D^3Nav: Data-Driven Driving Agents for Autonomous Vehicles in Unstructured Traffic

Aditya Nalgunda Ganesh
adityang.github.io
Gowri Srinivasa
gsrinivasa@pes.edu

PES Center for Pattern Recognition
Department of Computer Science and Engineering, PES University
Bengaluru, India

Abstract

Navigating unstructured traffic autonomously requires handling a plethora of edge cases, traditionally challenging for perception and path-planning modules due to scarce real-world data and simulator limitations. By employing the next-token prediction task, LLMs have demonstrated to have learned a world model. D^3Nav bridges this gap by employing a quantized encoding to transform high-dimensional video data (Fx3x128x256) into compact integer embeddings (Fx128) which are fed into our world model. D^3Nav's world model is trained on the next-video-frame prediction task and simultaneously predicts the desired driving signal. The architecture's compact nature enables real-time operation while adhering to stringent power constraints. D^3Nav's training on diverse datasets featuring unstructured data results in the model's proficient prediction of both future video frames and the driving signal. We make use of automated labeling to generate importance masks accentuating pedestrians and vehicles to aid our encoding system in focusing on objects of interest. These capabilities are an improvement in end-to-end autonomous navigation systems, particularly in the context of unstructured traffic environments. Our contribution includes our driving agent D^3Nav and our embeddings dataset of unstructured traffic. We make our code and dataset[1] public.

1 Introduction

Autonomous vehicle technology has rapidly evolved over the past decade, sparking significant interest both in academia and industry [2, 3, 12, 17, 26, 40]. The overarching aim has been realizing a fully autonomous system that can navigate complex traffic scenarios with the same dexterity as a human driver, in both structured and unstructured traffic. In the context of autonomous vehicles, "Unstructured Traffic" refers to environments where traffic rules and infrastructure are not clearly defined or predictable, such as roads without markings, areas with mixed traffic like pedestrians and cyclists, or unpredictable urban settings. These scenarios pose significant challenges for autonomous vehicles, which rely on predefined rules and algorithms, requiring advanced perception and decision-making capabilities to navigate effectively. The predominant approach to autonomous driving has been the integration of modular AI systems with hard-coded logic [20, 25, 29, 37, 42]. These systems, designed to handle specific tasks, were combined hierarchically, each module contributing its piece to the

[1] Please refer to the supplementary material

overall puzzle of autonomous driving. While this modular design provided granular control and allowed for specialized optimizations, it also introduced complexities. Such systems tend to be fragile by nature and their performance tends to suffer when they are taken out of structured traffic.

End-to-end systems emerged as an alternative, aiming to learn driving directly from raw sensor data to control commands [5, 7, 11, 18, 31, 45]. While conceptually appealing due to their simplicity, they have often shown limitations in grasping the multifaceted nuances of driving. For instance, when confronted with rare or previously unseen scenarios, such systems have failed to scale with data. The allure of large modular AI systems lies in their capacity to yield results rapidly, each module tailored for a specific task. However, the system can be quite fragile. Training each module to handle the vast spectrum of edge cases proficiently is a daunting task. Manually annotating data, especially for rare and complex scenarios, is time-consuming, expensive, and prone to human error. Furthermore, the inherent nature of these systems makes it arduous to capture the subtleties of human driving, particularly the implicit social contracts and non-verbal communication cues exchanged between drivers.

Amid these challenges, Generative Large Language Models (LLMs) powered by the GPT architecture [34] have heralded a new era in machine learning. By focusing on the next-token prediction task, LLMs have demonstrated proficiency in natural language tasks and have also been shown to possess an internal world model. What is important to note about the GPT architecture is that it is inherently an excellent sequence-to-sequence modeling tool and is not specific to language. Recent approaches have validated the domain-agnostic properties of GPT [6, 8, 35, 36].

Our Contributions. To build an agent with a world model suited to autonomous navigation, we train D^3Nav on the next-video-frame prediction task. D^3Nav stands for Data-Driven Driving and can be applied as an agent in the context of autonomous vehicles. Here, by Data-Driven Driving we refer to learning from raw unlabeled driving video data. D^3Nav applies the quantized video encoding for compression with an autoregressive architecture to generate future driving signals and future video frames based on past driving video inputs. By harnessing the strengths of generative pre-trained transformers (GPT) for sequence modeling and building an internal world model, D^3Nav provides an efficient system for driving signal generation. By making use of a compact architecture, our system can operate in real time and under a tight power budget making it suitable to be deployed on-vehicle.

