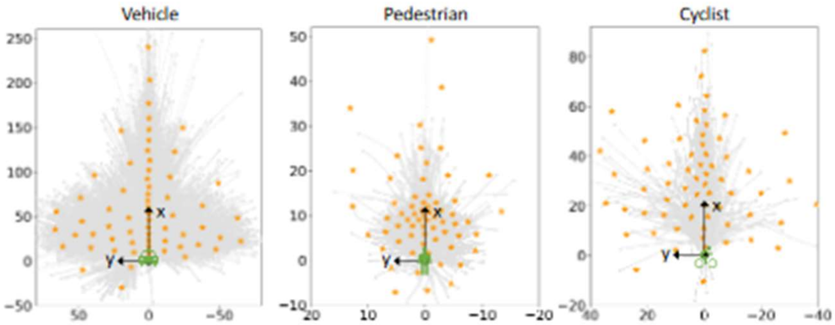
MTR++: Multi-Agent Motion Prediction with Symmetric Scene Modeling and Guided Intention Querying Paper Review

Paper Review by Tyler Kim

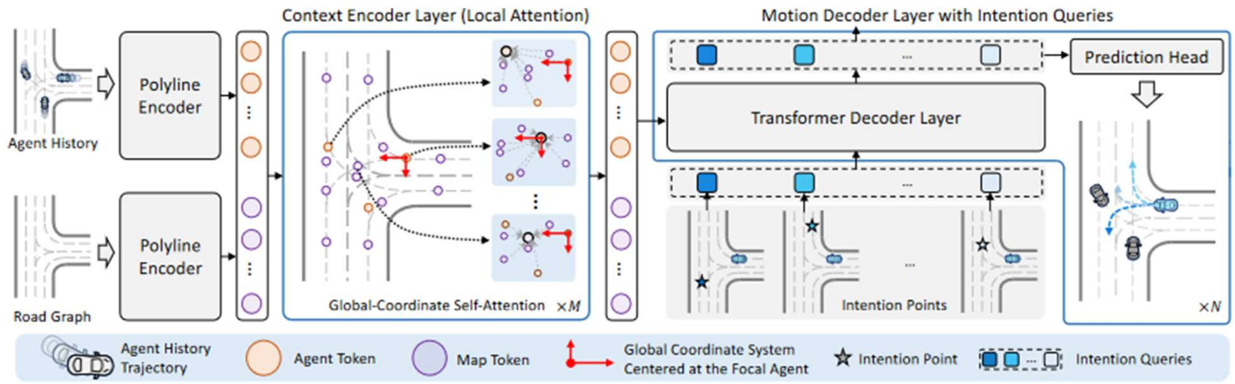
Summary

The paper *MTR++: Multi-Agent Motion Prediction with Symmetric Scene Modeling and Guided Intention Querying* by Shaoshuai Shi et al introduces a novel approach for multi-agent prediction. One problem involves inferencing on future actions of traffic participants which prove difficult due to multimodal behaviors of agents and the complexity of the environment. Current approaches to address the problems exhibit bias and the performance heavily depends on density of goal candidates. Therefore, the paper proposes Motion Transformer frameworks (MTR) which optimize identifying an agent's intentions and refine predicted trajectories. **The paper's main contributions are introducing the MTR frameworks which achieves state-of-the-art performance on the Waymo Open Motion Dataset and proposing MTR++ framework which won the Waymo Prediction Challenge.**

