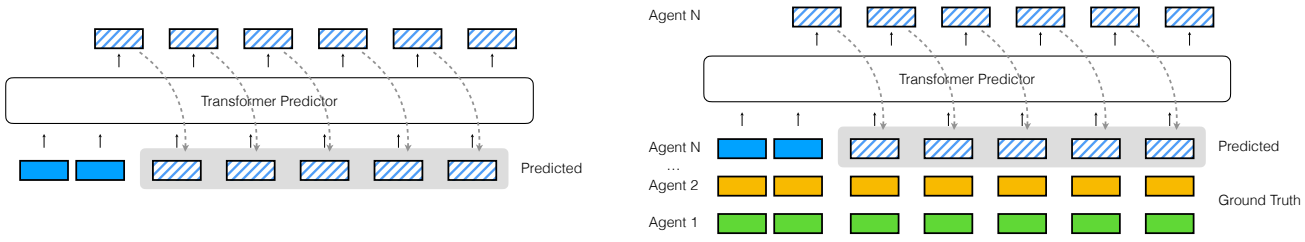
Poly-Autoregressive Prediction for Modeling Interactions

Neerja Thakkar[1] Tara Sadjadpour[1] Jathushan Rajasegeran[1] Shiry Ginosar[2,3] Jitendra Malik[1]
[1]UC Berkeley, [2]Toyota Technical Institute at Chicago, [3]Google DeepMind



(a) Autoregression (AR)

(b) Poly-autoregression (PAR)

Figure 1. Inference for (a) autoregressive (AR) models and (b) our proposed poly-autoregressive (PAR) model. Solid indicates ground-truth tokens which represent a tracked data modality such as action or 6DOF pose; striped represents predicted output tokens. Color denotes agent identity. Compared to AR models, the PAR model takes other agents' tokens as inputs when making a prediction for the next timestep.

Abstract

We introduce a simple framework for predicting the behavior of an agent in multi-agent settings. In contrast to autoregressive (AR) tasks, such as language processing, our focus is on scenarios with multiple agents whose interactions are shaped by physical constraints and internal motivations. To this end, we propose Poly-Autoregressive (PAR) modeling, which forecasts an ego agent's future behavior by reasoning about the ego agent's state history and the past and current states of other interacting agents. At its core, PAR represents the behavior of all agents as a sequence of tokens, each representing an agent's state at a specific timestep. With minimal data pre-processing changes, we show that PAR can be applied to three different problems: human action forecasting in social situations, trajectory prediction for autonomous vehicles, and object pose forecasting during hand-object interaction. Using a small proof-of-concept transformer backbone, PAR outperforms AR across these three scenarios.

1. Introduction

Large language models (LLMs) have been very successful with natural language processing (NLP) tasks, which require accurate reasoning over relationships words have within a body of text. A key component of LLMs is autoregressive (AR) modeling, where each word token is predicted based on a sequence of preceding word tokens. Building on the success of AR modeling in the NLP community, this work focuses on modeling the dynamic relationships between agents and entities in everyday interactions that occur in the physical world, such as social interactions, driving, and hand-object interactions.

Unlike language which is structured through grammar and semantics, interactions in the physical world are dictated by both the laws of physics (*e.g.* how a hand grasps an object) and the internal state of each agent (*e.g.* the trajectory an agent chooses to move a grasped object in) , a latent variable that we know nothing about. Furthermore, as opposed to text where each word follows another unidimensionally, the states of multiple agents are changing simultaneously. For example, in social situations, the history of a single person's past states does not alone determine the dynamics of their future states; we also need to consider the states of other agents. We argue, therefore, that AR modeling alone is insufficient.

In this paper, we introduce poly-autoregressive (PAR) modeling, a simple unifying approach to model the influence of other agents and entities on one's behavior. We model behavior as a temporal sequence of states and predict an ego agent's future behavior conditioned on the history of behavior of the ego agent as well as the rest of the agents. By considering other agents' behaviors, we demonstrate that our approach significantly improves upon the ill-posed problem

of single-agent prediction in interactive settings.

The PAR framework uses a transformer-based model for next-token prediction. Transformers have shown great success in language modeling and naturally lend themselves to predicting behavior over time. In an interaction scenario of N agents, our model predicts the future behavior of an ego (N^{th}) agent conditioned on its past behavior and the behavior of the other $N-1$ non-ego agents (see Figure 1b). Each prediction task may model a different behavior modality of interest, e.g. actions for social action prediction, or 6DoF pose for object forecasting in hand-object interactions. We apply PAR to three different case studies of common real-world interactions:

1. **Social action prediction.** We test our method on the AVA benchmark [19] for action forecasting. By incorporating both the ego and another agent using PAR, we get a overall $+1.9$ absolute mAP gain over AR, which only models the ego agent to predict its future behavior, and an absolute $+3.5$ mAP gain on 2-person interaction classes.
2. **Trajectory prediction for autonomous vehicles.** When forecasting future xy locations of an ego vehicle, incorporating the locations of neighboring vehicles with PAR outperforms AR, which only uses the ego vehicle's preceding trajectory as input. Specifically, on the nuScenes dataset [5], PAR outperforms AR with a relative improvement of 6.3% ADE and 6.4% FDE.
3. **6DoF object pose forecasting during hand-object interaction.** We use the DexYCB dataset [8], where we treat the object as the ego agent and the hand as the interacting agent. While PAR integrates that hand's 3D location and object pose history, AR only uses the object's pose history to predict the object's 6DoF pose. PAR outperforms AR with relative improvements of 8.9% and 41.0% for the rotation and translation predictions, respectively.

In all these settings, we find that incorporating the behavior of other agents in the scene improves predictions of the ego agent's behavior. All of these problems are modeled via the same simple PAR framework and implemented using the same proof-of-concept 4 million parameter transformer *without any modifications to the base framework or architecture*, only to data pre-processing and choice of tokenization. We also provide an example of a simple way to build on our architecture through a location positional encoding, Sec. 5 .

