# Resonance: Learning to Predict Social-Aware Pedestrian Trajectories as Co-Vibrations

Supplementary Material

# Appendix

## A. Metrics and Other Technical Details

We use the best Average/Final Displacement Error over $K$ trajectories (minADE$_K$/minFDE$_K$) to measure performance [1, 6]. In our main manuscript, we abbreviate them as "ADE" and "FDE" to save space. For an ego agent $i$, denote the $k$th model output as $\hat{\mathbf{Y}}_k^i$, and the predicted position on the $t$th future frame as $\hat{\mathbf{p}}_{k\,t}^{\,i}$, we have

$$\text{minADE}_K(i) = \min_{1 \leq k \leq K} \frac{1}{t_f} \sum_{t=t_h+1}^{t_h+t_f} \left\| \mathbf{p}_t^i - \hat{\mathbf{p}}_{k\,t}^{\,i} \right\|_2, \quad (1)$$

$$\text{minFDE}_K(i) = \min_{1 \leq k \leq K} \left\| \mathbf{p}_{t_h+t_f}^i - \hat{\mathbf{p}}_{k\,t_h+t_f}^{\,i} \right\|_2. \quad (2)$$

In addition, due to page limitations, some technical details are not presented in the main manuscript. These parts are unrelated to our contributions but may be used to reproduce our work:

The linear trajectory (including both the linear fit and the linear base) will be translated by adding a constant vector to ensure that it can intersect with the observed trajectory at the current observation moment ($t = t_h$) to maintain continuity. When computing the differential feature in the original Eq. (5), the outer product is also used to enhance the modeling capacities of trajectory spectrums in embedding networks $N_e$ and $N_{e,l}$ [19]. Also, following previous works [10], category labels of agents have been encoded through one simple linear layer when embedding their trajectories only in nuScenes to learn the significant interclass differences of heterogeneous agents (vehicles).

In the original Eq. (8), we use linear-speed interpolation to obtain the final forecasted self-biases since it could be difficult for either pedestrians or vehicles to make sudden changes in their velocities (amplitude and direction) while in motion. This means that when computing, we add the condition of equal velocities to the left and to the right of the interpolation keypoints.

Please also note that our used eth split from the ETH-UCY is actually the 6-frame-sampled version, similar to those used in SR-LSTM [24]. Other splits {hotel, univ*, zara*} used are still sampled with a 10-frame interval. The original eth video is accelerated. It means that only under this 6-frame sampling interval the observation and prediction periods are aligned with other ETH-UCY scenes (videos). Some researchers have used the 10-frame-sampled version to train their models and report their results, but we cannot make a clear distinction at this time because some of the researchers have not contributed open-source repositories, thus our reported results are their reported ones in their papers. We have provided our python script to generate 10-frame-sampled eth dataset in our code repo for training *Re* models that could be trained and compared with these approaches directly.

| Method | Performance (SDD) | $t_1$ | $t_{1\text{k}}$ | Parameter |
|---|---|---|---|---|
| E-V$^2$-Net [19] | 6.57/10.49 | 28 | 112 | 1,976,864 |
| SocialCircle [17] | 6.54/10.36 | 34 | 119 | 1,989,536 |
| SocialCircle+ [18] | 6.44/10.22 | 41 | 126 | 1,990,177 |
| *Re* (variation a2) | 6.93/10.26 | 29 | 91 | 2,046,428 |
| *Re* (variation a8) | 6.33/10.45 | 34 | 211 | 1,242,924 |
| *Re* (full) | 6.27/10.02 | 56 | 263 | 3,149,192 |

Table 1. Comparisons of the average inference times (on an Apple M1 Mac mini (2020, 8GB Memory)) under the batch size of 1 and 1000 (denoted as $t_1$ and $t_{1\text{k}}$ correspondingly, reported in milliseconds), and parameter amount.

Please refer to our code repo (`https://github.com/cocoon2wong/Re`) to learn how these details are implemented.

## B. Model Efficiency Analyses

We report the inference times and parameter amount of the proposed *Re* in comparison to several newly published approaches in Tab. 1. Due to the *vibration-like* prediction strategy, two trajectory biases should be forecasted (through two mirrored Transformer backbones) when making the final prediction. This means that the time-space efficiency of *Re* is naturally a bit disadvantaged compared to other methods, requiring roughly twice as much parameterization and inference time as other methods.

