2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Wayformer: Motion Forecasting via Simple & Efficient Attention Networks

Nigamaa Nayakanti*, Rami Al-Rfou*, Aurick Zhou, Kratarth Goel, Khaled S. Refaat, Benjamin Sapp[1]

***Abstract*— Motion forecasting for autonomous driving is a challenging task because complex driving scenarios involve a heterogeneous mix of static and dynamic inputs. It is an open problem how best to represent and fuse information about road geometry, lane connectivity, time-varying traffic light state, and history of a dynamic set of agents and their interactions into an effective encoding. To model this diverse set of input features, many approaches proposed to design an equally complex system with a diverse set of modality specific modules. This results in systems that are difficult to scale, extend, or tune in rigorous ways to trade off quality and efficiency.**

In this paper, we present Wayformer, a family of simple and homogeneous attention based architectures for motion forecasting. Wayformer offers a compact model description consisting of an attention based scene encoder and a decoder. In the scene encoder we study the choice of early, late and hierarchical fusion of input modalities. For each fusion type we explore strategies to trade off efficiency and quality via factorized attention or latent query attention. We show that early fusion, despite its simplicity, is not only modality agnostic but also achieves state-of-the-art results on both Waymo Open Motion Dataset (WOMD) and Argoverse leaderboards, demonstrating the effectiveness of our design philosophy.

I. INTRODUCTION

Predicting the future behavior of agents on the road is an essential task for safe and comfortable autonomous driving. The modeling needed for scene understanding is challenging for many reasons. For one, the *output* is highly unstructured and multimodal [43], [8]—*e.g.*, a person driving a vehicle could carry out one of many underlying intents unknown to an observer, and representing a distribution over diverse possible futures is required. A second challenge is that the *input* consists of a heterogeneous mix of modalities, including different agents' past physical state, interactions among them, static road information (*e.g.* location of lanes and their connectivity), and time-varying traffic light information.

Self-Attention based architectures are generic modeling primitives that are domain agnostic. In particular, recently, transformers have become the model of choice in the fields of Natural Language Processing [46], [11], [39], [4], Computer Vision [12], [34], [5] and Speech Recognition [18], [3], [16] because of their ability to achieve state of the art results. Recent works in motion forecasting, however, focused on hand engineering modality specific sub-components [8], [45]. While some efforts added transformers as sub-components [49], [15], [33], these designs hand engineered the order in which the modalities are processed.

* Equal Contribution
[1] Waymo, 1600 Amphitheatre Pkwy, Mountain View, California, USA. {nigamaa, rmyeid, aurickz, kratarth, krefaat, bensapp}@waymo.com

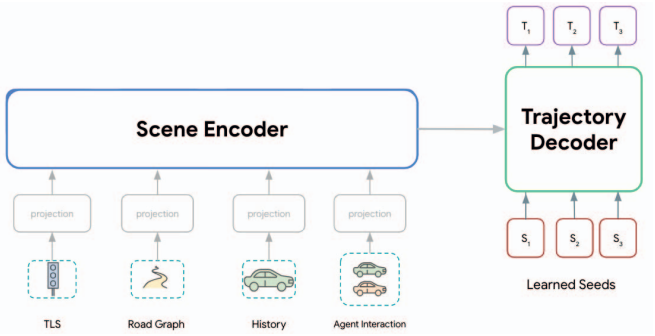


Fig. 1: The Wayformer architecture as a pair of encoder/decoder Transformer networks. This model takes multimodal scene data as input and produces multimodal distribution of trajectories.

In this work, we investigate if the complexity of modeling the multimodal *input space*, adopted by previous approaches, is necessary. We design Wayformer, a family of architectures that consist of self-attention networks to encode the scene and cross-attention network to predict trajectories (Fig 1).

First, we propose to simultaneously process all modalities with no specific order. Next, we vary the level of interaction across modalities by use of early, late, and hierarchical fusion types (Fig 2). Each fusion type controls how early modalities exchange information and as well as the assignment of model capacity between unimodal and fused multimodal encoding.

Processing all modalities simultaneously incurs high computational cost. For efficiency, we experiment with two self-attention proposals that trade off efficiency and expressivity: latent queries [29], [25] and factorized attention [2], [20]. Latent queries control input resolution while factorized attention limits information exchange across space and time.

We conduct thorough and comprehensive ablation studies on various combinations of fusion and attention types and demonstrate that our strategy of simplifying multimodal model design is successful at achieving state-of-art results on well studied benchmarks. In particular, the relative improvement over current[1] SOTA in Argoverse is the highest single advancement in the top 15 submissions. We analyze our model's performance through simulation, visualization and on an experimental robot.

We summarize our contributions as the following:

- We show that Wayformer-*Early Fusion*, the simplest and most modality-agnostic fusion type, achieves state-

[1]At the time when this work concluded

of-the-art results.
- We show that simultaneously encoding modalities independent of how early we fuse them, is sufficient to model motion forecasting.
- We study several proposals to enhance attention efficiency and identify latent queries as a promising approach to reduce computational footprint with no or minimal regression to quality.

