Literature Review: Self-Supervised Learning and World Models for Autonomous Driving

Tejdeep Chippa
Tandon School of Engineering
New York University
tc4263@nyu.edu

Venkat Kumar Laxmi Kanth Nemala
Tandon School of Engineering
New York University
vn2263@nyu.edu

Abstract

This project explores the use of Joint-Embedding Predictive Architectures (JEPA), specifically I-JEPA, for enabling label-efficient steering control in autonomous driving systems. We aim to pretrain I-JEPA on real-world camera data from the Waymo Open Dataset to learn semantic visual representations without requiring manual annotations. The pretrained encoder will be fine-tuned on a small, labeled subset of steering data and deployed in the CARLA simulator for inference-only evaluation. By experimenting with different training strategies such as frozen and partially fine-tuned encoders, we plan to assess the effectiveness of self-supervised representation learning under domain shift. This work-in-progress emphasizes the potential of JEPA-based models to serve as scalable, modular components in real-world autonomous driving pipelines, reducing reliance on large-scale labeled datasets.

.

1 Introduction

In recent years, self-supervised learning (SSL) has emerged as a powerful paradigm for representation learning, particularly in domains where labeled data is expensive or impractical to obtain. By leveraging the inherent structure and patterns within raw sensory inputs, SSL methods can learn rich, transferable features without the need for manual annotation. This capability is especially relevant in autonomous driving, where raw multimodal data such as camera images and LiDAR scans, is abundantly available, but acquiring fine-grained labels like steering commands, object trajectories, and driver intent is often labor-intensive and costly.

A notable advancement in this space is the Joint-Embedding Predictive Architecture (JEPA), which moves beyond traditional contrastive and generative SSL frameworks. Instead of relying on pixel-level reconstruction or negative sampling, JEPA focuses on predicting high-level latent representations of masked target regions from unmasked context. Models like I-JEPA, built on this principle, leverage a Vision Transformer (ViT) backbone to operate directly on image patches [1], enabling the model to learn spatial and semantic context efficiently. This results in better generalization, reduced reliance on handcrafted augmentations, and improved training stability and efficiency.

These characteristics make I-JEPA a strong candidate for vision-based decision-making systems in autonomous vehicles. Its ability to capture global scene structure without supervision aligns well with the inductive biases required for understanding road geometry, traffic agents, and motion intent, all of which are critical for planning and control tasks.

This project explores the use of I-JEPA, pretrained on the Waymo Open Dataset, as a label-efficient visual backbone for the downstream task of steering angle prediction. Unlike many prior approaches that rely on synthetic environments for both training and evaluation, our system is trained exclusively

on real-world driving data from Waymo and tested in the CARLA simulator under diverse conditions in an inference-only setup. By decoupling the training and evaluation environments, we aim to rigorously assess the generalization capability of self-supervised world models and investigate their viability for scalable, real-world deployment in autonomous driving systems.

2 Problem Definition

The primary downstream task addressed in this project is steering angle prediction for autonomous driving. The objective is to accurately infer the appropriate steering command from raw visual input captured by a front-facing camera. This is formulated as a regression problem, where the model predicts a continuous-valued steering angle corresponding to each input video frame.

The input modality consists of monocular RGB images from the front-facing camera, while the output is the corresponding steering angle, derived from the vehicle's motion data. During the training phase, both input images and steering labels are obtained from the Waymo Open Dataset. In the deployment and evaluation phase, the trained model is used in an inference-only mode within the CARLA simulator, where live front-camera frames are processed to produce steering predictions that are fed into CARLA's vehicle control API.

This task is designed under the following critical constraints:

- Real-Time Inference: The model must generate steering predictions with low latency to ensure safe and responsive control in the simulation environment.
- Cross-Domain Generalization: The model is trained exclusively on real-world Waymo data and is evaluated in the synthetic CARLA environment, necessitating strong generalization across domains and scene distributions.
- Label Efficiency: To reduce the burden of annotation, the model is fine-tuned using only a small subset of labeled Waymo data. The core visual representations are learned through self-supervised pretraining using I-JEPA, allowing effective learning from unlabeled data.

By addressing these constraints, the system is designed to serve as a scalable, efficient, and generalizable solution for vision-based control in autonomous driving applications.

3 Related Work

Early approaches to self-supervised learning (SSL) for visual representation learning have been primarily dominated by contrastive and generative paradigms. Contrastive methods, such as SimCLR and MoCo [2, 3], learn representations by maximizing the agreement between different augmented views of the same image while pushing apart unrelated samples. While effective, these methods are often computationally intensive and rely heavily on the careful design of data augmentations and negative sampling strategies, making them less scalable and brittle across domains.

In contrast, generative SSL approaches, such as Masked Autoencoders (MAE) [4], learn to reconstruct missing portions of the input data. Although these models are less reliant on negative sampling, they tend to prioritize low-level pixel fidelity, which may not align with the semantic understanding required for high-level downstream tasks like steering prediction or behavioral planning.

The Joint-Embedding Predictive Architecture (JEPA) was introduced to overcome these limitations by shifting the focus from input reconstruction to latent representation prediction. Specifically, I-JEPA learns to predict high-level representations of masked image patches using a context encoder [1], without relying on pixel-level targets or handcrafted augmentations [1]. This allows it to capture semantic abstractions that are more aligned with control and decision-making tasks. AD-L-JEPA, a AD-L-JEPA, a modality-specific extension for LiDAR, applies a similar approach [5] by predicting latent Bird's Eye View (BEV) representations, making it highly effective for spatial world modeling in autonomous systems [4].

This project builds upon the I-JEPA image-based framework, positioning it as a scalable and label-efficient alternative to traditional SSL methods in the context of autonomous driving. By leveraging self-supervised pretraining on real-world Waymo data and evaluating performance in CARLA with

minimal supervision, this work explores the transferability and practical utility of JEPA-based representations for real-time control tasks in novel domains.

4 Dataset and Data Processing

4.1 Pretraining Datasets

For self-supervised pretraining, we use the Waymo Open Dataset v1.4.1 [6], which offers a rich and diverse set of urban and suburban driving scenarios across multiple geographic regions, times of day, and weather conditions. The dataset includes high-resolution camera images, dense LiDAR point clouds, and vehicle motion metadata, making it well-suited for learning semantic representations in the context of autonomous driving.

While I-JEPA was originally pretrained on ImageNet-1K, we adapt or extend pretraining to the Waymo dataset to better align with the perceptual demands of real-world driving. We focus specifically on the front-facing RGB camera data, applying context–target masking strategies consistent with I-JEPA. The model is trained to predict latent representations of masked image patches using unmasked context, without relying on any human annotations.

4.2 Downstream Dataset: CARLA

The CARLA simulator is used exclusively for downstream inference-only evaluation. CARLA offers a high-fidelity simulation platform with configurable weather, lighting, and traffic scenarios, allowing for controlled testing of driving policies in diverse environments.

Our trained model, which has never seen CARLA data during training, receives simulated front-camera feeds as input and outputs steering angle predictions in real time. These predictions are passed to the CARLA vehicle control API, enabling us to evaluate the model's cross-domain generalization under realistic control dynamics.

4.3 KITTI vs. Waymo: Comparison and Dataset Choice

Both KITTI and Waymo were considered for this project. While KITTI has been influential [7], it presents limitations that make it less suitable for JEPA-based self-supervised learning and steering control tasks:

- **Dataset Size:** KITTI includes fewer sequences and frames compared to Waymo's extensive multi-city coverage.
- **Sensor Fidelity:** KITTI's LiDAR and camera setups are less dense and have lower temporal resolution.
- **Label Diversity and Richness:** Waymo provides richer annotations, including motion signals necessary for steering tasks.
- **BEV Readiness:** Waymo's format and spatial continuity are better suited for BEV representation learning in models like AD-L-JEPA.

Based on these criteria, we proceeded with the Waymo Open Dataset v1.4.1, which offers better scale, scene diversity, and sensor quality for both self-supervised representation learning and label-efficient fine-tuning.

4.4 Sensor Calibration and Synchronization

To ensure coherent data alignment, we use Waymo's intrinsic and extrinsic calibration parameters to geometrically align camera frames with vehicle-centric coordinates. Timestamps are used to synchronize RGB images with motion data such as yaw rate, steering angle, and lateral velocity. This synchronization is essential for constructing accurate input-label pairs during the supervised fine-tuning stage, where even minor misalignments could lead to training noise or label drift.

4.5 Preprocessing Pipelines (Image, LiDAR, Labels)

For image data, each frame is resized and normalized according to the requirements of the I-JEPA encoder (e.g., patchifying for Vision Transformer input). Context–target masking is applied to define the learning objective during pretraining. For steering labels, motion data such as yaw rate or lateral velocity is converted into steering angle equivalents using the vehicle's geometry and temporal context.

LiDAR data is not used in the current phase but remains compatible with future integration of AD-L-JEPA using BEV projections or voxelized representations. We will try to pursue this once we achieve the primary goals.

5 Model Architecture and System Design

5.1 Overview of Modular Architecture

The system is designed as a modular pipeline consisting of three main components:

1. A vision encoder based on I-JEPA for feature extraction.
2. A lightweight control head for steering prediction.
3. A CARLA-compatible interface for real-time inference.

This modularity ensures flexibility in experimenting with different encoder types or downstream heads and enables seamless integration with both real-world datasets and simulated environments.

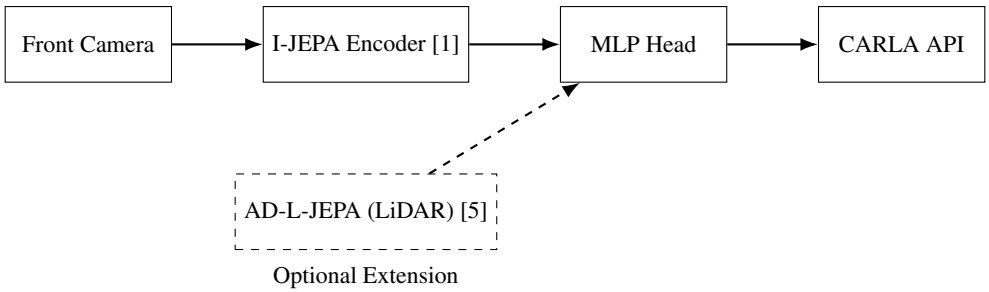

Figure 1: Compact architecture of the steering control system with optional AD-L-JEPA integration.

5.2 I-JEPA Vision Encoder

The core of the perception module is the I-JEPA encoder, a Vision Transformer (ViT)-based model that operates on masked image patches. It learns high-level semantic representations by predicting abstract embeddings of masked regions using the surrounding visible context. For this project, we use the encoder either as pretrained on ImageNet or fine-tuned on the Waymo Open Dataset using the I-JEPA objective. The input to the encoder is a normalized and resized front-facing camera image from Waymo or CARLA.

5.3 AD-L-JEPA LiDAR Encoder

While this project focuses primarily on camera-based perception, the architecture has been designed to optionally support a LiDAR branch in future extensions. The AD-L-JEPA encoder, which operates on Bird's Eye View (BEV) representations of LiDAR data, can be integrated in parallel to the I-JEPA encoder for multi-modal feature extraction. However, in the current scope, this branch remains unutilized.

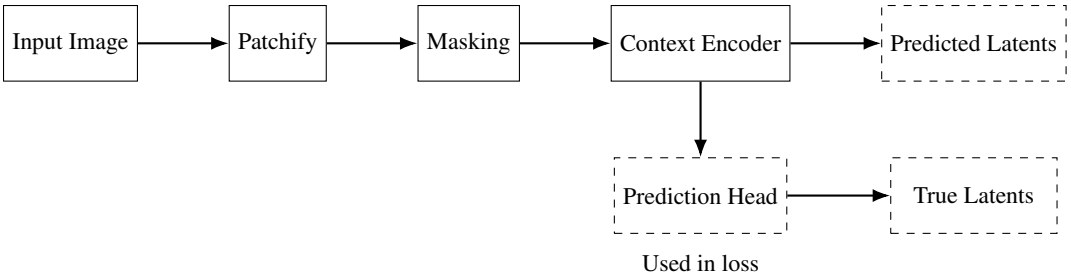

Figure 2: I-JEPA architecture: Context encoder predicts latent targets for masked patches using visible context only.

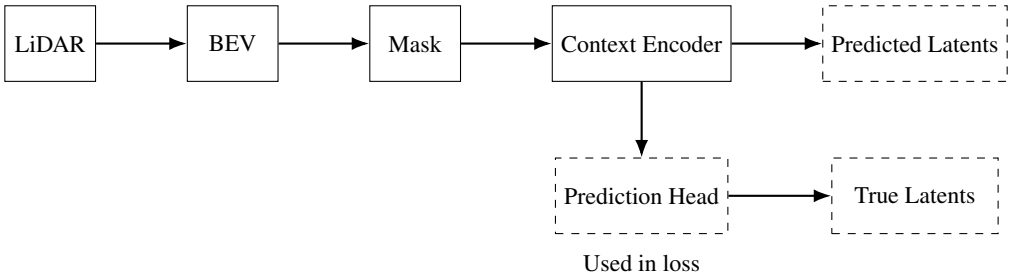

Figure 3: AD-L-JEPA architecture: Context encoder predicts latent targets for masked BEV LiDAR tiles.

5.4 Feature Fusion Strategies

Since this iteration of the project uses a vision-only pipeline, no multi-modal fusion is implemented at this stage. Future work may explore early or late fusion strategies for combining I-JEPA and AD-L-JEPA outputs, such as:

- **Early fusion:** Concatenation of modality-specific embeddings before the control head.
- **Late fusion:** Independent predictions with weighted ensembling or attention-based gating.

5.5 Control Head for Steering Prediction

On top of the I-JEPA encoder, we build a lightweight control head — a multi-layer perceptron (MLP) — to regress steering angles. The MLP receives the pooled image embedding (e.g., CLS token or average-pooled output) and outputs a single scalar value representing the predicted steering angle. This head is trained using a small, labeled subset of the Waymo dataset.

6 Learning Framework and Training Setup

6.1 Pretraining Strategy (Frozen or Finetuned Encoders)

Two modes of encoder usage are explored:

- **Frozen encoder:** The I-JEPA backbone remains unchanged during downstream fine-tuning; only the control head is trained.
- **Partially finetuned encoder:** A subset of the I-JEPA layers (e.g., final transformer blocks) is unfrozen for improved task adaptation.

The choice depends on the size of the labeled dataset and training stability.

6.2 Downstream Fine-Tuning Objectives

The downstream objective is to predict the steering angle associated with each camera frame using only a small labeled subset of the Waymo dataset. This label-efficient setup simulates real-world constraints, where annotated driving data is costly to acquire. The control head is trained using supervised regression on the extracted features from the pretrained encoder.

6.3 Loss Functions and Optimization

The primary loss function used is Mean Squared Error (MSE) between the predicted and ground-truth steering angles:

$$L_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{1}$$

where y_i is the true steering angle and $\hat{y}_i$ is the model's prediction. We use the AdamW optimizer with cosine learning rate decay and gradient clipping to ensure training stability.

Additional training strategies we might t:

- L2 regularization (weight decay) for generalization
- Gradient clipping to stabilize training
- Learning rate warmup and decay schedules

6.4 Planned Hyperparameter Search and Evaluation Criteria

Given the exploratory nature of this project, we will adopt an iterative experimental approach for selecting training parameters and evaluating model performance. Rather than fixing hyperparameters a priori, we plan to:

- Explore a range of learning rates, batch sizes, dropout values, and encoder freezing strategies
- Experiment with different pooling methods (e.g., mean-pooling vs. CLS token) to extract features from I-JEPA
- Evaluate label efficiency by varying the amount of supervised data used for fine-tuning
- Compare frozen vs. partially fine-tuned encoders to assess trade-offs between generalization and specialization

For evaluation, we intend to use a combination of:

- Offline metrics such as Mean Absolute Error (MAE) or Root Mean Squared Error (RMSE) on a validation split
- In-simulator metrics in CARLA, including lane deviation, maneuver success rate, and control smoothness under different environmental conditions
- This evaluation strategy will help guide hyperparameter tuning and assess the real-world viability of the learned model under domain shift from Waymo to CARLA.

7 Proposed Approaches

To evaluate the effectiveness of I-JEPA representations for steering control, we propose a small set of practical and complementary approaches. Each method is designed to explore a variation in pretraining, fine-tuning, or architecture configuration, with minimal supervision and direct deployment in CARLA.

7.1 Method 1: Baseline – Frozen I-JEPA + MLP Head

- Use the ImageNet-pretrained I-JEPA encoder, kept frozen during fine-tuning.
- Add a lightweight MLP on top to predict steering angle.
- Train only the MLP head using 1–5% labeled Waymo data.
- Deploy in CARLA using RGB front camera as input.

Goal: Establish a baseline performance with no encoder updates and minimal labeled supervision.

7.2 Method 2: Domain-Adaptive Pretraining on Waymo

- Continue I-JEPA pretraining on Waymo camera data (without labels).
- Use the same frozen encoder + MLP setup as Method 1.
- Fine-tune on the same small labeled subset.

Goal: Test whether domain-aligned pretraining improves downstream generalization to CARLA.

7.3 Method 3: Partial Fine-Tuning of I-JEPA

- Use the Waymo-pretrained I-JEPA encoder.
- Unfreeze the last few transformer blocks during fine-tuning, while keeping early layers fixed.
- Fine-tune with the same MLP head and small label budget.

Goal: Determine if limited fine-tuning improves performance without overfitting or needing large data.

7.4 Method 4: Pooling Strategy Variants

- Compare different ways of extracting the representation from the ViT:
  - CLS token
  - Mean-pooled token embeddings
- Keep encoder and head architecture fixed.

