PIAug - Physics Informed Augmentation for Learning Vehicle Dynamics for Off-Road Navigation

Parv Maheshwari^{1,2}, Wenshan Wang², Samuel Triest², Matthew Sivaprakasam², Shubhra Aich², John G. Rogers III³, Jason M. Gregory³ and Sebastian Scherer²

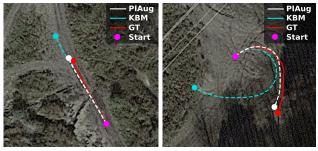
Abstract— Modeling the precise dynamics of off-road vehicles is a complex vet essential task due to the challenging terrain they encounter and the need for optimal performance and safety. Recently, there has been a focus on integrating nominal physics-based models alongside data-driven neural networks using Physics Informed Neural Networks. These approaches often assume the availability of a well-distributed dataset; however, this assumption may not hold due to regions in the physical distribution that are hard to collect, such as highspeed motions and rare terrains. Therefore, we introduce a physics-informed data augmentation methodology called PIAug. We show an example use case of the same by modeling highspeed and aggressive motion predictions, given a dataset with only low-speed data. During the training phase, we leverage the nominal model for generating target domain (medium and high velocity) data using the available source data (low velocity). Subsequently, we employ a physics-inspired loss function with this augmented dataset to incorporate prior knowledge of physics into the neural network. Our methodology results in up to 67% less mean error in trajectory prediction in comparison to a standalone nominal model, especially during aggressive maneuvers at speeds outside the training domain. In real-life navigation experiments, our model succeeds in 4x tighter waypoint tracking constraints than the Kinematic Bicycle Model (KBM) at out-of-domain velocities.

I. INTRODUCTION

Autonomous off-road driving has been a field of great interest [1–7] due to applications in various industries, like agriculture [8–10], search and rescue [11,12], and military operations [13–15]. It frequently involves performing challenging maneuvers, like high-speed travel over rough terrain and sharp turns at high yaw rates, where the dynamics of a robot, if not accurately modeled, can lead to significant errors and potential fatal accidents.

Traditional vehicle models enable robots to navigate and interact with their surroundings by incorporating general physics equations based on mechanics and dynamics. Hence, they do not overfit on any particular distribution but rather generalize equivalently across domains. However, their fundamental modeling assumptions can impose limitations in challenging scenarios. For example, the widely used physics-based vehicle model, the Kinematic Bicycle Model (KBM),

Accompanying Video: https://youtu.be/r0Dq61DIWZQ



(a) Moving on high pitched incline (b) Taking a turn aggressively with with $V_S=4.7 {\rm m/s}$, $V_M=3.5 {\rm m/s}$ $V_S=6.59 {\rm m/s}$, $V_M=5.42 {\rm m/s}$

Fig. 1: Trajectories predicted over 5s horizon by PIAug and KBM with respect to the ground truth (GT). Here, V_S and V_M denote the starting and average speed of ground truth. In both cases, PIAug shows robustness to out-of-domain scenarios with respect to speed since we only use low-speed ($\leq 3m/s$) data to train PIAug.

assumes negligible tire slip. This assumption gets violated during sharp turns and on low friction areas, resulting in large prediction errors. These models can only use low-dimensional modalities like odometry, wheel encoders, and IMU data. As an alternative to physics-based models, [1,16] have explored pure data-driven neural networks to bypass the assumptions and limitations of traditional models, possibly by leveraging multiple low and high-dimensional data like terrain maps. These neural network (NN) methods precisely capture the complex vehicle dynamics on in-domain samples but experience a rapid decline in prediction accuracy as gap widens between validation samples and training distribution.

To gain benefits of both neural networks and expert physics knowledge, recent works use Physics Informed Neural Network (PINN) [17] to model system dynamics of UAVs and soft robots [18–20]. By introducing "physics-inspired" loss, in surplus to standard task-specific loss, PINNs enhance the models' applicability while maintaining their accuracy within their specific domain. However, we find that this improvement in cross-domain generalization is limited by the dataset distribution used to determine the loss.

In off-road driving, it is hard to collect balanced data since driving at high speeds and aggressive turning is risky and tedious. In such cases, where access to certain distributions is not possible, augmenting the dataset can help achieve a more uniform distribution. In this work, exemplifying velocity distribution as different domains, we answer the question - "Can a vehicle model trained on low-velocity data (source

^{*} This work was supported by ARL awards #W911NF1820218 and #W911NF20S0005.

 $^{^1}$ Department of Mathematics, Indian Institute of Technology Kharagpur. parymaheshwari2002@iitkgp.ac.in

² Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA. {parvm,wenshanw,striest,msivapra,saich,basti}@andrew.cmu.edu

³ DEVCOM Army Research Laboratory, Adelphi, MD, USA. {john.g.rogers59.civ, jason.m.gregory1.civ}@army.mil

domain) excel in medium and higher-velocity (target domain) maneuvers?".