II. Related Work

a) Modality specialized architectures: Successful modeling techniques for motion forecasting fuse multi-modal inputs that represent different static, dynamic, social and temporal aspects of the scene [9], [22], [13], [40], [8], [45]. One class of models draws heavily from the computer vision literature, processing inputs as a multichannel rasterized top-down image using spatio-temporal convolutional networks[10], [8], [30], [21], [7], [52], [31], [51]. A popular alternative is to use an entity-centric approach, where agent state history is typically encoded via sequence modeling techniques like RNNs [36], [27], [1], [41] or temporal convolutions [32]. Road elements are approximated with basic primitives (e.g. piecewise linear segments) which encode direction and semantic information. Modeling relationships between entities is often presented as an information aggregation process, and models employ pooling [52], [14], [1], [19], [36], [30], soft-attention [36], [52] or graph neural networks [6], [32], [27].

b) Sequential modality fusion: A recent approach to encode multimodal data is to sequentially process one modality at a time [33], [25], [15]. [33] ingests the scene in the order {agent history, nearby agents, map}; they argue that it is computationally expensive to perform self-attention simultaneously over multiple modalities at once. [15] pre-encodes the agent history and contextual agents through self-attention and cross-attends to the map with agent encodings as queries. The order of self-attention and cross-attention relies heavily on the designer's intuition and has, to our knowledge, not been ablated before.

c) Efficient attention: Flattening high dimensional data leads to long sequences which make self-attention computationally prohibitive. [20] proposed limiting each attention operation to a single axes to alleviate the computational costs and applied this technique to autoregressive generative modeling for images. Similarly, [2] factorize the spatial and temporal dimensions of the video input when constructing their self-attention based classifier. This axis based attention, which gets applied in interleaved fashion across layers, has been adopted in Transformer-based motion forecasting models [15] and graph neural network approaches [48]. The order of applying attention over {temporal, social/spatial} dimensions has been studied with two different common patterns: (a) Temporal first [1], [19], [28] (b) Social/Spatial first [23], [42]. In Section IV-B, we study a 'sequential' mode and contrast it with interleaved mode where interleave dimensions of attention similar to [15]. For a complete discussion of previous works, we refer the reader to the comprehensive survey [29], [24], [44].

In contrast to previous works that fused modalities sequentially [33], [15], [37], we show that we can fuse all modalities simultaneously. Moreover, with early fusion's (Figure 2) success at achieving state-of-art results we demonstrate that there is no need to dedicate model capacity separately for each modality.

III. Multimodal Scene Understanding

Driving scenarios consist of multimodal data, such as road information, traffic light state, agent history, and agent interactions. In this section we detail the representation of these modalities in our setup. For readability, we define the following symbols: A denotes the number of modeled ego-agents, T denotes the number of past and current timesteps being considered in the history. Each modality m, has a feature size D_m and number of objects S_m. Hence, each modality is represented by a tensor of shape $[A, T, S_m, D_m]$.

a) Agent History: contains a sequence of past agent states along with the current state $[A, T, 1, D_h]$. For each timestep $t \in T$, we consider features that define the state of the agent e.g. x, y, velocity, acceleration, bounding box and so on. We include a context dimension $S_h = 1$ for homogeneity.

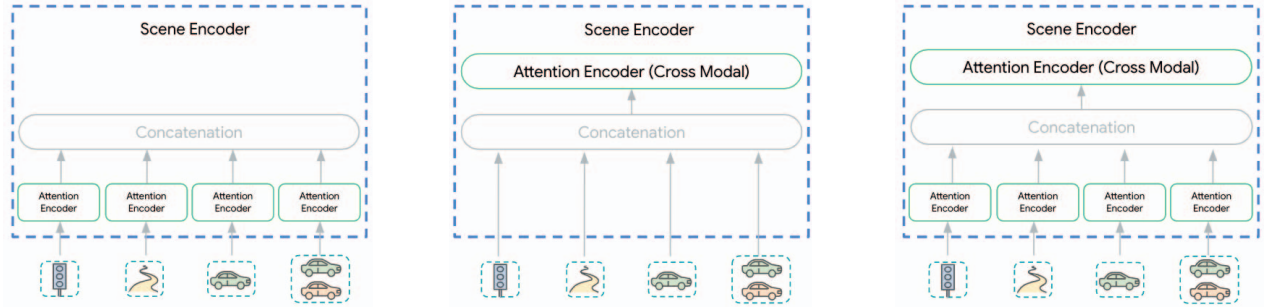
b) Agent Interactions: The interaction tensor $[A, T, S_i, D_i]$ represents the relationship between agents. For each modeled agent $a \in A$, a fixed number of the closest context agents $c_i \in S_i$ around the modeled agent are considered. These context agents represent the agents which influence the behavior of our modeled agent. The features in D_i represent the physical state of each context agents (as in D_h above), but transformed into the frame of reference of our ego-agent.

c) Roadgraph: The roadgraph $[A, 1, S_r, D_r]$ contains road features around the agent. Following [8], we represent roadgraph segments as polylines, approximating the road shape with collections of line segments specified by their endpoints and annotated with type information. We use S_r roadgraph segments closest to the modeled agent. Note that there is no time dimension for the road features, but we include a time dimension of 1 for homogeneity with the other modalities.

d) Traffic Light State: For each agent $a \in A$, traffic light information $[A, T, S_{tls}, D_{tls}]$ contains the states of the traffic signals that are closest to that agent. Each traffic signal point $tls \in S_{tls}$ has features D_{tls} describing the position and confidence of the signal.