2 Related Work

GPT. Training GPT at a large scale, as exemplified in models such as GPT-3 [6], GPT-4 [32] and LLAMA [41] has shown several noteworthy emergent properties. These properties include an enhanced capacity for natural language understanding and generation, improved performance in zero-shot and few-shot learning scenarios, and the ability to generate creative and contextually relevant responses across a wide range of topics. The remarkable ability of advanced GPT models to have an *internal world model* and generate contextually relevant information makes them particularly suitable for applications such as autonomous vehicle technology. However, the challenge in applying GPT to such a domain lies in having to input high-dimensional video data.

GPT for Vision. We have seen recent approaches of feeding video and images into GPT. ImageGPT [33] reveals that GPT's architecture, renowned for its efficacy in language models, can be adeptly adapted to process and generate visual data as well. ImageGPT is trained on the

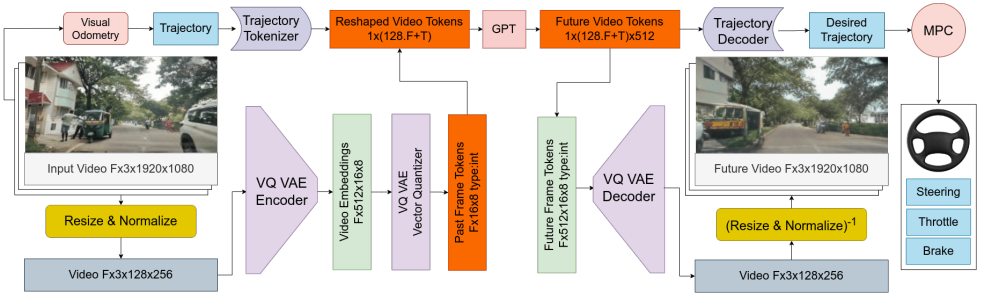

Figure 1: D^3Nav architecture takes the past F frames as input. These frames are resized, normalized, and input to the encoder. The resultant embeddings are tokenized and input to GPT which yields future tokens that are decoded to produce the next F frames of video output. The vehicle's recent trajectory is extracted using Visual Odometry and is fed in as context. Finally, the future trajectory tokens are decoded to produce the model prediction for the desired vehicle trajectory. We then use MPC to compute the driving signal. Note that this is the model being used in inference mode, during training, the Encoder-Decoder, Trajectory Quantizer, and the GPT are trained separately.

MNIST dataset and its scope is limited to handwritten digits in single frames. VideoGPT [46] and [14] apply GPT in the context of raw video generation. VideoGPT makes use of the BAIR Robot dataset, UCF-101, and the Tumbler GIF dataset. However, these approaches do not explore autonomous navigation datasets and the extraction of a driving signal.

Generative AI for Robotics. Wayve's closed-source Generative AI for Autonomy (GAIA-1) [19] shows an application of GPT-based generative video models in the domain of autonomy. GAIA-1's training dataset of about 4,700 hours was gathered in the structured traffic context of London, UK between 2019 and 2023. GAIA-1 is also a 9B parameter world model which is difficult to get working within a vehicle given the power constraints. Approaches like [14, 23, 38] show that it is possible to control and direct the generation of video frames from a generative model which is relatively compact when compared to GAIA-1. However, they do not focus on the domain of autonomous navigation. Finally, [22] shows how one can use Generative Adversarial Networks as neural simulators in the context of autonomous vehicles in structured traffic. Taking inspiration from these works, we build a power-efficient and open-source driving agent familiar with navigating unstructured traffic.

3 Proposed Work

3.1 Architecture

As described in Figure 1, D^3Nav takes the past F video frames as input and produces the next F video frames as output. The down-scaled video is then quantized by the encoder leading to efficient compression to produce video embeddings of dimensions $F \times 128$. The tokenized video embeddings are subsequently input into the GPT world model, known for its sequence-to-sequence transformation capabilities. The world model outputs future video tokens, which are prospective representations of driving signals and future video frames. The trajectory tokens are decoded and Model Predictive Control (MPC) [4] is used to compute the optimal series of actuator signals. The future video tokens are passed through a decoder, which is trained symmetrically with the encoder which reconstructs the video representation.

Figure 2: Importance Maps. The first column shows the input video frames and the second column presents the corresponding importance maps, with the entire scene's importance scaled from zero to one as a function of distance and semantics.