To generate a viable "target dataset," we need an alternative approach to standard synthetic generation techniques, i.e., random perturbations over training data and using the simulator to generate a balanced dataset. However, our task to transfer data from the low-velocity domain to a higher-velocity domain would require adding large perturbations to the state and action sequence, which cannot guarantee a kinodynamically feasible multi-step sequence. Moreover, using simulators becomes difficult with larger-scale robots due to the challenge of creating accurate vehicle simulation models.

As an alternative to these approaches, [21] demonstrates leveraging a physics prior for generating target domain data from the training data to counter domain shifts. However, they generate this "augmented data" before the training stage; hence, the knowledge of physics prior is never explicitly added to the NN. Taking inspiration from both PINNs and [21], we propose *PIAug - Physics Informed Augmentation* to generate a target dataset while extracting the complete potential of the physics prior and NNs. For this, we use a physics model to generate an augmented dataset during training. Unlike [21], this allows us to calculate a physics-inspired loss similar to PINN, which can then be backpropagated.

The main contributions of this work are as follows:

- To our knowledge, we designed the first-ever multimodal PINN using low and high-dimensional inputs to learn the system dynamics of our off-road robot.
- We introduce a data augmentation method, PIAug a physics model-based augmentation of the dataset during training, resulting in a physics-informed NN trained over out-of-distribution data.
- We compare the predictive performance of PIAug against several models. We also evaluate PIAug on fullscale navigation experiments and showcase its benefits over KBM in performing aggressive maneuvers even at out-of-domain velocities (Fig. 1).

II. RELATED WORK

Broadly, all the works for modeling system dynamics can be classified as - physics-based, data-driven, or a hybrid approach combining data-driven and physics models.

A. Physics-Based Vehicle Models

KBM and its variants are the prevailing models in this category, but to overcome their simplified kinematic assumptions, recent works [22–25] have explored high-fidelity models. However, a significant drawback of these models is their need for comprehensive data on the robot's mechanical geometry and actuator limitations. Obtaining these can be challenging in some cases. Jeon et al. [26] present a framework that enhances tire force estimation by using onboard navigation sensors to employ the Magic Formula [27]. However, this method requires wheel encoders, which limits its applicability to robots with these sensors. Also, highly nonlinear equations in such models result in slower run times as compared to KBM.

B. Data-Driven Neural Networks

Wang et al. [28] uses a data-driven NN for terrain traversal without perceptual information. Triest et al. [1] explored the role of additional modalities like First Person View (FPV) Image, top-down view RGB and height maps, and time series data like IMU in improving prediction accuracy. Our neural network architecture is similar to [2], with some modifications explained in Section III-B.

C. Hybrid Models

Similar to the approach in [29], there are works that utilize neural network outputs to predict inputs to a final Kino-dynamic layer and, therefore, are also bound by the assumptions of this layer. For instance, in [29], the last layer does not account for changes in the robot's altitude, roll, and pitch, along with their impact on other state variables, and hence the robot's motion is constrained to a 2D plane. Some works like [30] leverage a roll-pitch-yaw physics model along with terrain information to overcome the assumptions of a 2D final layer. A point of difference between our works is [30] only showcases their method over simulated data and cannot consider other modalities like RGB maps and FPV images or terrain maps.

An alternative approach to add physics knowledge to NNs is to use residual methods. Here, NN is employed to capture and model the uncertainties that exist between the predictions of the physics model and the ground truth. For instance, [31] demonstrates the application of this approach in the context of Unmanned Aerial Vehicles (UAVs).

More recent works [18] have shown better performance of PINNs over residual methods. Here, physics models are used to add explainability to NNs by adding a physics-inspired loss during training in addition to the standard data-driven loss. Another key distinction from residual methods is that PINNs predict absolute future states during inference instead of serving as a residual term to the physics model. While [32] uses a single high-dimensional modality as input to reconstruct a fluid's flow pattern, we can use of multiple high-dimensional modalities as inputs to a PINN.

III. METHODOLOGY

A. Extended Kinematic Bicycle Model

We employ an extended KBM (Eq. 1), which expands the typical state representation $[x_i, y_i, \psi_i]$ with additional variables $[v_i, \delta_i]$. x_i and y_i denote global positional coordinates, ψ_i and v_i represent the yaw and speed of the vehicle in the global frame, and δ denotes the steering angle of the wheels at timestep i. These supplementary state variables enable us to incorporate $[T_i, \delta_i^{target}]$ as actions (U_i) , where T_i represents throttle and δ_i^{target} denotes the target steering angle. Since throttle is a lower-level control, it is chosen as the desired action instead of a target velocity as in a standard KBM. While calculating the rate of change of speed (v_i) , we consider the effect of throttle, a combined velocity proportional effect of engine braking and drag force, along with friction, gravity (g) and pitch (θ_i) . These effects are scaled appropriately using constants (K_t, K_b, K_f, K_g) . Our