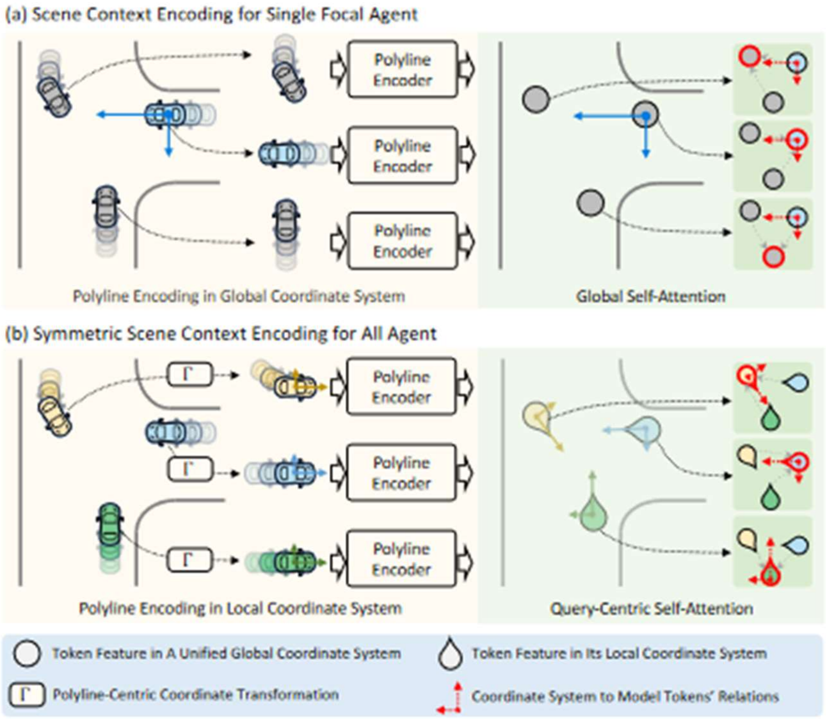
The MTR model uses *focal-agent-centric* approach where it takes in a normalization of all inputs to the global coordinate system centered on the agent and passes it through two polyline encoders: one for agent history and the second for the road map creating the polyline features. The polyline features are encoded using a multi-head self-attention layer which is fed into a motion decoder layer with intention queries to serve as context. Intention queries are simply trajectory predictions to infer the "goal" of an agent using clustering as displayed below.

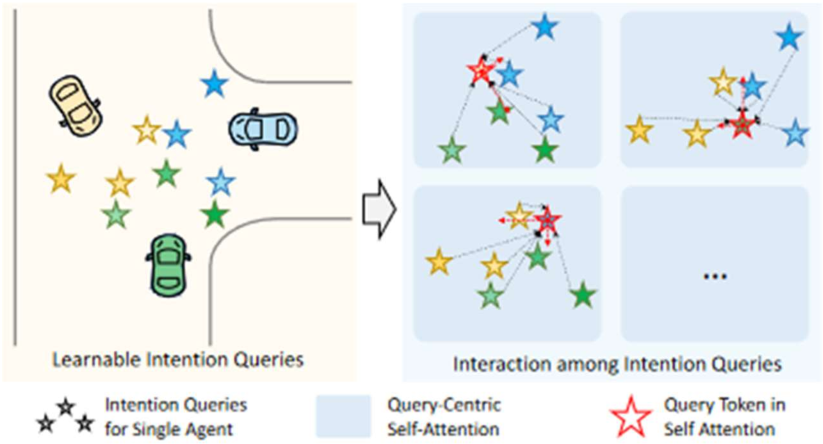


MTR decodes the encoded context with a learnable intention query integrated transformer decoder layer stack. Each decoder layer uses a self-attention module and multi-head cross-attention layer to generate the final query. The final query is passed through a multi-layer perceptron and then a Gaussian Mixture Model to represent the predicted trajectories at each timestep. The figure below describes the architecture of MTR.


Context Encoder Layer (Local Attention)
Motion Decoder Layer with Intention Queries
Agent History
Polyline Encoder
Road Graph
Polyline Encoder
Global-Coordinate Self-Attention ×M
Prediction Head
Transformer Decoder Layer
Intention Points
×N
Agent History Trajectory
Agent Token
Map Token
Global Coordinate System Centered at the Focal Agent
Intention Point
Intention Queries