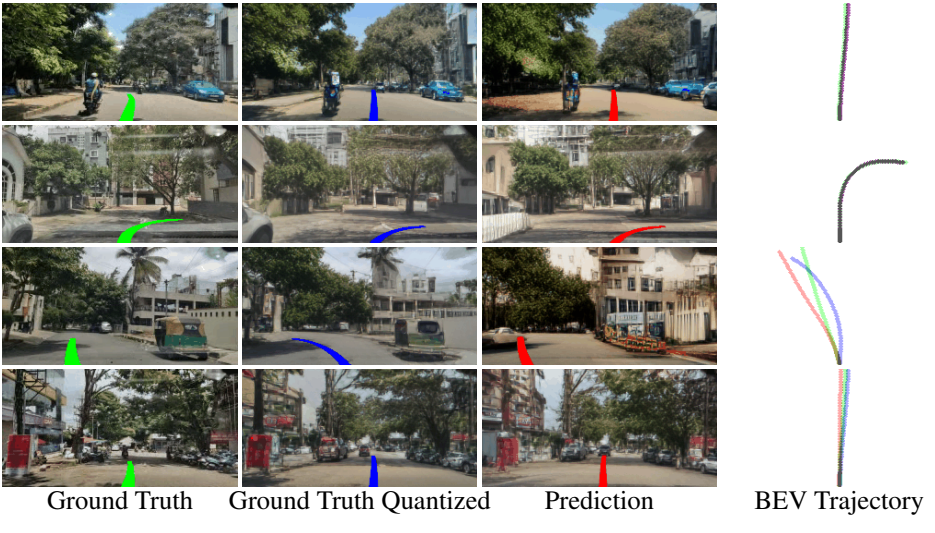

Figure 3: *D^3Nav* **Trajectory Output**. We have plotted out *D^3Nav*'s future video frame prediction and have overlayed the desired driving signal. The ground truth is plotted in green, the quantized ground truth is plotted in blue and the model prediction is plotted in red. We have projected the trajectories onto the image plane. We have also plotted the Bird's-Eye-View (BEV).

Figure 4: D^3Nav **Output**. Above are frames that are generated by D^3Nav when given a past video context as a prompt. The first column presents the last video frame from the input prompt. The subsequent three columns are the future frames predicted. The model is able to predict the flow of the unstructured traffic.

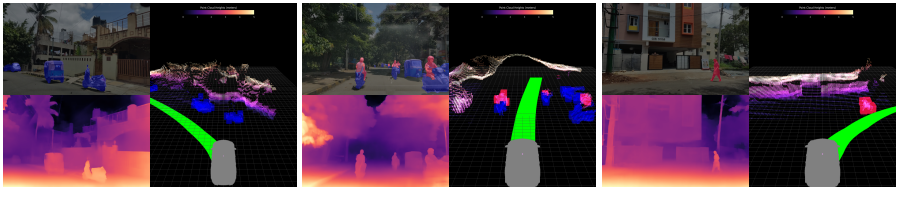

Figure 5: We extend the Bengaluru Driving Dataset [13] with semantic labels and trajectory labels. This trajectory has been calculated from visual odometry. Each panel consists of the RGB image with 2D semantic labels on the top left, the depth map on the bottom left, and the 3D plot on the right. The vehicle and pedestrian classes are colored blue and red respectively. Objects without classes have been plotted as a height map for the sake of visualization. The vehicle and its future trajectory have been plotted out in grey and green respectively.

3.2 Auto-Labeling Pipeline

To build a driving agent that scales well, we must ensure that the dataset generation pipeline also scales well. This reduces human involvement in generating the dataset. We make use of auto-labeling in the context of our semantic masks, depth maps, driving signal labels to generate importance maps. The importance masks only guide the model to have a more detailed representation of these regions as shown in Figure 2.

Driving Signal Labeling. We make use of Visual-Odometry, GPS, and IMU to extract the traversed ego-trajectory and speed from the video data as shown in Figure 5. We make use of the Shi-Tomasi Feature Extraction [39] to extract and track 2000 key points from the video and apply SLAM [30] to extract the trajectory.

Depth Boosting. Taking inspiration from the depth boosting techniques [13, 27, 28], we merge the depth maps from the various resolutions to generate high-resolution depth maps with global consistency. We use this method to generate depth labels for the Indian Driving Dataset as shown in Figure 5.

Semantic Segmentation auto-labeling. To produce high resolution 2D semantic labels, we take inspiration from PointRend [9]. We take an image as input and produce a coarse intermediate segmentation map using an existing segmentation approach MaskRCNN [16]. This coarse map is gradually up-sampled using bi-linear interpolation and only the regions of the resized map with high uncertainty are refined by a lightweight multi-layered perceptron. The uncertain regions typically include the boundaries of objects. As shown in Figure 5, we label the semantics and using the depth maps we are able to project them into 3D semantic occupancy grids.