KBM also models steering wheel actuation, using K_s , and eliminates the assumption that the target steering is achieved instantaneously as in a standard KBM, which directly uses δ_i^{target} instead of δ_i to predict yaw rate.

$$f(X_{i}, U_{i}) = \dot{X}_{i} = \begin{bmatrix} v_{i} * cos\psi_{i} \\ v_{i} * sin\psi_{i} \\ (v_{i} \tan \delta_{i})/L \\ \dot{v}_{i} \\ \dot{\delta}_{i} \end{bmatrix}$$

$$\dot{v}_{i} = K_{t} * T_{i} - K_{b} * v_{i} - K_{f} * sign(v_{i}) * \cos(\theta_{i}) \\ - K_{g} * g * \sin(\theta_{i})$$
(1a)

$$\dot{\delta_i} = K_s * (\delta_i^{target} - \delta_i) \tag{1c}$$

B. Data driven Vehicle Model

Building upon the ideas presented in [1], we employ a NN with multimodal inputs to estimate vehicle dynamics. One of the notable differences in our model is the incorporation of all components of linear v and angular w velocities, in addition to the current steering angle (δ) , into the state space. This choice was motivated by KBM, as it uses velocity and current steering angle to predict the future position and orientation of the robot. Further, we have chosen a recently introduced 6-state representation of rotation, denoted as \mathbf{r} , as proposed in [33]. This is a continuous representation, unlike quaternions (q), which possesses discontinuity. This choice is made to take advantage of the neural network's ability to perform better when dealing with continuous data over discontinuous data. Consequently, the new state vector for our neural networks consists of sixteen elements: $[\mathbf{p}, \mathbf{r}, \mathbf{v}, \mathbf{w}, \delta]$ compared to [1] which utilizes a seven-state vector: [**p**, **q**]. This enhanced state allowed us to achieve optimal predictive accuracy even without time series data such as wheel encoders and IMU. Additionally, we enhance our model by employing a 4-channel height map that encompasses [min, max, mean, std] values, in contrast to [1], which utilizes only a 2-channel height map consisting of [min, max].

C. Physics-Informed Vehicle Model

We formulate the Physics-Informed vehicle model as a combination of physics-inspired and data-driven loss functions (Eq 2). In our context, B_D represents a real-world dataset while B_P can either represent a real-world dataset for a PINN model or an augmented dataset for a PIAug model (Section III-D). We use B_P and B_D , for determining physics-informed (L_P) and data-driven (L_D) losses, respectively. Here, $\phi(X_k, O_k, U_k)$ represents the data-driven NN, and the physics model is signified by $F(X_k, U_k)$ and f (Eq. 1). k and T denote the current timestep and prediction horizon, respectively. λ_{PI} is the weight of L_P , in its linear combination(L) with L_D . For better accuracy, we use the Midpoint method [34] to evaluate the integral in Eq. 2b.

A data sequence is defined as $\{X_k, O_k, \boldsymbol{U_k}, \boldsymbol{Y_k}\}$ where X_k and O_k are the current state and observations defined as all the other multiple modalities required by the NN. In our case, this comprises {FPV Image, RGB map, Height map} similar to [1]. $U_k = [U_k, U_{k+1}..., U_{k+T-1}]$ are the T

actions, while $Y_k = [X_{k+1}, X_{k+2}..., X_{k+T}]$ denotes ground truth for the future trajectory. The predicted states of the data sequence using a model, m, are represented as $\hat{X}_{k}^{m} =$ $[\hat{X}_{k+1}^m,\hat{X}_{k+2}^m,...,\hat{X}_{k+T}^m]$, which can be calculated as shown in Eq. 2. An important note here is that since the state space of the NN and the physics model are different, we convert the prediction of the NN to KBM's state space in Eq. 2c. This is represented by $(\hat{X}_i^{NN})_{KBM}$.

$$\hat{X}_{k}^{NN} = \phi(X_{k}, O_{k}, U_{k}) \tag{2a}$$

$$\hat{\boldsymbol{X}}_{k}^{NN} = \phi(X_{k}, O_{k}, \boldsymbol{U}_{k})$$
 (2a)
$$\hat{\boldsymbol{X}}_{k}^{KBM} = F(X_{k}, \boldsymbol{U}_{k})$$
 (2b)

$$\begin{split} where \ \hat{X}_{k+l+1}^{KBM} \ = \ \hat{X}_{k+l}^{KBM} \ + \int_{t=0}^{t=1} f(X_{k+l}, U_{k+l}) \ dt, \\ \hat{X}_{k}^{KBM} \ = \ X_{k}, \ \text{and} \ l \in [0, 1..., T-1] \end{split}$$

$$L_P = \frac{1}{|B_P|} \sum_{i=1}^{|B_P|} ||(\hat{X}_i^{NN})_{KBM} - \hat{X}_i^{KBM}||^2 \quad (2c)$$

$$L_D = \frac{1}{|B_D|} \sum_{i=1}^{|B_D|} ||(\hat{X}_j^{NN} - Y_j)||^2$$
 (2d)

$$L = L_D + \lambda_{PI} L_P \tag{2e}$$

D. Physics-Informed Data Augmentation

To improve NN's generalization, the physics model can be used to generate multi-step ground truth predictions for outof-distribution scenarios. This allows the model to learn from the physics model, akin to PINN, while also augmenting the training dataset, allowing it to be robust to domain shift. We illustrate domain shifts in predictive and navigational performance across different mean speed groups.