IV. Wayformer

Wayformer is a simple family of models that consists of two main components: a Scene Encoder and a Trajectory Decoder (Figure 1) The scene encoder is mainly composed of one or more attention encoders that summarize the driving scene. The decoder is a stack of one or more standard transformer cross-attention blocks, in which learned initial queries are fed in, and then cross-attended with the scene encoding to produce trajectories. We assume no special treatment of different input modalities. Each modality is just represented as a sequence of tokens that are fed into the model. We formulate different variants of Wayformer models depending on the choice of fusion type and attention mechanism.



(a) Late Fusion (b) Early Fusion (c) Hierarchical Fusion

Fig. 2: Wayformer scene encoder fusing multimodal inputs at different stages. Late fusion dedicates an attention encoder per modality while early fusion process all inputs within one cross modal encoder. Finally, hierarchical fusion combines both the approaches.

a) Frame of Reference: As our model is trained to produce futures for a single agent, we transform the scene into an ego-centric frame of reference by centering and rotating the scene's spatial features around the ego-agent's position and heading at the current time step.

b) Projection Layers: Different input modalities may not share the same number of features, so we project them to a common dimension D before concatenating all modalities along the temporal and spatial dimensions $[S, T]$. We found the simple transformation $\text{Projection}(x_i) = \text{relu}(\mathbf{W}x_i + b)$, where $x_i \in \mathbb{R}^{D_m}$, $b \in \mathbb{R}^D$, and $\mathbf{W} \in \mathbb{R}^{D \times D_m}$, to be sufficient. Concretely, given an input of shape $[A, T, S_m, D_m]$ we project its last dimension producing a tensor of size $[A, T, S_m, D]$. After projection layers, positional embeddings are applied to different modalities.

A. Fusion

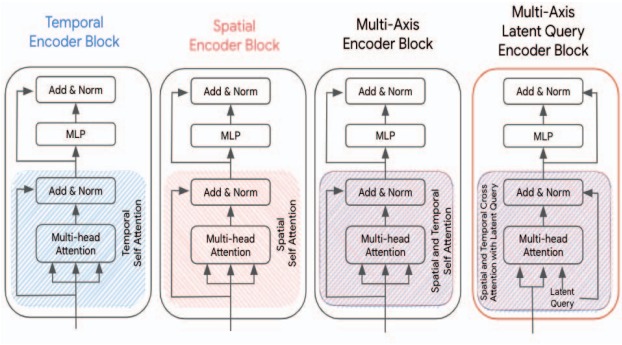
The scene encoder combines the information from all modalities to generate a representation of the environment. Concretely, we aim to learn a scene representation $\boldsymbol{Z} = Encoder(\{m_0, m_1, ..., m_k\})$, where $m_i \in \mathbb{R}^{A \times (T \times S_m) \times D}$, $\boldsymbol{Z} \in \mathbb{R}^{A \times L \times D}$, and L is a hyperparameter.

However, the diversity of input sources makes this integration a non-trivial task. Modalities might not be represented at the same abstraction level or scale: {pixels vs objects}. Therefore, some modalities might require more computation than the others. Splitting compute and parameter count among modalities is application specific and non-trivial to hand-engineer. We attempt to simplify the process by proposing three levels of fusion: {Late, Early, Hierarchical}.

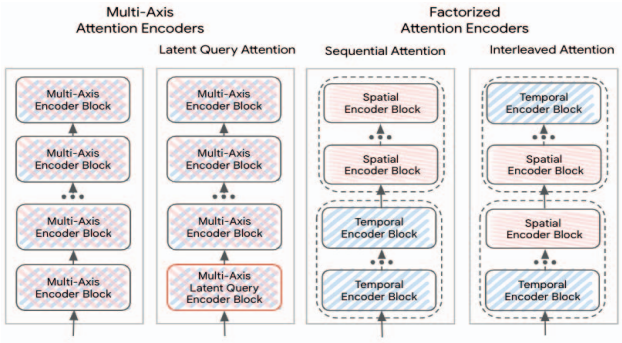
1) Late Fusion: This is the most common approach used by motion forecasting models, where each modality has its own dedicated encoder (Figure 2). We set the width of these encoders to be equal for simplicity. Moreover, we share the same depth across all encoders to narrow down the exploration space to a manageable scope. Transfer of information across modalities is allowed only in the cross-attention layers of the trajectory decoder.

2) Early Fusion: Instead of dedicating a self-attention encoder to each modality, early fusion reduces modality specific parameters to only the projection layers (See Figure 2). In this paradigm, the scene encoder consists of a single self-attention encoder ("Cross-Modal Encoder"), giving the network maximum flexibility in assigning importance across modalities with minimal inductive bias.

