A. Motion token vocabulary

Delta action space. The models presented in this paper use the following parameters for the discretized delta action space:

- Step frequency: 2 Hz
- Delta interval (per step): [-18.0 m, 18.0 m]
- Number of bins: 128

At 2 Hz prediction, a maximum delta magnitude of 18 m covers axis-aligned speeds up to 36 m/s ($\sim$80 mph), $> 99\%$ of the WOMD dataset.

Verlet-wrapped action space. Once the above delta action space has the Verlet wrapper applied, we only require 13 bins for each coordinate. This results in a total of $13^2 = 169$ total discrete motion tokens that the model can select from the Cartesian product comprising the final vocabulary.

Sequence lengths. For 8-second futures, the model outputs 16 motion tokens for each agent (note that WOMD evaluates predictions at 2 Hz). For the two-agent interactive split, our flattened agent-time token sequences (Section 3.2.2) have length $2 \times 16 = 32$.

B. Implementation details

B.1. Scene encoder

We follow the design of the early fusion network proposed by [31] as the scene encoding backbone of our model. The following hyperparameters are used:

- Number of layers: 4
- Hidden size: 256
- Feed-forward network intermediate size: 1024
- Number of attention heads: 4
- Number of latent queries: 92
- Activation: ReLU

B.2. Trajectory decoder

To autoregressively decode motion token sequences, we utilize a causal transformer decoder that takes in the motion tokens as queries, and the scene encodings as context. We use the following model hyperparameters:

- Number of layers: 4
- Hidden size: 256
- Feed-forward network intermediate size: 1024
- Number of attention heads: 4
- Activation: ReLU

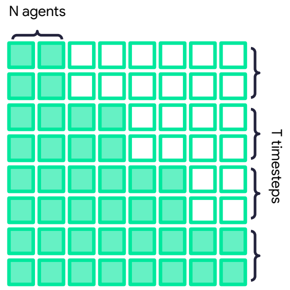

Figure 8. Masked causal attention between two agents during training. We flatten the agent and time axes, leading to an $NT \times NT$ attention mask. The agents may attend to each other's previous motion tokens (solid squares) but no future tokens (empty squares).

B.3. Optimization

We train our model to maximize the likelihood of the ground truth motion token sequences via teacher forcing. We use the following training hyperparameters:

- Number of training steps: 600000
- Batch size: 256
- Learning rate schedule: Linear decay
- Initial learning rate: 0.0006
- Final learning rate: 0.0
- Optimizer: AdamW
- Weight decay: 0.6

B.4. Inference

We found nucleus sampling [16], commonly used with language models, to be helpful for improving sample quality while maintaining diversity. Here we set the top-p parameter to 0.95.

C. Metrics descriptions

C.1. WOMD metrics

All metrics for the two WOMD [14] benchmarks are evaluated at three time steps (3, 5, and 8 seconds) and are averaged over all object types to obtain the final value. For joint metrics, a scene is attributed to an object class (vehicle, pedestrian, or cyclist) according to the least common type of agent that is present in that interaction, with cyclist being the rarest object class and vehicles being the most common. Up to 6 trajectories are produced by the models for each target agent in each scene, which are then used for metric evaluation.

mAP & Soft mAP mAP measures precision of prediction likelihoods and is calculated by first bucketing ground truth futures of objects into eight discrete classes of intent: straight, straight-left, straight-right, left, right, left u-turn, right u-turn, and stationary.

For marginal predictions, a prediction trajectory is considered a "miss" if it exceeds a lateral or longitudinal error threshold at a specified timestep T. Similarly for joint predictions, a prediction is considered a "miss" if none of the k joint predictions contains trajectories for all predicted objects within a given lateral and longitudinal error threshold, with respect to the ground truth trajectories for each agent. Trajectory predictions classified as a miss are labeled as a false positive. In the event of multiple predictions satisfying the miss criteria, consistent with object detection mAP metrics, only one true positive is allowed for each scene, assigned to the highest confidence prediction. All other predictions for the object are assigned a false positive.

To compute the mAP metric, bucket entries are sorted and a P/R curve is computed for each bucket, averaging precision values over various likelihood thresholds for all intent buckets results in the final mAP value. Soft mAP differs only in the fact that additional matching predictions (other than the most likely match) are ignored instead of being assigned a false positive, and so are not penalized in the metric computation.