For this, we will replace a real-world dataset with an augmented dataset to use as B_P . Fig. 2 provides an overview of our methodology. The physics model requires an initial state, initial observations, and a set of actions to achieve this. Taking inspiration from our physics model, it is known that speed and yaw rate can be approximated as linearly proportional. Therefore, to generate the desired initial state in the augmentation module (Yellow box in Fig. 2), we apply a transformation over the initial state of the minibatch (X_{RAW}) sampled from the training dataset (Here we drop the timestep subscript, k for a cleaner representation). This transformation is a random scaling sampled from a uniform distribution: U(2.5,4). The scaling is kept standard for all components of linear and angular velocity. The action sequences (U_{AUG}) used in evaluating this physics loss, however, are the same as those of the training batch (U_{RAW}) . A note here is that U_{AUG} can easily be generated using the Gaussian or Ornstein-Uhlenbeck process [35] if it is desired that the actions follow a preferred distribution.

The augmented initial state (X_{AUG}) , combined with the original observation (O_{RAW}) and actions (U_{AUG}) can then be used to predict the future states (Y_{AUG}) similar to \hat{X}_{i}^{KBM} in Eq. 2b. Fig. 3 shows the distribution of initial and mean velocity for $raw(B_D)$ and augmented (B_P) datasets where we use Y_{RAW} and Y_{AUG} to calculate mean velocities of the respective data.

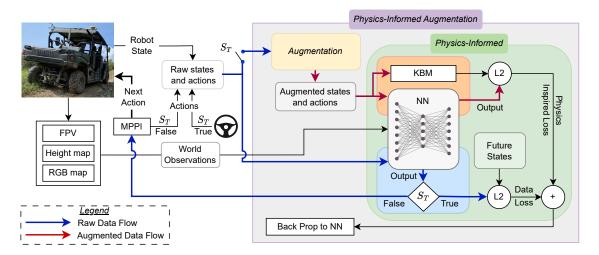


Fig. 2: An overview of our methodology. During training $(S_T = True)$, the raw states and actions (from a dataset) are fed to an augmentation methodology to generate an augmented state and sequence of actions. To generate the ground truth label of this augmented data, KBM (a physics model) is used. For both raw and augmented data sequences, a shared NN, along with common observations, are used to predict future states. We linearly combine the data-driven loss with the physics-informed loss. For clarity of the figure, we have not connected the backpropagation step back to the NN. At test time $(S_T = False)$, there is no augmentation, and MPPI generates the raw actions. The predicted ground truth for the raw data is then sent back to MPPI to evaluate their costs.

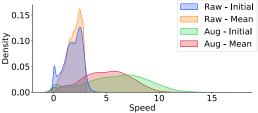


Fig. 3: The density (ρ) distribution of initial and mean speeds. "Raw" denotes the training dataset, which exclusively contains low-velocity data. "Aug" represents the augmented dataset created using KBM with a velocity-scaled initial state. It possesses a balanced distribution of initial and mean speeds in the trajectories over the imbalanced Raw dataset. Since there exists some data with a negative initial velocity $(\rho(Raw_{Initial} < 0) \neq 0)$, augmentation can also result in a negative mean speed $(\rho(Aug_{Mean} < 0) \neq 0)$.

IV. HARDWARE IMPLEMENTATION

We evaluated the performance of our models on a Yamaha Viking ATV modified for autonomous driving by Mai et al. [36]. We use a Velodyne UltraPuck LiDAR sensor for collecting point clouds, a Multisense S21 camera system for FPV Stereo and RGB images, and a NovAtel PROPAK-V3-RT2i GNSS unit for global positioning. Our navigation stack ran on a 12th-gen Intel i7 CPU and an NVIDIA 3080 laptop GPU. The navigation stack can be broken down into

- 1) Costmapping: We use Super Odometry [37] to align unregistered lidar scans in the vehicle's initial frame. These scans are then converted into gridmaps and further processed into costmaps based on a predefined lethal height threshold.
- 2) RGB and Height Maps: Similar to [1], we use TartanVO [38] to create top-down RGB maps and height maps,

Attribute	Symbol	Low	Med	High
v : Velocity	V	[0,3]	(3,5]	(5,7]
$ \theta $: Pitch	Θ	[0,0.05]	(0.05,0.12]	(0.12,1.57]
$ \dot{\psi} $: Yaw Rate	Ψ	[0,0.05]	(0.05,0.12]	(0.12,1.57]

TABLE I: Thresholds for categorizing various attributes of a sequence into low, medium, or high groups.

using RGB and Stereo FPV images. These maps serve as inputs for our neural network-based vehicle models.