The primary contribution of this work is a versatile framework that can be applied to a diverse range of settings, without modifications aside from domain-specific data processing. Our results suggest that PAR provides a simple formulation that, with a more complex transformer backbone and larger datasets, could enhance prediction of diverse multi-agent interactions across various problem domains. To facilitate further exploration and development, we have released our code, which contains the building blocks to use PAR for modeling other types of multi-agent interactions.

2. Related Work

Autoregressive models. Autoregressive modeling has a rich history in information theory and deep learning, tracing back to Shannon's paper on language prediction [45] and Attneave's study on visual perception [2]. These foundational works laid the groundwork for modern applications in deep learning, including [25], which revisits neural autoregressive models and [18, 50], which explore continuous-valued modeling. [53] developed PixelRNN and PixelCNN to autoregressively generate images one pixel at a time using RNN and CNN architectures, respectively.

The development of the transformer model [54] spurred progress in computer vision with the image transformer [36] and the vision transformer [14], and autoregressive models [9, 41] and more notably in NLP, where the GPT family of models [4, 37, 38] has demonstrated the power of large-scale unsupervised autoregressive pre-training. Recent research has focused on multimodal learning, exemplified by the Flamingo [1] or LlaVa [28] models, which combine vision and language processing capabilities, illustrating the versatility of autoregressive models across various domains in artificial intelligence. While these approaches operate on image patches and word tokens, we operate on symbolic representations extracted from in-the-wild videos showing natural interactions. A recent approach [39] frames humanoid locomotion as an autoregressive next-token prediction task that operates on two types of continuous tokens: observations and actions. This approach projects continuous tokens to the hidden dimension and uses a shifted loss, similar to the next-timestep prediction proposed in our framework.

Multi-agent regressive models. Several prior works addressed modeling specific multi-agent problems via regressive models as one-off case studies. We introduce the PAR framework to unify these efforts into a single cohesive framework. Many behavior prediction works focus on two agents engaging in social interaction, whether it be dyadic communication [32–34] or social dance [30, 46]. These studies primarily tackle the challenge of predicting the state of an interacting partner (Person B) based on the input from Person A's state, sometimes extending predictions into the future [20, 30]. While earlier works used architectures such as variational RNNs [3], recent works have predominantly adopted transformer architectures for social interaction modeling [10, 20, 32, 33, 46], with some works exploring diffusion [27], or diffusion with attention [17]. Our PAR framework focuses on transformer models.

Works encompassed by the PAR framework extend beyond human social interaction. Many multi-agent human or car trajectory prediction approaches use autoregressive prediction. For instance, MotionLM [44] utilizes a transformer decoder that processes multi-agent tokens, incorporating a

learned agent ID embedding. This methodology informs our approach across all our case studies. *Critically, in contrast to all prior multi-agent regressive works that designed solutions to address specific applications, we demonstrate that we can unify a diverse set of multi-agent regressive problems under a single PAR framework.* See Appendix Sec. 9 for case-study-specific works.

3. Poly-Autoregressive Modeling

Our goal is to model the behavior of an agent or entity while taking into account any other agents it interacts with, if any. To evaluate the performance of our model in capturing interaction dynamics, we predict the agent's future behavior and compare it against ground-truth data.

We define the following task: *In an interaction comprised of N agents, given the observed past states of the $N-1$ interacting agents, and the observed or previously-predicted past states of the N^{th} ego agent, predict the future states of the N^{th} ego agent.*

We define a transformer-based poly-autoregressive (PAR) predictor, $\mathcal{P}$, that learns to model temporally long-range interactions in the input sequence. The inputs to the predictor are the past states of the N interacting agents, and its output is the predicted future state of the N^{th} ego agent.

3.1. Problem Definition

Let $\mathbf{S} = \{\mathbf{s}_i\}_{i=1}^{T}$ be a temporal sequence of agent states, $\mathbf{s}_i$. We use $\mathbf{S}^N$ and $\mathbf{S}^{1:N-1}$ to denote the temporal sequences of states of the N_{th} agent and of the other $N-1$ agents, respectively. For each timestep $t \in [t_\pi, T]$, where $t_\pi \in [1, T]$ is the time we start predicting, we take as input all other $N-1$ agents' past observed state sequences $\mathbf{S}_{1:t-1}^{1:N-1}$ along with the N_{th} agent's past observed states up to t_π, $\mathbf{S}_{1:t_\pi}^{N}$, and any of its previously predicted past states $\hat{\mathbf{S}}_{t_\pi+1:t-1}^{N}$, if available (see Fig. 1). Our predictor, $\mathcal{P}$, then *poly-autoregressively* predicts the N_{th} agent's future states one time-step at a time:

$$\hat{\mathbf{s}}_t^N = \mathcal{P}(\mathbf{S}_{1:t-1}^{1:N-1}, \mathbf{S}_{1:t_\pi}^{N}, \hat{\mathbf{S}}_{t_\pi+1:t-1}^{N}). \quad (1)$$

$\mathcal{P}$ learns to model the distribution over the next timestep of the N_{th} agent's states, given all other agents' states:

$$p(\hat{\mathbf{s}}_t^N | \mathbf{S}_{1:t-1}^{1:N-1}, \mathbf{S}_{1:t-1}^{N}). \quad (2)$$

While we provide the observed ground truth states of other agents at inference, during training, we jointly maximize the likelihood of all N agents by computing losses on their future state predictions.

We train the predictor by maximizing the likelihood of the target state y at time t:

$$\mathcal{L}_{\mathcal{P}} = E_{y \sim p(y)}[-\log(p(\mathbf{s}_t^N))],$$

where the target state y at t is computed from the N_{th} agent ground truth future state.

3.2. The Poly-Autoregressive Framework

We address the problem of forecasting the future states of an agent (from time t to T) in a data-driven way, given a temporal sequence of past states (from time 1 to $t-1$). We assume that our agent has some feature, or a set of features, of interest in a video (e.g., 3D pose) that we can tokenize. We predict the future states of the agent in terms of this tokenized feature (or set of), where we use one token (or set of tokens) per time step. The predicted tokens can be discrete (i.e., an index into a feature codebook) or continuous (i.e., a vector of one or more continuous values). The loss ℓ will depend on the problem's specifics and the type of token used. To train the model to predict the future, we rely on all the interaction dynamics of length T in our training dataset as ground truth examples.