Miss rate Using the same definition of a "miss" described above for either marginal or joint predictions, miss rate is a measure of what fraction of scenarios fail to generate *any* predictions within the lateral and longitudinal error thresholds, relative to the ground truth future.

minADE & minFDE minADE measures the Euclidean distance error averaged over all timesteps for the closest prediction, relative to ground truth. In contrast, minFDE considers only the distance error at the final timestep. For joint predictions, minADE and minFDE are calculated as the average value over both agents.

C.2. Prediction overlap

As described in [29], the WOMD [14] overlap metric only considers overlap between predictions and ground truth. Here we use a *prediction overlap* metric to assess scene-level consistency for joint models. Our implementation is similar to [29], except we follow the convention of the WOMD challenge of only requiring models to generate (x, y) waypoints; headings are inferred as in [14]. If the bounding boxes of two predicted agents collide at any timestep in a scene, that counts as an overlap/collision for that scene. The final prediction overlap rate is calculated as the sum of per-scene overlaps, averaged across the dataset.

D. Additional evaluation

Ablations. Tables 5 and 6 display joint prediction performance across varying interactive attention frequencies and numbers of rollouts, respectively. In addition to the ensembled model performance, single replica performance is evaluated. Standard deviations are computed for each metric over 8 independently trained replicas.

Scaling analysis. Table 7 displays the performance of different model sizes on the WOMD interactive split, all trained with the same optimization hyperparameters. We vary the number of layers, hidden size, and number of attention heads in the encoder and decoder proportionally. Due to external constraints, in this study we only train a single replica for each parameter count. We observe that a model with 27M parameters overfits while 300K underfits. Both the 1M and 9M models perform decently. In this paper, our main results use 9M-parameter replicas.

Latency analysis. Table 8 provides inference latency on the latest generation of GPUs across different numbers of rollouts. These were measured for a single-replica joint model rolling out two agents.

E. Visualizations

In the supplementary zip file, we have included GIF animations of the model's greatest-probability predictions in various scenes. Each example below displays the associated scene ID, which is also contained in the corresponding GIF filename. We describe the examples here.

E.1. Marginal vs. Joint

- `286a65c777726df3`
  Marginal: The turning vehicle and crossing cyclist collide.
  Joint: The vehicle yields to the cyclist before turning.
- `440bbf422d08f4c0`
  Marginal: The turning vehicle collides with the crossing vehicle in the middle of the intersection.
  Joint: The turning vehicle yields and collision is avoided.
- `38899bce1e306fb1`
  Marginal: The lane-changing vehicle gets rear-ended by the vehicle in the adjacent lane.
  Joint: The adjacent vehicle slows down to allow the lane-changing vehicle to complete the maneuver.

	Ensemble				Single Replica			
Freq. (Hz)	minADE (↓)	minFDE (↓)	MR (↓)	mAP (↑)	minADE (↓)	minFDE (↓)	MR (↓)	mAP (↑)
0.125	0.9120	2.0634	0.4222	0.2007	1.0681 (0.011)	2.4783 (0.025)	0.5112 (0.007)	0.1558 (0.007)
0.25	0.9083	2.0466	0.4241	0.1983	1.0630 (0.009)	2.4510 (0.025)	0.5094 (0.006)	0.1551 (0.006)
0.5	0.8931	2.0073	0.4173	0.2077	1.0512 (0.009)	2.4263 (0.022)	0.5039 (0.006)	0.1588 (0.004)
1	0.8842	1.9898	0.4117	0.2040	1.0419 (0.014)	2.4062 (0.032)	0.5005 (0.008)	0.1639 (0.005)
2	**0.8831**	**1.9825**	**0.4092**	**0.2150**	**1.0345** (0.012)	**2.3886** (0.031)	**0.4943** (0.006)	**0.1687** (0.004)

Table 5. Joint prediction performance across varying interactive attention frequencies on the WOMD interactive validation set. Displayed are *scene-level* joint evaluation metrics. For the single replica metrics, we include the standard deviation (across 8 replicas) in parentheses.