3) Control: Much like the approach described in [39], we adapt MPPI (Model Predictive Path Integral) from Williams et al. [40] for navigation. Our modified MPPI incorporates multiple waypoints to improve cost-to-goal computation and makes effective use of our long prediction horizon.

V. EXPERIMENTS, RESULTS, AND ANALYSIS

A. Preliminaries

- 1) Dataset: We used a subset of TartanDrive [1] comprising only of the low-velocity trajectories as our training dataset. An additional dataset was gathered with a balanced velocity distribution to assess the capability of the model for both in and out-of-distribution scenarios.
- 2) Errors: All the reported prediction errors are the averages of errors within the multi-step horizon. For a fair comparison between KBM and the NN-based vehicle models, these errors encompass various aspects, including position (Δp) , yaw $(\Delta \psi)$, and velocity (Δv) . Δp quantifies the L_2 norm of prediction errors of both the Δx and Δy . We do not report steer angle error as an error of magnitude of the order of 10^{-2} rad is constantly exhibited.

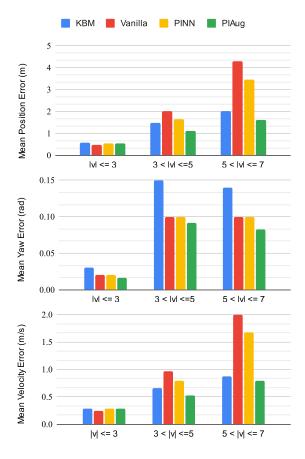


Fig. 4: Comparison of models trained on <3m/s trajectories. PIAug performs better than all other models in out-of-distribution trajectories.

B. Prediction Analysis over Different Velocity Domains

As mentioned in Section III-D, we will divide data sequences into domains majorly on the basis of the mean speed of their ground truth (|v|). Table I lists the different domains based on velocity. We evaluate the performance of various models in Fig. 4, in which the vanilla model denotes a data-driven variant of PINN where we use $\lambda_{PI} = 0$ during training (Eq. 2e). We can infer that even though PINN is better than its vanilla variant, the performance improvement is not as substantial as compared to the case of PIAug. This vast improvement between PINN and PIAug can be directly attributed to our method of scaling the initial state to other domains and hence also emphasizes the necessity to evaluate the physics-inspired loss on a well-distributed dataset. Apart from this, Fig. 4 also shows how the performance of Vanilla NN degrades quickly as the domain shift increases. We omit Vanilla NN and PINN from subsequent analyses due to these large errors.

C. Comparisons of vehicle models on different maneuvers

To highlight the complexity of different scenarios, apart from distributing them only on the basis of velocity as done in Section V-B, we categorize each sequence into more subgroups. This is done on the basis of combinations of the average of magnitudes of velocity (|v|), pitch $(|\theta|)$, and yaw rate

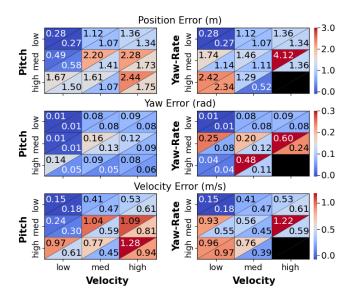


Fig. 5: Prediction errors in position (top), yaw (middle), and velocity (bottom) for KBM (upper half) and PIAug (lower half). Left: Navigating straight (Ψ_{low}) at varying speeds over slopes with different inclines. Right: Turning at different speeds and yaw rates on flat terrain (Θ_{low}). The black mask over ($V_{high}, \Psi_{high}, \Theta_{low}$) on the right represents no data collection to preserve driver safety. The most significant improvements (up to 67% achieved in case of position error of (V_{high}, Ψ_{med}) on the right) are near the bottom right half on each metric of both columns, which corresponds to the most difficult scenarios like driving fast on rugged terrains (left) and aggressive turning (right).

 $(|\dot{\psi}|)$. The threshold values of these groups are defined in Table I. This results in 27 subgroups denoted as $\{(V_p,\Theta_q,\Psi_r):p,q,r\in\{low,med,high\}\}$. Fig. 5 compares the prediction error of KBM and PIAug over two subsets of these groups $((V_p,\Theta_q,\Psi_{low}))$ and $(V_p,\Theta_{low},\Psi_r)$. The first rows of the left/right heatmaps in Fig. 5 are the same since these rows represent the scenarios $\in (V_{low},\Theta_{low},\dot{\Psi}_{low})$.