3) Hierarchical Fusion: As a compromise between the two previous extremes, capacity is split between modality-specific self-attention encoders and the cross-modal encoder in a hierarchical fashion. As done in late fusion, width and depth is common across attention encoders and the cross modal encoder. This effectively splits the depth of the scene encoder between modality specific encoders and the cross modal encoder (Figure 2).



(a) Encoder Blocks



(b) Encoders

Fig. 3: A summary of encoder architectures considered for Wayformer. (a) provides an overview of different encoder blocks and (b) explains how these blocks are arranged to construct the encoder.

B. Attention

Transformer networks do not scale well for large multidimensional sequences due to two factors: (a) Self-attention is quadratic in the input sequence length. (b) Position-wise Feed-forward networks are expensive sub-networks. In the following sections, we discuss different speedups to the transformer networks that will help us scale more effectively.
1) Multi-Axis Attention: This refers to the default transformer setting which applies self-attention across both spatial and temporal dimensions simultaneously (See Figure 3b), which we expect to be the most expensive computationally. Computational complexity of early, late and hierarchical fusions with multi-axis attention is $\mathcal{O}(S_m^2 \times T^2)$.
2) Factorized Attention: Computational complexity of the self-attention is a quadratic in input sequence length. This becomes more pronounced in multi-dimensional sequences, since each extra dimension increases the size of the input by a multiplicative factor. For example, some input modalities have both temporal and spatial dimensions, so the compute cost scales as $\mathcal{O}(S_m^2 \times T^2)$. To alleviate this, we consider factorized attention [2], [20] along the two dimensions. This exploits the multidimensional structure of input sequences by applying self-attention over each dimension individually, which reduces the cost of self-attention sub-network from $\mathcal{O}(S_m^2 \times T^2)$ to $\mathcal{O}(S_m^2) + \mathcal{O}(T^2)$. Note that the linear term still tends to dominate if $\sum_m \mathcal{S}_m \times T << 12 \times D$ [26].

While factorized attention has the potential to reduce computation compared to multi-axis attention, it introduces complexity in deciding the order in which self-attention is applied to each dimension. In our work, we compare two paradigms of factorized attention (see Figure 3b):

- **Sequential:** an N layer encoder consists of $N/2$ temporal encoder blocks followed by another $N/2$ spatial encoder blocks.
- **Interleaved:** an N layer encoder alternates between temporal and spatial encoder blocks $N/2$ times.

Another approach to address the computational costs of large input sequences is to use latent queries [29], [25] in the first encoder block, where input $x \in \mathbb{R}^{A \times L_{\text{in}} \times D}$ is mapped to latent space $z \in \mathbb{R}^{A \times L_{\text{out}} \times D}$. These latents $z \in \mathbb{R}^{A \times L_{\text{out}} \times D}$ are then processed further by a series of encoder blocks to produce an embedding of the same reduced size (see Figure 3a). This gives us full freedom to set the latent space resolution, reducing the computational costs of the both self-attention component and the position-wise feed-forward network of each block. We set the reduction value $(R = L_{\text{out}}/L_{\text{in}})$ to be a percentage of the input sequence length. Reduction factor R is kept constant across all the attention encoders in late and hierarchical fusions.

C. Trajectory Decoding

As our focus is on how to integrate information from different modalities in the encoder, we simply follow the training and output format of [8], [45], where the Wayformer predictor outputs a mixture of Gaussians to represent the possible futures an agent may take. To generate predictions, we use a Transformer decoder which cross attends a set of k learned initial queries $(\boldsymbol{S}_i \in \mathbb{R}^D)_{i=1}^k$ with the scene embeddings from the encoder in order to generate k embeddings for each component in the output mixture of Gaussians.

Given the embedding Y_i for a mixture component, we apply a linear projection layer to produces the unnormalized log-likelihood for mixture the component. To generate the trajectory, we project Y_i using another linear layer to output 4 time series: $T_i = \{\mu_x^t, \mu_y^t, \log \sigma_x^t, \log \sigma_y^t\}_{t=1}^T$ corresponding to the means and log-standard deviations of the predicted Gaussian at each timestep.

During training, we follow [8], [45] in decomposing the loss into separate classification and regression losses. Given k predicted Gaussians $(T_i)_{i=1}^k$, let $\hat{i}$ denote the index of the Gaussian with mean closest to the ground truth trajectory G. We train the mixture likelihoods on the log likelihood of selecting the index $\hat{i}$, and the Gaussian $T_{\hat{i}}$ to maximize the log-probability of the ground truth trajectory.

$$\max \underbrace{\log \Pr(\hat{i} \mid Y)}_{\text{classification loss}} + \underbrace{\log \Pr(G|T_{\hat{i}})}_{\text{regression loss}}. \tag{1}$$

This is followed optionally by a trajectory aggregation step (please refer to accompanying video for more details).

V. Experimental Setup

A. Datasets

a) Waymo Open Motion Dataset (WOMD): consists of 103K scenarios. The trajectories are sampled at 5Hz, with 1second of history and 8 seconds of future prediction horizon.
b) Argoverse Dataset: consists of 333K scenarios. The trajectories are sampled at 10Hz, with 2 seconds of history and a 3-second future prediction horizon.