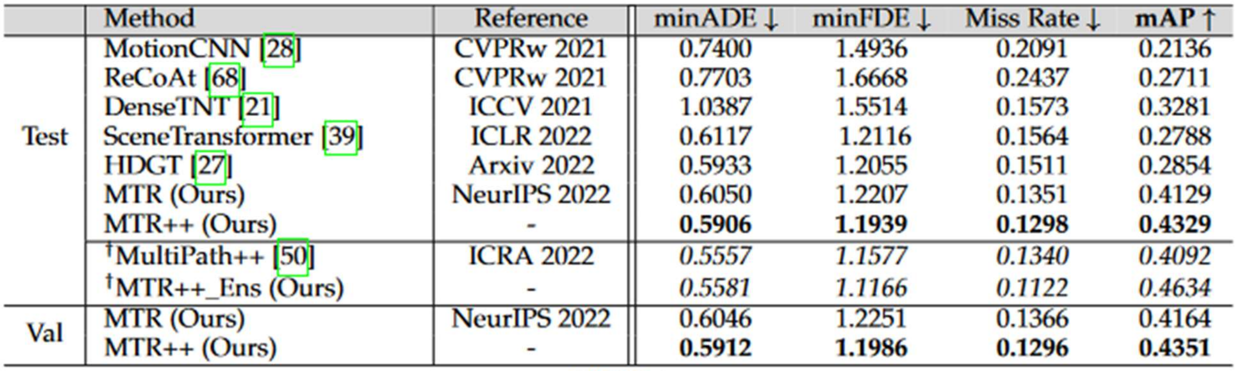
The paper proposes MTR++ which is essentially MTR except it models each scene symmetrically for each agent using *query-centric self-attention* module which models the relationship between all tokens in a symmetric manner. More specifically, it models the token into the local coordinate system. In addition, MTR++ uses a *mutually-guided intention querying* module which allows agents to interact and influence each other's behaviors. The figures below describe the symmetric scene context encoding and mutually-guided intention querying respectively. The loss function is the sum of the negative log-likelihood loss of Gaussian Mixture Model and L1 regression loss of the output of dense future prediction of future states.


(a) Scene Context Encoding for Single Focal Agent
Polyline Encoder
Polyline Encoder
Polyline Encoder
Polyline Encoding in Global Coordinate System
Global Self-Attention
(b) Symmetric Scene Context Encoding for All Agent
Polyline Encoder
Polyline Encoder
Polyline Encoder
Polyline Encoding in Local Coordinate System
Query-Centric Self-Attention
Token Feature in A Unified Global Coordinate System
Token Feature in Its Local Coordinate System
Polyline-Centric Coordinate Transformation
Coordinate System to Model Tokens' Relations


Learnable Intention Queries
Interaction among Intention Queries
...
Intention Queries for Single Agent
Query-Centric Self-Attention
Query Token in Self Attention

The experiment used the Waymo Open Motion Dataset for two tasks: independently predict motion of each agent and the joint future positions of two interacting agents.

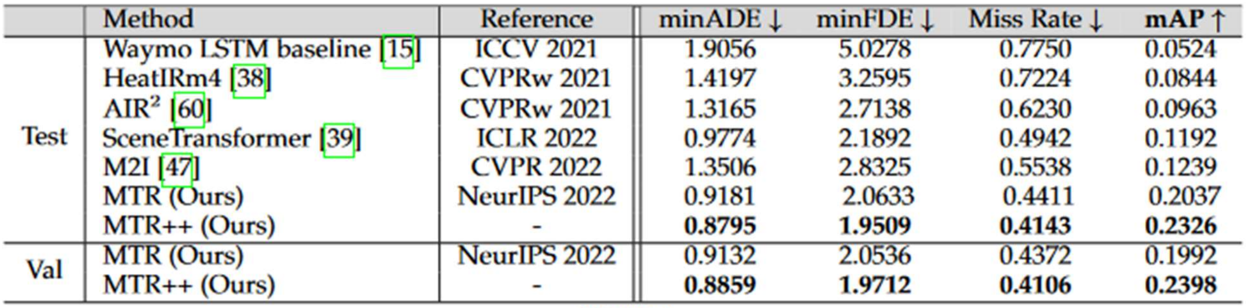
For predicting the motion of each individual agent, MTR achieves a mAP increase of +8.48% and reduces miss rate from 15.11% to 13.51%. MTR++ enhances MTR on all metrics increasing mAP by +2.00% compared to MTR. The figure below describes the results of the first task.