As a baseline, we consider the **single-agent autoregressive (AR)** paradigm, where a transformer is trained to perform next-token-prediction with teacher forcing. AR uses greedy sampling to generate sequences at inference time, predicting one next token at a time (Fig. 1(a)).

In contrast, our **multi-agent poly-autoregressive (PAR)** framework considers the other $N-1$ agents in the scene when predicting the future state of the Nth agent. In this setup, we tokenize the features of interest of all N agents, yielding N tokens at each timestep for a total of $N*T$ tokens. In practice, we operate on a flattened sequence of $N*T$ tokens. Instead of using the AR training procedure in this multi-agent case (as in Fig. 3a), we jointly model the N agents at each timestep by introducing the following features to our PAR framework.

Next-timestep prediction. A standard AR model predicts the next token. Given the flattened sequence of $N*T$ tokens our model operates on, next token prediction would take as input an agent k at timestep t and predict agent $k+1$'s state at the same timestep t (as in Fig. 3a). However, our goal is to predict the input agent k's future state at time $t+1$. Therefore, we perform *same-agent next-timestep* prediction rather than next-token prediction (see Fig. 3b for an illustration of same-agent next-timestep at training).

Learned agent identity embedding. When giving a model information corresponding to multiple agents, the model can benefit from knowing which token corresponds to which agent. We give the model this information with a learned agent ID embedding.

Joint training. We train the model to jointly predict the future of all agents by computing a loss on the predicted tokens of all agents (Fig. 3b). Please refer to Section 3.1 for our inference paradigm.

3.3. Task-Specific Considerations

Our simple PAR approach unifies diverse problems under a single framework and architecture without any modifications.

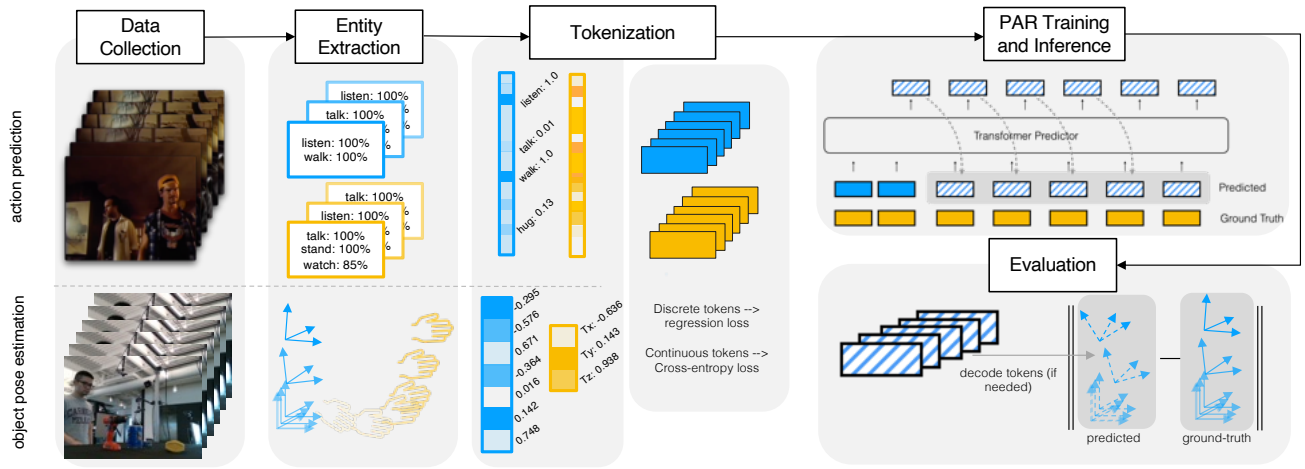
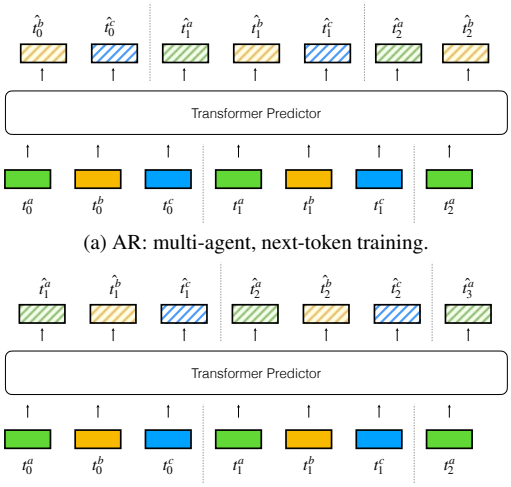


Figure 2. **The PAR Framework**. We begin by collecting a video dataset, such as AVA (top) or DexYCB (bottom). Then, using dataset labels or computer vision techniques, a trajectory of a given modality for our prediction task is extracted for each agent, such as action class labels (top) or object pose and 3D hand translation (bottom). Data is then tokenized, either through discretization or directly using continuous values, with our framework supporting both formats. Based on the tokenization and prediction task, we choose the appropriate loss function for PAR training. After training with PAR, predicted tokens can be decoded back to data space and evaluated with relevant metrics.



(a) AR: multi-agent, next-token training.

(b) PAR: same-agent, next-timestep training.

Figure 3. Training with teacher forcing for (a) multi-agent next-token prediction in autoregressive models and (b) multi-agent poly-autoregressive models. Solid vs striped indicates a ground-truth vs predicted token, respectively. Color denotes agent identity. The AR model is trained for next-token prediction, while the PAR model is trained to predict the next timestep of the same agent. Three agents are shown for ease of visualization, but the PAR model supports an arbitrary number of agents.

In order to formulate a problem as interaction-conditioned prediction, users must consider several task-specific details. Fig. 2 gives an overview of how the PAR framework disentangles multi-agent learning from problem-specific modeling.

Data. The input data source in our example tasks is always a collection of videos. From these videos, we extract various modalities relevant to the task at hand. These modalities can range from high-level features, such as action class labels, to low-level ones, such as 3D pose (Fig. 2 first two columns). We assume that each agent in the dataset is detected at each frame and is associated with an agent ID.

