images alone.

By combining the semantic information from camera data with the depth and structural detail from LiDAR data, the LiDAR-camera fusion approach (green lines) harnesses the strengths of both modalities. This fusion achieves superior results by providing a more comprehensive representation of the environment: LiDAR contributes precise spa-

tial geometry, while the camera supplies detailed scene semantics. Leveraging the fusion structure within H-VLO, the LiDAR-camera fusion method demonstrates the lowest translational errors across all sub-sequence lengths, suggesting that integrating these complementary data sources not only improves trajectory accuracy but also enhances robustness in complex environments with varying levels of occlusion, lighting, and scene intricacies.

#### 4.3.2 DVLO

Most learning-based approaches, such as H-VLO (Liu et al., 2024a), focus on feature-level fusion but often fail to capture the fine-grained pixel-to-point correspondences required for precise pose estimation. These approaches are further challenged by the inherent structural differences between sparse LiDAR points and dense camera pixels, resulting in data misalignment that limits the effectiveness of multi-modal fusion.

To address these limitations, DVLO (Liu et al., 2024b) introduces a novel local-to-global fusion network with bi-directional structure alignment, specifically designed to enhance the integration of LiDAR and visual features for more accurate pose estimation. In particular, DVLO clusters neighboring visual information in 2D images by projecting LiDAR depth data onto the 2D image plane. This clustering enables effective alignment of LiDAR points with visually similar pixels, improving spatial consistency.

As shown in Figure 9, pixels with similar texture information (yellow regions) are clustered by calculating the point-wise cosine similarity with designated cluster centers (red dots). This approach ensures that visual information is more cohesively grouped with corresponding depth data, leading to a more refined fusion of LiDAR and visual features and ultimately resulting in enhanced pose estimation accuracy.

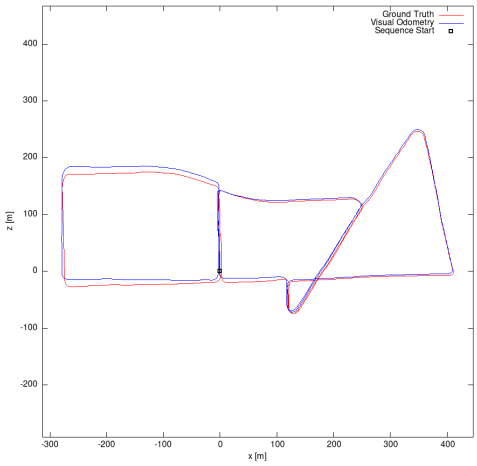

Figure 9: Visualization of DVLO's local clustering-based fusion mechanism within a certain cluster. Red points indicate the 2D positions of cluster centers. The yellow regions are clustered pixels around each center.

#### 4.3.3 SDV-LOAM

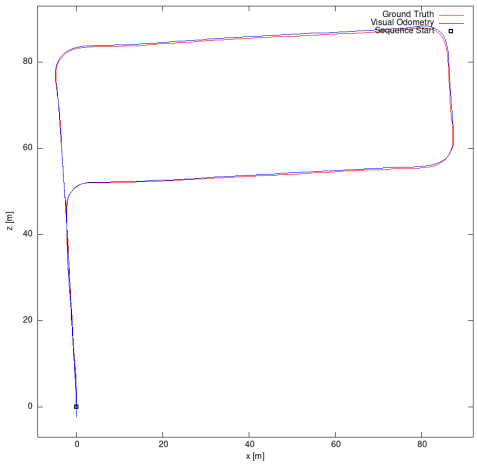
SDV-LOAM (Semi-Direct Visual-LiDAR Odometry and Mapping) (Yuan et al., 2023) was evaluated as one of the classical multimodal baselines. It is a robust odometry method that effectively combines visual and LiDAR data for accurate pose estimation and mapping. The system addresses common challenges in visual-LiDAR fusion by incorporating a semi-direct visual odometry approach and an adaptive sweep-to-map LiDAR odometry.

The visual module of SDV-LOAM employs a novel technique that directly extracts high-gradient pixels where 3D LiDAR points project for tracking, avoiding the need for explicit 3D-2D depth association. To handle large-scale differences between matching frames, it uses a point matching with propagation method, which propagates points from a host frame to an intermediate keyframe closer to the current frame. This approach significantly reduces scale differences and improves tracking accuracy.

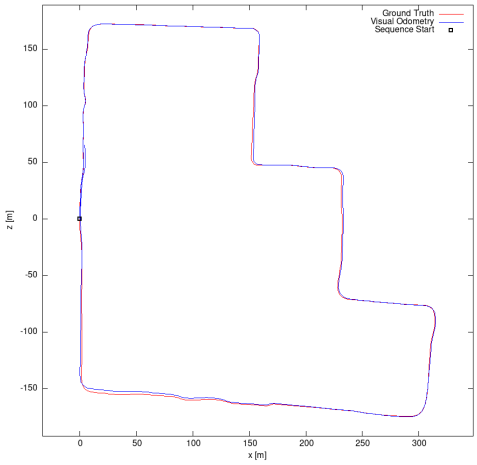
On the LiDAR side, SDV-LOAM introduces an adaptive sweep-to-map optimization method that dynamically chooses between optimizing 3 horizontal degrees of freedom (DOF) or 6 full DOF pose based on the richness of geometric constraints in the vertical direction. This adaptive approach helps reduce pose estimation drifts, particularly in the vertical direction. Some interesting results of the interesting inference runs are shown in figure 10.