	Method	Reference	minADE ↓	minFDE ↓	Miss Rate ↓	mAP ↑
Test	MotionCNN [28]	CVPRw 2021	0.7400	1.4936	0.2091	0.2136
	ReCoAt [68]	CVPRw 2021	0.7703	1.6668	0.2437	0.2711
	DenseTNT [21]	ICCV 2021	1.0387	1.5514	0.1573	0.3281
	SceneTransformer [39]	ICLR 2022	0.6117	1.2116	0.1564	0.2788
	HDGT [27]	Arxiv 2022	0.5933	1.2055	0.1511	0.2854
	MTR (Ours)	NeurIPS 2022	0.6050	1.2207	0.1351	0.4129
	MTR++ (Ours)	-	**0.5906**	**1.1939**	**0.1298**	**0.4329**
	†MultiPath++ [50]	ICRA 2022	*0.5557*	*1.1577*	*0.1340*	*0.4092*
	†MTR++_Ens (Ours)	-	*0.5581*	*1.1166*	*0.1122*	*0.4634*
Val	MTR (Ours)	NeurIPS 2022	0.6046	1.2251	0.1366	0.4164
	MTR++ (Ours)	-	**0.5912**	**1.1986**	**0.1296**	**0.4351**

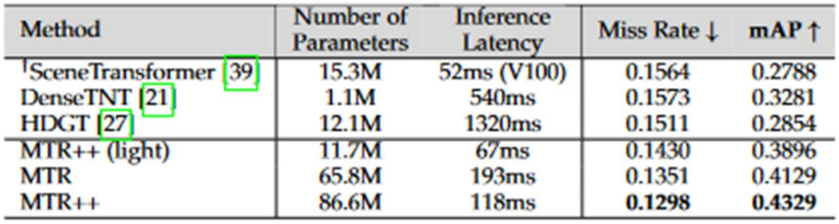
TABLE 1

For the second task, MTR reduces miss rates from 49.42% to 44.11% and enhances mAP from 12.39% to 20.37%. MTR++ increases MTR's performance by increasing mAP by +2.89% and reducing miss rate by 2.68%. Additionally, a lightweight MTR++ enhances mAP performance from 32.81% to 38.96% and uses fewer parameters. MTR framework performed significantly better on the Argoverse 2 dataset. The figures below display the results.



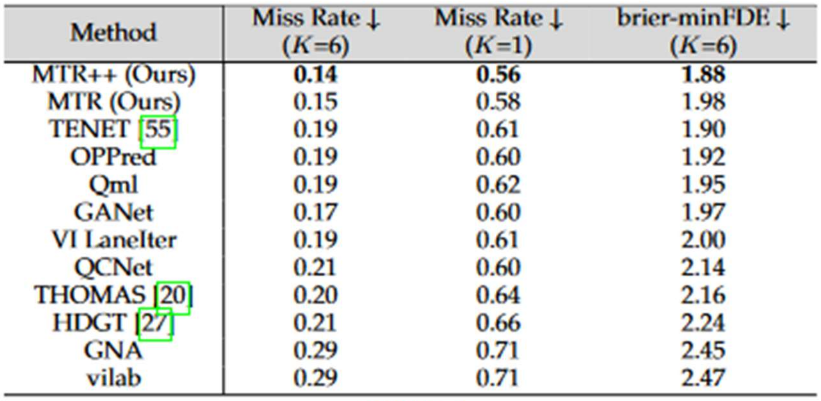
	Method	Reference	minADE ↓	minFDE ↓	Miss Rate ↓	mAP ↑
Test	Waymo LSTM baseline [15]	ICCV 2021	1.9056	5.0278	0.7750	0.0524
	HeatIRm4 [38]	CVPRw 2021	1.4197	3.2595	0.7224	0.0844
	AIR^2 [60]	CVPRw 2021	1.3165	2.7138	0.6230	0.0963
	SceneTransformer [39]	ICLR 2022	0.9774	2.1892	0.4942	0.1192
	M2I [47]	CVPR 2022	1.3506	2.8325	0.5538	0.1239
	MTR (Ours)	NeurIPS 2022	0.9181	2.0633	0.4411	0.2037
	MTR++ (Ours)	-	**0.8795**	**1.9509**	**0.4143**	**0.2326**
Val	MTR (Ours)	NeurIPS 2022	0.9132	2.0536	0.4372	0.1992
	MTR++ (Ours)	-	**0.8859**	**1.9712**	**0.4106**	**0.2398**

TABLE 2



Method	Number of Parameters	Inference Latency	Miss Rate ↓	mAP ↑
†SceneTransformer [39]	15.3M	52ms (V100)	0.1564	0.2788
DenseTNT [21]	1.1M	540ms	0.1573	0.3281
HDGT [27]	12.1M	1320ms	0.1511	0.2854
MTR++ (light)	11.7M	67ms	0.1430	0.3896
MTR	65.8M	193ms	0.1351	0.4129
MTR++	86.6M	118ms	**0.1298**	**0.4329**

