Trajectory Prediction with Linguistic Representations

Yen-Ling Kuo*,[1,2], Xin Huang[2], Andrei Barbu[2], Stephen G. McGill[1]
Boris Katz[2], John J. Leonard[1,2], Guy Rosman[1]

***Abstract*— Language allows humans to build mental models that interpret what is happening around them resulting in more accurate long-term predictions. We present a novel trajectory prediction model that uses linguistic intermediate representations to forecast trajectories, and is trained using trajectory samples with partially-annotated captions. The model learns the meaning of each of the words without direct per-word supervision. At inference time, it generates a linguistic description of trajectories which captures maneuvers and interactions over an extended time interval. This generated description is used to refine predictions of the trajectories of multiple agents. We train and validate our model on the Argoverse dataset, and demonstrate improved accuracy results in trajectory prediction. In addition, our model is more interpretable: it presents part of its reasoning in plain language as captions, which can aid model development and can aid in building confidence in the model before deploying it.**

I. INTRODUCTION

Predicting future trajectories of agents on the road is indispensable for autonomous driving, allowing a vehicle to plan safe and effective actions. Prediction requires understanding of how agents relate to a variety of concepts, such as map features, other agents, maneuvers, rules, to name a few.

Language allows us to reason about such concepts easily. Humans can use language to richly describe events in the world. These descriptions can involve other agents, actions, goals, and objects in the environment. They also focus our attention on entities, relationships, and properties, cutting through a sea of irrelevant details. Finally, language is also flexible in its treatment of time, as it can seamlessly describe the past and multiple possible futures within a single utterance.

Thus far, models for reasoning about road scenarios have not availed themselves of the added level of abstraction that language provides. A concrete example where language can help is shown in Fig. 1. The ego-vehicle is approaching an intersection while a cyclist is about to pass through the intersection. The car must predict the future trajectory of the cyclist and its own movements in response to that trajectory. The hypothetical futures of the ego-vehicle can be described in sentences such as "I should slow down waiting for the cyclist to cross and then turn left" or "I can turn left because the cyclist is turning right". Having access to the descriptions of two scenarios enables one to quickly check which of them is happening by removing many irrelevant details. This is the power that language brings to trajectory prediction, it abstracts away irrelevant non-actionable information, and focuses the computation on relevant parts such as the temporal order of crossings and interactions with other road agents.

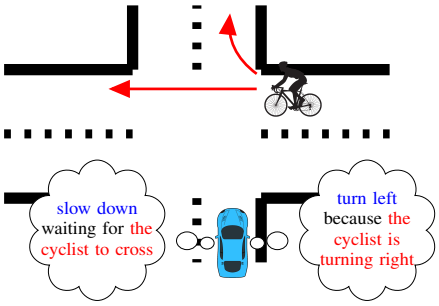


Fig. 1. An illustration of the idea that an ego-car can use language to hypothesize possible futures by applying different affordances of language. The words in blue shows the ego-car's reasoning about its own actions conditioned on the other agent's behavior, shown in red. The words such as "waiting for" and "because" allow the ego-car to reason about the temporal relationship and causal relationship in this interacting scenario.

Taking the notion of language as an abstraction for agents, interactions, and temporal reasoning, we propose a predictor, illustrated in Fig. 2, which leverages a learned linguistic representation to attend to the relevant agent and the relevant parts of descriptions at each time step. We co-train the sentence generator with the attention-based trajectory predictor to generate sentences that hypothesize future trajectories. Every generated word is grounded in the future trajectories of the agents. We train and test the proposed predictor on the Argoverse dataset with a set of synthetic linguistic tokens. Inspecting the attention on each word reveals 1) what concepts are relevant for the current predictions, and 2) what properties or interactions those concepts are based on. This greatly accelerates model building: rather than dealing with the failure of a black box one can discover that a particular word is not well trained. It also provides an auditing mechanism, which will help us to ensure that autonomous cars are making the right decisions for the right reasons, rather than incidentally succeeding on benchmarks, thus risking failure in the real world. Finally, it provides an audit that traces the reasoning of the model which can be helpful when understanding accidents.

This work makes the following contributions.

1) We develop trajectory-prediction datasets augmented with linguistic descriptions.
2) We present a novel predictor that leverages language as one of its internal representations.

[1]Toyota Research Institute, Cambridge, MA 02139, USA
{guy.rosman,stephen.mcgill,john.leonard}@tri.global
[2]CSAIL & CBMM MIT, Cambridge, MA 02139, USA
{ylkuo,huangxin,abarbu,boris}@mit.edu
*Work done as an intern at TRI.

3) We demonstrate how a spatio-temporal attention on top of that language can help trajectory prediction.