B. Training Details and Hyperparameters

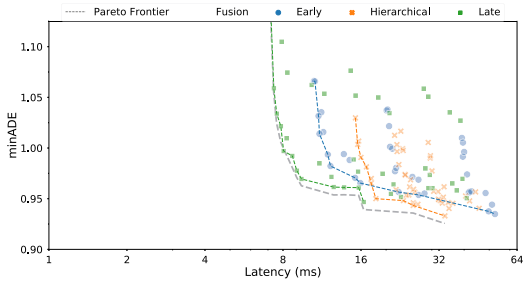
We compare models using competition specific metrics associated with these datasets (see accompanying video for more details).

For all metrics, we consider only the top $\hat{k} = 6$ most likely modes output by our model (after trajectory aggregation) and use only the mean of each mode.

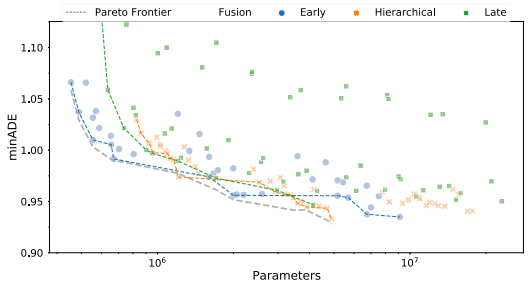
For all experiments, we train models using the AdamW optimizer [35] with an initial learning rate of 2e-4 and linearly decaying to 0 over 1M steps. We train models using 16 TPU-v3 cores each, with a batch size of 16 per core, resulting in a total batch size of 256 examples per step.

To vary the capacity of the models, we consider hidden sizes among $\{64, 128, 256\}$ and depths among $\{1, 2, 4\}$ layers. We fix the intermediate size in the feedforward network of the Transformer block to be either 2 or 4 times the hidden size.

For our architecture study in Sections (VI-A-VI-C), each predictor outputs a mixture of Gaussians with $k = 6$ components, with no trajectory aggregation. For our benchmark results in Section VI-D, each predictor outputs a mixture



(a) Efficiency



(b) Capacity

Fig. 4: MinADE of models with multi-axis attention.

of Gaussians with $k = 64$ components, and we prune the mixture components using the trajectory aggregation scheme described accompanying video. In ablation studies, we use a small number Gaussian mixture components to speed up model training. We increase the number of modes and utilize aggregation to improve the target metrics for the final submitted models.

VI. Results

We present experiments that demonstrate the trade-offs of combining different fusion strategies with vanilla self-attention (multi-axis) and more optimized methods such as factorized attention and learned queries. In our ablation studies (Section VI-A-VI-C), we trained models with varying capacities (0.3M-20M parameters) for 1M steps on WOMD. Each point within an architectural family represents a model with a different hidden size, depth, or intermediate size. We also report their inference latency on a current generation GPU, capacity, and minADE as a proxy of quality.

A. Multi-Axis Attention

We train Wayformer models on early, hierarchical and late fusion in combination with multi-axis attention. In Figure (4a), we show that for models with low latency ($x \leq 16$ ms), late fusion represents an optimal choice. These models are computationally cheap since there is no interaction between modalities during the scene encoding step. Adding the cross modal encoder for hierarchical models unlocks further quality gains for models in the range ($16\text{ms} < x < 32\text{ms}$). Finally, we can see that early fusion can match hierarchical fusion at higher computational cost ($x > 32\text{ms}$). We then study the model quality as a function of capacity, as measured by the number of trainable parameters (Figure 4b). Small models perform best with early fusion, but as model capacity increases, sensitivity to the choice of fusion decreases dramatically.

B. Factorized Attention

To reduce the computational budget of our models, we train models with factorized attention instead of jointly attending to spatial and temporal dimensions together. When combining different modalities together for the cross modal encoder, we first tile the roadgraph modality to a common temporal dimension as the other modalities, then concatenate modalities along the spatial dimension. After the scene encoder, we pool the encodings over the time dimension before feeding to the predictor.

We study two types of factorized attention: sequential, interleaved (Figure 5). First, we observe that both sequential and interleaved factorized attention perform similarly across all types of fusion. Second, we are surprised to see quality gains from applying factorized attention to the early and late fusion cases (Figures 5a, 5b). Finally, we only observe latency improvements for late fusion models (Figure 5b), since tiling the road graph to the common temporal dimension in cross-modal encoder used in early and hierarchical fusion significantly increases the count of tokens.

