A Detailed Results

A.1 Quantitative Results: Forecasting Glancing Behavior

Table 3 depicts the NLL and head orientation error metrics for our experiments on the task of forecasting glancing behavior using synthetic data. All models are evaluated under the *random* context regime and *no-pool* configuration. The sinusoids are interpreted to represent a horizontal head rotation between $-90°$ and $90°$. To provide further insight into model performance, in Figure 4 we plot the MAE in predicted and expected mean forecasts averaged over $\boldsymbol{t}_{\text{fut}}$ against the phase of the sinusoids in the dataset. We observe that the SP-GRU error plot is smoother with respect to small phase changes, with lower errors overall.

Table 3: Mean (Std.) Metrics on the Synthetic Glancing Behavior Dataset. The metrics are averaged over timesteps; mean and std. are then computed over sequences. Lower is better. Boldface indicates best overall.

	NLL	Head Ori. MAE (°)
Baseline		
NP-latent	-0.281 (0.239)	19.631 (7.260)
Ours		
SP-latent (MLP)	-0.361 (0.197)	19.461 (7.049)
SP-latent (GRU)	$\mathbf{-0.552}$ $(\mathbf{0.230})$	$\mathbf{18.55}$ $(\mathbf{7.109})$

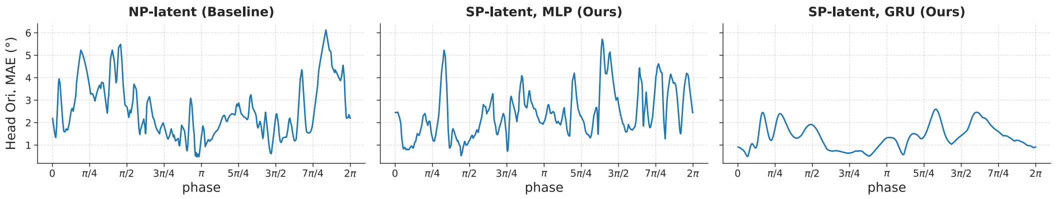

Figure 4: Error in forecast mean and expected mean orientation (average of the two ground-truth futures) for every sequence in the Synthetic Glancing dataset. Each sequence is denoted by the phase of the sinusoid.

A.2 Per Timestep Metrics

In Figure 5 we plot the evaluation metrics per timestep averaged over sequences in the Synthetic Glancing Behavior dataset. In Figure 6 we do the same for sequences in the Haggling Test Sets.

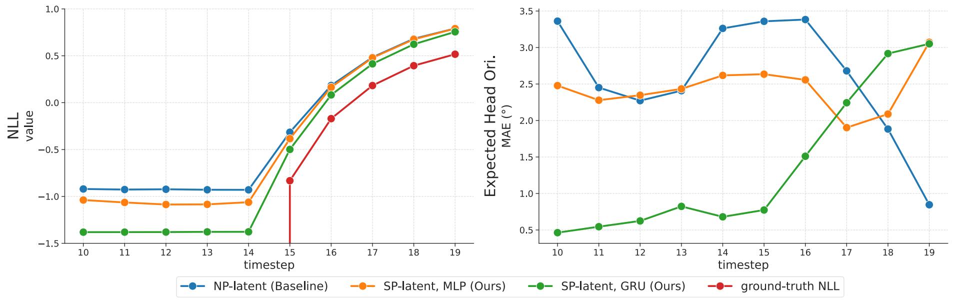

Figure 5: Mean Per Timestep Metrics over the Sequences in the Synthetic Glancing Dataset. NLL is expected to increase over timesteps where ground-truth futures diverge, being $-\infty$ when the future is certain. Head orientation error is computed between the predicted mean and the expected mean (average of the two ground-truth futures). We observe that the SP-GRU model performs best, especially when the future is certain, learning both the best mean and std. over those timesteps.