II. RELATED WORK

A. Trajectory Prediction with Attention and Structures

Trajectory prediction of road agents has become an important research topic in recent years. Different inputs to trajectory prediction models such as providing past trajectories [1], maps [2–4], and images [5–7] have been considered. Structural priors that encapsulate temporal and inter-agent reasoning have been incorporated, implicitly with attention [8–12] and explicitly via maneuvers and rules [13–15, 7, 16–18] and goals [19, 20]. Language is a structural prior, but it is one that spans all of these topics (naturally and flexibly combining interactions, maneuvers, and goals) and integrates seamlessly with attention. This is not just for the ease of communication but also allows us to express all possible occurrences in the physical world without being confined to only object references or any specific temporal structure.

Similar to attention-based predictors [21, 12], attention plays a key role in our model and guides the trajectory predictions. Although the scope of the attention module is gated by the linguistic descriptions [22–24], agents that don't co-occur in the same description don't need to be jointly attended to. Language guides attention which ultimately removes extraneous agents and improves performance. In addition, the pooled agent states are combined with the word embeddings of descriptions to generate attention over the relevant language tokens that represents sequences of events. Our language-guided attention approach efficiently combines both social and temporal reasoning for trajectories while prior work only considers one of these at a time.

B. Language and Planning

Most prior work that uses language in robotics involves planning to follow natural language commands in various tasks including manipulation [25–27, 24, 28] and navigation [25, 29, 30]. These planners mainly learn from demonstrations paired with input linguistic commands, i.e. learning an action distribution conditioned on linguistic input. Our approach explores another language use, where language is implicit not explicit — no linguistic input is provided; instead language acts as an internal representation for hypotheses about future trajectories. To that end, our approach includes a sentence generator while these planners do not. There are approaches that generate captions for videos [31–33] but they have not yet been considered in planning. The input linguistic commands for planners mostly specify goals and referents, e.g., landmarks, for navigation or locating the objects to interact with. Some planners learn visual attention maps conditioned on language to predict a goal map [28], filter relevant objects [24], or estimate a visitation map [29] for a single agent. However, none of these planners considers language describing interactions with other agents and manners of performing an action, such as quickly or slowly. These linguistic attributes are critical for trajectory predictions. In addition to grounding words to visual maps, our model grounds words to agents' own trajectories and interactions with others.

C. Neuro-symbolic Networks & Architectures

Our approach relates to different uses of neuro-symbolic architectures in other fields such as analysis of images [34–36] and videos [37, 38]. Recent work by Chitnis et al. [39] uses symbolic planning to guide the continuous planning process. These architectures combine neural components with symbolic representations such as programs or domain specific languages. Our approach similarly uses a symbolic component to guide the continuous component, i.e. trajectory prediction, but our approach works not only for a domain specific language but also for natural languages.

III. MODEL

A. Problem Formulation

a) Trajectory Prediction: Given a sequence of the observed states for an agent $\mathbf{s}_P = \{s_{T'+1}, \cdots, s_0\}$, our goal is to predict N possible future trajectories, $\mathbf{s}_F$. Each agent trajectory is a sequence, $\mathbf{s}_F = \{s_1, \cdots, s_T\}$, where t is time and s_t denotes the state of an agent at time t. $t = 0$ corresponds to the last observed time, i.e. the time at which we make the predictions about the future. The agent's past interactions with other agents and map elements are encoded into an embedding h_0 as shown in Fig. 2.

b) Learning Intermediate Representation: In addition to predicting future trajectories given the observed states $\mathbf{s}_P$, we aim to learn to sample N possible linguistic descriptions $Y_{1:M}$ that matches how people describe the agent's future rollouts and use the descriptions to predict trajectories $\mathbf{s}_F$. This linguistic description is a sequence of tokens that describe the potential future states of the agent, $Y_{1:M} = (y_1, \cdots, y_M), y_m \in \mathcal{Y}$, where $\mathcal{Y}$ is the vocabulary of candidate tokens to describe a driving scenario and M is the maximum number of tokens in the description.

This description about the future is often generated recursively from the past context [40] as:

$$\log p(Y_{1:M}|\mathbf{s}_P) = \sum_{m=1}^{M} \log p(y_m|y_{1:m-1}, h_0) \qquad (1)$$

While it is possible to generate exhaustive descriptions for every detail in every scene, we consider descriptions from a single agent's point of view with a length cap that ensures that only the most relevant concepts will be included. As each generated description describes a hypothetical future, we can generate a sample of future trajectory $\mathbf{s}_F^n$ based on each description: $p(\mathbf{s}_F^n|Y_{1:M}^n)$.