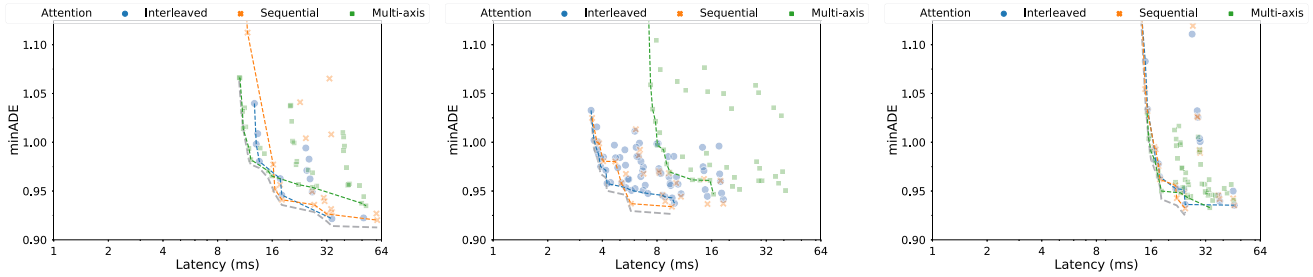
C. Latent Queries

In this study, we train models with multi-axis latent query encoders with varying levels of input sequence length reduction in the first layer as shown in Figure 5. The number of the latent queries is calculated to be a percentage of the input size of the Transformer network with 0.0% indicating the baseline models (multi-axis attention with no latent queries as presented in Figure 4). We experiment with reducing the original input resolution by $0.25, 0.5$, 0.75 and 0.9 times the original sequence length.

Figure 6 shows the results of applying latent queries, which speeds up all fusion models by 2x-16x times with minimal to no quality regression. Early and hierarchical fusion still produce the best quality results, showing the importance of the cross modal interaction stage.

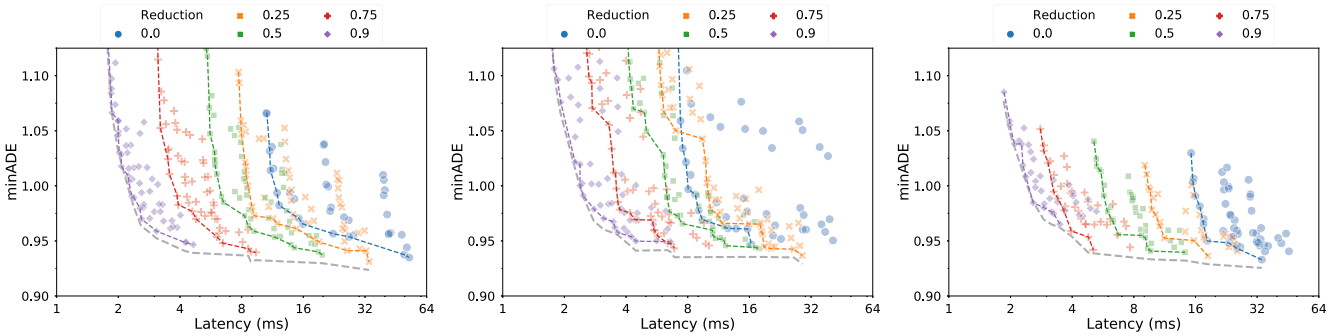
D. Benchmark Results

We validate our learnings by comparing Wayformer models to competitive models on popular benchmarks of motion forecasting. We choose early fusion models since they match the quality of the hierarchical or late fusion models while being the simplest. Moreover, as models' capacity increases they are less sensitive to the choice of fusion (See Figure 4b). We use latent queries since they speed up models without noticeable quality regression, and in one variant, we combine it with the sequential variant of factorized attention (see accompanying video for more details) since that improved the quality further in our ablation studies. We further apply ensembling, a standard practice for producing SOTA results for leaderboard submissions. Full hyperparameters for Wayformer models reported on benchmarks are reported in accompanying video.



(a) Early (b) Late (c) Hierarchical

Fig. 5: Factorized attention improves quality, but only speeds up late fusion models.



(a) Early (b) Late (c) Hierarchical

Fig. 6: Latent queries reduce models' latency without significant degradation to the quality.

Models		Waymo Open Motion Dataset minFDE (↓)	minADE (↓)	MR (↓)	Overlap (↓)	mAP* (↑)	Argoverse Dataset Brier-minFDE* (↓)	minFDE (↓)	MR (↓)	minADE (↓)
SceneTransformer [38]		1.212	0.612	0.156	0.147	0.279	1.8868	1.2321	0.1255	0.8026
DenseTNT [17]		1.551	1.039	0.157	0.178	0.328	1.9759	1.2858	0.1285	0.8817
MultiPath [8]		2.040	0.880	0.345	0.166	0.409	-	-	-	-
MultiPath++ [45]		1.158	0.556	0.134	0.131	0.409	1.7932	1.2144	0.1324	0.7897
LaneConv [32]		-	-	-	-	-	2.0539	1.3622	0.1600	0.8703
LaneRCNN [50]		-	-	-	-	-	2.1470	1.4526	0.1232	0.9038
mmTransformer [33]		-	-	-	-	-	2.0328	1.3383	0.1540	0.8346
TNT [52]		-	-	-	-	-	2.1401	1.4457	0.1300	0.9400
DCMS [47]		-	-	-	-	-	1.7564	**1.1350**	**0.1094**	**0.7659**
Wayformer Early Fusion	**Attention**									
	LQ + Multi-Axis	1.128	**0.545**	**0.123**	**0.127**	**0.419**	**1.7408**	1.1615	0.1186	0.7675
	LQ + Factorized	**1.126**	**0.545**	**0.123**	**0.127**	0.412	1.7451	1.1625	0.1192	0.7672

TABLE I: Wayformer models and select SOTA baselines on Waymo Open Motion Dataset 2021 and Argoverse 2021. * denotes the metric used for leaderboard ranking. LQ denotes latent query.