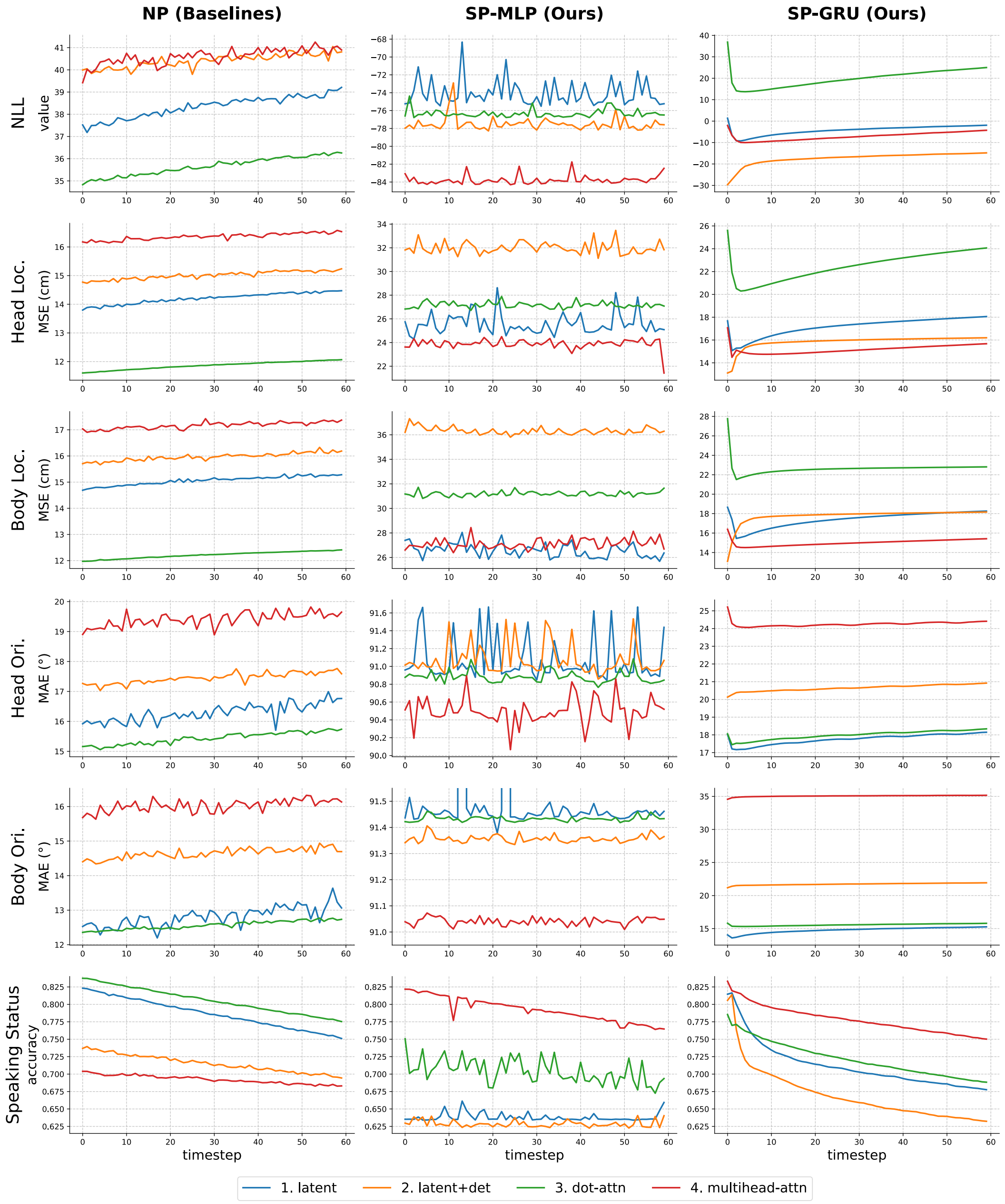

Figure 6: Mean Per Timestep Metrics over the Sequences in the Haggling Test Sets. Note that the y-axes do not share the same scale, except for speaking status accuracy. We observe that the SP-GRU model predicts smooth futures unlike the MLP models. There is a slight trend that the models get worse at forecasting over the duration of $\boldsymbol{t}_{\text{fut}}$.