Importance Maps. We use depth and semantic labels to assign higher importance to regions of interest in the image. These maps are used to bias the loss of our Encoder-Decoder to focus more on objects of importance (semantics) and those closer to the camera (depth). We accept as input an RGB frame, a corresponding segmentation mask, and a depth map, along with several parameters that guide the mask generation. The output is an importance map (F_{IM}) which assigns weights to different parts of the image based on depth and semantic cues as shown in Figure 2. Further details on the computation of these maps are presented in Appendix A2.

3.3 Training

Our architecture is split into three main components. The Encoder-Decoder which condenses the image input into a quantized embedding space. The trajectory quantizer takes the trajectory as input and tokenizes it. The GPT world dynamics model takes in the quantized image embeddings and trajectories as input and learns the world dynamics. These three are trained separately and are integrated together in inference mode as shown in Figure 1.

Encoder-Decoder. We train the Encoder-Decoder system inspired by VQ-VAE [43] to learn a compact representation of unstructured traffic video frames. At first, the model learns the dataset reasonably well, but most of the frames it generates tend to be a bit blurry and unfocused. To mitigate this, we apply the importance maps to guide it to focus on pedestrians, vehicles, and nearby objects. This produces a model that is able to transform between the image space (3x128x256) and the integer latent space of (128). We experiment with latent spaces of dimensions 128 and 512, with a compression ratio of 256x and 64x respectively. We ended up using the smaller latent space as it allowed us to feed in more frames as context into the GPT module. With each frame taking up 128 tokens and setting our GPT to have a

maximum context length of 4096, of which the first 2048 are taken up by the input and the last 2048 are the output. This allows us to feed in 16 frames as context.

GPT. Once we generate our embeddings dataset from the Encoder-Decoder pair, we begin training the GPT module. We feed in 3 to 6 frames as input and query for 3 to 6 frames as output respectively. To feed in the frames, we structured our prompt with delimiters and end tokens. Training with multiple frames as context provided temporally consistent results.

Driving Signal. We are able to extract the past vehicle trajectory using Visual Odometry. This trajectory is encoded into tokens for every frame using a set of template trajectories and K-Means Clustering. Initially, we observe that the model takes a large number of epochs to learn the trajectory tokens since they are a small fraction ($< 10\%$) of the total number of tokens. We fix this by weighting the loss associated with the trajectory tokens to increase their importance. To extract the final driving signal, we apply Model Predictive Control (MPC) [4] to compute the optimal steering, throttle and braking values.

Loss. Our loss function is a convex combination of three components: image, video and trajectory reconstruction losses. Since the trajectory tokens make up a smaller fraction of the total number of tokens produced, they have a higher weight in the loss function. We use the Cross-Entropy loss to supervise the tokens predicted. Further details are in Appendix A3.

4 Experiments

We train D^3Nav on a laptop with an Intel i7-12700H (20 threads) and NVIDIA GeForce RTX 3070 Mobile GPU with 8 GB VRAM. We make use of A100 GPU clusters to train larger and deeper networks. To focus performance in unstructured traffic, our network has been trained on the Indian Driving Dataset [44] and the Bengaluru Driving Dataset [13].

4.1 Datasets

Indian Driving Dataset [44] . The IDD has a total of about 7974 frames with 6993 and 981 frames for training and testing respectively.

Bengaluru Driving Dataset [13]. The raw video BDD has a total of about 71 thousand frames. We split it to have 10% for testing.

CommaVQ Dataset [47]. The dataset consists of 100,000 heavily compressed driving videos which we use as a base to fine-tune our Encoder-Decoder pair.

Bengaluru Embeddings and Trajectory Dataset (Ours). We extend BDD with image embeddings and vehicle trajectory labels. The image embeddings allow us to feed the video data into GPT as a condensed and quantized set of embeddings. Each image can be represented as a set of 128 tokens. Extending the BDD video dataset gives us around 9 million tokens to train on. The vehicle's ego-motion is extracted from the video dataset using Visual Odometry.

4.2 Quantitative Results

We evaluate D^3Nav in the domains of image reconstruction, next-video-frame tokens prediction and driving signal (trajectory) accuracy.