In Table I, we present results on the Waymo Open Motion Dataset and Argoverse Dataset. We use the standard metrics used for the each dataset for their respective evaluation (see accompanying video). For the Waymo Open Motion Dataset, both Wayformer early fusion models outperform other models across all metrics; early fusion of input modalities results in better overall metrics independent of the attention structure (multi-axis or factorized attention). For comparative analysis of Wayformer Vs MultiPath++ predictions, we include visualizations of 12 scenarios in accompanying video. We further include remaining shortcomings that could be addressed in future work and additional results without ensembling.

Finally, we are providing, in the supplementary materials, videos that show an experimental robot being driven by a demo stack which incorporates Wayformer motion forecasting of surrounding agents as an input to path planning demonstrating the ability to handle real world scenarios.

VII. Conclusion

Contrary to the majority of existing works, which focus on custom models for the domain, we demonstrate attention based networks are sufficient for processing multi-dimensional (temporal and spatial) inputs from heterogeneous sources and develop state-of-the-art motion forecasting models, without any special domain knowledge, and winning in two popular leaderboards. To speed up Wayformer models, we observed that introducing latent queries in the first layer not only resulted in SOTA results but also provide an effective way of scaling with limited latency requirements.

ACKNOWLEDGEMENT

We thank Balakrishnan Varadarajan for help on ensembling strategies; Dragomir Anguelov and Eugene Ie for their helpful feedback on the paper.

REFERENCES

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–971, 2016.

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6836–6846, October 2021.

[3] Alexei Baevski, Henry Zhou, Abdel rahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *ArXiv*, abs/2006.11477, 2020.

[4] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher 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. *CoRR*, abs/2005.14165, 2020.

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Herv'e J'egou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9630–9640, 2021.

[6] Sergio Casas, Cole Gulino, Renjie Liao, and Raquel Urtasun. Spagnn: Spatially-aware graph neural networks for relational behavior forecasting from sensor data. In *IEEE Intl. Conf. on Robotics and Automation*. IEEE, 2020.

[7] Sergio Casas, Wenjie Luo, and Raquel Urtasun. Intentnet: Learning to predict intention from raw sensor data. In *Conf. on Robot Learning*, 2018.

[8] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In *CoRL*, 2019.

[9] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, and James Hays. Argoverse: 3d tracking and forecasting with rich maps. *CoRR*, abs/1911.02620, 2019.

[10] Henggang Cui, Vladan Radosavljevic, Fang-Chieh Chou, Tsung-Han Lin, Thi Nguyen, Tzu-Kuo Huang, Jeff Schneider, and Nemanja Djuric. Multimodal trajectory predictions for autonomous driving using deep convolutional networks. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2090–2096. IEEE, 2019.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2021.

[13] Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Benjamin Sapp, Charles R. Qi, Yin Zhou, Zoey Yang, Aurelien Chouard, Pei Sun, Jiquan Ngiam, Vijay Vasudevan, Alexander McCauley, Jonathon Shlens, and Dragomir Anguelov. Large scale interactive motion forecasting for autonomous driving : The waymo open motion dataset. *CoRR*, abs/2104.10133, 2021.

[14] Jiyang Gao, Chen Sun, Hang Zhao, Yi Shen, Dragomir Anguelov, Congcong Li, and Cordelia Schmid. VectorNet: Encoding hd maps and agent dynamics from vectorized representation. In *CVPR*, 2020.

[15] Roger Girgis, Florian Golemo, Felipe Codevilla, Jim Aldon D'Souza, Samira Ebrahimi Kahou, Felix Heide, and Christopher J. Pal. Latent variable nested set transformers & autobots. *CoRR*, abs/2104.00563, 2021.

[16] Yuan Gong, Yu-An Chung, and James R. Glass. Ast: Audio spectrogram transformer. *ArXiv*, abs/2104.01778, 2021.

[17] Junru Gu, Chen Sun, and Hang Zhao. Densetnt: End-to-end trajectory prediction from dense goal sets. *CoRR*, abs/2108.09640, 2021.

[18] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. Conformer: Convolution-augmented transformer for speech recognition. *ArXiv*, abs/2005.08100, 2020.

[19] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[20] Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*, 2019.

[21] Joey Hong, Benjamin Sapp, and James Philbin. Rules of the road: Predicting driving behavior with a convolutional model of semantic interactions. *CoRR*, abs/1906.08945, 2019.

[22] John Houston, Guido Zuidhof, Luca Bergamini, Yawei Ye, Ashesh Jain, Sammy Omari, Vladimir Iglovikov, and Peter Ondruska. One thousand and one hours: Self-driving motion prediction dataset. *CoRR*, abs/2006.14480, 2020.

[23] Yingfan Huang, Huikun Bi, Zhaoxin Li, Tianlu Mao, and Zhao qi Wang. Stgat: Modeling spatial-temporal interactions for human trajectory prediction. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6271–6280, 2019.

[24] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Andrew Brock, Evan Shelhamer, Olivier J. H'enaff, Matthew M. Botvinick, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver io: A general architecture for structured inputs & outputs. *ArXiv*, abs/2107.14795, 2021.