A.3 Ablations

Table 4: Mean (Std.) NLL for the Ablation Experiments with the SP-latent+det GRU Model. The reported mean and std. are over sequences in the Haggling Test Sets. Lower is better.

	Context	
	Random	**Fixed-Initial**
Full Model	-17.38 (50.5)	-16.08 (52.2)
Encoding Partner Behavior		
no-pool	8.02 (75.5)	12.39 (97.5)
pool-oT	-4.67 (26.9)	-4.50 (26.7)
No Deterministic Decoding		
Shared Social Encoders	-30.65 (39.3)	-29.45 (40.4)
Unshared Social Encoders	-3.81 (28.3)	-1.79 (27.3)

Table 5: Mean (Std.) Errors in Predicted Means for the Ablation Experiments with the SP-latent+det GRU Model. The reported mean and std. are over sequences in the Haggling Test Sets. Lower is better for all except for speaking status accuracy.

	Random Context					Fixed-Initial Context				
	Head Loc. MSE (cm)	**Body Loc.** MSE (cm)	**Head Ori.** MAE (°)	**Body Ori.** MAE (°)	**Speaking** Accuracy	**Head Loc.** MSE (cm)	**Body Loc.** MSE (cm)	**Head Ori.** MAE (°)	**Body Ori.** MAE (°)	**Speaking** Accuracy
Full Model	15.84 (5.5)	17.76 (7.5)	20.65 (19.9)	21.73 (29.5)	0.671 (0.22)	16.53 (6.0)	18.20 (8.0)	20.74 (19.5)	21.31 (28.9)	0.674 (0.22)
Encoding Partner Behavior										
no-pool	18.20 (6.7)	18.05 (7.7)	16.76 (12.8)	14.30 (20.9)	0.690 (0.21)	18.64 (6.7)	18.45 (7.4)	16.85 (12.9)	14.29 (20.5)	0.687 (0.21)
pool-oT	17.42 (6.2)	19.31 (6.3)	23.39 (24.9)	17.68 (26.9)	0.743 (0.21)	17.83 (6.2)	19.23 (6.3)	23.53 (24.3)	17.51 (25.7)	0.735 (0.22)
No Deterministic Decoding										
Shared Social Encoders	15.76 (7.2)	16.34 (6.6)	45.54 (44.6)	21.87 (25.0)	0.644 (0.22)	16.93 (8.1)	17.15 (7.0)	45.49 (44.3)	21.83 (24.7)	0.637 (0.22)
Unshared Social Encoders	17.40 (6.9)	18.33 (6.7)	18.62 (14.7)	14.54 (20.2)	0.704 (0.23)	18.54 (7.9)	19.18 (7.1)	18.68 (14.9)	14.44 (20.0)	0.700 (0.23)