TABLE 3



Method	Miss Rate ↓ (K=6)	Miss Rate ↓ (K=1)	brier-minFDE ↓ (K=6)
MTR++ (Ours)	**0.14**	**0.56**	**1.88**
MTR (Ours)	0.15	0.58	1.98
TENET [55]	0.19	0.61	1.90
OPPred	0.19	0.60	1.92
Qml	0.19	0.62	1.95
GANet	0.17	0.60	1.97
VI LaneIter	0.19	0.61	2.00
QCNet	0.21	0.60	2.14
THOMAS [20]	0.20	0.64	2.16
HDGT [27]	0.21	0.66	2.24
GNA	0.29	0.71	2.45
vilab	0.29	0.71	2.47

TABLE 4

For the ablation study, they tested the effectiveness of each component in MTR/MTR++. The paper found that the learnable intention query significantly improves mAP metric, the iterative trajectory refinement reduces miss rate by 1.48% and improves performance of mAP by +1.6%, local attention for context encoding proves more memory efficient, symmetric scene context reduces inference latency and memory cost, mutually guided intention query strategy enhances improvements, and query-centric self-attention increases performance for the mAP. Other results of the ablation studies have revealed that higher intention queries lead to fewer miss rates and better performance for the mAP, dense future predictions increase performance of MTR++, and greater performance improvement with more decoder layers from 1 – 6.

The paper discusses a couple of failure cases and future challenges. One problem is that the model achieves a relatively lower average precision while maintaining a favorable miss rate. Another challenge the paper discusses is the model generates homogenized future trajectories that fail to encompass the genuine intentions of the agent. Future works the paper mentions includes investigating methods to refine alignment between predicted scores and quality of the corresponding trajectory and development of methods that yield comprehensive multimodal behaviors.

Strengths

The paper provides a methodology that reportedly works better than state-of-the-art architectures for predicting trajectories of other agents. This contribution is particularly valuable because understanding the intentions of other agents proves a serious obstacle for all sorts of applications involving some sort of obstacle evasion. Another strength of this paper includes the memory efficiency and state-of-the-art inferencing on interactions between agents. As evidenced in the paper, MTR++ performed the best when tasked to predict the interaction between multiple agents. One thing I learned from reading the paper was different approaches to predicting trajectories of other agents. One way is to use *focal-agent-centric* approach which focuses on using the GPS coordinates of an agent to predict their intentions. Another way is the *query-centric self-attention* approach which converts each token into a local position. Another thing I learned was about intention queries which essentially predict the target location of an agent.

Potential Improvements

There are a couple of things I think the paper and/or the methodology could improve on to reach its stated contributions/goals and leverage the latest techniques or models to improve the method. I think the paper is wordy and does not clearly explain various terminologies which could hinder the paper from properly communicating its stated goals. One possible improvement the paper discusses includes a problem with lower average precision while maintaining a favorable miss rate. The paper argues that the discrepancy is due to the inadequacy of current quality score estimations in accurately reflecting the quality of each predicted trajectory. One future direction that the paper discusses is investigating methods to refine the alignment between predicted scores and the quality of the corresponding trajectory. Another challenge is that the model may generate homogenized future trajectories which could be fixed with a more diverse dataset. The discussed improvements may help the model reach its stated contribution/goal. As of now, the MTR++ is state-of-the-art but potential improvements for leveraging the latest techniques or models to improve the method is to use a diverse dataset with different types of vehicles and road systems. Another technique could be to use diffusion models to predict possible intention queries of each agent which may help generalize the model to even edge cases.

Extensions

One idea I want to try with this paper is to test it on a different dataset, specifically from different countries such as India or South Korea. Since a cornerstone of the model is the intention query, I would be curious to know how the model would generate queries for unconventional roads such as those in India. Another extension would be to apply the method to different agents such as drones in air space. Due to the multi-dimensional movements of drones in the air, I believe it would pose a unique challenge to the model. Finally, I wonder if there are other things to encode other than polylines for the encoder portion of the model.