Tokenization. Our framework supports both discrete, quantized tokens and continuous vector tokens. The choice between discrete and continuous depends on the nature of the task. In the case of discrete tokens, we use a standard embedding layer to project to the hidden dimension. For continuous tokens, we train a projection layer to project the token into the hidden dimension of the transformer. For instance, if our continuous token is a 3D vector with an (x, y, z) 3D location coordinate and our hidden dimension is 128, our projection layer will project from 3 to 128 dimensions. We also train an un-projection layer that reverts the hidden dimension to the original token dimension.

Loss. The type of token and task-specific considerations dictate the loss function ℓ applied during model training. For discrete tokens, a classification loss is appropriate. For continuous tokens, we use a regression loss on the original token dimension.

Baselines. We compare to the following baselines, where applicable on a case-by-case basis:

• *Random token*: pick random tokens from the best available token space and use as the prediction.

• *Random trajectory*: pick at random a trajectory from the training dataset to use as the prediction.

• *NN*: Given an input agent A's trajectory history, find the closest trajectory to it in the training set, belonging to A^T. Use A^T's future as the predicted future.

• *Multiagent NN*: In a dataset with two interacting partners A and B, where B is the ego agent, given an input agent

A's trajectory history, find the closest trajectory to it in the training set, belonging to A^T. Use A^T's interaction partner's B^T's future as the prediction.

• *Mirror*: In a dataset with two interacting partners A and B, use the ground truth future of agent B as the predicted future for agent A.

3.4. Framework Implementation Details

We keep the following implementation details constant for all case studies (see also Sec. 10).

Learned agent ID embedding. Our learned agent ID embedding consists of the integer agent ID mapped to a hidden dim-sized vector, and summed to the token embedding.

Architecture. For all case studies, we use the Llama [51] transformer decoder architecture with 8 layers, 8 attention heads, and a hidden and intermediate dimension of 128. The decoder has ∼4.4M learned parameters, not including learned embedding layers which add a few thousand more parameters. A rotary positional encoding [47] is used in addition to other summed encodings (i.e. agent ID embedding, locational positional encoding in Sec. 5). We train using teacher forcing. The only hyperparameter that changes between case studies is the learning rate.

4. Case Study 1: Social Action Forecasting

Our first case study involves forecasting human actions. Human behaviors are fundamentally social; for instance, individuals frequently walk in groups and alternate between speaking and listening roles when conversing. Certain actions, like hugging or handshaking, are intrinsically multi-person. Therefore, modeling human interactions should help improve action forecasting performance, especially on multi-person actions, which we show in this case study.

4.1. Experimental Setup

Dataset. The Atomic Visual Actions (AVA) dataset [19] comprises 235 training and 64 15-minute validation videos from movies. Annotations are provided at a 1Hz frequency, detailing bounding boxes and tracks for individuals within the frame, and each person's actions within a 1-second timeframe. Individuals may engage in multiple concurrent actions from a repertoire of 60 distinct action classes (e.g., sitting and talking simultaneously). For our analysis, we select clips featuring a continuous sequence of an agent's actions spanning at least 4s, splitting sequences exceeding 12s. We use the first half of each clip as history to predict the second half. For any ego agent trajectory, we pick a second agent by selecting the person present in the scene for the longest subset of the ego agent's trajectory.

Task-specific considerations. Each agent's token $\mathcal{A}$ represents an 60-dimensional vector that corresponds to the actions performed at a specific timestep. Each element denotes the probability of a particular action class being enacted; ground-truth inputs are a binary vector. We implement an embedding layer that projects these tokens into the transformer's hidden dimension, as well as an un-projection layer that reverts them back to the original 60D token space for the purposes of loss calculation and output generation. We do not explicitly require the outputs to be values between 0 and 1. We use a MSE regression loss on the 60D action tokens: $\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}(\mathcal{A}_i - \hat{\mathcal{A}}_i)^2$. Our evaluation metric is the mean average precision (mAP) on the 60 AVA classes.

We implement all baselines described in 3.3, where *Random Token* corresponds to a random 60D vector sampled from 0 to 1. *NN* and *Multiagent NN* use Hamming distance as the distance metric.

Method	Timestep pred	Ag-ID embd	mAP ↑
1-agent AR	N/A	N/A	40.7
2-agent AR	✗	✗	38.0
2-agent PAR*	✗	✓	40.2
2-agent PAR*	✓	✗	40.0
2-agent PAR	✓	✓	**42.6**

Table 1. **PAR action forecasting performance on AVA** We evaluate 1 and 2-agent AR methods, two 2-agent PAR ablations (rows 3 and 4, PAR*), and our PAR method. Without next-timestep prediction (see Fig. 3) or a learned agent ID embedding, our model struggles with multi-agent reasoning, performing worse than the AR baseline. With both components, the 2-agent PAR model achieves a +1.9 mAP gain over the AR method (see Fig. 11 and Fig. 5 for class breakdown).

Baseline	Agents	mAP ↑
Random Token	1	3.46
Random Training Traj	1	3.44
Nearest Neighbor	1	13.17
Multiagent NN	2	5.10
Mirror	2	7.97

Table 2. **AVA baselines** While the nearest neighbor baseline performs best among baselines, it is still significantly worse than the AR model.

4.2. Results

We report the performance of a single-agent AR model as a baseline, in the first line of Table 1. The AR model is significantly better than our baselines (see Table 2), the strongest baseline being the single-agent NN. We compare these baselines to our 2-agent PAR model (last line) and various ablations where we remove the agent ID embedding and perform next-token rather than same-agent next-timestep prediction. The second line of the table corresponds to multi-agent next-token prediction (Fig. 3a). We see that this approach confuses the model, and the performance is significantly worse than just training on and considering a single agent. However,