B. Model Overview

The overall network structure is shown in Fig. 2. Our model is based on a multi-agent encoder-decoder GAN [9], with an LSTM for each agent. The generator uses an encoder-decoder structure to produce samples of future trajectories. For the latent space representation in the encoder-decoder, in addition to using a regular fixed-dimension tensor output by the encoder, we generate linguistic descriptions to represent

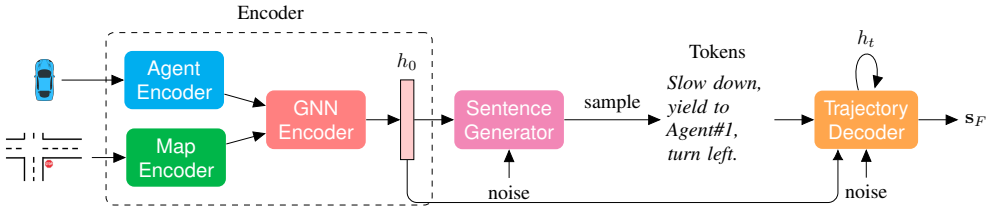


Fig. 2. An overview of the model with language as its intermediate representation. The encoder consists of a model which embeds the observed trajectory of each agent along with the map into h_0. A sentence is predicted from h_0. Together, the sentence and h_0 are used to refine the trajectory using the trajectory decoder; details of which are provided in Fig. 3. The result is a predicted trajectory $\mathbf{s}_F$. We show only one agent in this figure but all agents (up to four agents in our experiment) in the scene are encoded using the same encoder.

the future trajectory which is ultimately decoded by the trajectory decoder. The encoder uses a graph neural network to encode inter-agent interactions and agents' relation to the map. We use a joint random noise vector in sentence generation and the decoder to enable joint probabilistic modeling. The discriminator determines whether the generated trajectory is realistic or not.

C. Encoder

We encode map information with attention over the road graph [3]. This map encoder takes a map input as a set of lane centerlines and uses self-attention to combine the encoded tensors from all lane centerlines. The map encodings are concatenated with the position encoding at each time point and then passed through an LSTM [8] to get the hidden state vector h_0 at the last observed time $t = 0$. This hidden vector encapsulates the agent's observations up until $t = 0$ and is used to generate linguistic descriptions and future trajectories.

D. Language Generation

The sentence generator is an LSTM. It takes each agent's encoded hidden state vector h_0 along with a noise vector as input, to make M-step rollouts to generate a description of length M. When there are multiple sentences in the description, we use two special tokens <bos> and <eos> to indicate the beginning and the end of a sentence and <pad> to pad the sequence if the description has less than M tokens. At each rollout step, the LSTM outputs a distribution over the vocabulary and we leverage Gumbel-softmax [41] to sample a token from the predicted distribution.

E. Language Conditioned Decoder

The decoder for each agent takes the language token sequence and embeds each of the tokens with an embedding layer. We co-train the word embedding with the decoder. The sentence description contains words that refer to other agents and the structure to relate the dependency of events. We design the attention mechanisms to capture these language properties as illustrated in Fig. 3.

a) Agent Attention: When generating descriptions, we refer to an agent by its index number in the scenario. Inside the decoder, when we see a word embedding y_m that corresponds to an agent index a, we use an MLP layer with softmax output to generate attention over the agent to pool the relevant agent

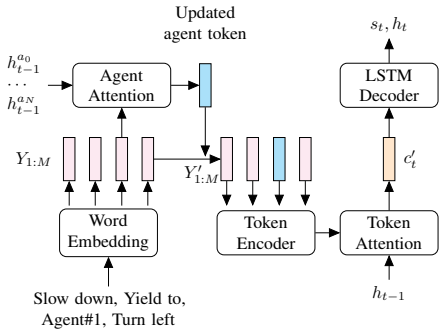


Fig. 3. Attention computation inside the trajectory decoder block in Fig. 2. We use the tokens that refer to other agents to pool the agents' states first and then employ attention over tokens to determine the tokens that are relevant to the current time point to make a prediction.

states in the previous time step $t-1$ to update the word embedding:

$$y'_m = Softmax(MLP(y_m)) \cdot h^{\mathcal{A}}_{t-1} \tag{2}$$

where $h^{\mathcal{A}}_{t-1}$ is the tensor of decoder hidden states of all agents in the scene. This update allows us to ground the tokens to the relevant agents' states so that the decoded trajectories can respond to the change of other agents.

b) Token Attention: To determine which tokens are relevant to the trajectory at the current time step, we consider the attention structure that has been used widely in machine translation [42] to generate the attention over the input tokens for the decoder. We re-encode the word embeddings at every time step using an LSTM token encoder since the word embeddings are updated with relevant agents' state. The token attention module takes the last output c_t from the LSTM token encoder along with the current agent's trajectory decoder state in the previous time step h_{t-1} to generate attention over tokens as in [42]. This attention is used to pool the word embeddings of the generated language tokens to provide the context vector c'_t to decode the trajectory:

$$c'_t = Attention(c_t, h_{t-1}) \cdot Y'_{1:M} \tag{3}$$

Finally, this context vector is concatenated with the agent's state to decode the trajectory s_t and update the agent's decoder state h_t at time t using an LSTM.