Encoder-Decoder. We evaluate our Encoder-Decoder on RMSE, a_1, a_2, a_3, and compression while tracking the hyper-parameters such as learning rate, loss weight distribution, number of epochs and batch size as shown on Table 2. Once the Encoder-Decoder achieved a satisfactory score, we fine-tuned it using the importance maps to focus the model on the important objects on the road. While the RMSE score on the image increases by about 37%,

Size	Hyperparameters					Metrics				
	L	D_E	D_R	D_A	LR	$F1$	$Prec$	DTW	CE	FPS
XS	6	0.200	0.300	0.0003	0.00003	0.318	0.318	27.2	3.159	35.021
S	12	0.500	0.500	0.0003	0.0003	0.370	0.370	26.1	2.766	32.683
M	24	0.100	0.500	0.0003	0.0002	0.395	0.395	24.6	2.522	24.570
L	36	0.200	0.200	0.100	0.0003	0.458	0.458	23.2	2.277	20.276
XL	48	0.500	0.300	0.300	0.003	0.462	0.462	18.5	2.230	17.702

Table 1: **Quantitative Results** on our proposed architecture comparing the optimal hyper-parameters and metrics achieved. The table shows the hyper-parameters Number of Layers L, Embeddings Dropout D_E, Residual Dropout D_R, Attention Dropout D_A, and Learning Rate LR. We have evaluated on the metrics $F1$, Precision $Prec$, Dynamic Time Warping Distance [1] DTW, Cross Entropy CE and Frame Rate FPS.

Model	Hyperparameters			Metrics					
	LR	β	BS	$RMSE$	M. $RMSE$	a_1	a_2	a_3	$Comp.$
$V_{16\times8}$	0.0003	0.1	32	0.3920	0.3920	0.7952	0.9549	0.9514	256x
$V_{32\times16}$	0.00003	0.25	32	0.3649	0.3649	0.7979	0.9637	0.9590	64x
$V^{IM}_{16\times8}$	0.0003	0.1	32	0.5396	0.3945	0.7231	0.8892	0.9051	256x
$V^{IM}_{32\times16}$	0.00003	0.25	32	0.4982	0.3587	0.7418	0.8979	0.9134	64x

Table 2: Our Encoder-Decoder pair was trained on our video datasets to learn an efficient embedding space. We optimize for Learning Rate LR, Beta β, Batch Size BS. Beta decides the weight given to the commitment loss[43]. We evaluate on the metrics $RMSE$, Masked $RMSE$, $a1$, $a2$, $a3$ and $Compression$. a_i is the fraction of predictions where the threshold maximum between gt/pred or pred/gt is less than 1.25^i. Models with superscript IM were trained with importance masking. We use the $V^{IM}_{16\times8}$ as our primary encoder.

the masked RSME score is largely unaffected. This indicates that the fine-tuned model has learned the masked regions better.

GPT. We evaluate the GPT sub-module on the metrics of F1 score, Perplexity, Precision, Recall, Cross Entropy, and measure its frame rate. We track the hyper-parameters: number of layers, the various dropout values, forward expansion, learning rate, and weight decay as shown in Table 1. We also visualize the future frames predicted in Figure 4. Through our hyper-parameter sweep, we observe that increasing the number of frames of context has a positive correlation with the performance of the model. An increase in the number of layers is shown to increase accuracy. We also evaluate the speed of each variant of our model and show that D^3Nav_M strikes a balance between speed and accuracy. While D^3Nav_{XL} is not as fast, large models can be used as simulators to supervise smaller models. We compare our approach to Action-RNN [10], SAVP [24], WorldModel [15], GameGAN [21], and DriveGAN [22] in terms of image reconstruction as shown in Table 3 and we visualize the same in Table 4.

Driving Signal. Once D^3Nav had been trained on the larger embeddings dataset, we fine-tuned it on the trajectory dataset. We observe that our model can predict the desired trajectory token with high accuracy as shown in the DWT Distance metric in Table 1. We visualize the same in Figure 3. We plot the ground truth trajectory and the quantized ground truth trajectory along with the model's prediction. We observe that the model is able to associate trajectory templates that are similar to each other as even if the model does not predict the exact trajectory correctly, it tends to get the general direction correct.

Model	Abs. Rel.	Sq. Rel.	RMSE	a1	a2	a3
Action-RNN [10]	2.874	1.459	8.615	0.605	0.807	0.902
SAVP [24]	2.663	1.290	8.359	0.607	0.890	0.901
WorldModel [15]	2.984	1.790	9.472	0.406	0.614	0.736
GameGAN [21]	3.056	1.481	8.541	0.589	0.786	0.884
DriveGAN [22]	2.368	1.329	8.679	0.586	0.785	0.881
Ours	**2.293**	**1.126**	**7.790**	**0.723**	**0.889**	**0.905**

Table 3: A comparison of the next frame image reconstruction metrics of D^3Nav with the published visuals of Action-RNN, SAVP, WorldModel, GameGAN, and DriveGAN on Real World Driving (RWD) [22].