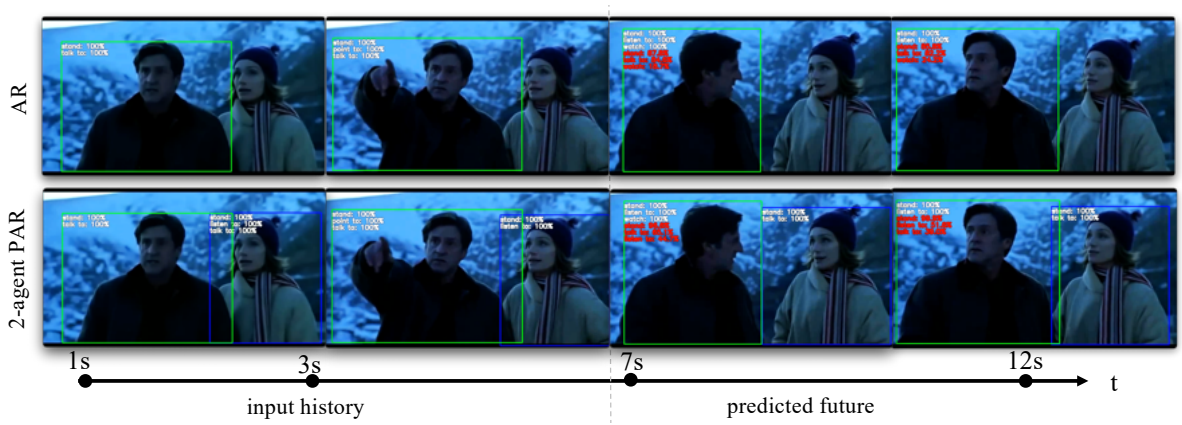


Figure 4. **Action forecasting example.** The distribution over ground truth actions are in white, and our predictions in red. A 6s action history (1Hz) is input, and 6s of future actions predicted. In the scene, the man and woman alternate between talking and listening. Initially, the man is talking. The AR model predicts he will continue talking, while the 2-agent PAR model recognizes the woman is talking and predicts more accurate turn-taking behavior.

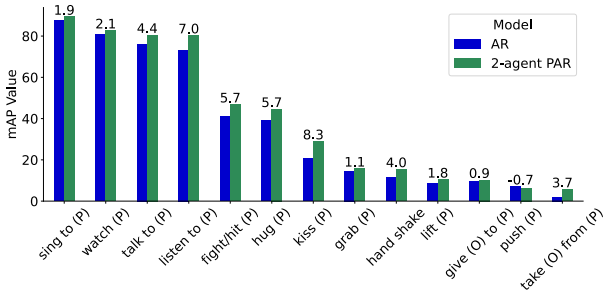


Figure 5. **Per-class mAP for AVA 2-person actions**. We see performance improvement on almost all 2-person AVA action classes ((P) stands for "a person"). Some absolute mAP gains are particularly significant: *listen to* +7.0, *kiss* +8.3, *fight/hit* +5.7, *talk to* +4.4, *hug* +5.7, and *hand shake* +4.0.

as we add various components of our PAR approach, the performance improves, and with both the next timestep prediction and agent ID embedding, we get a +1.9 mAP gain. When only considering 2-person action classes (enumerated in Fig. 5), our mAP is 36.3 on the single agent PAR model and 39.8 on the 2-agent PAR model, a +3.5 mAP gain.

In Fig. 4 we see an example of action forecasting. In the input history, the man talks and the woman listens. In the future, the woman talks, and the man listens. Our 2-agent PAR model (bottom row) better understands that talking and listening actions are complementary actions, while the AR model doesn't learn this correlation. We see quantitative evidence of this in Fig. 5, with per-class mAPs for our AR vs 2-agent PAR model for 2-person action classes. Here, *talk to* gets a +4.4 mAP gain and *listen to* gets a +7.0 mAP gain when we train a multi-agent model. We see a significant boost on many other interaction-related action classes—for instance, *kiss a person* +8.3 and *fight/hit a person* +5.7 mAP—and on single-person actions, see Fig. 11.

5. Case Study 2: Multiagent Car Trajectory Prediction

Our second case study focuses on predicting car trajectories. Trajectory prediction requires a vehicle to be aware of other cars on the road to avoid collisions and promote cooperative behavior. This study demonstrates how our framework enables the joint modeling of multiple vehicles' movements.

5.1. Experimental Setup

Dataset. We use nuScenes [5] , inputting 2 seconds of positions to forecast vehicle positions 6 seconds ahead. Specifically, our objective is to predict the xy coordinates of each agent, exclusively considering vehicles as agents. We use the `trajdata` interface [23] to load and visualize the data.

Task-specific considerations. Instead of discretizing the xy position space, we discretize the motion, resulting in discrete velocity or acceleration tokens. These integer tokens are projected to the transformer hidden dimension using the Llama token embedding layer. Inputting only these tokens results in our PAR model knowing what speed the other agents are going at, but not where they are. It is important the model has this awareness (it should know if two agents are going to collide), so our model needs to reason over this second modality of location. We implement this by passing locations relative to the agent we are predicting into a sin-cosine positional embedding (see details in Sec. 12.1), which we denote a location positional encoding (LPE). The LPE is summed to our token embeddings.

We use a cross-entropy classification loss on our discrete tokens: $\mathcal{L} = E_{y\sim p(y)}[-\log(p(\mathbf{s}_{t_\pi}^T)]$. We use the standard average displacement error (ADE) and final displacement error (FDE) to evaluate our predicted trajectories. For our baselines (Sec. 3.3), we use the closest agent at the current timestep for *Multiagent NN* and *Mirror*. For *NN* and *Multiagent NN* we use MSE as the distance metric.

Token type	LPE	Method	ADE ↓	FDE ↓
Velocity	✗	1-agent AR	1.50	3.64
Velocity	✗	3-agent PAR	1.45	3.51
Accleration	✗	1-agent AR	1.44	3.57
Accleration	✗	3-agent PAR	1.40	3.44
Accleration	✓	3-agent PAR	**1.35**	**3.34**

Table 3. **Car trajectory prediction performance.** Using acceleration tokens and 3-agent PAR results in a stronger performance over velocity tokens and single-agent AR. Adding location via a positional encoding (LPE) further improves results.

Baseline	Agents	ADE ↓	FDE ↓
Random Trajectory	1	8.89	16.51
NN	1	1.80	4.13
Multiagent NN	N	6.40	12.04
Mirror	N	11.59	14.93

Table 4. **Car trajectory prediction baselines.** Nearest neighbor performs best overall, but our learned single-agent AR models outperform all baselines.