F. Training the Language-based Predictor

We train the generator and the discriminator end-to-end. When generating trajectories $\hat{\mathbf{s}}_F$, similar to [9], we compute

Time horizon	2s past, 3s future
Total # of trajectories	7,753,060
# of trajectories with descriptions	1,168,963
# of descriptions referring to other agents	272,877
Average number of tokens per description	3.17
Vocabulary size	16

Fig. 4. The overall statistics of the dataset augmented with language descriptions. We do not augment all trajectories with captions because some trajectories are too short or the agent does not move.

the Minimum over N (MoN) losses using the average displacement error (ADE) to encourage the model to cover the ground-truth positions:

$$\mathcal{L}_{MoN} = \min_{n}(ADE(\hat{\mathbf{s}}_F)^{(n)}) \tag{4}$$

When the ground-truth description is available, we augment the trajectory generation loss by a cross entropy loss for the generated descriptions:

$$\mathcal{L}_{lang} = \frac{1}{M}\sum_{m=0}^{M} CrossEntropy(y_m, \hat{y}_m) \tag{5}$$

The total loss for the generator is a combination of the losses above:

$$\mathcal{L}_G = \lambda_1 \mathcal{L}_{MoN} + \lambda_2 \mathcal{L}_{lang} \tag{6}$$

Note that this loss structure affords training of the generator even when the ground-truth descriptions do not cover all trajectories, and merely uses descriptions as an additional supervision. This is especially true for multiple agents – all agents share the same latent structure, but only one agent has a ground-truth description loss associated with it. The parameter sharing ensures that other agents are also explainable. Finally, the discriminator uses a binary cross entropy loss to classify the generated trajectories and the ground-truth trajectories.

IV. EXPERIMENTS

To train and evaluate the model with linguistic representations, we describe the procedure to augment the existing datasets for training. Then we describe the baseline models to test the ability to predict future trajectories and demonstrate the scenarios where language can improve predictions.

A. Dataset

There is no existing motion prediction dataset that contains annotated language descriptions. To train a predictor that utilizes the language as a latent representation, we first produce a dataset that contains trajectory and description pairs. We augment the Argoverse v1.1 [43] with synthetic language descriptions. The statistics of the training dataset are summarized in Table 4. We consider trajectories from all agents in a scenario as candidates for annotation with linguistic descriptions, not just the trajectories from the ego vehicles.

We generate the description for each trajectory by applying a set of predefined filters to identify meaningful parts of trajectories. Augmenting with synthetic descriptions gives us higher language coverage in trajectories than is feasible with human annotations. However, the token and structure are bounded by the types of filters we employ.

We generate tokens `MoveFast`, `MoveSlow`, `Stop`, `TurnLeft`, `TurnRight`, `SpeedUp`, `SlowDown` based on the velocity, angular velocity, and acceleration of trajectories. We generate tokens for lane changes (`LaneKeep`, `LaneChangeLeft`, `LaneChangeRight`) based on the change of the closest lane centerlines. Two tokens involving other agents are generated: `Follow Agent#k` and `Yield Agent#k`, based on intersection tests and overlaps. In the case of `Yield`, the target agent needs to slow down and arrive at the intersecting area later than the agent it yields to.

The tokens are organized in a sequence according to their temporal order. If there are multiple tokens active at the same time, we randomly select one of them, to imitate how humans selectively describe some properties of trajectories instead of providing every minute detail. We do not label trajectories if they are too short or if the generated tokens are oscillating between tokens with opposite meanings.

B. Experiment Setup

1) Baselines: As described in the related work section, no existing model is trained with language descriptions or uses language as intermediate representations for trajectory predictions. We compare our model to baselines and perform two ablations. These baseline models all use the same encoder as described here but implement different decoders. The first baseline is a vanilla LSTM decoder without employing any structure, modules, or attention mechanism. The second baseline is an LSTM decoder with multi-head attention [12]. These attention heads attend to other agents' LSTM states at time $t-1$. This represents the predictors that model the interactions between agents explicitly and is the model closest to ours, with standard methods of capturing social interactions [9, 11], map information [3], and attention structures [12].

Additionally, we consider two ablations. *Ours (no attention)* is our model without both attention modules in the predictor. The decoder LSTM directly concatenates the last output from the token encoder to generate trajectories. This shows the performance of language embeddings but doesn't consider how language relates to trajectories. *Ours (no agent attention)* ablates the agent's attention but keeps the token attention. This ablation doesn't update the token that refers to other agents with the pooled agent states. This shows the performance when only the temporal aspects of language are considered.