	Ensemble				Single Replica			
# Rollouts	minADE (↓)	minFDE (↓)	MR (↓)	mAP (↑)	minADE (↓)	minFDE (↓)	MR (↓)	mAP (↑)
1	1.0534	2.3526	0.5370	0.1524	1.9827 (0.018)	4.7958 (0.054)	0.8182 (0.003)	0.0578 (0.004)
2	0.9952	2.2172	0.4921	0.1721	1.6142 (0.011)	3.8479 (0.032)	0.7410 (0.003)	0.0827 (0.004)
4	0.9449	2.1100	0.4561	0.1869	1.3655 (0.012)	3.2060 (0.035)	0.6671 (0.003)	0.1083 (0.003)
8	0.9158	2.0495	0.4339	0.1934	1.2039 (0.013)	2.7848 (0.035)	0.5994 (0.004)	0.1324 (0.003)
16	0.9010	2.0163	0.4196	0.2024	1.1254 (0.012)	2.5893 (0.031)	0.5555 (0.005)	0.1457 (0.003)
32	0.8940	2.0041	0.4141	0.2065	1.0837 (0.013)	2.4945 (0.035)	0.5272 (0.005)	0.1538 (0.004)
64	0.8881	1.9888	0.4095	0.2051	1.0585 (0.012)	2.4411 (0.033)	0.5114 (0.005)	0.1585 (0.004)
128	0.8851	1.9893	0.4103	0.2074	1.0456 (0.012)	2.4131 (0.033)	0.5020 (0.006)	0.1625 (0.004)
256	0.8856	1.9893	**0.4078**	0.2137	1.0385 (0.012)	2.3984 (0.031)	0.4972 (0.007)	0.1663 (0.005)
512	**0.8831**	**1.9825**	0.4092	**0.2150**	**1.0345** (0.012)	**2.3886** (0.031)	**0.4943** (0.006)	**0.1687** (0.004)

Table 6. Joint prediction performance across varying numbers of rollouts per replica on the WOMD interactive validation set. Displayed are *scene-level* joint evaluation metrics. For the single replica metrics, we include the standard deviation (across 8 replicas) in parentheses.

Parameter count	Miss Rate (↓)	mAP (↑)
300K	0.6047	0.1054
1M	0.5037	0.1713
9M	0.4972	0.1663
27M	0.6072	0.1376

Table 7. Joint prediction performance across varying model sizes on the WOMD interactive validation set. Displayed are *scene-level* joint mAP and miss rate for 256 rollouts for a single model replica (except for 9M which displays the mean performance of 8 replicas).

- `2ea76e74b5025ec7`
  Marginal: The cyclist crosses in front of the vehicle leading to a collision.
  Joint: The cyclist waits for the vehicle to proceed before turning.

- `55b5fe989aa4644b`
  Marginal: The cyclist lane changes in front of the adjacent vehicle, leading to collision.
  Joint: The cyclist remains in their lane for the duration of the scene, avoiding collision.

Number of rollouts	Latency (ms)
16	19.9 (0.19)
32	27.5 (0.25)
64	43.8 (0.26)
128	75.8 (0.23)
256	137.7 (0.19)

Table 8. Inference latency on current generation of GPUs for different numbers of rollouts of the joint model. We display the mean and standard deviation (in parentheses) of the latency measurements for each setting.

E.2. Marginal vs. Conditional

"Conditional" here refers to temporally causal conditioning as described in the main text.

- `5ebba77f351358e2`
  Marginal: The pedestrian crosses the street as a vehicle is turning, leading to a collision.
  Conditional: When conditioning on the vehicle's turning trajectory as a query, the pedestrian is instead predicted to remain stationary.

- `d557eee96705c822`

Marginal: The modeled vehicle collides with the lead vehicle.
Conditional: When conditioning on the lead vehicle's query trajectory, which remains stationary for a bit, the modeled vehicle instead comes to a an appropriate stop.

- `9410e72c551f0aec`
  Marginal: The modeled vehicle takes the turn slowly, unaware of the last turning vehicle's progress.
  Conditional: When conditioning on the query vehicle's turn progress, the modeled agent likewise makes more progress.

- `c204982298bda1a1`
  Marginal: The modeled vehicle proceeds slowly, unaware of the merging vehicle's progress.
  Conditional: When conditioning on the query vehicle's merge progress, the modeled agent accelerates behind.

E.3. Temporally Causal vs. Acausal Conditioning

- `4f39d4eb35a4c07c`
  Joint prediction: The two modeled vehicles maintain speed for the duration of the scene.
  Conditioning on trailing agent:
  - **Temporally causal:** The lead vehicle is indifferent to the query trailing vehicle decelerating to a stop, proceeding along at a constant speed.
  - **Acausal:** The lead vehicle is "influenced" by the query vehicle decelerating. It likewise comes to a stop. Intuitively, this is an incorrect direction of influence that the acausal model has learned.
  Conditioning on lead agent:
  - **Temporally causal:** When conditioning on the query lead vehicle decelerating to a stop, the modeled trailing vehicle is likewise predicted to stop.
  -**Acausal:** In this case, the acausal conditional prediction is similar to the temporally causal conditional. The trailing vehicle is predicted to stop behind the query lead vehicle.