We can see that KBM exhibits significant prediction errors in scenarios containing high difficulty of any attribute due to its reliance on kinematic assumptions. In contrast, PIAug demonstrates considerably higher prediction accuracy (up to 67%) even on subgroups containing V_{high} . This is intriguing as PIAug was trained exclusively on data from V_{low} . Even beyond these scenarios, PIAug consistently outperforms KBM in most groups, with comparable performance in the remaining cases.

D. Real-Time Frequency Comparison

NN-based vehicle models like PIAug enhance predictions and, if optimized, reduce inference time compared to KBM. This accelerates the navigation stack, improving performance and enabling more samples in MPPI for better control solutions. Table II compares inference times for KBM and a NN-based vehicle model during MPPI trajectory rollouts. NN-

	Number of MPPI Samples					
Model Type	1024		2048		4096	
	Time ¹	Mem^2	Time	Mem	Time	Mem
KBM_CPU	0.058	0.00	0.066	0.00	0.132	0.00
KBM_GPU	0.074	0.598	0.072	0.622	0.069	0.622
NN	0.166	3.806	N/A	N/A	N/A	N/A
NN_OPT	0.027	2.064	0.035	2.510	0.052	4.996

TABLE II: NN-based models are more precise than KBM, and an optimized implementation provides 2x faster inference time. N/A represents insufficient (>6GB) GPU memory.

based model rollouts, implemented with GPU-based RNNs [41], can offer a speed advantage over solving integrals in the case of KBMs if they are well-optimized. In MPPI, since all samples share an initial state, it allows encoding a single sample during inference and using the result with all sampled actions. This approach (denoted as NN_OPT) significantly reduces the required GPU memory and inference time. To optimize control solutions with accurate PIAug, referencing Table II, we select 2048 MPPI rollout samples. This choice mitigates GPU memory challenges tied to PIAug's accuracy, which could occur with 4096 samples.

E. Hardware Experiments

We now demonstrate the impact of an accurate vehicle model on real-world navigation by navigating a challenging figure-8 trajectory (Fig. 6) at a max speed of 5m/s, allowing us to test PIAug's robustness to domain shift. A human first drives the desired maneuver to collect waypoints, as shown in Fig. 6. Table III reports the results of our experiments. Note that while calculating the average velocity, we do not include the acceleration phase (Start to Goal 1) and deceleration phase (Goal 11 to Finish). Due to a long prediction horizon, the vehicle starts decelerating after it reaches Goal 11, as it can see the Finish point within its reach.

We can observe that when using a 4m goal radius, while PIAug and KBM both have a 100% completion rate, the average velocity for PIAug is significantly higher than KBM. This happens since a high error in trajectory prediction, as is the case of KBM, leads the robot to correct its path often due to a significant difference between the actual state as compared to what was predicted in the past. This frequent correction can lead to disruptions when accelerating or maintaining high speed. Furthermore, the importance of a more accurate vehicle model is highlighted in the case of a 1m goal radius, where the experiment can only be completed in the presence of precise dynamics modeling. The average speed of PIAug increases when following a larger radius as the vehicle cuts a lot of corners and is not expected to follow the waypoints very precisely. This allows the vehicle to navigate the trajectory at a higher speed.

VI. CONCLUSION AND FUTURE WORK

We introduce PIAug, a novel data augmentation method that uses a physics model to transfer data from the source to

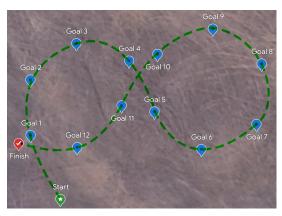


Fig. 6: Figure-8 trajectory with waypoints spaced at 10m is used for real-world navigational experiments. Taking such sharp turns at a maximum speed of 5m/s tests the limits of all vehicle models

Goal Radius (m)	Vehicle Model	No. of Successes (out of 3)	Mean Speed (m/s)
4 4	KBM PIAug	3 3	3.91 4.36
1	KBM PIAug	0 3	N/A 3.47

TABLE III: Effect of using KBM and PIAug on the overall Navigation Performance. N/A represents that the average speed cannot be calculated since no run was successful.

the target domain while integrating physics priors into NN training. We demonstrate its potential by developing a robust model for velocity distribution domain shift. A comprehensive comparison of our model is done with other physics, data-driven, and hybrid models. Our approach outperforms the nominal model by reducing mean prediction error by upto 67%. In real-life navigation experiments requiring precise trajectory tracking while traveling at out-of-domain velocities, our model is able to operate even under 4x tighter waypoint tracking constraints, surpassing KBM's limitations. These aggressive maneuvers highlight our methodology's robustness against domain-shifting. However, in contrast to low dimensional vehicle states, it is a challenging task to use physics priors to transfer environment modalities (e.g., FPV, terrain maps) to a sufficiently different target domain (e.g., transferring FPV data captured in a forest domain to an open area domain)

The future scope of this research is three-fold. First, physics-informed augmentation can reduce training time and data requirements for transferring a vehicle model across different robots using the target robot's nominal model. Second, incorporating complex vehicle models [27] can narrow the gap between augmented and real-world datasets. This is viable as the slower run times of these models only affect the training but not the inference time for PIAug-trained models. Third, we can enhance safety by sampling augmentations from a task and constraint-informed prior distribution rather than random scaling.