2) Model Details: The encoder LSTM has a hidden dimension size of 32 and output dimension of 32. In the decoder, we use a dimension of 4 for the token generator LSTM hidden state and the token embedding. The attention modules are one-layer MLPs with hidden dimension 4. We co-train the word embeddings with the rest of the network. We use 0.2 dropout for the token generation, encoder, and attention modules. For the multi-head attention baseline, we follow Mercat et al. [12] in using six 10-dimensional attention heads. The loss coefficients are selected to be 1. The model is optimized using Adam and trained on an NVIDIA Tesla V100 GPU with learning rate 1e-3 and batch size 32. At every training epoch, we rebalance the training examples that

Model	ADE	FDE	Entropy
LSTM decoder	0.69	1.46	2.66
Multihead attention decoder	0.67	1.39	2.51
Ours (no attention)	0.67	1.44	2.61
Ours (no agent attention)	0.67	1.39	2.56
Ours	0.65	1.35	2.41

Fig. 5. MoN average displacement errors (ADE) and final displacement errors (FDE) of our method and baseline models with N=6 samples on the Argoverse dataset predicting 3 seconds in the future.

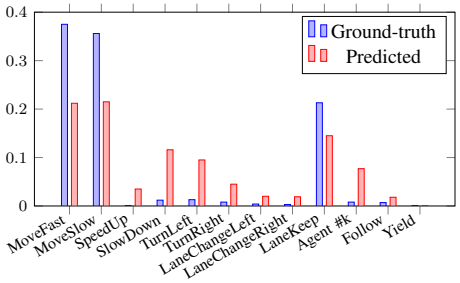


Fig. 6. Distributions of ground-truth and generated tokens in the validation set. In our experiment, k={1,2,3,4} indicating the sentence can refer to up to four agents. Note that we generate language tokens for all trajectories but only have ground-truth for the confident examples, resulting in gaps between the predicted and ground-truth distributions.

contain and don't contain linguistic descriptions to make sure the model receives smooth gradients from the language data.

3) Metrics: We measure the correctness of the predicted language by computing its recall, i.e. the fraction of ground-truth tokens which are predicted. We choose recall because the additional predicted tokens may not be wrong, but are just missing in the ground-truth (the predefined filters might not have identified the maneuvers or interactions). The prediction performance of trajectories is evaluated with minimum average displacement error (ADE) and final displacement error (FDE) [9] in meters. The ADE measures the average distance between the predicted and the ground-truth trajectories. The FDE measures the final distance at the selected prediction horizons. We use minimum-of-N (MoN, variety loss) with $N = 6$ unless stated otherwise.

In order to compare how language improves trajectories at long time horizons, we report displacement errors at 1 and 3 seconds. We produce results with the number of samples, N, set to 6. We also compute the entropy of the predicted trajectory samples to understand how language influence the trajectory choices to more relevant ones. Similar to Trajectron++ [17], we perform Gaussian kernel density estimation on the samples to compute the entropy of the samples. For baseline models, the entropy is $H(\mathbf{s}_F)$; for the language-based models, this is the conditional entropy $H(\mathbf{s}_F|Y_{1:M})$.

C. Results

1) Quantitative Results: The recall of the generated linguistic descriptions is 85.7%. Fig. 6 shows the distribution of ground-truth tokens along with the generated tokens in the validation set. Even though the ground-truth tokens mostly cover simple maneuvers such as speed, our training regime identify the trajectories with rarer tokens while mimicking the ground-truth distributions.

Fig. 5 summarizes the displacement errors and entropy for Argoverse at 3 seconds. Our model produces the lowest FDE and ADE compared to baselines. Ablation of any attention module shows poor performance and higher entropy as the trajectory decoder does not know what the encoded token vectors mean.

We also computed the FDE and ADE at 1 second. All baseline models have 0.29 FDE and 0.40 ADE. Our model has 0.28 FDE and 0.40 ADE. Comparing the displacement errors at 1 vs. 3 seconds confirms that the influence of language is more prominent at longer horizons. We observe the reduction of entropy compared to the baselines. In order to gauge if this reduction is from the information stored in the generated sentences, we compute the information gain based on the entropy of predictions conditioned on the generated descriptions and all-padded descriptions. The average information gain for language conditioned predictions is 0.48 bits. This is a significant influence as many trajectories in Argoverse have one or two tokens in the description. Furthermore, we did not have the linguistic representation bottlenecked [44, 45] in the predictor, which may lead to underestimating the mutual information.