B Qualitative Visualizations

B.1 Glancing Behavior

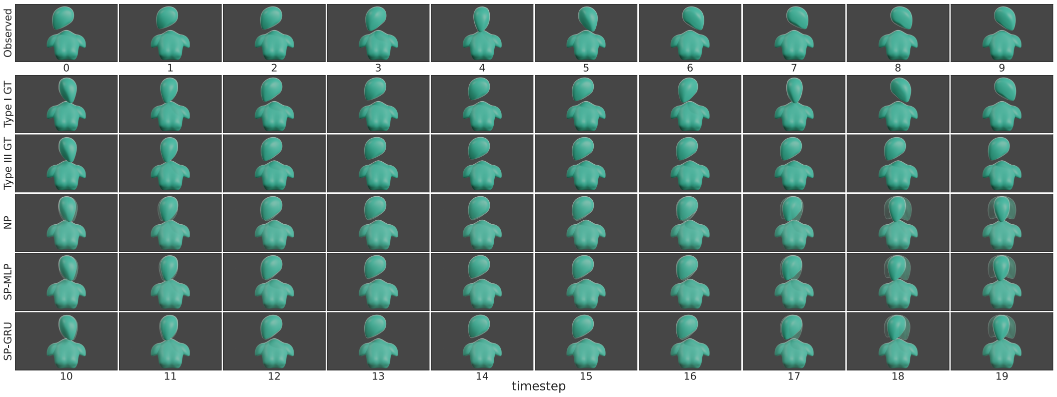

Figure 7: Forecasting Glancing Behavior for a Sequence in the Context Set. We visualize the same sinusoid within the context set as plotted in Figure 3 (phase $= 4.2$), here interpreted as a horizontal head rotation between $-90°$ and $90°$. The bottom three rows depict predictions, with the solid head denoting the mean, and the translucent heads the std. *GT* stands for *Ground-Truth*. The SP models learn better uncertainty estimates, especially over the timesteps where the future is certain (see timestep 11, for instance).

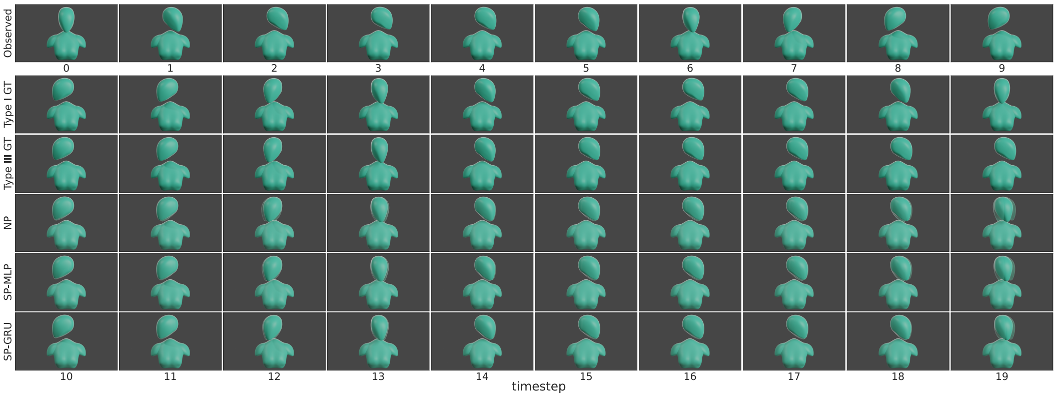

Figure 8: Forecasting Glancing Behavior for a Sequence Not in the Context Set. We visualize the same sinusoid not in the context set as plotted in Figure 3 (phase $= 0.005$). See the Figure 7 caption for details.