(a) Sequence 13




(b) Sequence 14




(c) Sequence 15

Figure 10: Results of SDV-LOAM on KITTI Sequences 13, 14, 15 (Geiger et al., 2012).

## 5 Proposed Model

### 5.1 Overall model structure

The model first consists of a preprocessing step for both the RGB images and LiDAR point clouds. The images are sent through an optical flow regression model to get an intermediary flow representation, and the point clouds are projected onto the image to enforce explicit matching between the 2D pixels and the corresponding 3D points. These processed modalities are then sent through their respective ViT encoders and then combined with a ViT decoder and a projection head to output pose. The model diagram is presented in Figure 11. Some key changes from out initial proposed model include projecting our LiDAR points into pointmaps instead of depth, and adding the initial RGB images to the flow as input.

### 5.2 Unimodel Processing

#### 5.2.1 RGB

The key information we seek from the RGB input is the presence of strong visual features that allow us to match between frames and establish correspondences. To achieve this, we process the images by running them through a flow model, which computes optical flow to provide dense correspondences between frames. TartanVO trains its odometry model solely using a flow model as its encoder, so we have evidence that this approach retrains information needed for odometry. We then concatenate the two frames with their corresponding flow to form the final processed input from the RGB data to retain all information for the decoder.

#### 5.2.2 LiDAR

We preprocess the LiDAR point cloud by projecting it onto the image space, such that the 3D points are in the shape of the image but contain $(x, y, z)$ values instead of $(R, G, B)$ as shown in DVLO (Liu et al., 2024b). This "point-map" 3D representation has been shown to be very effective in relating 2D RGB images to 3D points, as seen in works like DUSt3R (Wang et al., 2023). Since the pointmap is in the same shape as the RGB image, the points can be concatenated, establishing strong correspondence between RGB pixels and their corresponding LiDAR points. We can then use the same encoder as used for the RGB images on these point maps, thus making the shared space between the two modalities more homogeneous.

### 5.3 Multimodal Decoder

The decoder is a ViT decoder architecture that performs cross-attention between the intermediate flow and pointmap encodings. This mechanism will allow for effective global information sharing between the two modalities. The output of this module is then sent to separate heads for rotation and translation similar to the TartanVO architecture (Wang et al., 2020a), enabling precise recovery of pose parameters. We have also tried using a ResNet backbone as the decoder model, and we observed that the model performance is similar. We believe the strong pixel-wise correspondences of the concatenated flow, RBG, and LiDAR data enables for easier modality fusion, making the exact architecture of the decoder less significant.

### 5.4 Loss Functions

The primary objective used to train the model will be a simple mean-squared error on the predicted translation and rotation.

$$L_{pose} = \|\hat{T} - T\| + \|\hat{R} - R\| \tag{1}$$

This is based on the objective used for TartanVO (Wang et al., 2020a), but uses absolute translation error instead of up-to-scale error since we wish to determine absolute pose.

An auxiliary loss that we believe greatly helps in inducing relevant biases for the model are optical flow.

$$L_{flow} = \sum_{i=1}^{N} ||\mathbf{f}_{gt} - \mathbf{f}_i||_1 \tag{2}$$

Without the flow loss, we show that the pretrained RGB flow model will forget its flow capabilities, losing its inductive biases for finding correspondences. This auxiliary loss ensures that the model retains that flow knowledge while also learning to estimate pose. This is shown in Figure 32

In combination, the final loss will look as follows:

$$L_{total} = L_{pose} + \lambda_1 * L_{flow} \tag{3}$$

We hypothesize that the composition of these relevant auxiliary tasks will induce a robust and accurate objective for odometry.