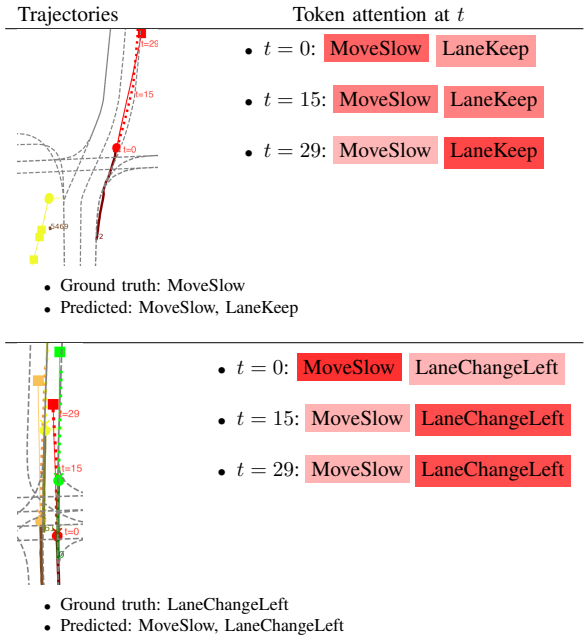


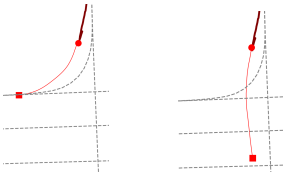
Fig. 7. Example predicted trajectories and linguistic descriptions. Red lines on the left correspond to the target agent. The dark red line is the observed past trajectory. The solid red line is the predicted future sample and the dotted red line is the ground-truth trajectory. We also show the attention weights of the generated language tokens at different time steps on the right. Tokens with darker background are the ones attended to more when generating trajectory points at that time step. The changes in the background color, i.e. in attention, indicates a transition between maneuvers.

2) Qualitative Results: In Fig. 7, we present qualitative examples that demonstrate the effectiveness of linguistic representations to predict trajectories. We visualize the attention weights for the predicted tokens at different time steps (start, middle, and end of the trajectory) to show which tokens are in the focus as trajectories are generated.

This visualization also shows the utility of attention for understanding the inferences of the model. The generated descriptions may not match all the ground-truth tokens but that is not necessarily a prediction error. For example, the first row in Fig. 7 only has `MoveSlow` as ground truth while the predicted tokens include `LaneKeep`. This is because we only include the most confident tokens in the description and sample only one token if tokens overlap in time. The model is able to recover missing tokens during training.

We can also directly modify the generated descriptions to see how changing this internal reasoning affects the predicted trajectories. This allows us to reveal the meaning of different tokens. In Fig. 8, we demonstrate that when we observe the same past trajectory but change the tokens from "turn left" to "turn right", the predicted trajectories of the agent change accordingly.

This representation also provides a mechanism to introspect the predictor's failures; see Fig. 9 for example predicted tokens and trajectories in this case. When there is a prediction error, this is evident from the predicted description. We can also find the relevant parts of trajectories that correspond to different errors by inspecting the attention weights on different tokens; see Fig. 9(b) for an example of interpreting interactions between two agents. In the example in Fig. 9(a), the cause of the error can be identified from the generated description *TurnLeft*. Practically, this can enable one to collect more targeted training examples. An end-to-end model without an intermediate representation that can be inspected and understood by humans would not provide such clear guidance about what went awry and how to fix it.

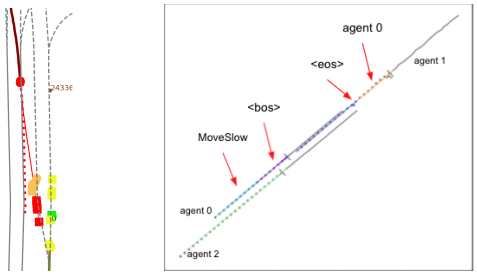


(a) turn right (b) turn left

Fig. 8. The trajectory changes after changing a token in the linguistic description: turn right vs. turn left. The dark red line is the observed past trajectory and the thin red line is the generated future trajectories.

V. EXTENDING TO NATURAL LANGUAGE

While the predictors shown thus far are based on a synthetic language, we want to understand if it is feasible to train the model with natural language. We sampled 40,000 trajectories from the Waymo dataset [46] and had humans annotate them with captions. We observed that the annotated sentences use a much more diverse vocabulary than the synthetic language. For example, humans use "drive after" or "drive behind" to describe following another agent. These captions also contain temporal and spatial relationships such as "before", "left", and "right". When processing natural language, we employ an English tokenizer to preprocess the human-generated captions and then embed the predicted tokens using GloVe [47]. The rest of the network is unchanged aside from using a hidden dimension of 16 for the language generator and encoder



(a) Example failure
- Ground truth: LaneKeep
- Predicted: TurnLeft

(b) Example token association: The paths for agent 0 and 1 are colored and marked with the most relevant tokens showing that agent 0 slowed down so agent 1 needed to attend to it.

Fig. 9. Example of introspection using language. (a) The error in the predicted trajectories is reflected in the generated sentence. The dotted red line is the ground truth. (b) The most relevant tokens at different part of trajectories.

The agent starts to slow down.	The agent slows down behind an agent.
The agent turns behind an agent.	The agent slows down because light.
The agent crosses intersection fast.	The agent slows down waiting an agent.
The agent halts on crosswalk.	The agent moves fast passing an agent.
The agent passing bumps slow.	The agent takes left while crossing.

Fig. 10. Example of predicted sentences in the Waymo dataset.

modules because the space of possible words, i.e. tokens, has increased dramatically. This extension generates naturalistic sentences for the predictor as in Fig. 10.