B.2 Haggling

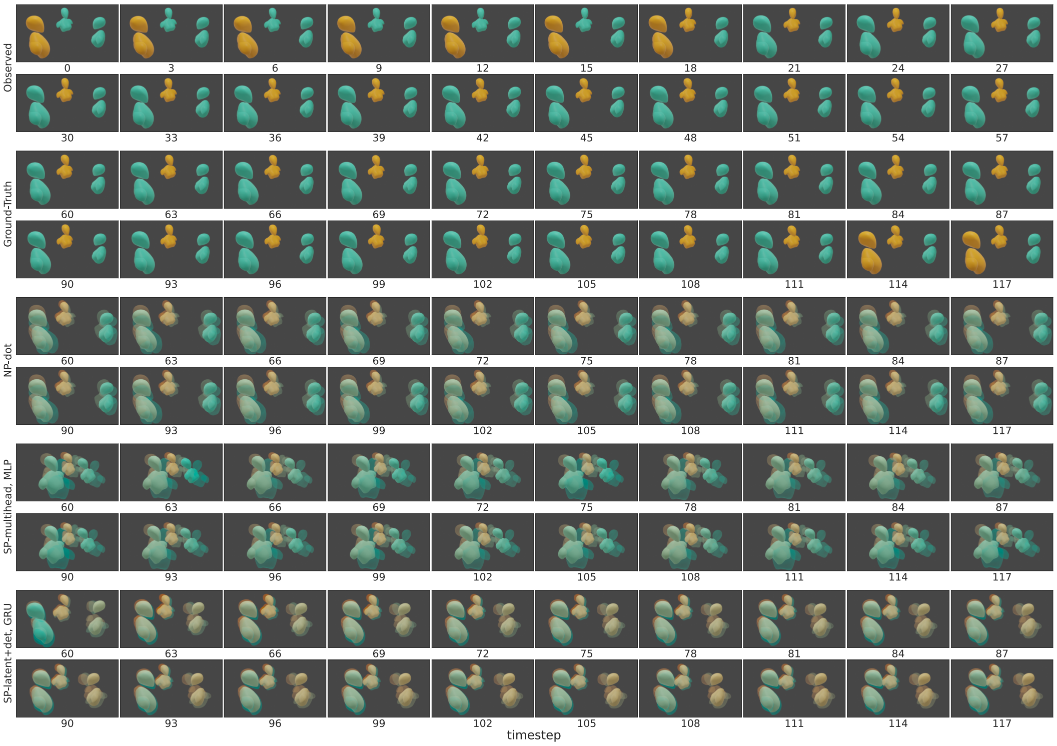

Figure 9: Forecasts for a Sequence from the Haggling Test Group *170221-b1-group3*. Note that these are features from real-world data visualized using 3D models. Speakers are depicted in orange and listeners in green. The predicted speaking status mean is visualized as an interpolated shade between the two colors. The translucent models in the forecasts denote the mean ± std. pose and speaking status. We observe that the NP forecasts are almost completely static. The SP-GRU forecasts are comparatively dynamic with lower uncertainties overall. The SP-MLP model seems to be learning an overall average orientation, forecasting all participants to be facing in the direction of the two sellers. Note that the pose changes are far more subtle than in the glancing behavior dataset. Interaction videos reveal that the participants significantly rely on gaze changes to direct attention. See Section 7 for a discussion.