By comparing several newly published methods that obtained similar prediction performance, it can be seen that *Re* still has considerable time-space efficiency. Please note that these efficiency experiments are conducted on one Apple M1 Mac mini (2020, 8GB Memory), whose computing performance is roughly the same as current (2024) smartphones. Computations on such devices may be more in line with trajectory prediction application scenarios, providing a more valuable reference for efficiency analyses, rather than on high-performance server clusters. Whether the batch size is 1 or 1000, it still could forecast trajectories within the *low-latency* [9] thresholds, *i.e.*, it could forecast trajectories within each adjacent sampling interval

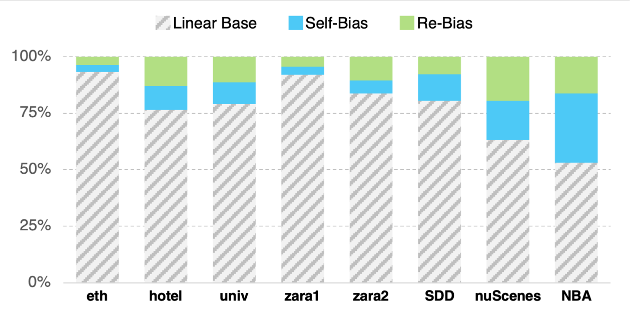

Figure 1. Energy shares of trajectory biases on different datasets.

$\Delta t = 0.4$ seconds, even on the Apple M1 that performs similar to current iPhones. Notably, its variation a2 (predict linear base + self-bias, see the original Tab. 4) takes only about 30% inference time of the full model, still with acceptable quantitative performance, especially with better FDE than E-V$^2$-Net and SocialCircle. In addition, another variation, a8 (predict linear base + re-bias), owns the minimum number of parameters and also achieves better ADE than these baselines, indicating comparable prediction performance. Therefore, depending on the application's needs, whether it is a fast calculation or an accurate one, the proposed *Re* method may cope with it.

# C. Additional Discussions on Trajectory Biases

## C.1. Contributions of Vibrations (Biases)

In the main manuscript, we have analyzed different trajectory biases and their spatial distributions in a visualized way. Here, we further discuss how these bias terms contribute quantitatively to final forecasted trajectories. We use the energy of trajectories as a measurement to simplify the calculation. It is computed as the square sum of each point in the trajectory $\mathbf{X}$, *i.e.*, the $\|\mathbf{X}\|^2$. Fig. 1 reports these biases' average percentage energy shares across different datasets. It can be seen that the linear base occupies the most energy, while other terms may change over datasets. For example, the linear base occupies less energy on SDD, nuScenes, and NBA than those on ETH-UCY, indicating that there might be more non-linear trajectories, especially those caused by socially interactive behaviors.

This phenomenon aligns with previous research [11] that the simulation of trajectories themselves (the non-interactive terms, *i.e.*, the linear base plus the self-bias in the proposed *Re*) has become the most important optimization objective for most trajectory prediction networks. Under this circumstance, we notice that the re-bias (under social excitations) has not been compressed significantly, particularly in comparison to the self-bias (Newtonian excitations). Instead, self-bias and re-bias provide almost identical contributions. This implies that the predicted trajectories have been decomposed into distinct biases that hold approximately the same energy (instead of being drowned out by other terms) except for their linear portions, thus roughly validating the usefulness of these biases.

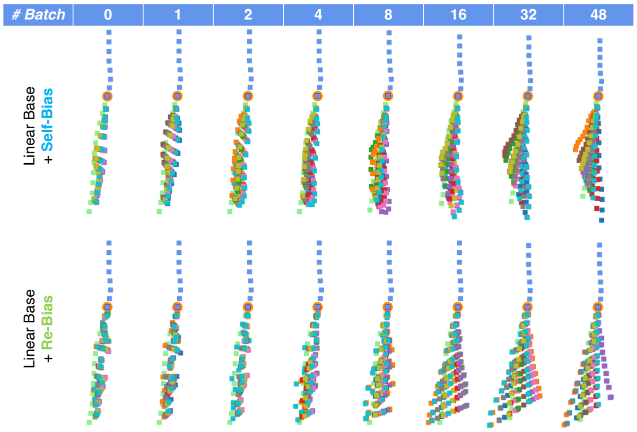

Figure 2. The changes of self-biases and re-biases during training (after how much training batches before finishing training 1 epoch of the whole dataset).