¹Inference time (in seconds) during the MPPI rollout stage.

²Amount of GPU memory required (in GB)

REFERENCES

- S. Triest, M. Sivaprakasam, S. J. Wang et al., "Tartandrive: A large-scale dataset for learning off-road dynamics models," arXiv preprint arXiv:2205.01791, 2022.
- [2] X. Meng, N. Hatch, A. Lambert *et al.*, "Terrainnet: Visual modeling of complex terrain for high-speed, off-road navigation," 2023.
- [3] J. Tremblay, T. Manderson, A. Noca et al., "Multimodal dynamics modeling for off-road autonomous vehicles," 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 1796–1802, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID: 227151313
- [4] S. Thrun, "Stanley: The robot that won the darpa grand challenge," *Journal of Field Robotics*, vol. 23, 2006. [Online]. Available: https://api.semanticscholar.org/CorpusID:1438204
- [5] K. Chu, M. Lee, and M. Sunwoo, "Local path planning for offroad autonomous driving with avoidance of static obstacles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1599–1616, 2012.
- [6] T. Chang, T. Hong, M. N. Abrams, and M. Shneier, "An intelligent world model for autonomous off-road driving," *Computer Vision and Image Understanding*, 2001. [Online]. Available: https://api.semanticscholar.org/CorpusID:118762171
- [7] M. G. Castro, S. Triest, W. Wang et al., "How does it feel? self-supervised costmap learning for off-road vehicle traversability," in 2023 IEEE International Conference on Robotics and Automation (ICRA), 2023, pp. 931–938.
- [8] A. Ghobadpour, G. Monsalve, A. Cardenas, and H. Mousazadeh, "Off-road electric vehicles and autonomous robots in agricultural sector: Trends, challenges, and opportunities," *Vehicles*, vol. 4, no. 3, pp. 843–864, 2022. [Online]. Available: https://www.mdpi. com/2624-8921/4/3/47
- [9] H. Mousazadeh, "A technical review on navigation systems of agricultural autonomous off-road vehicles," *Journal of Terramechanics*, vol. 50, no. 3, pp. 211–232, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022489813000220
- [10] T. Duckett, S. Pearson, S. Blackmore, and B. D. Grieve, "Agricultural robotics: The future of robotic agriculture," *ArXiv*, vol. abs/1806.06762, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:49300462
- [11] A. Gadekar, S. Fulsundar, P. Deshmukh *et al.*, "Rakshak: A modular unmanned ground vehicle for surveillance and logistics operations," *Cognitive Robotics*, vol. 3, pp. 23–33, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2667241323000083
- [12] K. Berns, A. Nezhadfard, M. Tosa et al., Unmanned Ground Robots for Rescue Tasks, 08 2017.
- [13] J. E. Naranjo, M. Clavijo, F. Jiménez et al., "Autonomous vehicle for surveillance missions in off-road environment," in 2016 IEEE Intelligent Vehicles Symposium (IV), 2016, pp. 98–103.
- [14] A. Bezzina, L. Xuereb, S. G. Fabri, and C. J. Debono, "Development of an autonomous off-road vehicle for military applications," *Robotics* and Autonomous Systems, vol. 94, pp. 52–61, 2017.
- [15] E. Krotkov and J. Blitch, "The defense advanced research projects agency (darpa) tactical mobile robotics program," *The International Journal of Robotics Research*, vol. 18, no. 7, pp. 769–776, 1999.
- [16] J. Tremblay, T. Manderson, A. Noca et al., "Multimodal dynamics modeling for off-road autonomous vehicles," 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 1796–1802, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID: 227151313
- [17] M. Raissi, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/ S0021999118307125
- [18] A. Saviolo, G. Li, and G. Loianno, "Physics-inspired temporal learning of quadrotor dynamics for accurate model predictive trajectory tracking," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10256–10263, oct 2022. [Online]. Available: https://doi.org/10.1109%2Flra.2022.3192609
- [19] S. Sanyal and K. Roy, "Ramp-net: A robust adaptive mpc for quadrotors via physics-informed neural network," 2023 IEEE International Conference on Robotics and Automation (ICRA), pp. 1019–1025, 2022. [Online]. Available: https://api.semanticscholar.org/ CorpusID:252367666