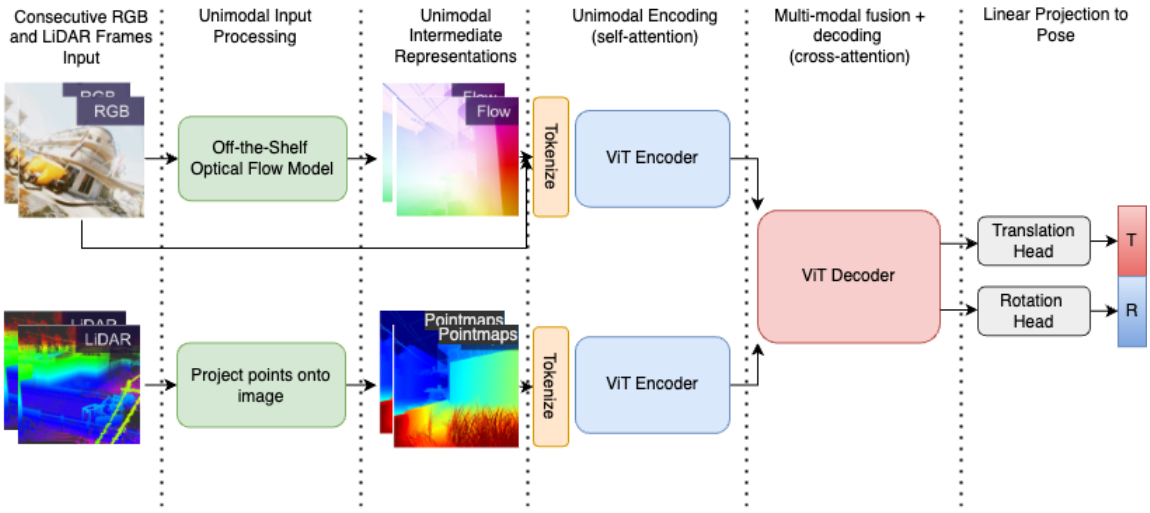



Figure 11: The VOL model diagram. Given RGB and LiDAR frames from two consecutive timesteps, the model computes optical flow and aligns the LiDAR points to the point map. These intermediate representations are then processed by the separate encoder and combined into a decoder, and lastly sent through translation and rotation heads to recover pose.

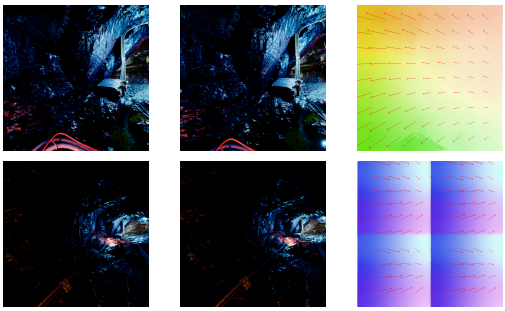


Figure 12: The top row shows the model intermediate flow output during training at step 1 when training start. We observe that the pretrained flow model outputs reasonable flow. The bottom row shows the flow output after 5000 steps without flow supervision. We see that the flow output degrades, and we also observed that the model's performance on odometry also drops.

### 5.5 Changes to training data

Given that we trained solely on a synthetic dataset, some key augmentations to the training data were required to enable better generalization. For the LiDAR point clouds, dropout was applied to simulate their inherent sparsity, as real-world LiDAR data often lacks points for every pixel due to occlusions and sensor limitations. Additionally, random padding was introduced to the RGB images during training, enabling the model to handle variable aspect ratios and better generalize to diverse real-world RGB data. These augmentations were crucial for bridging the gap between synthetic and real-world scenarios. Visualizations of the final training data can be seen in Figure 13

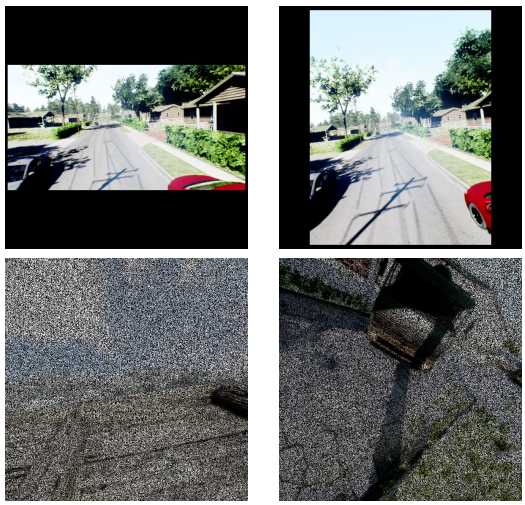


Figure 13: The top row shows the random padding augmentation to enable random aspect ratio inputs. The bottom row shows LiDAR dropout applied to RGB images for visualization.

### 5.6 Hyperparameters and their effects

Key hyperparameters were the loss alphas, the addition of the flow loss, and the training curriculum. As shown in Figure 32, the addition of flow supervision, and thereby the weighting of the flow loss, played a significant role in the model's ability to extract useful features from the RGB images. Additionally, we found that a key technique needed to successfully learn seperate skills such as flow and odometry was curriculum learning. Our train-

ing curriculum for flow was designed to progressively guide the model toward effective odometry performance. Initially, we trained the model with ground truth flow, enabling the decoder to learn pose prediction from accurate flow and LiDAR inputs. Next, we used a frozen, pretrained GMFlow model, allowing the decoder to adjust to predicted flow while preserving the pretrained flow representations. Finally, we unfroze the flow module and added flow supervision, training the model end-to-end to fine-tune flow predictions for odometry. This staged approach ensured the model effectively balanced learning flow and pose while adapting to the training dataset. More details, along with visual results, are shown in the Analysis section.