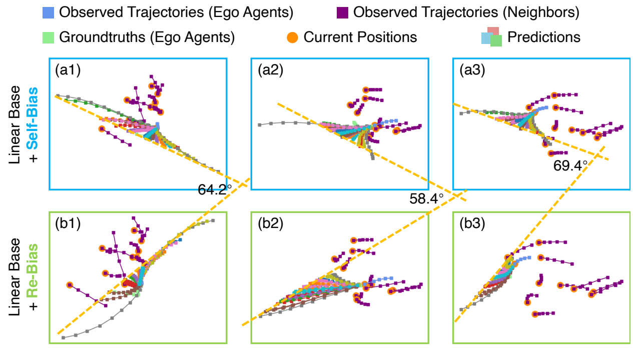

Figure 3. Visualized trajectory biases ($K = 20$) on NBA. Different from other datasets, we found that self-biases and re-biases vibrate in directions about $\pi/3$ (60 degrees) away from each other.

## C.2. Spatial Distributions of Vibrations

We have concluded in the main manuscript that self-bias and re-bias vibrate almost vertically in different directions. We now further discuss this phenomenon. Please note that this phenomenon is different from our angle-based approach to gather resonance features when forecasting re-biases, even though words like "angles" or "directions" are evolved in both these approaches. Such a vertically-vibrating phenomenon is directly caused by our *vibration-like* trajectory prediction strategy that forecasts trajectories with multiple trajectory biases. For the convenience of representation, we define the *vibration direction* of a trajectory bias as the acute angle between the fitting line of the predicted $K = 20$ trajectory bias points on the last predicted