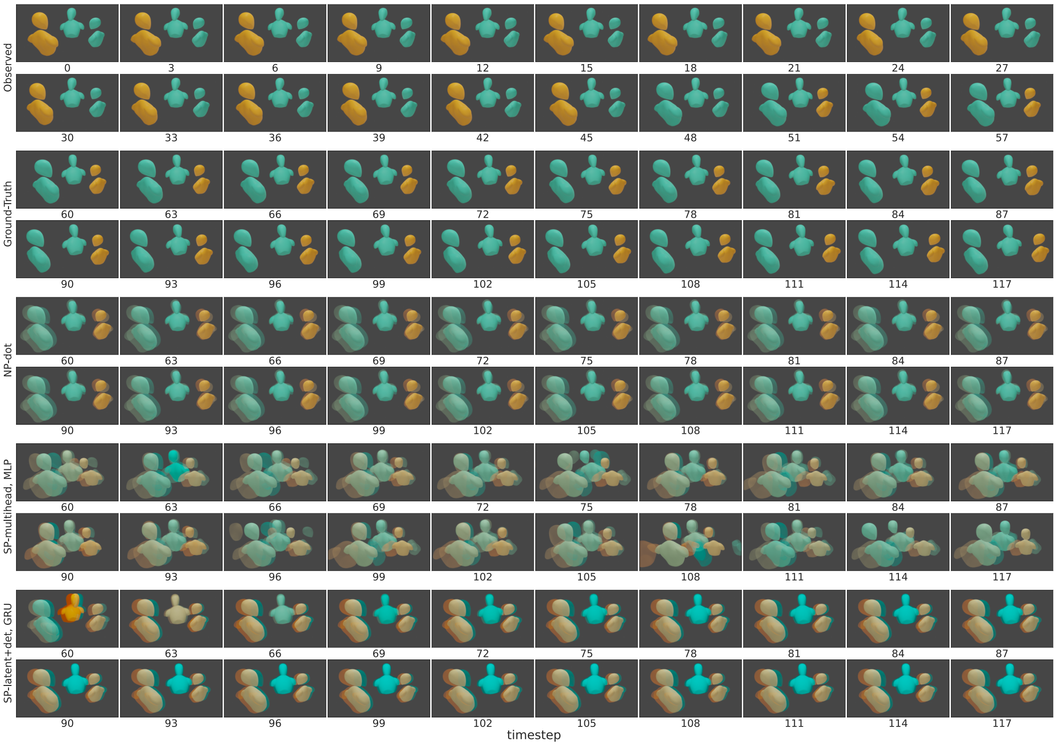

Figure 10: Forecasts for a Sequence from the Haggling Train Group *170224-a1-group1*. We see a similar pattern to the model forecasts as in Figure 9: NP forecasts are static, SP-GRU predicts more dynamic futures, while the SP-MLP forecast average orientations. A turn change has occurred at the end of the observed window. We observe that the SP-GRU model forecasts an interesting continuation to the turn. It anticipates the buyer (middle) to quickly interject the last observed speaking seller, before falling silent and directing attention between the sellers, both of whom it expects to then speak simultaneously. While this is not the ground-truth future in this instance, we believe that the forecast still indicates that the model is capable of learning believable haggling turn dynamics from the overall training data. See the Figure 9 for details on the visualization setup.

C Implementation Details

C.1 Neural Architectures

The data dimension for the experiments on the Haggling dataset is 15, while that for the toy glancing experiment is 1. Table 6 specifies the network architecture hyperparameters for the Haggling dataset experiments. For the toy experiment, all the hidden and representation dimensions are fixed at 32.

The goal of our experiments is to evaluate the relative impact of our modeling choices on performance, rather than finding the best possible model for benchmarking. Consequently, we chose a set of architecture hyperparameters such that the simplest *-latent* variants have a comparable number of parameters for cross-family comparison. These hyperparameters were then kept fixed for the variants within each family for fair intra-family comparison. The hyperparameters we chose resulted from light tuning through 5-fold cross-validation and showed improved performance for all models, but improved absolute performance might be obtained through more extensive tuning.

C.2 Training and Evaluation

We construct batches for training by bucketing samples such that all sequences in a batch share the same $\boldsymbol{t}_{\text{obs}}$, and the same $\boldsymbol{t}_{\text{fut}}$ length. Note that since the MLP models are operationalized by collapsing the timestep and feature dimensions, the length of $\boldsymbol{t}_{\text{obs}}$ and $\boldsymbol{t}_{\text{fut}}$ is the same for these models across batches. However, since the recurrent models can handle sequences of different lengths, we allow for forecasting different length futures across batches resulting in a few more training batches. Following the training practices suggested by Le et al. [76], we construct the context set at training as a random

Table 6: Architecture Hyperparameters for the Haggling Dataset Experiments.

Hyperparameter	**NP**	**SP-MLP**	**SP-GRU**
Sequence Encoder/Decoder			
Number of layers	2	2	1
Hidden dim	416	64	320
Partner Pooler $\psi(\boldsymbol{x}_j)$			
Number of MLP layers	—	2	2
MLP hidden dim	—	64	64
Output dim	—	32	32
$\boldsymbol{z}$ **Encoder**			
Number of layers	2	2	2
Hidden dim	64	64	64
Representations			
$\boldsymbol{e}, \boldsymbol{r}, \boldsymbol{s}, \boldsymbol{z}$ dim	64	64	64
Multi-Head Attention			
Query/Key dim	32	32	32
Number of heads	8	8	8
Number of parameters in *-latent* variant	2.8M	2.2M	3.0M

subset of the batch. Consequently, we further constrain samples in a batch to correspond to the same interacting group (see Section 2 for the underlying meta-learning intuition). For the same reason, we also ensure that a batch contains unique observed sequences, so that a single observed sequence does not dominate the aggregation of representations over context. This is because a single observed sequence has multiple associated future sequences at different offsets, and could show up multiple times in a batch through random sampling if not handled explicitly.