VI. CONCLUSION & FUTURE WORK

We propose a new trajectory prediction model that employs language as an intermediate representation. Our method generates interpretable linguistic tokens and is sensitive to these tokens. The attention weights associated with each token is meaningful and can be used to understand what each part of a linguistic description is referring to. While we don't explore this here, the learned language model can provide an interface by which humans can include their preferences into the reasoning of the car. Such interfaces can also provide a level of comfort to users by creating meaningful explanations for the behavior of the car.

Our proposed model is based on single agent's observations and view. For predicting multiple agents in a scene, disagreements between the linguistic descriptions made by different agents, the Rashomon effect [48], may arise when predicting multiagent interactions. Contrasting our linguistic descriptions with whole-scene multimodal language-vision representations could be an interesting step forward.

ACKNOWLEDGMENT

This work was supported by the Center for Brains, Minds and Machines, NSF STC award 1231216, the MIT CSAIL Systems that Learn Initiative, the CBMM-Siemens Graduate Fellowship, the DARPA Artificial Social Intelligence for Successful Teams (ASIST) program, the United States Air Force Research Laboratory and United States Air Force Artificial Intelligence Accelerator under Cooperative Agreement Number FA8750-19-2-1000, and the Office of Naval Research under Award Number N00014-20-1-2589 and Award Number N00014-20-1-2643. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

REFERENCES

[1] C. Schöller, V. Aravantinos, F. Lay, and A. Knoll, "What the constant velocity model can teach us about pedestrian motion prediction," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1696–1703, 2020.

[2] I. Gilitschenski, G. Rosman, A. Gupta, S. Karaman, and D. Rus, "Deep context maps: Agent trajectory prediction using location-specific latent maps," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5097–5104, 2020.

[3] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid, "Vectornet: Encoding HD maps and agent dynamics from vectorized representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 525–11 533.

[4] M. Liang, B. Yang, R. Hu, Y. Chen, R. Liao, S. Feng, and R. Urtasun, "Learning lane graph representations for motion forecasting," in *European Conference on Computer Vision*, 2020, pp. 541–556.

[5] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[6] A. Amini, G. Rosman, S. Karaman, and D. Rus, "Variational end-to-end navigation and localization," in *2019 IEEE International Conference on Robotics and Automation*, 2019, pp. 8958–8964.

[7] X. Huang, S. G. McGill, B. C. Williams, L. Fletcher, and G. Rosman, "Uncertainty-aware driver trajectory prediction at urban intersections," in *2019 IEEE International Conference on Robotics and Automation*, 2019, pp. 9718–9724.

[8] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 961–971.

[9] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: Socially acceptable trajectories with generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2255–2264.

[10] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese, "Sophie: An attentive gan for predicting paths compliant to social and physical constraints," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1349–1358.

[11] J. Amirian, J.-B. Hayet, and J. Pettré, "Social ways: Learning multi-modal distributions of pedestrian trajectories with GANs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[12] J. Mercat, T. Gilles, N. El Zoghby, G. Sandou, D. Beauvois, and G. P. Gil, "Multi-head attention for multi-modal joint vehicle motion forecasting," in *2020 IEEE International Conference on Robotics and Automation*, 2020, pp. 9638–9644.

[13] N. Deo and M. M. Trivedi, "Multi-modal trajectory prediction of surrounding vehicles with maneuver based LSTMs," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1179–1184.

[14] X. Huang, S. G. McGill, J. A. DeCastro, L. Fletcher, J. J. Leonard, B. C. Williams, and G. Rosman, "DiversityGAN: Diversity-aware vehicle motion prediction via latent semantic sampling," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5089–5096, 2020.

[15] X. Li, G. Rosman, I. Gilitschenski, C.-I. Vasile, J. A. DeCastro, S. Karaman, and D. Rus, "Vehicle trajectory prediction using generative adversarial network with temporal logic syntax tree features," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3459–3466, 2021.

[16] C. Tang and R. R. Salakhutdinov, "Multiple futures prediction," *Advances in Neural Information Processing Systems*, vol. 32, pp. 15 424–15 434, 2019.

[17] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in *ECCV 2020*, 2020, pp. 683–700.

[18] X. Huang, G. Rosman, I. Gilitschenski, A. Jasour, S. G. McGill, J. J. Leonard, and B. C. Williams, "HYPER: Learned hybrid trajectory prediction via factored inference and adaptive sampling," in *2022 IEEE International Conference on Robotics and Automation*, 2022.

[19] N. Rhinehart, R. McAllister, K. Kitani, and S. Levine, "Precog: Prediction conditioned on goals in visual multi-agent settings," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2821–2830.

[20] H. Zhao, J. Gao, T. Lan, C. Sun, B. Sapp, B. Varadarajan, Y. Shen, Y. Shen, Y. Chai, C. Schmid, C. Li, and D. Anguelov, "TNT: Target-driven trajectory prediction," in *Conference on Robot Learning*, 2020.