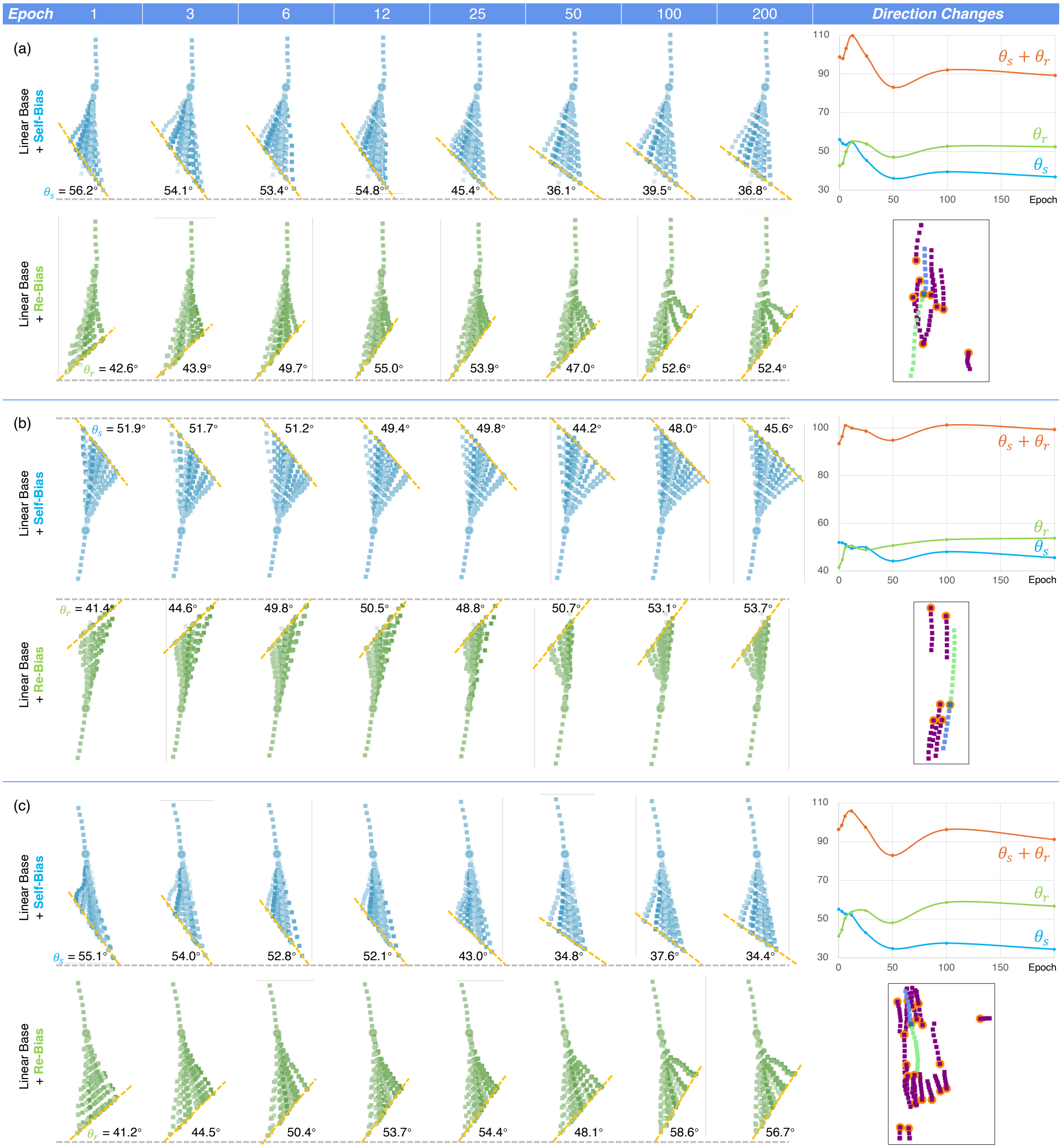

Figure 4. Vibration directions (defined as the acute angle between the fitting line of the predicted $K = 20$ trajectory bias points on the last predicted moment and the horizontal direction, including the angle of self-biases $\theta_s$ and the angle of re-biases $\theta_r$) on different epochs when training on the pedestrian dataset ETH-UCY. We can consider self-bias and re-bias to be vibrating vertically when $\theta_s + \theta_r$ approaches $\pi/2$.

moment and the horizontal direction. Thus, we have angles $\theta_s$ and $\theta_r$ to represent how self-biases and re-biases vibrate.

Fig. 4 shows how angles $\theta_s$ and $\theta_r$ change with training epochs during one training progress. It can be seen that each angle, whether $\theta_s$ or $\theta_r$, gradually converges to a particular value for each sample during training. These convergence processes may be followed by numerical oscillations, and the convergence values may not be the same for every sample. Especially, we observe that the convergence value of $\theta_s + \theta_r$ is around $\pi/2$ (90 degrees) for most pedestrian samples, which is our described vertically-vibrating phenomenon in the main manuscript. In addition, it also shows that the nonlinearities of these trajectory biases have been reassigned during training. For example, we can see stronger nonlinearities in the predicted self-biases after 3 training epochs than those in re-biases in Fig. 4 (a) and (c). However, self-biases become more linearly distributed over time (as expected when wiring the Transformer $T_s$). Such nonlinearities seem to be *taken over* by the other re-biases, better seen by comparing the forecasted re-biases at epochs 50 to 200 in Fig. 4 (a) and (c).

Fig. 2 represents how these trajectory biases are distributed and how they change during training at several initial training steps (after how many batches of training data). It can be seen that these biases are almost randomly distributed at the very first training step, with none of the linearity/nonlinearity or vertically vibrating properties presented. We also observe that it is the re-bias that leads the training. For example, the phenomenon that predicted points in re-biases distributed around a straight line has already appeared after 16 batches in Fig. 2, while it appears for self-biases until training after 32 batches.