| **Method** | 06 | | 07 | | 09 | | 10 | |
|---|---|---|---|---|---|---|---|---|
| | $t_{rel}$ | $r_{rel}$ | $t_{rel}$ | $r_{rel}$ | $t_{rel}$ | $r_{rel}$ | $t_{rel}$ | $r_{rel}$ |
| *Visual Odometry Methods:* | | | | | | | | |
| ORB-SLAM (-) | 18.68 | 0.26 | 10.96 | 0.37 | 15.3 | **0.26** | 3.71 | **0.3** |
| TartanVO (*) | 4.72 | 2.95 | 4.32 | 3.41 | 6.0 | 3.11 | 6.89 | 2.73 |
| *LiDAR Odometry Methods:* | | | | | | | | |
| ICP-SLAM (-) | 1.95 | 1.59 | 5.17 | 3.35 | 6.93 | 2.89 | 8.91 | 4.74 |
| LO-Net (*) | 1.04 | 0.69 | 0.71 | 0.50 | 2.12 | 0.77 | 1.80 | 0.93 |
| *Multimodal Odometry Methods:* | | | | | | | | |
| H-VLO (*) | 0.75 | 0.30 | 0.79 | 0.48 | 1.89 | 0.34 | 1.36 | 0.43 |
| DVLO (*) | **0.33** | **0.17** | **0.46** | **0.33** | 0.85 | 0.36 | 0.88 | 0.46 |
| DV-LOAM (-) | 0.65 | 0.33 | 0.51 | **0.33** | 0.73 | 0.32 | 0.87 | 0.38 |
| SDV-LOAM (-) | 0.50 | 0.27 | 0.84 | 0.53 | **0.63** | 0.34 | **0.68** | 0.41 |
| *Proposed Method:* | | | | | | | | |
| **Ours** | 0.4345 | 0.49 | 0.1639 | 0.495 | 0.343 | 0.495 | 0.234 | 0.495 |

# 6 Results (1 page)

## 6.1 Training Results

We successfully trained the model and observed that it learns the task rather quickly, in surprisingly few steps. We provide the training and validation losses in Figure 14 and 15, respectively. The pose losses dropped sharply during the initial stages of training and soon stabilized at values that are competitive with the training losses reported for TartanVO. This rapid convergence highlights the model's ability to effectively utilize the flow and LiDAR inputs to learn pose estimation efficiently, demonstrating the power of these modalities and the aligned representation for odometry learning. We also observe that the training loss curves and the validation loss curves have similar shape and values, indicating that the model is not overfitting.

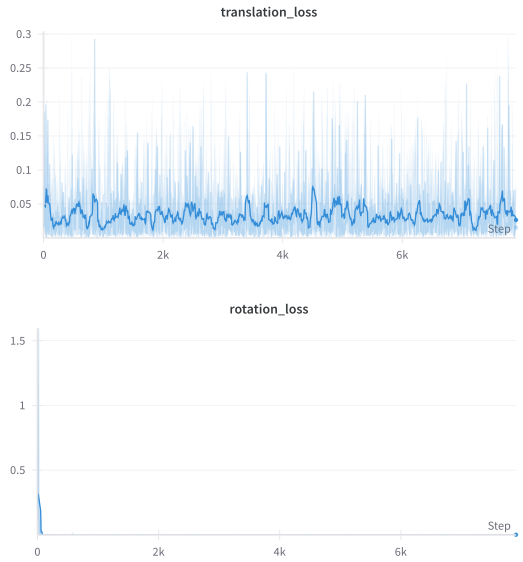



Figure 14: The top plot shows the training translation loss and the bottom plot shows the training rotation loss.

## 6.2 KITTI Evaluation

We evaluated the model architecture using the KITTI odometry dataset, primarily to benchmark its performance against other baselines. Table 2 summarizes the model's performance on sequences 6 to 10, evaluated over the initial 1000 steps. The Average Translation Error (ATE) across all sequences ranges from 0.16m to 0.435m, while the Average Rotation Error (ARE) falls between 0.490 rad and 0.495 rad.

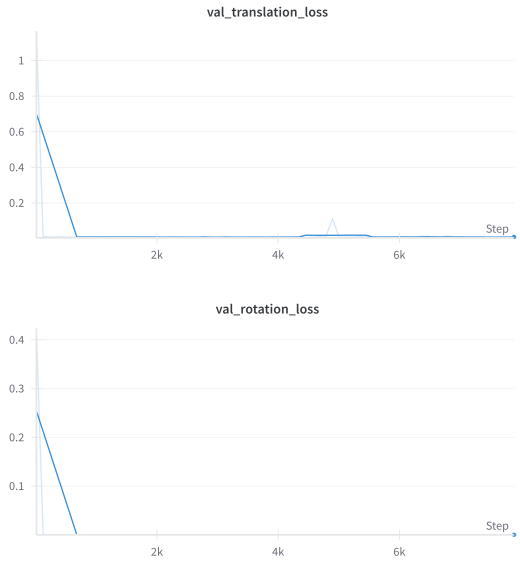



Figure 15: The top plot shows the validation translation loss and the bottom plot shows the validation rotation loss.

### 6.2.1 Required Steps

- **DataLoader:** To facilitate this evaluation, we developed a KITTI sequence loader to load the requested data sequences along with the specified sequence length. To ensure compatibility with the PyTorch DataLoader, we designed a pipeline that loads images, lidar point clouds, and calibration data. This data includes the extrinsics between the lidar and camera, which were used to transform the lidar information into the camera frame. Additionally, the camera intrinsics were provided as output, enabling the evaluation script to project lidar points onto the camera image.