[25] Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver: General perception with iterative attention. In *ICML*, 2021.

[26] Jared Kaplan, Sam McCandlish, T. J. Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeff Wu, and Dario Amodei. Scaling laws for neural language models. *ArXiv*, abs/2001.08361, 2020.

[27] Siddhesh Khandelwal, William Qi, Jagjeet Singh, Andrew Hartnett, and Deva Ramanan. What-if motion prediction for autonomous driving. *ArXiv*, 2020.

[28] Vineet Kosaraju, Amir Sadeghian, Roberto Martín-Martín, Ian D. Reid, Seyed Hamid Rezatofighi, and Silvio Savarese. Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. In *NeurIPS*, 2019.

[29] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *ICML*, 2019.

[30] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B. Choy, Philip H. S. Torr, and Manmohan Krishna Chandraker. DESIRE: distant future prediction in dynamic scenes with interacting agents. *CoRR*, abs/1704.04394, 2017.

[31] Junwei Liang, Lu Jiang, Kevin Murphy, Ting Yu, and Alexander Hauptmann. The garden of forking paths: Towards multi-future trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10508–10518, 2020.

[32] Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph representations for motion forecasting. *arXiv preprint arXiv:2007.13732*, 2020.

[33] Yicheng Liu, Jinghuai Zhang, Liangji Fang, Qinhong Jiang, and Bolei Zhou. Multimodal motion prediction with stacked transformers. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7573–7582, 2021.

[34] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002, 2021.

[35] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[36] Jean Pierre Mercat, Thomas Gilles, Nicole El Zoghby, Guillaume Sandou, Dominique Beauvois, and Guillermo Pita Gil. Multi-head attention for multi-modal joint vehicle motion forecasting. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9638–9644, 2020.

[37] Jiquan Ngiam, Benjamin Caine, Vijay Vasudevan, Zhengdong Zhang, Hao-Tien Lewis Chiang, Jeffrey Ling, Rebecca Roelofs, Alex Bewley, Chenxi Liu, Ashish Venugopal, David Weiss, Benjamin Sapp, Zhifeng Chen, and Jonathon Shlens. Scene transformer: A unified multi-task model for behavior prediction and planning. *CoRR*, abs/2106.08417, 2021.

[38] Jiquan Ngiam, Benjamin Caine, Vijay Vasudevan, Zhengdong Zhang, Hao-Tien Lewis Chiang, Jeffrey Ling, Rebecca Roelofs, Alex Bewley, Chenxi Liu, Ashish Venugopal, David Weiss, Benjamin Sapp, Zhifeng Chen, and Jonathon Shlens. Scene transformer: A unified multi-task model for behavior prediction and planning. *CoRR*, abs/2106.08417, 2021.

[39] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[40] Nicholas Rhinehart, Kris Kitani, and Paul Vernaza. R2P2: A reparameterized pushforward policy for diverse, precise generative path forecasting. In *ECCV*, 2018.

[41] Nicholas Rhinehart, Rowan McAllister, Kris Kitani, and Sergey Levine. Precog: Prediction conditioned on goals in visual multi-agent settings. In *ECCV*, 2019.

[42] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Multi-agent generative trajectory forecasting with heterogeneous data for control. *CoRR*, abs/2001.03093, 2020.

[43] Yichuan Charlie Tang and Ruslan Salakhutdinov. Multiple futures prediction. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2019. Curran Associates Inc.

[44] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *ArXiv*, abs/2009.06732, 2020.

[45] Balakrishnan Varadarajan, Ahmed S. Hefny, Avikalp Srivastava, Khaled S. Refaat, Nigamaa Nayakanti, Andre Cornman, Kan Chen, Bertrand Douillard, C. P. Lam, Drago Anguelov, and Benjamin Sapp. Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction. In *ICRA*, 2021.

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[47] Maosheng Ye, Jiamiao Xu, Xunnong Xu, Tongyi Cao, and Qifeng Chen. Dcms: Motion forecasting with dual consistency and multi-pseudo-target supervision, 2022.

[48] Cunjun Yu, Xiao Ma, Jiawei Ren, Haiyu Zhao, and Shuai Yi. Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In *ECCV*, 2020.

[49] Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris Kitani. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. *ArXiv*, abs/2103.14023, 2021.

[50] Wenyuan Zeng, Ming Liang, Renjie Liao, and Raquel Urtasun. Lanercnn: Distributed representations for graph-centric motion forecasting. *CoRR*, abs/2101.06653, 2021.

[51] Wenyuan Zeng, Wenjie Luo, Simon Suo, Abbas Sadat, Bin Yang, Sergio Casas, and Raquel Urtasun. End-to-end interpretable neural motion planner. In *CVPR*, 2019.

[52] Hang Zhao, Jiyang Gao, Tian Lan, Chen Sun, Benjamin Sapp, Balakrishnan Varadarajan, Yue Shen, Yi Shen, Yuning Chai, Cordelia Schmid, et al. Tnt: Target-driven trajectory prediction. *arXiv preprint arXiv:2008.08294*, 2020.