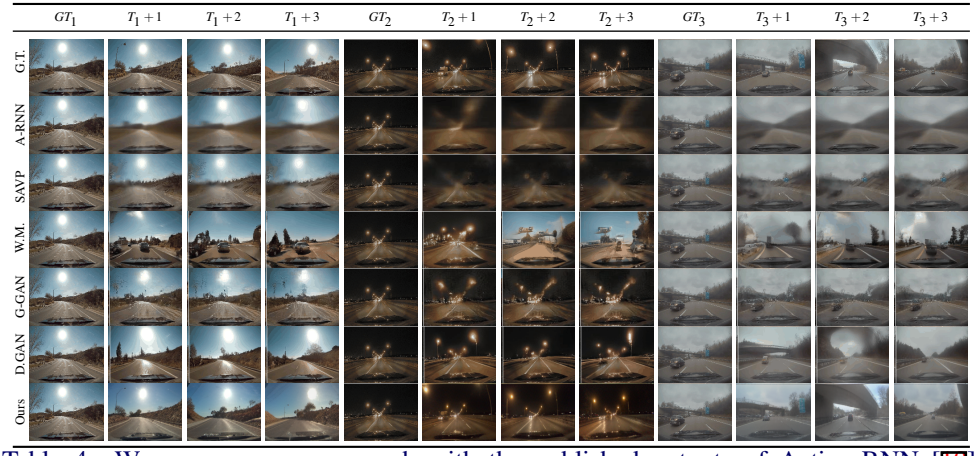

Table 4: We compare our approach with the published outputs of Action-RNN [10], SAVP [24], WorldModel [15], GameGAN [21], and DriveGAN [22]. Above we have provided three examples being GT_1, GT_2 and GT_3 along with the corresponding model outputs for the same at $T+1$, $T+2$ and $T+3$

4.3 Discussion and Limitations

As demonstrated by the results, D^3Nav produces temporally coherent video output given a video context prompt. The use of importance maps focuses the Image Encoder on semantics (discernible objects) and depth (higher importance to objects closer to the camera). Further, the results demonstrate that D^3Nav can predict driving signals with high accuracy. Presently, the Image Encoder compresses frames to a latent space of size 16×8. This limit was chosen due to memory constraints while training and in consideration of the number of frames that must be input to the model as context. Increasing the context length would allow us to increase the number of tokens per frame. This would increase the latent space dimensions, thereby increasing the image reconstruction quality. While the proposed D^3Nav model is relatively lightweight, we expect the next frame prediction and trajectory prediction system's performance to scale with the size of the dataset.

5 Conclusions

D^3Nav offers a compelling solution to the inherent challenges of autonomous navigation in unstructured traffic environments. By utilizing quantized encodings, our system efficiently compresses high-dimensional video and trajectory data into embeddings that retain essential visual information. The automated importance labeling mechanism is pivotal in highlighting critical elements such as pedestrians and vehicles, enabling the predictive model to focus on key aspects of the traffic scene without the need for exhaustive human labeling efforts. The ability of D^3Nav to efficiently predict future video frames and the desired control signal with minimal human intervention through the entire training pipeline marks an advancement in the field. This progress is particularly beneficial for building a driving agent or simulating training scenarios where real-world data is scarce or incomplete. The generated dataset of embeddings and trajectory labels presents a valuable asset for further research and development of autonomous driving agents and simulators.

References

[1] *Dynamic Time Warping*, pages 69–84. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. ISBN 978-3-540-74048-3. doi: 10.1007/978-3-540-74048-3_4.

[2] James M. Anderson, Nidhi Kalra, Karlyn D. Stanley, Paul Sorensen, Constantine Samaras, and Oluwatobi A. Oluwatola. *Autonomous Vehicle Technology: A Guide for Policymakers*. RAND Corporation, 2014. ISBN 9780833083982. URL http://www.jstor.org/stable/10.7249/j.ctt5hhwgz.

[3] Claudine Badue, Ranik Guidolini, Raphael Carneiro, Pedro Azevedo, Vinicius Cardoso, Avelino Forechi, Luan Ferreira Reis de Jesus, Rodrigo Berriel, Thiago Paixão, Filipe Mutz, Lucas Veronese, Thiago Oliveira-Santos, and Alberto De Souza. Self-driving cars: A survey. *Expert Systems with Applications*, 165:113816, 08 2020. doi: 10.1016/j.eswa.2020.113816.

[4] Alberto Bemporad. Model predictive control design: New trends and tools. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 6678–6683, 2006. doi: 10.1109/CDC.2006.377490.

[5] Mariusz Bojarski, Davide Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Larry Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars, 04 2016.

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[7] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2722–2730, 2015. doi: 10.1109/ICCV.2015.312.

[8] Jun Chen, Han Guo, Kai Yi, Boyang Albert Li, and Mohamed Elhoseiny. Visualgpt: Data-efficient adaptation of pretrained language models for image captioning. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18009–18019, 2021.