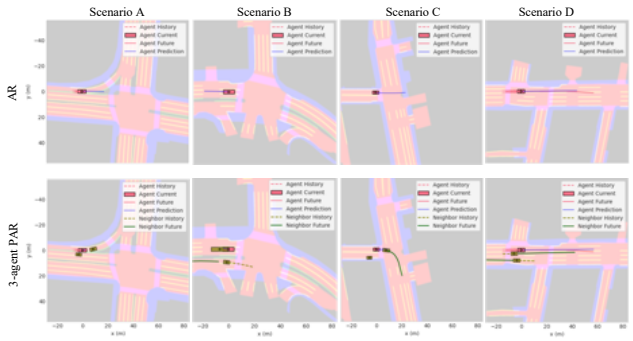
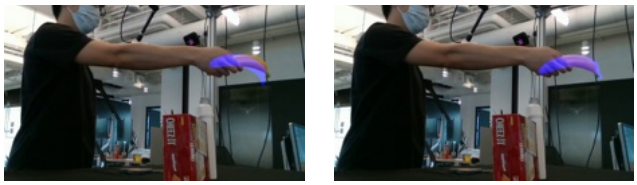


Figure 6. Example results from our single-agent AR model (top row) and three-agent PAR model with location positional encoding (bottom row) on nuScenes. The predicted agent's ground truth trajectory is in pink, and the predicted future in blue. For the PAR model, the other two agents' ground truth states are in green. Qualitatively, the PAR model handles situations where single-agent predictions might lead to collisions (A, B), uses other agents' behavior to better adhere to road areas (A, C) without environment data, and predicts based on the speed changes of other cars (D).

5.2. Results

We train AR and 3-agent PAR models using velocity tokens, acceleration tokens, and acceleration tokens combined with our location positional encoding. The results can be seen in Table 3. Note that the 3-agent PAR model uses the agent ID embedding and next timestep prediction. Acceleration tokens consistently outperform velocity tokens both for agent AR and 3-agent PAR models. This could be because the vocabulary size for acceleration tokens is much smaller and therefore easier to optimize. Regardless, both ways of tokenizing result in models that outperform our baselines (see



1-Agent AR 2-Agent PAR

Figure 7. **Rotation forecasting qualitative result on test set.** 3D predictions are projected onto the image, isolating rotation results by showing the ground-truth translation. Incorporating the hand agent in the PAR framework (right) improves object pose prediction over object-only AR (left).

1-Agent AR 2-Agent PAR

Figure 8. **Translation forecasting qualitative result on test set.** 3D predictions are projected onto the image, isolating translation results by showing the ground-truth rotation. Using the PAR framework (right) instead of AR (left) improves object pose prediction.

Table 4 - NN has a relatively low error on this dataset), and highlight that our framework is flexible such that a user can experiment with different ways of representing entities. For both token types, the 3-agent PAR model that is blind to location outperforms the AR model. While location information should help the model, it is possible that simply knowing whether other agents are slowing down or accelerating can help the model make better predictions. When adding location information via the LPE to our 3-agent PAR model, we see another performance gain in ADE and FDE.

Qualitative examples of the AR model (top row) and 3-agent location-aware PAR model (bottom row) can be seen in Figure 6. Our method uses no image or environment data (e.g., lanes) as input. However, by reasoning over multiple agents, its predictions lead to fewer collisions and better reasoning about speed changes and driveable areas based solely on other agents' behaviors.

6. Case Study 3: Object Pose Forecasting During Hand-Object Interaction

Our final case study explores how hand-object interaction can be leveraged for object pose estimation. We define the human hand and the interacting object as two agents, with tokens representing distinct state types. We show that our PAR framework can jointly model these agents, improving 3D translation and rotation predictions for the object.

6.1. Experimental Setup

Dataset. We use the DexYCB dataset, which includes 1000 videos of 10 subjects performing object manipulation tasks with 20 distinct objects from the YCB-Video dataset. The data is split into 800 training, 40 validation, and 160 testing videos. We use one of 8 provided camera views. In each trial, subjects pick up and lift objects in randomized conditions. Labels include the object's SO(3) rotation and 3D translation, and the hand's 3D translation. We focus on predicting the object's rotation or translation.

Task-specific considerations. We tokenize object information in object-only experiments and both object and hand information in hand-object experiments. The object is represented as a 4D token for rotation forecasting (quaternion from SO(3) rotation) or a 3D token for translation forecasting (Euclidean coordinates). In hand-object experiments, the hand token is included with a 3D translation vector, and agent ID embeddings distinguish between the hand and object. Normalization is applied to all 3D translation vectors in both AR and PAR experiments; quaternions are normalized by definition and require no additional processing. An embedding layer projects the tokens into the transformer's hidden dimension, and another layer projects them back for prediction.

For rotation-only forecasting, the loss is $\mathcal{L}_{rot} = 1 - |\hat{q} \cdot q|$, where $\hat{q}$ is the predicted quaternion and q the ground-truth quaternion. For translation-only forecasting, the loss $\mathcal{L}_t$ is the mean squared error (MSE) between predicted and ground-truth translations. For PAR we predict relative object-to-hand translations at each frame, using the current hand position as origin, while for AR, we predict absolute object translations without considering the interacting agent. For PAR models, we add the loss $\mathcal{L}_h$, a MSE on hand translation. The object-only AR rotation model is optimized with $\mathcal{L}_{rot}$, while the PAR rotation model combines $\mathcal{L}_{rot} + \mathcal{L}_h$; similarly, the object-only translation model is trained with $\mathcal{L}_t$, and the hand-object translation model uses $\mathcal{L}_t + \mathcal{L}_h$. At inference, the first half of each video is provided, and object predictions are autoregressively generated for the second half. Translation is evaluated using MSE, while rotation is measured using geodesic distance (GEO) on SO(3).

6.2. Results

We compare the object-only AR models to the hand-object PAR models in Table 5 for the two prediction tasks. We also present the baselines described in Sec. 3.3 in Table 6. Figures 7 and 8 show qualitative results on the rotation and translation predictions, respectively. In both prediction tasks, we observe that incorporating the human hand's interaction with the object enhances accuracy: for rotation, PAR results in a relative improvement of 8.9% over AR, and for translation, 41%. See Section 11.2 for additional qualitative results with more sampled frames.