- **Evaluation Script:** For model inference on the KITTI dataset, the input data must be correctly formatted. Since our training dataset images differed in shape from those in the KITTI dataset, we resized the images while maintaining the aspect ratio to match the model's required input shape. Figure 3 illustrates a KITTI image, while Figure 2 shows the reshaped input image fed into the model. Similarly, for the lidar input, we projected the lidar points onto the image plane using camera intrinsics. Afterward, the projected lidar points were rescaled to match the input shape of the model. Once this preprocessing was complete, we proceeded with model inference testing.

Figure 16: KITTI Sequence 0, Image 1

Figure 17: Processed Image for the model, Image 1

Further, some prediction (upto 100 frames) error plots for the challenging sequences are shown in figures 25 and 21.

## 7 Analysis (2 pages)

Figure 26 - 30 presents the Average Translation and Rotation Error plots for KITTI dataset sequences 6 to 10. The model demonstrated better performance in predicting the rotational component of visual odometry compared to the translational component. Notably, sequences 7 and 10 exhibited lower Average Translation Errors compared to other trajectories. This could be attributed to the challenging nature of these sequences, which include abrupt turns that were likely underrepresented in the training dataset. Additionally, the model's performance for both the translation and rotation components of odometry is influenced by its inability to account for real-world sensor dynamics, such as lidar skew during vehicle motion and sensor noise, as it was trained on a synthetic dataset. We provide the

Table 2: Average Translation Error (ATE) and Average Rotation Error (ARE) for KITTI Sequences 6 to 10

| Sequence | ATE | ARE (rad) |
|---|---|---|
| 6 | 0.4345 | 0.490 |
| 7 | 0.1639 | 0.495 |
| 8 | 0.2570 | 0.495 |
| 9 | 0.3430 | 0.495 |
| 10 | 0.2340 | 0.495 |

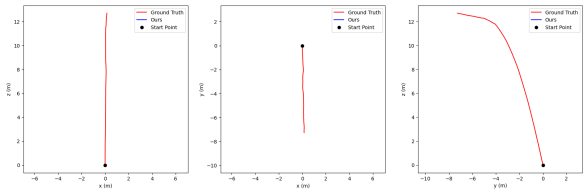


Figure 18: Predicted Path

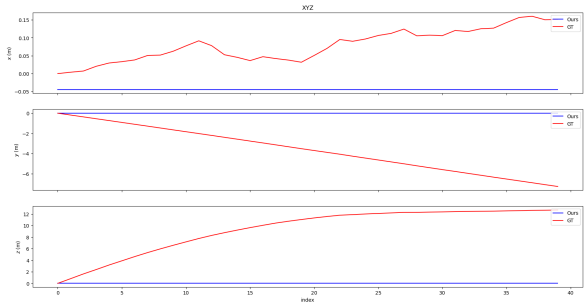


Figure 19: Translation Error

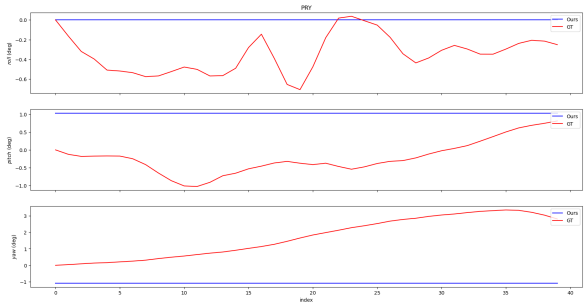


Figure 20: Rotation Error

Figure 21: Prediction Errors for Sequence 08.

comparison against the baselines in Table 5.6. We observe that our model is competitive with state-of-the-art learning-based visual-LiDAR method even though even it was trained on KITTI, so its zero-shot generalization is very promising.

### 7.1 Intrinsic Metrics

The key intrinsic metric we measured was flow, which plays a crucial role in ensuring accurate correspondences between frames. It was essential to retain flow information during training while carefully balancing the extent to which the model focuses on odometry without forgetting flow. We achieved this balance through a carefully designed curriculum training strategy. Initially, we trained the model with the flow module frozen and no aug-

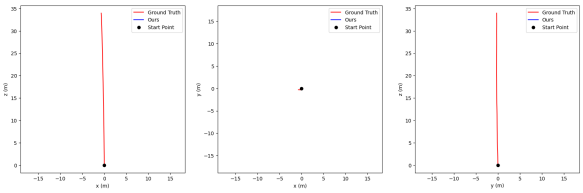



Figure 22: Predicted Path

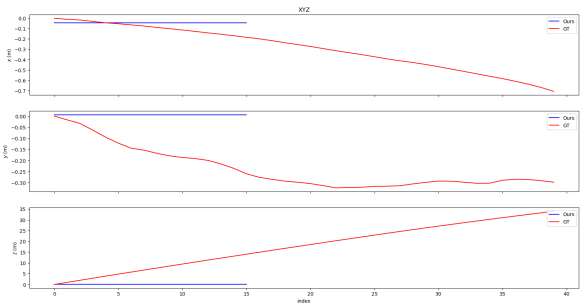



Figure 23: Translation Error

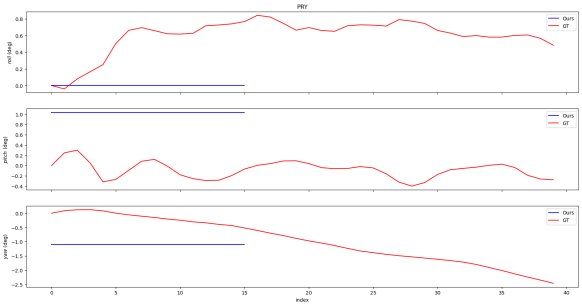



Figure 24: Rotation Error

Figure 25: Prediction Errors for Sequence 03.