[9] Bowen Cheng, Omkar Parkhi, and Alexander Kirillov. Pointly-supervised instance segmentation. *ArXiv*, 2021.

[10] Silvia Chiappa, Sébastien Racanière, Daan Wierstra, and Shakir Mohamed. Recurrent environment simulators. *CoRR*, abs/1704.02254, 2017. URL http://arxiv.org/abs/1704.02254.

[11] Felipe Codevilla, Matthias Miiller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, page 1–9. IEEE Press, 2018. doi: 10.1109/ICRA.2018.8460487.

[12] Daniel Fagnant and Kara Kockelman. Preparing a nation for autonomous vehicles: Opportunities, barriers and policy recommendations. *Transportation Research Part A: Policy and Practice*, 77, 07 2015. doi: 10.1016/j.tra.2015.04.003.

[13] Aditya N Ganesh, Dhruval Pobbathi Badrinath, Harshith Mohan Kumar, Priya S, and Surabhi Narayan. Octran: 3d occupancy convolutional transformer network in unstructured traffic scenarios. Spotlight Presentation at the Transformers for Vision Workshop, CVPR, 2023. Transformers for Vision Workshop, CVPR 2023.

[14] Songwei Ge, Thomas Hayes, Harry Yang, Xi Yin, Guan Pang, David Jacobs, Jia-Bin Huang, and Devi Parikh. Long video generation with time-agnostic vqgan and time-sensitive transformer. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 102–118, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-19790-1.

[15] David Ha and Jürgen Schmidhuber. World models. *CoRR*, abs/1803.10122, 2018. URL http://arxiv.org/abs/1803.10122.

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. doi: 10.1109/ICCV.2017.322.

[17] Kersten Heineke, Philipp Kampshoff, Armen Mkrtchyan, and Emily Shao. Self-driving car technology: When will the robots hit the road?, May 2017.

[18] Bilal Hejase, Ekim Yurtsever, Teawon Han, Baljeet Singh, Dimitar P. Filev, H. Eric Tseng, and Umit Ozguner. Dynamic and interpretable state representation for deep reinforcement learning in automated driving. *IFAC-PapersOnLine*, 55(24):129–134, 2022. ISSN 2405-8963. doi: https://doi.org/10.1016/j.ifacol.2022.10.273. 10th IFAC Symposium on Advances in Automotive Control AAC 2022.

[19] Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving, 2023.

[20] Joel Janai, Fatma Güney, Aseem Behl, and Andreas Geiger. Computer vision for autonomous vehicles: Problems, datasets and state-of-the-art. *Found. Trends Comput. Graph. Vis.*, 12:1–308, 2017.

[21] Seung Wook Kim, Yuhao Zhou, Jonah Philion, Antonio Torralba, and Sanja Fidler. Learning to simulate dynamic environments with gamegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[22] Seung Wook Kim, Jonah Philion, Antonio Torralba, and Sanja Fidler. Drivegan: Towards a controllable high-quality neural simulation. *CoRR*, abs/2104.15060, 2021. URL https://arxiv.org/abs/2104.15060.

[23] Guillaume Le Moing, Jean Ponce, and Cordelia Schmid. Ccvs: Context-aware controllable video synthesis. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 14042–14055. Curran Associates, Inc., 2021.

[24] Alex X. Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018.

[25] John Leonard, Jonathan How, Seth Teller, Mitch Berger, Stefan Campbell, Gaston Fiore, Luke Fletcher, Emilio Frazzoli, Albert Huang, Sertac Karaman, Olivier Koch, Yoshiaki Kuwata, David Moore, Edwin Olson, Steve Peters, Justin Teo, Robert Truax, Matthew Walter, David Barrett, Alexander Epstein, Keoni Maheloni, Katy Moyer, Troy Jones, Ryan Buckley, Matthew Antone, Robert Galejs, Siddhartha Krishnamurthy, and Jonathan Williams. A perception-driven autonomous urban vehicle. *Journal of Field Robotics*, 25(10):727–774, 2008. doi: https://doi.org/10.1002/rob.20262. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.20262.

[26] Todd Litman. Autonomous vehicle implementation predictions: Implications for transport planning, January 2020. URL https://trid.trb.org/view/1678741.

[27] S. Mahdi H. Miangoleh, Sebastian Dille, Long Mai, Sylvain Paris, and Yağız Aksoy. Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9680–9689, 2021. doi: 10.1109/CVPR46437.2021.00956.

[28] Seyed Mahdi Hosseini Miangoleh. Boosting monocular depth estimation to high resolution. Master's thesis, Simon Fraser University, 2022.