Type	Method	MSE ↓	GEO (rad) ↓
Translation	1-agent AR	3.68×10^{-3}	-
Translation	2-agent PAR	$\mathbf{2.17 \times 10^{-3}}$	-
Rotation	1-agent AR	-	0.919
Rotation	2-agent PAR	-	**0.837**

Table 5. **Test set results on DexYCB dataset.** For both rotation and translation forecasting, the 2-agent PAR model, which treats the hand as an additional agent, improves results.

Baseline	Translation - MSE (m^2) ↓	Rotation - GEO (rad) ↓
Random	0.244	2.196
Random Trajectory	1.60×10^{-2}	2.146
NN	1.69×10^{-2}	2.179
Multiagent NN	1.71×10^{-2}	2.170
Mirror	1.20×10^{-2}	-

Table 6. **Test set results for DexYCB baselines.** We cannot provide rotation results for the Mirror baseline, because the ground-truth does not include hand rotation, only 3D translation.

7. Discussion

This work introduced the Poly-Autoregressive (PAR) framework, a unifying approach to prediction for multi-agent interactions. By applying the same transformer architecture and hyperparameters across diverse tasks, including action forecasting in social settings, trajectory prediction for autonomous vehicles, and object pose forecasting during hand-object interaction, we have demonstrated the versatility and robustness of our framework.

Our findings underscore the crucial importance of considering the influence of multiple agents in a scene for prediction tasks. By modeling interactions, we significantly improved prediction accuracy over single-agent approaches on all three problems we considered. While we achieved promising results with a simple architecture, we have only provided a starting point that can be built upon extensively. For instance, incorporating environmental context or tokenizing pixel patches, especially as a way to relax our assumption on high-quality tracking, are avenues for further research using PAR. It would be interesting to experiment with scaling the data and model. We do not specifically consider the relative importance of neighboring agents, which is an interesting future research direction (ex. dynamic attention mechanism).

Our PAR framework provides a simple and generalizable foundation of universal building blocks, ready for extension or refinement in future tasks. The PAR framework holds potential for advancing AI systems, enhancing prediction capabilities and enabling more accurate navigation and operation in real-world, multi-agent interactions.

8. Acknowledgements

We thank Jane Wu, Himanshu Singh, Georgios Pavlakos, and João Carreira for useful discussions and feedback. This work was supported by ONR MURI N00014-21-1-2801 and NSF Graduate Fellowships to NT and TS.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022. 2

[2] Fred Attneave. Some informational aspects of visual perception. *Psychological review*, 61(3):183, 1954. 2

[3] Murchana Baruah and Bonny Banerjee. A multimodal predictive agent model for human interaction generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 1022–1023, 2020. 2

[4] Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 2

[5] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 2, 6, 1

[6] Sergio Casas, Wenjie Luo, and Raquel Urtasun. Intentnet: Learning to predict intention from raw sensor data. In *Conference on Robot Learning*, pages 947–956. PMLR, 2018. 1

[7] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8748–8757, 2019. 1

[8] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, et al. Dexycb: A benchmark for capturing hand grasping of objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9044–9053, 2021. 2, 1

[9] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International conference on machine learning*, pages 1691–1703. PMLR, 2020. 2

[10] Baptiste Chopin, Hao Tang, Naima Otberdout, Mohamed Daoudi, and Nicu Sebe. Interaction transformer for human reaction generation. *IEEE Transactions on Multimedia*, pages 1–13, 2023. 2

[11] Can Cui, Yunsheng Ma, Xu Cao, Wenqian Ye, Yang Zhou, Kaizhao Liang, Jintai Chen, Juanwu Lu, Zichong Yang, Kuei-Da Liao, et al. A survey on multimodal large language models for autonomous driving. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 958–979, 2024. 1

[12] 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. 1

[13] Xinke Deng, Arsalan Mousavian, Yu Xiang, Fei Xia, Timothy Bretl, and Dieter Fox. Poserbpf: A rao–blackwellized particle filter for 6-d object pose tracking. *IEEE Transactions on Robotics*, 37(5):1328–1342, 2021. 1

[14] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 1

[15] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6824–6835, 2021. 1

[16] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019. 1

[17] Anindita Ghosh, Rishabh Dabral, Vladislav Golyanik, Christian Theobalt, and Philipp Slusallek. Remos: 3d motion-conditioned reaction synthesis for two-person interactions. In *European Conference on Computer Vision (ECCV)*, page 3, 2024. 2

[18] Karol Gregor, Ivo Danihelka, Andriy Mnih, Charles Blundell, and Daan Wierstra. Deep autoregressive networks. In *International Conference on Machine Learning*, pages 1242–1250. PMLR, 2014. 2

[19] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6047–6056, 2018. 2, 5

[20] Wen Guo, Xiaoyu Bie, Xavier Alameda-Pineda, and Francesc Moreno-Noguer. Multi-person extreme motion prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13053–13064, 2022. 2

[21] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 1

[22] Yanjun Huang, Jiatong Du, Ziru Yang, Zewei Zhou, Lin Zhang, and Hong Chen. A survey on trajectory-prediction methods for autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 7(3):652–674, 2022. 1

[23] Boris Ivanovic, Guanyu Song, Igor Gilitschenski, and Marco Pavone. trajdata: A unified interface to multiple human trajectory datasets. In *Proceedings of the Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks*, New Orleans, USA, 2023. 6

[24] Bolin Lai, Sam Toyer, Tushar Nagarajan, Rohit Girdhar, Shengxin Zha, James M Rehg, Kris Kitani, Kristen Grau-

man, Ruta Desai, and Miao Liu. Human action anticipation: A survey. *arXiv preprint arXiv:2410.14045*, 2024. 1

[25] Hugo Larochelle and Iain Murray. The neural autoregressive distribution estimator. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 29–37. JMLR Workshop and Conference Proceedings, 2011. 2