mentations applied, allowing the model to focus on foundational learning. Subsequently, we introduced augmentations and unfroze the flow module, adding flow supervision to guide learning. This approach significantly improved model performance, as the flow outputs demonstrated that the model successfully learned the augmentations, developing some invariance to varying aspect ratios as shown in Figure 32. Consequently, flow served as a valuable auxiliary measure for evaluating robustness to these augmentations. We observe that the flow loss also closely follows the odometry loss curve, indicating a strong correlation as shown in Figure 31.

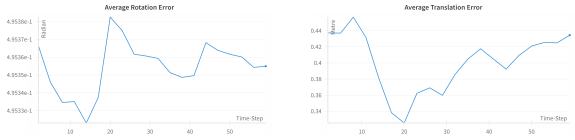



Figure 26: Average Rotation(left) and Translation Error Plots(right) for Sequence 6

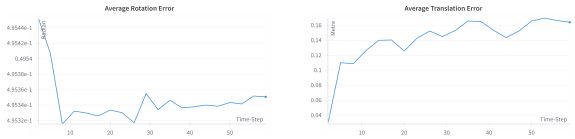



Figure 27: Average Rotation(left) and Translation Error Plots(right) for Sequence 7

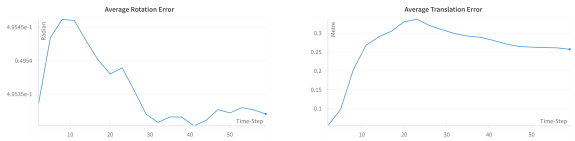



Figure 28: Average Rotation(left) and Translation Error Plots(right) for Sequence 8

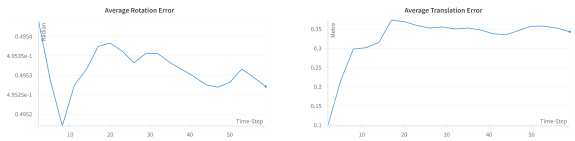



Figure 29: Average Rotation(left) and Translation Error Plots(right) for Sequence 9

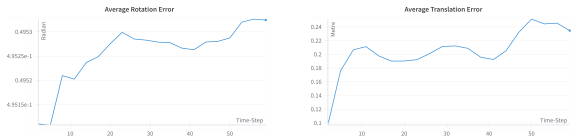



Figure 30: Average Rotation(left) and Translation Error Plots(right) for Sequence 10

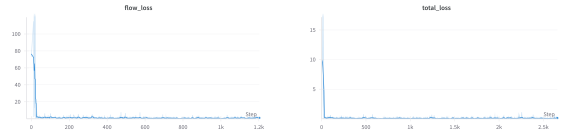



Figure 31: The first image shows the flow loss and the second image shows the total loss. Observe that both loss curves are highly correlated.

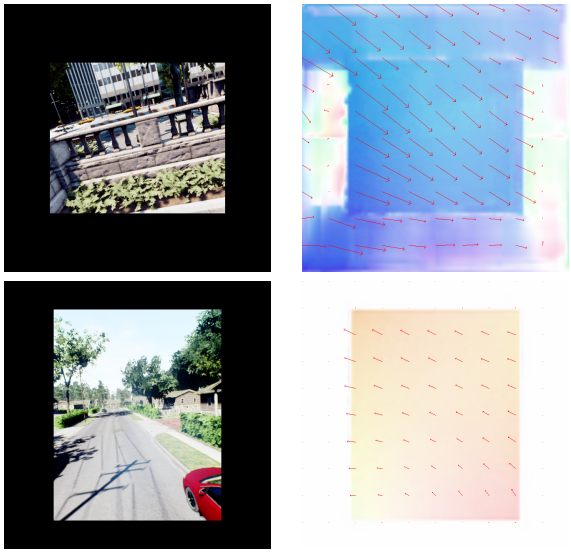


Figure 32: The top row shows the pretrained flow model's flow output at the start of training, and the second row shows the model's flow output at the the end of training. The model shows that it has learned to be invariant to different aspect ratios.

Figure 33: The top image displays the model input from the left image of KITTI Odometry Trajectory 10 at 10 seconds, while the bottom image shows the corresponding optical flow output.

### 7.2 Qualitative Analysis and Examples

In figure 33, the bottom image, illustrates the optical flow representation while executing the TartanVO baseline on KITTI Odometry Trajectory - Sequence 10. This visualization provides valuable qualitative, insights showing that atleast our flow model is able to generalize to different data. Because the largest distribution shift we expect to see if from RGB images, and given that flow seems to be robust to this shift, we are more confident in our model's ability to perform in out-of-distribution settings. We also projected the model's odometry into 2D, and visualized the model's predicted odometry as flow with motion arrows. We provide more visualizations of the model's odometry outputs in Figure 34.