[29] Michael Montemerlo, Jan Becker, Suhrid Bhat, Hendrik Dahlkamp, Dmitri Dolgov, Scott Ettinger, Dirk Haehnel, Tim Hilden, Gabe Hoffmann, Burkhard Huhnke, Doug Johnston, Stefan Klumpp, Dirk Langer, Anthony Levandowski, Jesse Levinson, Julien Marcil, David Orenstein, Johannes Paefgen, Isaac Penny, Anna Petrovskaya, Mike Pflueger, Ganymed Stanek, David Stavens, Antone Vogt, and Sebastian Thrun. Junior:

The stanford entry in the urban challenge. *Journal of Field Robotics*, 25(9):569–597, 2008. doi: https://doi.org/10.1002/rob.20258.

[30] Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. doi: 10.1109/TRO.2015.2463671.

[31] Aditya NG, Dhruval PB, Jehan Shalabi, Shubhankar Jape, Xueji Wang, and Zubin Jacob. Thermal voyager: A comparative study of rgb and thermal cameras for night-time autonomous navigation. In *Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA)*, Yokohama, Japan, 2024.

[32] OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023.

[33] Rushikesh Pupale, Adarsh Shrivastava, and Pradeep Singh. Image generation using gpt-2. In Lalit Garg, Dilip Singh Sisodia, Nishtha Kesswani, Joseph G. Vella, Imene Brigui, Peter Xuereb, Sanjay Misra, and Deepak Singh, editors, *Information Systems and Management Science*, pages 131–141, Cham, 2023. Springer International Publishing. ISBN 978-3-031-13150-9.

[34] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. Technical report, OpenAI, 2018.

[35] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gómez Colmenarejo, Alexander Novikov, Gabriel Barth-maron, Mai Giménez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. A generalist agent. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=1ikK0kHjvj. Featured Certification, Outstanding Certification.

[36] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 12 2020. doi: 10.1162/tacl_a_00349.

[37] Wilko Schwarting, Javier Alonso-Mora, and Daniela Rus. Planning and decision-making for autonomous vehicles. *Annual Review of Control, Robotics, and Autonomous Systems*, 1(1):187–210, 2018. doi: 10.1146/annurev-control-060117-105157.

[38] Younggyo Seo, Kimin Lee, Fangchen Liu, Stephen James, and Pieter Abbeel. Harp: Autoregressive latent video prediction with high-fidelity image generator. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 3943–3947, 2022. doi: 10.1109/ICIP46576.2022.9897982.

[39] Jianbo Shi and Tomasi. Good features to track. In *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994. doi: 10.1109/CVPR.1994.323794.

[40] Sebastian Thrun. Toward robotic cars. *Commun. ACM*, 53(4):99–106, apr 2010. ISSN 0001-0782. doi: 10.1145/1721654.1721679.

[41] Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, M.A. Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian X. Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288, 2023.

[42] Chris Urmson, Joshua Anhalt, Drew Bagnell, Christopher Baker, Robert Bittner, M. N. Clark, John Dolan, Dave Duggins, Tugrul Galatali, Chris Geyer, Michele Gittleman, Sam Harbaugh, Martial Hebert, Thomas M. Howard, Sascha Kolski, Alonzo Kelly, Maxim Likhachev, Matt McNaughton, Nick Miller, Kevin Peterson, Brian Pilnick, Raj Rajkumar, Paul Rybski, Bryan Salesky, Young-Woo Seo, Sanjiv Singh, Jarrod Snider, Anthony Stentz, William "Red" Whittaker, Ziv Wolkowicki, Jason Ziglar, Hong Bae, Thomas Brown, Daniel Demitrish, Bakhtiar Litkouhi, Jim Nickolaou, Varsha Sadekar, Wende Zhang, Joshua Struble, Michael Taylor, Michael Darms, and Dave Ferguson. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics*, 25(8):425–466, 2008. doi: https://doi.org/10.1002/rob.20255. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.20255.

[43] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6309–6318, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[44] G. Varma, A. Subramanian, A. M. Namboodiri, M. Chandraker, and C. V. Jawahar. IDD: A dataset for exploring problems of autonomous navigation in unconstrained environments. *CoRR*, abs/1811.10200, 2018. URL http://arxiv.org/abs/1811.10200.

[45] Huazhe Xu, Yang Gao, Fisher Yu, and Trevor Darrell. End-to-end learning of driving models from large-scale video datasets. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3530–3538, 2017. doi: 10.1109/CVPR.2017.376.

[46] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers, 2021.

[47] Yassine Yousfi. Commavq dataset: 100,000 heavily compressed driving videos for machine learning research. https://github.com/commaai/commavq and https://huggingface.co/datasets/commaai/commavq, 2023.