[26] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. Deepim: Deep iterative matching for 6d pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 683–698, 2018. 1

[27] Han Liang, Wenqian Zhang, Wenxuan Li, Jingyi Yu, and Lan Xu. Intergen: Diffusion-based multi-human motion generation under complex interactions. *International Journal of Computer Vision*, pages 1–21, 2024. 2

[28] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023. 2

[29] Siyuan Brandon Loh, Debaditya Roy, and Basura Fernando. Long-term action forecasting using multi-headed attention-based variational recurrent neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2419–2427, 2022. 1

[30] Vongani Maluleke, Lea Müller, Jathushan Rajasegaran, Georgios Pavlakos, Shiry Ginosar, Angjoo Kanazawa, and Jitendra Malik. Synergy and synchrony in couple dances. *arXiv preprint arXiv:2409.04440*, 2024. 2

[31] Nigamaa Nayakanti, Rami Al-Rfou, Aurick Zhou, Kratarth Goel, Khaled S Refaat, and Benjamin Sapp. Wayformer: Motion forecasting via simple & efficient attention networks. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2980–2987. IEEE, 2023. 1

[32] Evonne Ng, Hanbyul Joo, Liwen Hu, Hao Li, Trevor Darrell, Angjoo Kanazawa, and Shiry Ginosar. Learning to listen: Modeling non-deterministic dyadic facial motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20395–20405, 2022. 2

[33] Evonne Ng, Sanjay Subramanian, Dan Klein, Angjoo Kanazawa, Trevor Darrell, and Shiry Ginosar. Can language models learn to listen? In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023. 2

[34] Evonne Ng, Javier Romero, Timur Bagautdinov, Shaojie Bai, Trevor Darrell, Angjoo Kanazawa, and Alexander Richard. From audio to photoreal embodiment: Synthesizing humans in conversations. In *ArXiv*, 2024. 2

[35] Jiquan Ngiam, Benjamin Caine, Vijay Vasudevan, Zhengdong Zhang, Hao-Tien Lewis Chiang, Jeffrey Ling, Rebecca Roelofs, Alex Bewley, Chenxi Liu, Ashish Venugopal, et al. Scene transformer: A unified architecture for predicting multiple agent trajectories. *arXiv preprint arXiv:2106.08417*, 2021. 1

[36] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Łukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer, 2018. 2

[37] Alec Radford. Improving language understanding by generative pre-training. 2018. 2

[38] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 2

[39] Ilija Radosavovic, Bike Zhang, Baifeng Shi, Jathushan Rajasegaran, Sarthak Kamat, Trevor Darrell, Koushil Sreenath, and Jitendra Malik. Humanoid locomotion as next token prediction. *arXiv preprint arXiv:2402.19469*, 2024. 2

[40] Jathushan Rajasegaran, Georgios Pavlakos, Angjoo Kanazawa, Christoph Feichtenhofer, and Jitendra Malik. On the benefits of 3d pose and tracking for human action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 640–649, 2023. 1

[41] Jathushan Rajasegaran, Ilija Radosavovic, Rahul Ravishankar, Yossi Gandelsman, Christoph Feichtenhofer, and Jitendra Malik. An empirical study of autoregressive pre-training from videos. *arXiv preprint arXiv:2501.05453*, 2025. 2

[42] Chaitanya Ryali, Yuan-Ting Hu, Daniel Bolya, Chen Wei, Haoqi Fan, Po-Yao Huang, Vaibhav Aggarwal, Arkabandhu Chowdhury, Omid Poursaeed, Judy Hoffman, et al. Hiera: A hierarchical vision transformer without the bells-and-whistles. In *International Conference on Machine Learning*, pages 29441–29454. PMLR, 2023. 1

[43] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pages 683–700. Springer, 2020. 1

[44] Ari Seff, Brian Cera, Dian Chen, Mason Ng, Aurick Zhou, Nigamaa Nayakanti, Khaled S Refaat, Rami Al-Rfou, and Benjamin Sapp. Motionlm: Multi-agent motion forecasting as language modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8579–8590, 2023. 2, 1

[45] Claude E Shannon. Prediction and entropy of printed english. *Bell system technical journal*, 30(1):50–64, 1951. 2

[46] Li Siyao, Tianpei Gu, Zhitao Yang, Zhengyu Lin, Ziwei Liu, Henghui Ding, Lei Yang, and Chen Change Loy. Duolando: Follower gpt with off-policy reinforcement learning for dance accompaniment. *arXiv preprint arXiv:2403.18811*, 2024. 2

[47] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. 5

[48] Chen Sun, Abhinav Shrivastava, Carl Vondrick, Rahul Sukthankar, Kevin Murphy, and Cordelia Schmid. Relational action forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 273–283, 2019. 1

[49] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 1

[50] Lucas Theis and Matthias Bethge. Generative image modeling using spatial lstms. *Advances in neural information processing systems*, 28, 2015. 2

[51] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste

Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. 5

[52] Ameni Trabelsi, Mohamed Chaabane, Nathaniel Blanchard, and Ross Beveridge. A pose proposal and refinement network for better 6d object pose estimation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2382–2391, 2021. 1

[53] Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pages 1747–1756. PMLR, 2016. 2

[54] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. 2

[55] Gu Wang, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16611–16621, 2021. 1

[56] Bowen Wen, Chaitanya Mitash, Baozhang Ren, and Kostas E Bekris. se (3)-tracknet: Data-driven 6d pose tracking by calibrating image residuals in synthetic domains. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10367–10373. IEEE, 2020. 1

[57] Bowen Wen, Jonathan Tremblay, Valts Blukis, Stephen Tyree, Thomas Müller, Alex Evans, Dieter Fox, Jan Kautz, and Stan Birchfield. Bundlesdf: Neural 6-dof tracking and 3d reconstruction of unknown objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 606–617, 2023.

[58] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17868–17879, 2024. 1

[59] Jane Wu, Georgios Pavlakos, Georgia Gkioxari, and Jitendra Malik. Reconstructing hand-held objects in 3d. *arXiv preprint arXiv:2404.06507*, 2024. 1

[60] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017. 1

[61] Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris M Kitani. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9813–9823, 2021. 1