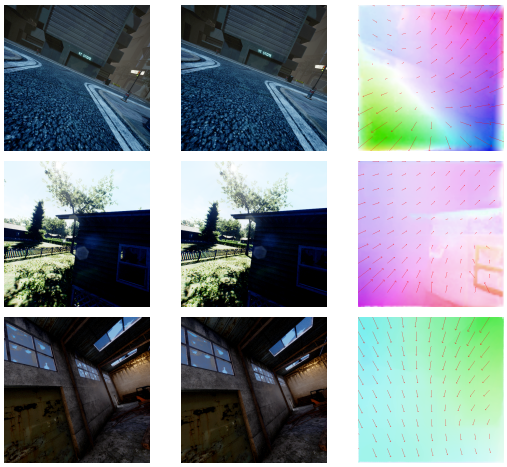


Figure 34: Model odometry output vizualizations.

## 8 Future work and Limitations (1 page)

Our work takes promising steps towards addressing the current limitations of visual-LiDAR odometry, but it does currently have some drawbacks of its own. We detail some of the key problems we hope to address in future works. **Projection Issues** One major limitation of our approach lies in the projection of 3D 360-degree LiDAR points onto a 2D image plane. This process inherently results in the loss of valuable information, as LiDAR points outside the camera's field of view are ignored. While this method provides direct correspondences between LiDAR points and image pixels, it neglects the rich spatial context of unprojected points. Future work could explore strategies to better leverage the full 3D point cloud, potentially through 3D feature learning or hybrid representations that combine 2D and 3D modalities.

**Sim-to-Real Gap** Our reliance on synthetic training data introduces a notable sim-to-real gap. Synthetic LiDAR data assumes perfect 3D points, whereas real-world LiDAR sensors are affected by latency, noise, and skew. This distribution shift is particularly significant for LiDAR, as real-world sensors vary in type, configuration, and noise characteristics. While the flow-based intermediate representation helps mitigate the sim-to-real gap for RGB data by bounding the distribution shift, the LiDAR data remains more susceptible. Future work should focus on incorporating domain adaptation techniques or leveraging real-world LiDAR data to enhance model robustness and generalizability.

**Dynamic Objects** Our model implicitly disregards dynamic objects in the scene, treating them as outliers or noise. While this simplifies the learning process, it introduces limitations when dynamic objects play a significant role in the scene or occlude key features. Future research could address this by explicitly modeling dynamic objects, allowing the system to adapt to their presence and even utilize their motion for additional context in odometry estimation.

**Different Datasets** Our training and evaluation rely on a limited set of datasets, which may restrict the model's ability to generalize across different camera parameters, image shapes, and LiDAR configurations. Adding diverse datasets with varying sensor setups—such as different camera focal lengths, aspect ratios, and LiDAR types—can help the model become more robust to diverse real-world scenarios. This would enable better handling of varied intrinsic and extrinsic parameters, further reducing overfitting to specific dataset characteristics.

**LiDAR Processing** Currently, our approach directly projects LiDAR points onto the 2D image frame for correspondences, without additional processing of the LiDAR data. This simplistic method may overlook useful spatial and structural information in the LiDAR point cloud. Future work could involve explicitly encoding LiDAR data into higher-dimensional features, such as LiDAR-specific embeddings or learned 3D features. These representations could improve the model's understanding of spatial geometry and enhance its performance in odometry estimation. Techniques from works like point cloud learning could be explored to better utilize LiDAR data.

**Better Fusion** Our current approach relies on cross-attention mechanisms applied to concatenated inputs of image, flow, and LiDAR data. While effective to some extent, this implicit fusion may not fully capture the complementary information between modalities. More explicit fusion techniques, such as learning shared feature spaces or designing fusion layers that integrate geometric and visual features at multiple stages, could significantly improve performance. Future work should explore tailored fusion architectures that better align and combine the strengths of each modality, leading to richer and more informative feature representations for odometry estimation.

By addressing these limitations, future iterations of the system can achieve better performance and broader applicability in real-world, dynamic, and diverse environments.

## 9 Ethical Concerns and Considerations

Visual-LiDAR odometry, especially in its application to autonomous vehicles, raises some ethical concerns. Intentionally, misuse of the technology for surveillance or military purposes can infringe on privacy and human rights. Unintentionally, biases in training data, such as underrepresentation of diverse environments or certain groups of people, can lead to accidents or harm in scenarios not well-represented during development. Direct ethical concerns include the potential for malfunctions causing harm to passengers or pedestrians, while indirect concerns involve societal impacts, such as job displacement in industries reliant on human drivers. Ensuring robust, equitable, and transparent development and deployment of these technologies is critical to addressing these challenges.

## 10 Team member contributions

**Pujith Kachana** contributed the proposed model architecture. Wrote the dataloading and processing code, the flow prediction code, the model and training code, the visualization code, and trained the model checkpoint. Wrote the proposed model, limitations, and ethical concerns section, and contributed to writing of introduction, related works, baselines, results, and analysis sections.