We optimize the models using Adam [77]. For the NP and SP-MLP models we use a batch size of 128, an initial learning rate of $3\mathrm{e}{-5}$, and a weight decay of $5\mathrm{e}{-4}$, and a dropout rate of 0.25. For the MLP-GRU models we use a batch size of 64, an initial learning rate of $1\mathrm{e}{-5}$, and a weight decay of $1\mathrm{e}{-3}$. The entire system was implemented using Pytorch [78] and Pytorch Lightning [79]. Every model was trained on a single NVIDIA GPU on an internal cluster depending on availability; one of Geforce GTX 970 (4 GB) or 1080 (8 GB), or Quadro P4000 (8 GB).

We validate the hyperparameters using 5-fold cross-validation, in the *random* context regime. At test, we use the same context sequences across models for fair comparison. The final model parameters for testing are obtained by averaging the parameters from the five best models during training. All testing was done with a batch size of 128 for consistency. All evaluation metrics are computed after destandardizing the location dimensions (orientation is already denoted by a unit quaternion, and therefore not standardized). The predicted std. deviations are scaled by the same value as the predicted means during destandardization.

D Additional Dataset Details

D.1 Synthesized Glancing Behavior Dataset

The set of pristine sinusoids representing the *Type I* glances is computed by evaluating the sine function at the bounds of 19 equally spaced partitions of $[0, 3\pi + \phi)$, for phase values ϕ in $[0, 2\pi)$ with a step size of 0.001. More concretely, this is the set

$$g = \{r : r = \sin(x),\ x = n{\times}(3\pi{+}\phi)/19,\ n \in \{0, 1, \dots 19\},\ \phi = p{\times}0.001,\ p \in \{0, 1, \dots 6283\}\},$$

which results in 6284 sequences. The *Type III* glances are represented by identical sinusoids with clipped amplitudes for the last six timesteps, resulting in the final dataset of 12568 sequences. We train with batches of 100 sequences, using a randomly sampled 25 % of the batch as context. For evaluation, we fix 785 randomly sampled phase values as context for all models. For each phase, samples corresponding to both types of glances are included in the context set, effectively using 25 % of all samples as context at evaluation.

D.2 Preprocessing the Panoptic Haggling Dataset

We begin by converting the orientation normals into unit quaternions. While quaternions afford many benefits over other representations of rotation, their one downside is that they are not injective—the quaternion $\mathbf{q}$ denotes an identical rotation to $-\mathbf{q}$. We address this by constraining every first quaternion of a sequence to the same hemisphere in quaternion space. To ensure smooth interpolation, the quaternion at every subsequent frame is chosen to be the one in $\{\mathbf{q}_t, -\mathbf{q}_t\}$ that is the shortest distance from $\mathbf{q}_{t-1}$ along the unit hypersphere. As discussed in Section 5,we then split the interaction data into pairs of $\boldsymbol{t}_{\text{obs}}$ and $\boldsymbol{t}_{\text{fut}}$ windows to construct the samples for forecasting. Motivated by the domain focus on the organization of turn-taking, we consider window lengths of 2 seconds supported by dataset statistics and literature. The dataset duration of contiguous speech follows a mean of 2.13 s ($\sigma = 2.61$ s), which is close to the mean measure of 1.68 s found in turn-taking analysis [20, 80]. We generate sliding windows with an overlap of 0.8, constraining the offset between $\boldsymbol{t}_{\text{obs}}$ and $\boldsymbol{t}_{\text{fut}}$ to a maximum of 5 s. This is to roughly restrict candidate future windows to those starting after two turn changes. In total, we obtain about 140K observed-future sequence pairs for training, and about 40K pairs for testing. We standardize the location features to have zero mean and unit variance, using the train statistics to standardize the test sets.