[21] J. Li, F. Yang, H. Ma, S. Malla, M. Tomizuka, and C. Choi, "Rain: Reinforced hybrid attention inference network for motion forecasting," *arXiv preprint arXiv:2108.01316*, 2021.

[22] J. Mun, M. Cho, and B. Han, "Text-guided attention model for image captioning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[23] P. Wang, Q. Wu, J. Cao, C. Shen, L. Gao, and A. v. d. Hengel, "Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1960–1968.

[24] Y.-L. Kuo, B. Katz, and A. Barbu, "Deep compositional robotic planners that follow natural language commands," in *2020 IEEE International Conference on Robotics and Automation*, 2020, pp. 4906–4912.

[25] S. Tellex, T. Kollar, S. Dickerson, M. Walter, A. Banerjee, S. Teller, and N. Roy, "Understanding natural language commands for robotic navigation and mobile manipulation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 25, no. 1, 2011.

[26] R. Paul, A. Barbu, S. Felshin, B. Katz, and N. Roy, "Temporal grounding graphs for language understanding with accrued visual-linguistic context," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2017, pp. 4506–4514.

[27] C. Paxton, Y. Bisk, J. Thomason, A. Byravan, and D. Foxl, "Prospection: Interpretable plans from language by predicting the future," in *2019 IEEE International Conference on Robotics and Automation*, 2019, pp. 6942–6948.

[28] S. Stepputtis, J. Campbell, M. Phielipp, S. Lee, C. Baral, and H. Ben Amor, "Language-conditioned imitation learning for robot manipulation tasks," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 13 139–13 150.

[29] V. Blukis, D. Misra, R. A. Knepper, and Y. Artzi, "Mapping navigation instructions to continuous control actions with position-visitation prediction," in *Conference on Robot Learning*, 2018, pp. 505–518.

[30] A. Magassouba, K. Sugiura, and H. Kawai, "CrossMap transformer: A crossmodal masked path transformer using double back-translation for vision-and-language navigation," *arXiv preprint arXiv:2103.00852*, 2021.

[31] A. Barbu, A. Bridge, Z. Burchill, D. Coroian, S. Dickinson, S. Fidler, A. Michaux, S. Mussman, S. Narayanaswamy, D. Salvi, L. Schmidt, J. Shangguan, J. M. Siskind, J. Waggoner, S. Wang, J. Wei, Y. Yin, and Z. Zhang, "Video in sentences out," in *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, 2012, pp. 102–112.

[32] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu, "Video paragraph captioning using hierarchical recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4584–4593.

[33] J. Chen, Y. Pan, Y. Li, T. Yao, H. Chao, and T. Mei, "Temporal deformable convolutional encoder-decoder networks for video captioning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 8167–8174.

[34] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3128–3137.

[35] L. A. Hendricks, S. Venugopalan, M. Rohrbach, R. Mooney, K. Saenko, and T. Darrell, "Deep compositional captioning: Describing novel object categories without paired training data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1–10.

[36] S. Amizadeh, H. Palangi, A. Polozov, Y. Huang, and K. Koishida, "Neuro-symbolic visual reasoning: Disentangling "Visual" from "Reasoning"," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119, 2020, pp. 279–290.

[37] S. Kulal, J. Mao, A. Aiken, and J. Wu, "Hierarchical motion understanding via motion programs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6568–6576.

[38] K. Yi*, C. Gan*, Y. Li, P. Kohli, J. Wu, A. Torralba, and J. B. Tenenbaum, "CLEVRER: Collision events for video representation and reasoning," in *International Conference on Learning Representations*, 2020.

[39] R. Chitnis, T. Silver, J. B. Tenenbaum, T. Lozano-Perez, and L. P. Kaelbling, "Learning neuro-symbolic relational transition models for bilevel planning," *arXiv preprint arXiv:2105.14074*, 2021.

[40] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning

with neural networks," in *Advances in Neural Information Processing Systems*, 2014, pp. 3104–3112.

[41] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," in *International Conference on Learning Representations*, 2017.

[42] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1412–1421. [Online]. Available: https://aclanthology.org/D15-1166

[43] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan *et al.*, "Argoverse: 3d tracking and forecasting with rich maps," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8748–8757.

[44] G. Chechik, A. Globerson, N. Tishby, Y. Weiss, and P. Dayan, "Information bottleneck for gaussian variables." *Journal of machine learning research*, vol. 6, no. 1, 2005.

[45] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016, pp. 2180–2188.

[46] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. Qi, Y. Zhou, Z. Yang, A. Chouard, P. Sun, J. Ngiam, V. Vasudevan, A. McCauley, J. Shlens, and D. Anguelov, "Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset," *arXiv preprint arXiv:2104.10133*, 2021.

[47] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1532–1543.

[48] K. G. Heider, "The Rashomon effect: When ethnographers disagree," *American Anthropologist*, vol. 90, no. 1, pp. 73–81, 1988.