In addition to our descriptions of the original Fig. 8 in the main manuscript, the changes in these biases can also be roughly explained as a two-player cooperative game. In detail, our goal is to learn to forecast two trajectory biases (the linear base is not trainable), $\Delta\hat{\mathbf{Y}}^i_s$ and $\Delta\hat{\mathbf{Y}}^i_r$, to fit the given trajectory $\mathbf{Y}^i - \hat{\mathbf{Y}}^i_l$. Also, these two biases share almost equal positions in the prediction network, since whether their mirrored Transformer structures or they are exactly the same additive weight (both equal to 1, since $\hat{\mathbf{Y}}^i = \hat{\mathbf{Y}}^i_l + \Delta\hat{\mathbf{Y}}^i_s + \Delta\hat{\mathbf{Y}}^i_r$). Intuitively, these biases would not have the same amplitude and would not vibrate in the same direction, as this would cause the whole network to get confused and not be able to distinguish samples that have similar ego trajectories but with different neighbors and different future trajectories, since self-biases are learned and predicted only by the ego agent, whereas re-biases are by the ego agent and all neighbors together. In other words, only re-biases could *tell* such socially different differences. This means that these two biases are more inclined to vibrate in different directions, and re-biases would be the first ones to distinguish different training samples, where the differences of their future trajectories are mostly caused by their differently distributed neighbors. Under this circumstance, the differences between self-biases and re-biases may increase as the training progresses. Thus, after a period of cooperation, these biases would likely show the greatest difference in vibrational directions, leading to our observed vertically vibrating phenomenon.

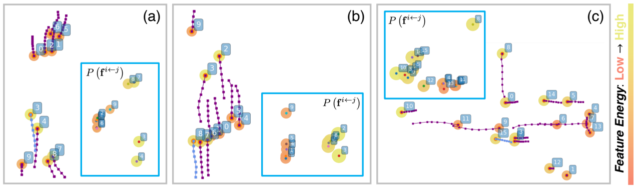

Figure 5. Further comparisons of resonance features in more complex interactive scenes. Colored dots represent the value of feature energy $\|\mathbf{f}^{i\leftarrow j}\|^2$, and each blue box represents resonance features in the feature space $\{\mathbf{f}^{i\leftarrow j} | 1 \leq j \leq N_a, j \neq i\}$.

Note that these explanations are only meant to provide a more intuitive understanding of two trajectory biases vibrating vertically to each other and are not strict proofs. After our validation, similar phenomena have been presented on pedestrian datasets ETH-UCY and SDD. However, as we show in the original Fig. 7 (a6) and (c6) in the main manuscript, most NBA samples do not match this rule, with about 60 degrees of direction differences between these two trajectory biases. We attempted to explain this phenomenon by assuming that in NBA scenarios, where linear bases are difficult to apply (due to the rapidly changing motion states of the players and the intent of the game), there may actually be three *actual* trajectory biases, and thus the best training is achieved when these three biases vibrate at a 60-degree difference from each other. However, two *actual* biases were estimated simultaneously in self-biases or re-biases, thus causing this vibration phenomenon with a 60-degree difference between each other (as shown in Fig. 3). This phenomenon still needs further study.

## D. Further Discussions on Resonance Features and Social Interactions

### D.1. Distributions of Resonance Features

In the main manuscript, we have analyzed resonance features of different neighboring agents $j$ relative to the ego $i$ and have observed a clear tendency to describe all neighbors in distinct groups. However, analyzing high-dimensional features using the energy metric results in a large amount of information loss and only partially reflects their nature. In Fig. 5, we use PCA (Principal Component Analysis) to reduce the dimension of each $\mathbf{f}^{i\leftarrow j}$, thus further visualizing how they distribute in the feature space. It can be seen that agents with similar motion states share closer features. Meanwhile, agents with similar features may probably behave as groups in real scenarios, like the group {0, 1, 2, 5, 6} in Fig. 5 (a) and {0, 1, 4, 5} in Fig. 5 (b), independent of their relevant positions (since the relative positional information has been detached in these features, see the original Eq. (9)). It is worth noting whether or not following a group is actually an important social event and has been studied a lot by previous researchers like GroupNet [20]. The proposed *Re* could learn such behaviors without constructing graph structures and human annotations, further verifying our thought that social interactions are associated with the spectral properties of trajectories.

Also, we can see from Fig. 5 that agents with different preferences could be distinguished, like bikers $\{6, 7, 9, 11\}$ and $\{8\}$ owns remarkable differences to other pedestrians in Fig. 5 (c). This means that the proposed *Re* has the ability to learn different social responses of the same ego agent to neighbors with different states. It also indicates that *Re* has learned social behaviors that are not limited to finding or locating which neighbors have similar spectral characteristics to itself, but focuses more on those with different spectrums to itself, like the feature distances between group $\{0, 1, 2, 5, 6\}$ and others in the feature space in Fig. 5 (a). Thus, our assumptions about the *social* resonance can be verified, which regards that social interactions are associated with the spectral properties of trajectories, and it could produce the maximum effects when the trajectory spectrums of a neighbor and the ego agent both cover some common frequency bands (*i.e.*, higher spectral similarities) or no spectral overlap at all (*i.e.*, have a longer feature distance in the feature space).