- [20] W. Sun, N. Akashi, Y. Kuniyoshi, and K. Nakajima, "Physics-informed recurrent neural networks for soft pneumatic actuators," *IEEE Robotics* and Automation Letters, vol. 7, no. 3, pp. 6862–6869, 2022.
- [21] W. Luo, Z. Yan, Q. Song, and R. Tan, "Physics-directed data augmentation for deep model transfer to specific sensor," ACM Trans. Sen. Netw., vol. 19, no. 1, dec 2022. [Online]. Available: https://doi.org/10.1145/3549076
- [22] N. Seegmiller and A. Kelly, "High-fidelity yet fast dynamic models of wheeled mobile robots," *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 614–625, 2016.
- [23] Z. Han, S. Yuan, X. Li, and J. Zhou, "Enhanced closed-loop systematic kinematics analysis of wheeled mobile robots," *International Journal* of Advanced Robotic Systems, vol. 16, no. 4, p. 1729881419863242, 2019. [Online]. Available: https://doi.org/10.1177/1729881419863242
- [24] S. Kakkar and M. A. Minor, "Fast and reliable motion model for articulated wheeled mobile robots on extremely rough and rocky terrains," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2252–2259, 2019.
- [25] F. Rubio, C. Llopis-Albert, F. Valero, and A. J. Besa, "A new approach to the kinematic modeling of a three-dimensional car-like robot with differential drive using computational mechanics," *Advances in Mechanical Engineering*, vol. 11, no. 3, p. 1687814019825907, 2019. [Online]. Available: https://doi.org/10.1177/1687814019825907
- [26] S.-Y. Jeon, R. Chung, and D. Lee, "Tire force estimation of dynamic wheeled mobile robots using tire-model based constrained kalman filtering," in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 2470–2477.
- [27] H. B. Pacejka and E. Bakker, "The magic formula tyre model," Vehicle System Dynamics, vol. 21, pp. 1–18, 1991. [Online]. Available: https://api.semanticscholar.org/CorpusID:108456393
- [28] S. J. Wang, S. Triest, W. Wang et al., "Rough terrain navigation using divergence constrained model-based reinforcement learning," in *Proceedings of the 5th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, A. Faust, D. Hsu, and G. Neumann, Eds., vol. 164. PMLR, 08–11 Nov 2022, pp. 224–233. [Online]. Available: https://proceedings.mlr.press/v164/wang22c.html
- [29] T. Kim, H. Lee, and W. Lee, "Physics embedded neural network vehicle model and applications in risk-aware autonomous driving using latent features," in 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2022, pp. 4182–4189.
- [30] H.-W. Lee, T. Kim, J. Mun, and W. Lee, "Learning terrain-aware kinodynamic model for autonomous off-road rally driving with model predictive path integral control," ArXiv, vol. abs/2305.00676, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID: 258426349
- [31] K. Y. Chee, T. Z. Jiahao, and M. A. Hsieh, "Knode-mpc: A knowledge-based data-driven predictive control framework for aerial robots," *IEEE Robotics and Automation Letters*, vol. 7, pp. 2819–2826, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID: 237485443
- [32] G. Raynaud, S. Houde, and F. P. Gosselin, "Modalpinn: An extension of physics-informed neural networks with enforced truncated fourier decomposition for periodic flow reconstruction using a limited number of imperfect sensors," *J. Comput. Phys.*, vol. 464, p. 111271, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID: 237259912
- [33] Y. Zhou, C. Barnes, J. Lu et al., "On the continuity of rotation representations in neural networks," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5738–5746, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID: 56178817
- [34] E. Süli and D. F. Mayers, "An introduction to numerical analysis," 2003. [Online]. Available: https://api.semanticscholar.org/CorpusID: 118602067
- [35] G. E. Uhlenbeck and L. S. Ornstein, "On the theory of the brownian motion," *Physical review*, vol. 36, no. 5, p. 823, 1930.
- [36] J. Mai, "System design, modelling, and control for an off-road autonomous ground vehicle," Master's thesis, Carnegie Mellon University, Pittsburgh, PA, July 2020.
- [37] S. Zhao, H. Zhang, P. Wang et al., "Super odometry: Imu-centric lidar-visual-inertial estimator for challenging environments," in 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2021, pp. 8729–8736.
- [38] W. Wang, Y. Hu, and S. Scherer, "Tartanvo: A generalizable learning-based vo," Conference on Robot Learning, 2020.

- [39] S. Triest, M. G. Castro, P. Maheshwari et al., "Learning risk-aware costmaps via inverse reinforcement learning for off-road navigation," in 2023 IEEE International Conference on Robotics and Automation (ICRA), 2023, pp. 924–930.
- (ICRA), 2023, pp. 924–930.
 [40] G. Williams, N. Wagener, B. Goldfain et al., "Information theoretic mpc for model-based reinforcement learning," in 2017 IEEE International Conference on Robotics and Automation. IEEE, 2017, pp. 1714–1721.
- [41] A. Paszke, S. Gross, F. Massa et al., "Pytorch: An imperative style, high-performance deep learning library," in Advances in Neural Information Processing Systems 32. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library. pdf