**Shashwat Chawla** contributed to the ideation of the project and design of the proposed model architecture, implemented the initial lidar projection pipeline for model training, helped in debugging the model decoder, and developed the KITTI evaluation and preprocessing code along with visualization plots. Also contributed to the Results, Analysis, and Future Work sections of the report.

**Ming-Fong Li** contributed to setting up the report, training code, introductions, literatutre review, evaluations codes and model implementations.

**Yatharth Ahuja** contributed to model brainstorming, baseline analysis and exploration, literature review, problem setup, prediction evaluation, and visulaizations. Also worked on setting up the report and evaluation codes.

## References


Yi An, Jin Shi, Dongbing Gu, and Qiang Liu. 2022. Visual-lidar slam based on unsupervised multi-channel deep neural networks. *Cognitive Computation*, 14(4):1496–1508.

Siyu Chen, Kangcheng Liu, Chen Wang, Shenghai Yuan, Jianfei Yang, and Lihua Xie. 2024. Salient sparse visual odometry with pose-only supervision.

Jakob Engel, Vladlen Koltun, and Daniel Cremers. 2017. Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):611–625.

Jakob Engel, Thomas Schöps, and Daniel Cremers. 2014. Lsd-slam: Large-scale direct monocular slam. In *European Conference on Computer Vision (ECCV)*. Springer.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Johannes Graeter, Alexander Wilczynski, and Martin Lauer. 2018. Limo: Lidar-monocular visual odometry. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 7872–7879. IEEE.

Shi-Sheng Huang, Ze-Yu Ma, Tai-Jiang Mu, Hongbo Fu, and Shi-Min Hu. 2020. Lidar-monocular visual odometry using point and line features. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1091–1097. IEEE.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. 2024. Grounding image matching in 3d with mast3r.

Qing Li, Shaoyang Chen, Cheng Wang, Xin Li, Chenglu Wen, Ming Cheng, and Jonathan Li. 2020a. Lo-net: Deep real-time lidar odometry.

Zhen Li, Yiming Wang, Yi Xu, Yang Chen, Yung-Hsiang Liu, Xian Zhu, and Yi Li. 2020b. Lo-net: Deep real-time lidar odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2065–2074.

Jiuming Liu, Dong Zhuo, Zhiheng Feng, Siting Zhu, Chensheng Peng, Zhe Liu, and Hesheng Wang. 2024a. Dvlo: Deep visual-lidar odometry with local-to-global feature fusion and bi-directional structure alignment.

Jiuming Liu, Dong Zhuo, Zhiheng Feng, Siting Zhu, Chensheng Peng, Zhe Liu, and Hesheng Wang. 2024b. Dvlo: Deep visual-lidar odometry with local-to-global feature fusion and bi-directional structure alignment.

Raul Mur-Artal and Juan D Tardós. 2017. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5):1255–1262.

Raúl Mur-Artal and Juan D. Tardós. 2017. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262.

Tixiao Shan and Brendan Englot. 2018. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain.

Shihao Shen, Yilin Cai, Wenshan Wang, and Sebastian Scherer. 2023. Dytanvo: Joint refinement of visual odometry and motion segmentation in dynamic environments.

Sai Shubodh, Mohammad Omama, Husain Zaidi, Udit Singh Parihar, and Madhava Krishna. 2024. Lip-loc: Lidar image pretraining for cross-modal localization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 948–957.

Zachary Teed, Lahav Lipson, and Jia Deng. 2023. Deep patch visual odometry. In *Advances in Neural Information Processing Systems*, volume 36, pages 39033–39051. Curran Associates, Inc.

Sundararajan Vijayanarasimhan, Simone Ricco, Christoph Schmid, Rahul Sukthankar, and Katerina Fragkiadaki. 2017. Sfm-net: Learning of structure and motion from video. *arXiv preprint arXiv:1704.07804*.

Ignacio Vizzo, Tiziano Guadagnino, Benedikt Mersch, Louis Wiesmann, Jens Behley, and Cyrill Stachniss. 2023. Kiss-icp: In defense of point-to-point icp – simple, accurate, and robust registration if done the right way. *IEEE Robotics and Automation Letters*, 8(2):1029–1036.

Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. 2018. End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks. *The International Journal of Robotics Research*, 37(4-5):513–542.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. 2023. Dust3r: Geometric 3d vision made easy.

Wei Wang, Jun Liu, Chenjie Wang, Bin Luo, and Cheng Zhang. 2021. Dv-loam: Direct visual lidar odometry and mapping. *Remote Sensing*, 13(16):3340.

Wenshan Wang, Yaoyu Hu, and Sebastian Scherer. 2020a. Tartanvo: A generalizable learning-based vo.

Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. 2020b. Tartanair: A dataset to push the limits of visual slam.

Zikang Yuan, Qingjie Wang, Ken Cheng, Tianyu Hao, and Xin Yang. 2023. Sdv-loam: semi-direct visual–lidar odometry and mapping. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9):11203–11220.

Ji Zhang and Sanjiv Singh. 2015. Visual-lidar odometry and mapping: Low-drift, robust, and fast. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2174–2181. IEEE.

Zichao Zhang and Davide Scaramuzza. 2018. A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry.