

Human Insights Driven Latent Space for Different Driving Perspectives: A Unified Encoder for Efficient Multi-Task Inference

Huy-Dung Nguyen¹, Anass Bairouk¹, Mirjana Maras¹, Wei Xiao², Tsun-Hsuan Wang², Patrick Chareyre¹,
 Ramin Hasani², Marc Blanchon¹, Daniela Rus²

Abstract—Autonomous driving systems require a comprehensive understanding of the environment, achieved by extracting visual features essential for perception, planning, and control. However, models trained solely on single-task objectives or generic datasets often lack the contextual information needed for robust performance in complex driving scenarios. In this work, we present a unified encoder trained across a diverse set of computer vision tasks essential for urban driving, including depth estimation, pose estimation, 3D scene flow estimation, and semantic, instance, panoptic, and motion segmentation. This single-encoder approach not only integrates these complementary visual cues, inspired by the diversity of visual cues used in human driving perception, but also enables a compact and inference-efficient model that embeds a rich, navigation-relevant latent space. Indeed, the unified encoder learns to embed multi-task knowledge into a shared representation, allowing for better downstream task adaptation, particularly for steering estimation. To ensure the efficient learning across tasks within a unified encoder, we propose a multi-scale pose decoder and employ knowledge distillation from a multi-backbone teacher model. Our experiments demonstrate that (1) the unified encoder achieves strong generalization across all visual tasks, comparable to state-of-the-art dedicated models, and (2) its frozen latent representations significantly outperform both fine-tuned models and ImageNet-pretrained baselines for steering estimation. These results underscore how multi-task feature learning, inspired by the diversity of perceptual cues used in human driving, offers an efficient and context-rich foundation for autonomous driving systems.

I. INTRODUCTION

The advancement of self-driving cars has gained significant public attention in recent years, although the development of this technology began decades ago. Early innovations include vehicle-to-vehicle communication via radio waves in the 1920s [1] and electromagnetic guidance in the 1930s [2]. The primary goal of autonomous vehicles is to improve road safety and efficiency by minimizing human error, which causes more than 90% of vehicle accidents, while mechanical failures account for only 2% [3]. Achieving human-level autonomy requires comprehensive environmental understanding, robust control, and reliable real-time decision making.

A fully autonomous driving system extracts diverse visual cues—such as depth, motion, and segmentation—to support tasks including object detection, path planning, and behavior prediction. Beyond explicit outputs, intermediate visual representations (*e.g.*, from CNNs) that capture spatial structure, temporal consistency, and scene dynamics are

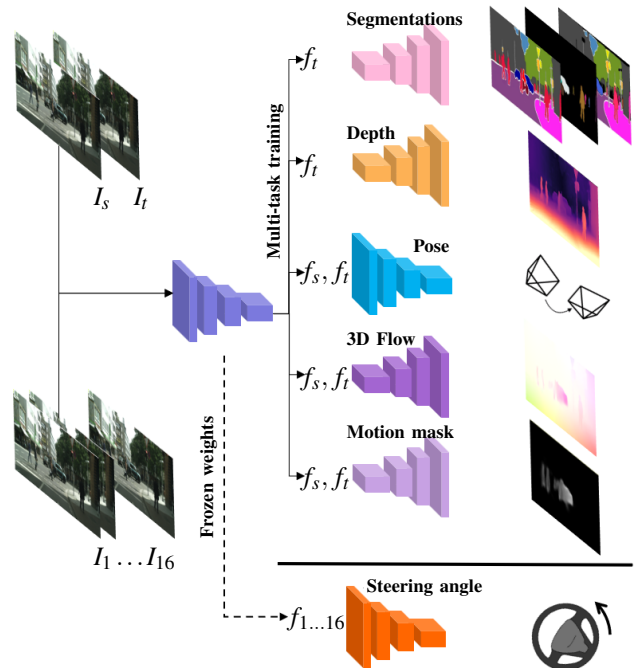


Fig. 1: Our multi-task training strategy. I_s , I_t , $I_{1...16}$ represent the source, target, and 16 sequential images, respectively. Their features, denoted as f_s , f_t , $f_{1...16}$, are extracted (and concatenated when necessary) using our single encoder.

critical for downstream tasks such as obstacle avoidance, adaptive control, and trajectory prediction [4], [5], [6], [7].

Despite their importance, many existing learning-based driving methods solely focus on single-task objectives, such as steering estimation, and are trained using RGB inputs with task-specific supervision [8], [9], [10], [11]. While these models can learn task-relevant features, their reliance on narrow objectives and input modalities limits the diversity of visual cues captured in the learned representations. Prior work [12] has shown that incorporating additional motion information, such as optical flow, can significantly improve steering performance, indicating that richer visual signals can enhance both task accuracy and scene understanding.

In this paper, we propose a novel training strategy for a unified encoder that captures diverse driving-relevant visual cues through multi-task learning. Rather than optimizing isolated objectives, we jointly incorporate tasks that humans implicitly rely on while driving—semantic labeling, object boundaries, depth estimation, and motion cues—into a single

¹Hybrid Intelligence, part of Capgemini Engineering.

²Computer Science and Artificial Intelligence Laboratory (CSAIL), MIT.

perception pipeline (see Figure 1). This unified design enables compact, inference-efficient multi-task prediction while embedding a rich, navigation-relevant latent representation.

Training a shared encoder across multiple supervised and self-supervised tasks presents significant optimization challenges. In particular, strong supervised segmentation losses can dominate weaker self-supervised objectives such as scene flow and motion mask estimation, destabilizing training and degrading depth accuracy. Addressing these issues is essential to ensure stable convergence and effective feature sharing.

To address these challenges, we introduce a multi-scale pose decoder and apply knowledge distillation from a multi-encoder teacher to balance gradient contributions across tasks. While the resulting training procedure involves multiple stages and an auxiliary teacher model, the additional computational cost is incurred only once during pretraining. Once trained, the unified encoder replaces the multiple task-specific backbones and enables multi-task prediction with a single forward pass, reducing inference-time complexity and deployment overhead. In our setup, the pretraining is conducted on a single NVIDIA A5000 GPU prior to deployment.

Our contributions are summarized as follows:

- **A multi-scale pose decoder:** We design a pose decoder that leverages multi-scale features from the shared encoder, improving depth estimation in dynamic scenes.
- **Knowledge distillation for stability:** We apply knowledge distillation from a multi-encoder teacher model to guide the learning of self-supervised tasks and prevent gradient imbalances.
- **Comprehensive evaluation of shared latent space:** We conduct an evaluation of the unified encoder’s shared representation on a downstream navigation task (steering estimation). Our results demonstrate that the frozen unified encoder outperforms its fine-tuned counterpart and the same architecture pretrained solely on generic datasets such as ImageNet.
- **Human-inspired feature learning:** Our training strategy encourages the encoder to capture richer, human-like perceptual features, promoting contextual awareness and scene understanding closer to how human drivers perceive the surrounding environment.

II. RELATED WORK

A. Image segmentation

Image segmentation involves separating an image into segments, grouping pixels by specific criteria. Semantic segmentation assigns pixels to broad classes like roads and buildings, while instance segmentation focuses on distinct objects like cars and people [13]. Traditionally, these tasks were handled separately. To integrate them, Kirillov *et al.* [14] introduced panoptic segmentation, which organizes pixels into amorphous background regions (“stuff”) and distinct objects (“things”). However, this led to an additional specialized task rather than unifying these two, with performance falling short of state-of-the-art results in dedicated tasks.

Recent methods have proposed unified models for all three tasks showing high performance but still requiring separate training for each. In a significant advancement, Jain *et al.* introduced OneFormer [15], a model that, given an image and a text prompt specifying the task, produces the corresponding type of segmentation. Evaluated on several public datasets, OneFormer set new state-of-the-art benchmarks for all three tasks with a single, jointly trained model. In this paper, we use OneFormer decoders to generate the three mentioned segmentation outcomes.

B. Monocular Depth & Pose Estimation

Monocular depth estimation is a task of predicting the depth of a scene from a single 2D image. Unlike stereoscopic methods, which use multiple viewpoints, monocular depth estimation must infer depth from a single viewpoint, making it particularly challenging. Traditional methods mostly based on hand-crafted features, which can lead to inaccuracies in complex scenarios [16]. In a first attempt using deep learning, Eigen *et al.* introduced a CNN-based model to predict depth maps directly from single images, achieving superior results [17]. However, this method requires ground truth for depth, typically obtained with expensive hardware like LiDAR, limiting its practicality. To overcome this, recent research has shifted to unsupervised methods that use the inherent structure of unlabeled images. Notably, Zhou *et al.* [18] and Godard *et al.* [19] proposed unsupervised methods which employ video sequences to simultaneously learn depth and camera motion, reducing the dependency on labeled data and adapting better to dynamic scenes.

However, unsupervised methods often consider the assumption of a static world, which is not true in most real scenarios. To this end, recent advancements have integrated flow (*e.g.*, 2D or 3D) and motion segmentation in addition to depth and pose estimation, enabling better handling of dynamic objects in the scene [20].

C. Scene Flow and Motion Segmentation

Scene flow estimation is similar to depth estimation but in addition, it can capture 3D motion between consecutive images, while motion segmentation identifies dynamic objects. These tasks, when jointly trained with depth and pose estimation, improve depth accuracy [21], [22]. Jiao *et al.*’s EfficScene network [23] trains on these tasks using stereo images, leveraging the shared geometric structure of scene depth and object movement. Recently, Sun *et al.* [20] propose DynamoDepth framework that further improves flexibility with the capacity of training on only video frames.

Building on DynamoDepth, we propose a single encoder model that requires only two images to estimate scene flow and motion segmentation, reducing complexity compared to the three-image requirement in the original method.

D. Steering estimation

The two main research directions for steering estimation are model-based and model-free. Model-based methods rely on vehicle dynamics models, while model-free

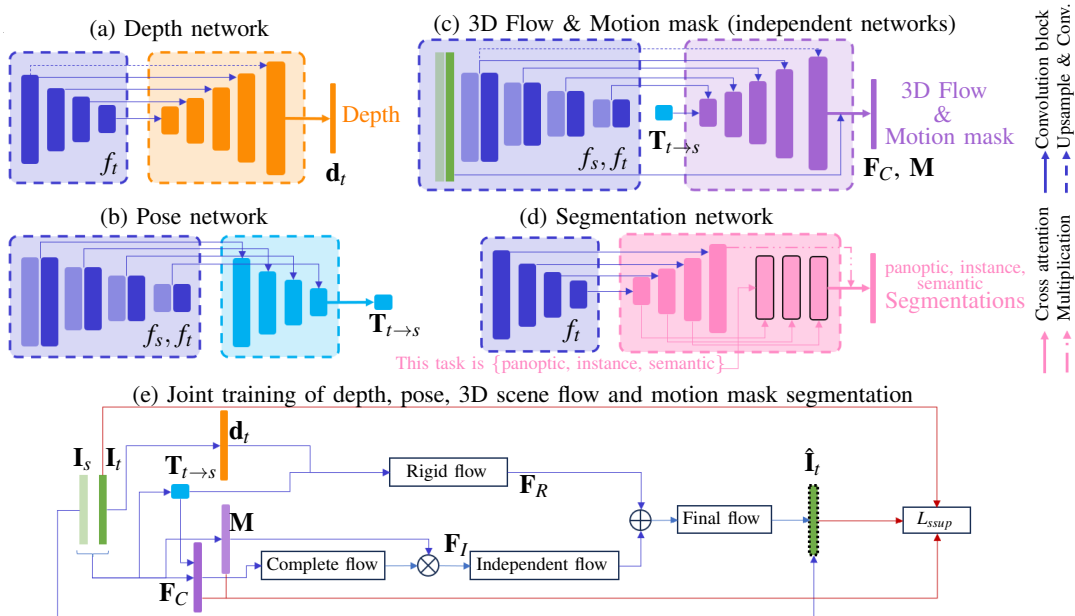


Fig. 2: Simplified architecture of our model: (a) Depth network using target image features f_t to output depth d_t , (b) Multi-scale pose network using source and target image features f_s, f_t to output relative pose $T_{t \rightarrow s}$, (c) 3D Scene Flow F_C and Motion mask M networks using RGB images and features f_s, f_t , (d) Segmentation network outputting panoptic, instance, and semantic segmentations, and (e) Loss computation L_{ssup} for joint training of depth, pose, 3D scene flow, and motion mask segmentation. We denote rigid flow F_R , independent flow F_I , final flow, and sampled target image \hat{I}_t .

methods leverage data-driven techniques like deep learning. This paper focuses only on the model-free approach. Bojarski *et al.* [10] introduced an end-to-end steering estimation using CNN with a single RGB camera input. Capito *et al.* [12] improved the navigation capability by incorporating optical flow into the input. The importance of temporal information has also been highlighted in many works. Indeed, Eraqi *et al.* [24] employed LSTM networks to enhance steering control, while Xu *et al.* [6] combined fully convolutional networks with LSTM and semantic segmentation for improved road condition interpretation. Lechner *et al.* [4] proposed Neural Circuit Policies (NCPs), which provide interpretable decision maps from high-dimensional inputs. These studies indicate that integrating temporal information between frames can improve steering accuracy.

E. Multi-Task Learning in Computer Vision

Recent works have explored multi-task learning (MTL) frameworks to simultaneously address multiple vision tasks, leveraging shared representations for improved performance. For instance, online knowledge distillation techniques have been proposed to mitigate negative transfer between tasks such as semantic segmentation and depth estimation, enhancing overall learning stability [25]. Joint-confidence-guided MTL frameworks have also been introduced for 3D reconstruction, combining depth prediction, semantic labeling, and surface normal estimation to improve feature fusion [26]. Moreover, collaborative MTL approaches have demonstrated effectiveness in handling object detection, segmentation, and tracking tasks by establishing associative connections among

task heads [27].

Inspired by this synergy, we take a different perspective. Instead of demonstrating how MTL can enhance the performance of one or a few task through training, we hypothesize that the shared features learned from a diverse set of navigation tasks are sufficient not only for achieving good performance on each individual task but also for supporting a related navigation task—steering estimation—which is not present during training.

III. METHODS

Figure 1 illustrates our model architecture, which includes a Swin Tiny encoder [28] and six decoders: panoptic, instance, and semantic segmentation, depth and pose, 3D scene flow estimation, motion mask segmentation, and steering command prediction. Training was performed in two stages. In the first stage, we pretrained the encoder with five decoders (excluding the steering command decoder) to learn generalized features across tasks. In the second stage, we froze the encoder and added a prediction head for steering estimation to evaluate its performance in navigation.

Given the challenge of finding an annotated dataset for all the targeted tasks, we selected the CityScapes dataset [29] for training due to its high-quality ground truth segmentation and structured video sequences. This sequential format is particularly suitable for self-supervised training in depth, optical flow, and motion mask estimation without requiring explicit labels. However, we did not evaluate the steering prediction in CityScapes. Instead, this evaluation is performed on the MIT dataset [4] after fine-tuning a prediction head (with the

frozen encoder). This MIT dataset provides the appropriate annotations for this task. CityScapes dataset serves as a crucial foundation for training the unified encoder, allowing it to capture rich visual and motion cues that contribute to downstream tasks (*e.g.*, steering estimation).

A. Panoptic, Instance, Semantic Segmentations

For these three supervised segmentation tasks, we base on the OneFormer approach. This is a task-conditioned universal image segmentation model that achieves state-of-the-art performance across all segmentation tasks with a unified model and architecture. The OneFormer framework adopts a task-conditioned joint training strategy that allows simultaneous training on all segmentation tasks using a single model, reducing resource requirements significantly.

Figure 2d provides a simplified illustration of the OneFormer architecture. This model consists of an encoder and two distinct decoders. The first decoder processes image features to generate all segment-related features. The output from this decoder, along with an embedding of a text prompt defining the task (*e.g.*, "The task is {panoptic, instance, semantic}"), is then passed to the second decoder. The second decoder uses this information to produce the desired segmentation type. The supervised loss function combines cross-entropy loss for classification, binary cross-entropy for mask predictions, and Dice loss for accurate mask boundary predictions. We trained this model end-to-end on panoptic annotations, from which semantic and instance labels are derived. The overall loss function L_{sup} integrates multiple components to ensure task-specific accuracy:

$$L_{\text{sup}} = \lambda_{\text{cls}}L_{\text{cls}} + \lambda_{\text{bce}}L_{\text{bce}} + \lambda_{\text{dice}}L_{\text{dice}} + \lambda_{\text{contrast}}L_{\text{contrast}}, \quad (1)$$

where:

- L_{cls} is the cross-entropy loss for mask (*i.e.*, region) classification accuracy.
- L_{bce} is the binary cross-entropy loss applied to mask.
- L_{dice} is the Dice loss, helps to improve mask boundaries.
- L_{contrast} is the contrastive loss between object and text queries to make sure the text query is taken into account.

B. Depth, Pose, 3D Flow Estimation, Motion Mask

For depth estimation, we employ the depth decoder architecture from [30]. In addition, we introduce a novel pose decoder, which is discussed in detail in the next section. For 3D scene flow estimation and motion mask segmentation, we build upon the DynamoDepth approach [20], enhancing it with convolutional blocks including additional non-linearity to increase the model’s representational capacity. Concretely, instead of considering the 3D scene flow as a linear combination of the encoder features f and the pose $\mathbf{T}_{t \rightarrow s}$, we consider it as a non linear combination $\mathbf{F}_i = \mathbf{U}(\mathbf{F}_{i+1}) + \text{ConvELU}([a_i, \text{Conv}(a_i)])$ where $a_i = \text{Conv}([\mathbf{U}(\mathbf{F}_{i+1}), f_i])$, $\mathbf{F}_5 = \mathbf{T}_{t \rightarrow s}$, and \mathbf{F} and \mathbf{U} are respectively flow and upscaling operation. By doing this, we increase the complexity of the decoders, which can help the unified encoder to learn simpler and more generalizable features that can support downstream

tasks. Figure 2a, 2b, and 2c provides simplified illustrations of our networks.

Starting with source \mathbf{I}_s and target frames \mathbf{I}_t , we estimate depth \mathbf{d}_t and relative pose $\mathbf{T}_{t \rightarrow s}$. Using $\mathbf{T}_{t \rightarrow s}$, we calculate 3D rigid flow \mathbf{F}_R . The source and target features are concatenated and passed through the 3D scene flow and motion mask segmentation decoders along with $\mathbf{T}_{t \rightarrow s}$, producing the complete flow \mathbf{F}_C and binary motion mask \mathbf{M} . The independent flow \mathbf{F}_I is computed as $\mathbf{F}_I = \mathbf{M} \times (\mathbf{F}_C - \mathbf{F}_R)$. The final flow $\mathbf{F}_F = \mathbf{F}_R + \mathbf{F}_I$ is used to estimate the target frame $\hat{\mathbf{I}}_t$ from \mathbf{I}_s . The self-supervision loss L_{ssup} is:

$$L_{\text{ssup}} = L_{\text{recon}} + L_s + \lambda_c L_c + \lambda_m L_m + \lambda_g L_g, \quad (2)$$

where:

- L_{recon} is the reconstruction loss that combines SSIM and L1 norms weighted by α .
- L_s is the smoothness loss [19] which is a weighted sum of edge-aware smoothness losses for the inverse depth, the \mathbf{F}_C and \mathbf{M} .
- L_c is the motion consistency loss that penalize flow discrepancy $\mathbf{F}_D = \|\mathbf{F}_C - \mathbf{F}_R\|_1$ for static pixels.
- L_m is the motion sparsity loss that penalizes the mask \mathbf{M} with low flow discrepancies using cross-entropy to favor zero motion mask.
- L_g is the above-ground loss that penalizes points projected below the ground plane using RANSAC.

Figure 2e illustrates the data flow for computing the self-supervised loss described above.

C. Towards a Unified Multi-Task Encoder

Unlike DynamoDepth, which employs three separate encoders—each specialized for depth estimation, pose estimation, and 3D scene flow with motion mask segmentation—we propose a single, unified encoder capable of handling all these tasks simultaneously, along with semantic, instance, and panoptic segmentation. This unified encoder design is not only for compactness, it is driven by the hypothesis that a unified feature space, with visual and motion cues learned across tasks, can better support both perception and downstream driving-related tasks such as steering estimation. This unified model contrasts with prior systems like OneFormer, which unify only segmentation tasks, or DynamoDepth, which limits multi-task learning to motion and geometry tasks, without extending to semantic scene understanding. Our proposed multi-task pipeline introduces:

a) *Multi-Scale Pose Decoder*: To address the challenge of low depth estimation accuracy in shared encoder configurations, we introduce a multi-scale pose decoder that progressively refines pose estimates using features at different resolutions. This design may mitigate the ambiguity that arises when pose estimation relies only on low-resolution features, a limitation seen in prior shared-encoder methods like Monodepth2 [19].

The multi-scale pose decoder takes as input the multi-resolution features extracted from the shared encoder for both source I_s and target I_t images. These feature pairs f are concatenated at each resolution level and processed by



Fig. 3: Qualitative results. Left to right: Input, panoptic, instance, semantic output, depth, motion mask, independent flow.

convolutional blocks with skip connections. The output at each level is computed as:

$$out = \text{ReLU}(\text{ConvBNReLU}(\text{ConvBN}(f)) + \text{Shortcut}(f)), \quad (3)$$

Two consecutive blocks are applied per level, with higher-resolution outputs downsampled and concatenated with lower-resolution inputs at each stage. This design preserves both global and local spatial relationships critical for accurate pose estimation, particularly in dynamic environments.

b) Knowledge Distillation for Stability: Training a unified encoder across both supervised (*e.g.*, segmentation) and self-supervised (*e.g.*, scene flow, motion segmentation) tasks presents significant challenges due to the gradient imbalance between these learning types. Supervised tasks, which have strong labeled signals, can dominate the shared feature learning, hindering the convergence of self-supervised tasks like motion mask estimation.

To overcome this, we propose to use knowledge distillation. We first train a teacher model consisting of three encoders and five decoders similar to [20]: one Swin encoder for depth estimation and the three supervised segmentation tasks, one ResNet18 encoder for pose estimation, and one ResNet18 encoder for 3D scene flow and motion mask segmentation. We then use the encoder-decoder pair responsible for 3D scene flow and motion mask segmentation in the teacher model to supervise these two tasks in the unified encoder model. The associated distillation loss is:

$$L_{distil} = \beta_1 \cdot \|F_C^{teacher} - F_C^{student}\|_1 + \beta_2 \cdot \|M^{teacher} - M^{student}\|_1, \quad (4)$$

Unified Multi-Task Loss The final training objective for the unified encoder integrates supervised, self-supervised, and distillation losses into a single balanced objective:

$$L_{total} = \lambda_1 \cdot L_{sup} + \lambda_2 \cdot L_{ssup} + \lambda_3 \cdot L_{distil}, \quad (5)$$

D. Steering Estimation from Dense Latent Space

Steering angle prediction is used here as a proxy task to evaluate whether the learned latent representation captures navigation-relevant structure. While steering alone does not cover the full complexity of driving behavior, it requires implicit understanding of scene geometry, semantics, and motion, making it a suitable first validation target.

For the steering command, we used the pre-trained encoder, which was frozen, followed by an attentive pooling mechanism [31] to process the steering input. The encoder's outputs were fed into the attentive pooler, which takes a

sequence of 16 images to compute the steering angle. During training, we optimized the loss L_{pred} , which is defined as follows:

$$L_{pred} = \sum_i w_i (\hat{\mathbf{y}}^{(i)} - \mathbf{y}^{(i)})^2 / \sum_i w_i, \quad (6)$$

where $w_i = \exp(\lambda \cdot |\mathbf{y}^{(i)}|)$, with λ representing the factor modulating the influence of the steering command's magnitude, $|\mathbf{y}^{(i)}|$, on the loss. To evaluate the model's performance, we employed the mean squared error (MSE).

IV. EXPERIMENTAL RESULTS

A. Training Setup

We use the KITTI Eigen split for initial ablation (trained only with the depth and the pose decoders) and CityScapes for the remaining experiments, with image resolutions of 192×640 and 192×512 , respectively. Images are pre-processed into triples using scripts from [18]. For CityScapes, the lower 25% of images is cropped to exclude the front car [32]. The entire CityScapes dataset is used for supervised segmentation tasks. Pretraining is conducted on a single NVIDIA RTX A5000 with a batch size of 6 (3 images for the supervised tasks, *i.e.*, segmentations, and 3 triples for the self-supervised tasks) over four steps: (1) training the shared encoder with depth, pose, and segmentation decoders for 60,000 steps, (2) training the 3D scene flow decoder for 40,000 steps with other components frozen, (3) training the pose decoder, 3D scene flow decoder, and motion mask segmentation decoder for 40,000 steps with other components frozen, and (4) training the whole network for 250,000 steps. After pretraining, we follow the tenfold cross-validation procedure by Lechner *et al.* [4] to fine-tune the model on the downstream steering estimation task. The values of hyperparameters are: $\lambda_{cls} = 2, \lambda_{bce} = \lambda_{dice} = \lambda_c = 5, \lambda_{contrast} = 0.5, \lambda_m = \lambda_g = \beta_1 = 0.1, \beta_2 = 1e-3, \lambda_1 = \lambda_2 = \lambda_3 = 1$.

B. Ablation Studies

Multi-scale pose decoder. To evaluate the effectiveness of our proposed multi-scale pose decoder, we conducted depth estimation experiments using the KITTI Eigen split dataset. This dataset was selected because accurate depth estimation can be achieved with just a depth and a pose network, allowing us to more clearly isolate and assess the contribution of our multi-scale pose decoder. Table I-1 presents the depth estimation results on the KITTI dataset. Our findings reveal that a naive approach of using a shared

encoder, by concatenating the lowest-level features of source and target images and passing them through a pose prediction head, results in a drop in depth estimation performance. This trend is evident in both the Monodepth2 model [19] and our Swin encoder. In contrast, using our multi-scale pose decoder maintains depth performance comparable to using a separate ResNet18 encoder specifically for pose estimation.

Model	MS	FM	KD	Error (\downarrow)		Acc. (\uparrow)
				Abs_{rel}	$rmse_{log}$	$\delta < 1.25$
$MD2_{sep}$				0.115	0.193	0.877
$MD2_{sh}$				0.125	0.201	0.857
Our_{sep}				0.108	0.183	0.884
Our_{sh}				0.117	0.190	0.872
Our_{sh}	✓			0.109	0.184	0.884
Our_{sep}				0.103	0.157	0.885
Our_{sh}	✓			0.126	0.182	0.850
Our_{sh}	✓	✓		0.134	0.183	0.833
Our_{sh}	✓	✓	✓	0.106	0.158	0.888

TABLE I: **1)** (Upper part) Ablation on multi-scale pose decoder using KITTI dataset. **2)** (Lower part) Analysis on knowledge distillation on CityScapes. MS, FM, and KD refers multi-scale pose decoder, 3D scene Flow & Motion mask, and Knowledge Distillation. *sep* and *sh* refer to the separate and shared version. For the separate encoder, a ResNet18 was employed similar to [19].

	PQ (\uparrow)	AP (\uparrow)	IoU (\uparrow)
OneFormer [15]	55.8	28.4	74.3
Our multi-task model	56.0	28.6	74.2

TABLE II: Ablation study on panoptic, instance, and semantic segmentation tasks.

Knowledge distillation for 3D flow & motion mask.

We evaluate the effectiveness of knowledge distillation in training our unified encoder on the CityScapes dataset, which includes many dynamic objects, making 3D scene flow and motion mask networks crucial. Table I-2 shows the depth error and accuracy for different models. First, using a shared encoder with our multi-scale pose decoder achieves good performance compared to using separate encoders for different tasks. Second, training the shared encoder without knowledge distillation leads to suboptimal performance, even worse than the shared encoder model that doesn't consider 3D scene flow and motion masks. However, incorporating knowledge distillation improves the shared encoder's performance, making its performance inline with models with separate encoders. This highlights the effectiveness of our strategy.

Panoptic, instance, and semantic segmentations. To make a fair comparison, we trained OneFormer using the same batch size as our multi-task model. Table II shows the results for Panoptic Quality (PQ), Average Precision (AP), and Intersection over Union (IoU) respectively for panoptic, instance, and semantic segmentation tasks. We can observe that the segmentation performance of our multi-task model closely matches that of the OneFormer model, confirming the state-of-the-art capability of our approach.

C. Comparison with State-of-the-Art Methods

Since our multi-task model aligns closely with the state-of-the-art OneFormer model for segmentation tasks, we only focus on comparing depth estimation with other leading methods. Table III presents depth estimation results on the KITTI and CityScapes datasets.

On KITTI, due to the lack of labeled segmentation data, our model is trained without supervision but still surpasses the average performance of state-of-the-art methods. Notably, it outperforms several established methods (e.g., Struct2Depth) across all metrics, even without independent motion and segmentation support. Similarly, on CityScapes, our model shows competitive results, performing better than some methods and approaching the best results for each metric. It is important to highlight that most other methods rely on off-the-shelf segmentation models or use multiple encoders for pose and/or motion estimation.

Finally, Figure 3 shows qualitative results of our multi-task model. The panoptic, instance, and semantic segmentation results demonstrate high quality, consistent with the strong quantitative results. Additionally, the quality of the estimated flow and motion mask demonstrate the contribution of these outputs to improved depth estimation compared to the model without the 3D scene flow and motion mask.

D. Dense Latent Space to Steering: Evaluation

Table IV summarizes the performance of various models for steering prediction, including our proposed Swin-AttnPool approach and several existing CNN-based [4] and VAE-based models [7]. To evaluate the effectiveness of the features learned through our training strategy, we present three variants of the same encoder architecture:

- ImageNet-pretrained (frozen): The encoder is pretrained on ImageNet, frozen, and fine-tuned on the steering estimation task.
- Our pretrained (unfrozen): The encoder is initialized with our pretrained weights and unfrozen during fine-tuning.
- Our pretrained (frozen): The encoder is initialized with our pretrained weights and is frozen during fine-tuning.

The results reveal several key insights. First, our frozen encoder (variant 3) outperforms the frozen encoder pretrained on ImageNet (variant 1), indicating that the visual features learned through our multi-task training are more relevant for navigation tasks. Second, the frozen encoder (variant 3) also surpasses its unfrozen counterpart (variant 2), suggesting that exclusively fine-tuning for the steering task may overlook valuable features learned from other navigation-related tasks, which can positively contribute to steering performance.

Finally, although Table IV shows that our training error (variant 3) is relatively high, the test error remains competitive compared to the average performance of all methods. Indeed, our frozen encoder outperforms the VAE-LSTM (19 units) model and performs comparably to the CNN-GRU (64 units) approach. The relatively high error could be due to architectural aspects, either within the encoder or the

Method	IM	Sem.	#f	D	Error metric (\downarrow)				Accuracy metric (\uparrow)		
					Abs_{rel}	Sq_{rel}	RMSE	$RMSE_{log}$	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Monodepth2 [19]			1	K	0.115	0.903	4.863	0.193	0.877	0.959	0.981
LiteMono [33]			1	K	0.101	0.729	4.454	0.178	0.897	0.965	0.983
Struct2Depth [34]	✓	✓	1	K	0.141	1.026	5.290	0.215	0.816	0.945	0.979
RM-Depth [35]	✓		1	K	0.107	0.687	4.476	0.181	0.883	0.964	0.984
Dynamo-Depth [20]	✓		1	K	0.112	0.758	4.505	0.183	0.873	0.959	0.984
Ours (wo 3D scene flow)			<u>1</u>	<u>K</u>	<u>0.109</u>	<u>0.818</u>	<u>4.654</u>	<u>0.184</u>	<u>0.884</u>	<u>0.963</u>	<u>0.983</u>
Struct2Depth [34]	✓	✓	1	CS	0.145	1.737	7.280	0.205	0.813	0.942	0.978
Gordon <i>et al.</i> [36]	✓	✓	1	CS	0.127	1.330	6.960	0.195	0.830	0.947	0.981
RM-Depth [35]	✓		1	CS	0.100	0.839	5.774	0.154	0.895	0.976	0.993
Zhong <i>et al.</i> [37]	✓		2	CS	0.098	0.946	5.553	0.148	0.908	0.977	0.992
ManyDepth [32]	✓		2	CS	0.114	1.193	6.223	0.170	0.875	0.967	0.989
Ours	✓	✓	<u>1</u>	<u>CS</u>	<u>0.106</u>	<u>1.033</u>	<u>5.913</u>	<u>0.158</u>	<u>0.888</u>	<u>0.974</u>	<u>0.982</u>

TABLE III: Depth evaluation on the KITTI (K), and CityScapes (CS) Dataset. **IM** and **Sem.** stand for independent motion and semantics. #f indicates the number of frames during inference. D is the used dataset for training and evaluation. **Bold** is best. Underline is ours.

prediction head, that may not be fully optimized for the steering estimation task.

All experiments are conducted on standard urban driving datasets with predominantly daytime and clear-weather conditions. Evaluating robustness under adverse weather, low-light scenarios, and different geographic domains is an important direction for future work.

V. DISCUSSION

Training a unified encoder for multiple tasks introduces significant optimization challenges, particularly in balancing supervised and self-supervised losses. One notable issue was the dominance of segmentation loss over motion-based tasks, resulting in imbalanced feature learning. To address this, we proposed to use knowledge distillation from a multi-encoder teacher, which guided the learning of motion segmentation and scene flow, ensuring stable convergence. Additionally, the proposed multi-scale pose decoder enhanced depth estimation, mitigating inaccuracies in driving scenes. These improvements preserved the advantages of multi-task learning while addressing convergence issues.

Our approach represents an attempt to draw inspiration from human driving perception by integrating visual tasks, such as semantics, geometry, and motion, into a unified representation. We do not explicitly encode cognitive constraints or human priors. Instead, the human-inspired aspect lies in the choice and combination of perception tasks that humans implicitly rely on when driving.

Although the strategy shows promising results, higher numerical errors were observed in certain steering evaluations. These might come from two factors: (1) the used

encoder architecture may not be fully optimized for the steering estimation task, and (2) the prediction head was not extensively analyzed, as we only employed an attention pooler among many possible alternatives. Future work should explore other head architectures and refine the model to align more closely with the downstream task, potentially reducing these errors while preserving the benefits of multi-task learning. In addition, future studies can extend the downstream evaluation to other tasks such as trajectory planning, and joint lateral-longitudinal control to explore the generality of the learned representation.

Additionally, while our approach demonstrates the potential of multi-task learning, it introduces more computational complexity due to the simultaneous training of multiple tasks. This additional complexity is primarily incurred during pretraining and is paid only once. In contrast, inference benefits from a single unified encoder instead of multiple task-specific networks. Future work will investigate other distillation strategies and more efficient training to further reduce the computational problem.

Finally, a systematic evaluation of domain generalization, including adverse weather, low-light conditions, and different geographic regions, remains an important direction for future work.

VI. CONCLUSION

In this work, we present a unified encoder for autonomous driving, trained using multiple human-inspired visual perception cues necessary for navigation. By integrating knowledge from computer vision tasks—including depth, pose, 3D scene flow estimation, and various segmentation tasks—into a single encoder, our approach enables efficient and compact multi-task inference.

To address the challenges of multi-task training, we introduced a multi-scale pose decoder and applied knowledge distillation from a multi-encoder teacher, ensuring stable convergence and competitive performance for all the pretraining tasks. Our results show that this unified design achieves performance comparable to dedicated models for each task,

Model	Training error	Test error
CNN-GRU 64 units [4]	1.25 ± 1.02	5.06 ± 6.64
CNN-LSTM 64 units [4]	0.19 ± 0.05	3.17 ± 3.85
VAE-LSTM 64 units [7]	0.54 ± 0.26	4.70 ± 4.80
VAE-LSTM 19 units [7]	0.60 ± 0.30	6.75 ± 8.33
(1) Swin-AttnPool (ImageNet pretrained)	2.91 ± 2.23	11.03 ± 10.03
(2) Swin-AttnPool (Encoder unfrozen)	9.62 ± 3.24	16.46 ± 11.13
(3) Swin-AttnPool (Encoder frozen)	1.64 ± 1.63	5.41 ± 6.06

TABLE IV: Results from the passive lane-keeping evaluation across tenfold cross-testing. **Bold** is best. Underline is ours.

while enabling coherent steering estimation directly from the shared representation.

Beyond performance, this work highlights the importance of human-inspired perception in autonomous driving. By structuring the encoder to capture rich, contextual information similar to human visual processing, we take a step toward more interpretable and human-centric driving systems. Future work should further explore how human-like perception and reasoning mechanisms can enhance safety and adaptability in complex real-world scenarios.

REFERENCES

- [1] The Milwaukee Sentinel, "'phantom auto' will tour city," *The Milwaukee Sentinel*, 1926.
- [2] The Victoria Advocate, "Power companies build for your new electric living," *The Victoria Advocate*, 1957.
- [3] S. Singh, "Critical reasons for crashes investigated in the national motor vehicle crash causation survey," National Highway Traffic Safety Administration, Technical Report, 2015.
- [4] M. Lechner, R. Hasani, A. Amini, T. Henzinger, D. Rus, and R. Grosu, "Neural circuit policies enabling auditable autonomy," *Nature Machine Intelligence*, vol. 2, pp. 642–652, 2020.
- [5] J. Kim and J. Canny, "Interpretable learning for self-driving cars by visualizing causal attention," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2961–2969.
- [6] H. Xu, Y. Gao, F. Yu, and T. Darrell, "End-to-end learning of driving models from large-scale video datasets," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3530–3538.
- [7] A. Bairouk, M. Maras, S. Herlin, A. Amini, M. Blanchon, R. Hasani *et al.*, "Exploring latent pathways: Enhancing the interpretability of autonomous driving with a variational autoencoder," *arXiv preprint arXiv:2404.01750*, 2024.
- [8] V. Rausch, A. Hansen, E. Solowjow, C. Liu, E. Kreuzer, and J. Hedrick, "Learning a deep neural net policy for end-to-end control of autonomous vehicles," in *2017 American Control Conference (ACC)*, 2017, pp. 4914–4919.
- [9] S. Sharma, G. Tewolde, and J. Kwon, "Behavioral cloning for lateral motion control of autonomous vehicles using deep learning," in *2018 IEEE International Conference on Electro/Information Technology (EIT)*, 2018, pp. 490–495.
- [10] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.
- [11] T.-D. Do, M.-T. Duong, Q.-V. Dang, and M.-H. Le, "Real-time self-driving car navigation using deep neural network," in *2018 4th International Conference on Green Technology and Sustainable Development (GTSD)*, 2018, pp. 7–12.
- [12] L. Capito, U. Ozguner, and K. Redmill, "Optical flow based visual potential field for autonomous driving," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 885–891.
- [13] A. Majid, S. Kausar, S. Tehsin, and A. Jameel, "A fast panoptic segmentation network for self-driving scene understanding," *Computer Systems Science & Engineering*, vol. 43, no. 1, 2022.
- [14] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, "Panoptic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9404–9413.
- [15] J. Jain, J. Li, M. Chiu, A. Hassani, N. Orlov, and H. Shi, "Oneformer: One transformer to rule universal image segmentation," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 2989–2998.
- [16] A. Bhoi, "Monocular depth estimation: A survey," *arXiv preprint arXiv:1901.09402*, 2019.
- [17] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Neural Information Processing Systems*, 2014.
- [18] T. Zhou, M. Brown, N. Snavely, and D. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1851–1858.
- [19] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3828–3838.
- [20] Y. Sun and B. Hariharan, "Dynamo-depth: Fixing unsupervised depth estimation for dynamical scenes," *arXiv preprint arXiv:2310.18887*, vol. abs/2310.18887, 2023.
- [21] J. Hur and S. Roth, "Self-supervised monocular scene flow estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7396–7405.
- [22] —, "Self-supervised multi-frame monocular scene flow," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2684–2694.
- [23] Y. Jiao, T. Tran, and G. Shi, "Effiscene: Efficient per-pixel rigidity inference for unsupervised joint learning of optical flow, depth, camera pose and motion segmentation," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 5534–5543.
- [24] H. M. Eraqi, M. N. Moustafa, and J. Honer, "End-to-end deep learning for steering autonomous vehicles considering temporal dependencies," *arXiv preprint arXiv:1710.03804*, 2017.
- [25] G. M. Jacob, V. Agarwal, and B. Stenger, "Online knowledge distillation for multi-task learning," *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2358–2367, 2023.
- [26] Y. Wang, Q. Zhao, Y. Gan, and Z. Xia, "Joint-confidence-guided multi-task learning for 3d reconstruction and understanding from monocular camera," *IEEE Transactions on Image Processing: A Publication of the IEEE Signal Processing Society*, 2023.
- [27] Y. Cui, C. Han, and D. Liu, "Collaborative multi-task learning for multi-object tracking and segmentation," *Journal on Autonomous Transportation Systems*, vol. 1, pp. 1–23, 2023.
- [28] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang *et al.*, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10012–10022.
- [29] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3213–3223.
- [30] D. Han, J. Shin, N. Kim, S. Hwang, and Y. Choi, "Transdssl: Transformer based depth estimation via self-supervised learning," *IEEE Robotics and Automation Letters*, vol. 7, pp. 10969–10976, 2022.
- [31] F. Chen, G. Datta, S. Kundu, and P. Beerel, "Self-attentive pooling for efficient deep learning," 2022. [Online]. Available: <https://arxiv.org/abs/2209.07659>
- [32] J. Watson, O. Mac Aodha, V. Prisacariu, G. Brostow, and M. Firman, "The temporal opportunist: Self-supervised multi-frame monocular depth," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1164–1174.
- [33] N. Zhang, F. Nex, G. Vosselman, and N. Kerle, "Lite-mono: A lightweight cnn and transformer architecture for self-supervised monocular depth estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 18537–18546.
- [34] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova, "Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, July 2019, pp. 8001–8008.
- [35] T. W. Hui, "Rm-depth: Unsupervised learning of recurrent monocular depth in dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1675–1684.
- [36] A. Gordon, H. Li, R. Jonschkowski, and A. Angelova, "Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8977–8986.
- [37] J. Zhong, X. Huang, and X. Yu, "Multi-frame self-supervised depth estimation with multi-scale feature fusion in dynamic scenes," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 2553–2563.