Goal: Identify the most effective feature summarization method for steering regression.

7.5 Method 5: Data Subset Variation (Label Efficiency Sweep)

- Fix model setup (e.g., frozen I-JEPA + MLP).
- Train using different fractions of labeled Waymo data: 1%, 5%, 10%, 20%.

Goal: Evaluate model robustness to label scarcity and identify the point of diminishing returns.

We will try to iterate on as many of these approaches along with any new approaches that we might find better along the process of experimentation.

8 Experiment Design and Evaluation Plan

To assess the effectiveness, generalization, and label-efficiency of the proposed approaches, a comprehensive experimental plan is outlined. The experiments are structured to isolate the impact of self-supervised pretraining, modality selection, and fine-tuning strategies.

8.1 Experimental Setup in CARLA

Evaluation is conducted in the CARLA simulator (v0.9.x), using pre-defined driving scenarios such as lane-following, urban navigation, and curved turns under varying lighting and weather conditions. The simulation is configured to stream front-facing RGB images to the trained model, which outputs steering angle predictions that are applied through CARLA's control API in real time.

Metrics such as lane-keeping quality, deviation from centerline, and completion rate of test routes are logged to assess the practical viability of the learned steering policy.

8.2 Baselines for Comparison

The following baselines are used for evaluating the proposed models:

- Baseline (frozen I-JEPA + MLP)
- Waymo-pretrained I-JEPA
- Partial fine-tuning
- CLS vs. mean-pooling
- Label efficiency sweep

These baselines help quantify the added value of JEPA-based pretraining and label efficiency.

8.3 Data-Efficiency Evaluation

To test the label efficiency of self-supervised pretraining, models are fine-tuned on varying subsets of the labeled Waymo data: 1%, 5%, 10%, and 100%. Performance is measured in CARLA to evaluate how well the models generalize when trained with minimal supervision.

This helps benchmark I-JEPA's core promise of learning effective representations from unlabeled data.

8.4 Domain Shift and Robustness Tests

Because the model is trained on Waymo (real-world) and evaluated in CARLA (simulated), domain shift is inevitable. To assess robustness:

- Performance is measured across different CARLA towns and weather settings.
- Domain adaptation techniques like color jittering or style transfer (optional) are tested for mitigating performance drops.

These tests evaluate the model's generalization to unseen environments, a key requirement for real-world deployment.

9 Implementation Plan

We will try to implement the project using the below tools and frameworks and extensively evaluate it to understand the shortcomings and move to the next iteration.

9.1 Tools and Frameworks

- **PyTorch**: Core deep learning framework for model implementation and training.
- **Waymo Open Dataset Toolkit**: For data extraction and preprocessing.
- **CARLA Simulator (v0.9.x)**: For real-time inference and downstream evaluation.
- **Hugging Face Transformers / Meta's I-JEPA repo**: For leveraging existing pretrained models or replicating architecture.
- **OpenCV & NumPy**: For image processing and synchronization.
- **FFmpeg**: For converting simulation episodes to visual outputs.

10 Conclusion

This project proposes a modular, label-efficient steering control system that leverages the power of Joint-Embedding Predictive Architectures (JEPA) for real-world autonomous driving applications. By pretraining I-JEPA on the Waymo Open Dataset and evaluating the fine-tuned model in the CARLA simulator, the approach examines the generalization and practical deployment potential of self-supervised vision models.

Unlike traditional supervised pipelines that rely heavily on labeled data, this framework emphasizes representation learning from unlabeled sequences, enabling efficient downstream adaptation with minimal supervision. The evaluation in a domain-shifted environment (CARLA) further highlights the robustness of JEPA-based SSL models. This work demonstrates a promising step toward scalable, efficient, and adaptable machine learning systems for real-world autonomous driving.

References

[1] Mahmoud Assran, Ishan Misra, Yossi Botbol, Lam Dinh, Alexander Kirillov, Mathilde Caron, Gabriel Synnaeve, Julien Mairal, Yann LeCun, and Armand Joulin. Self-supervised learning from images with a joint-embedding predictive architecture. *arXiv preprint arXiv:2301.08243*, 2023.

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

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

[4] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2022.

[5] Haoran Zhu, Zhenyuan Dong, Kristi Topollai, and Anna Choromanska. Ad-l-jepa: Self-supervised spatial world models with joint embedding predictive architecture for autonomous driving with lidar data, 2025.

[6] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Yuning Chou, Aurelien Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuxiang Chai, Ben Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. `https://waymo.com/open`, 2021.

[7] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2021 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012.