### D.2. Resonance Features and Neighbor Positions

As described in the original Eq. (9), the relative positional information of all neighbors relative to the ego agent has not been considered in resonance features. In other words, we regard the resonance features to represent the *pure* spectral properties of agents' trajectories, since it can easily be verified that a positional offset introduces more low-frequency (or base frequency) interference into the corresponding spectrum for the linear transform $\mathcal{T}$ we used. Instead, in the original Eq. (11), we encode the corresponding positional information outside from the resonance feature $\mathbf{f}^{i\leftarrow j}$ as the $\mathbf{f}_p^{i\leftarrow j}$. In our main manuscript, we have used term $c(x, y|\mathbf{X}^m)$ to verify how the resonance feature and the positional information collaborate to *modify* the final predicted trajectories:

Define the set of observed trajectories of all agents as $\mathcal{X} = \{\mathbf{X}^i | 1 \leq i \leq N_a\}$, and the computation of a prediction network as $\mathcal{N}(\cdot)$, its prediction for the $i$th ego agent can be represented by

$$\hat{\mathbf{Y}}^i := \left(\hat{\mathbf{p}}^i_{t_h+1}, ..., \hat{\mathbf{p}}^i_{t_h+t_f}\right)^\top = \mathcal{N}(\mathcal{X}). \tag{3}$$

Considering a manual neighbor [17], *i.e.*, a neighbor with manually set observed trajectories and has been put manually into the prediction scene, located at $(x, y)$ with the *pure* trajectory $\mathbf{X}^m$, we denote the newly predicted trajectory after applying the *intervention* $do(\overline{\mathcal{X}} = \mathcal{X} \cup \{\mathbf{X}^m + (x, y)^\top\})$ as

$$\overline{\hat{\mathbf{Y}}^i} := \left(\overline{\hat{\mathbf{p}}}^i_{t_h+1}, ..., \overline{\hat{\mathbf{p}}}^i_{t_h+t_f}\right)^\top = \mathcal{N}(\overline{\mathcal{X}}). \tag{4}$$

Then, the absolute trajectory modification $c(x, y|\mathbf{X}^m)$ caused by the new trajectory $\mathbf{X}^m + (x, y)^\top$ is defined as

$$c(x, y|\mathbf{X}^m) = \max_{t_h+1 \leq t \leq t_h+t_f} \left\| \overline{\hat{\mathbf{p}}}^i_t - \hat{\mathbf{p}}^i_t \right\|. \tag{5}$$

This is actually an intervention to verify the model's ability to represent *causalities* [5] between social interactions and predicted trajectories. If the model cannot represent such a causal relationship, then the predicted trajectories may not change no matter how much the new neighbor changes.

Such modification describes how the *originally* predicted trajectories are modified by additionally considering a new neighbor agent and potential socially interactive behaviors with it. For clarity, we abbreviate such modifications as *social modifications* in the following sections. Thus, by manually moving (translating) the trajectory $\mathbf{X}^m$ to any position $(x, y)$ around the ego agent, we can verify how the resonance feature describes a neighbor with different positions relative to the ego agent. Spatial distributions of $c(x, y|\mathbf{X}^m)$ corresponding to more structured manual trajectories ($\mathbf{X}^m$) are attached in Fig. 6, which represent how resonance features (how the manual trajectory is structured) and positional information (where the manual neighbor is located, and its relative position to other neighbors) collaborate to modify the original forecasted trajectories.

### D.3. Additional Discussions on Social Modifications

The above trajectory modification $c(x, y|\mathbf{X}^m)$ could represent how a trajectory prediction model responds to change its predictions as a consideration of social interactions. Naturally, the representation of social behaviors can be considered better when such social modifications of a model are more sensitive and consistent with human intuitions. Fig. 7 presents the spatial distribution of social modifications $c(x, y|\mathbf{X}^m)$ obtained from one of the current state-of-the-art approaches SocialCircle[17], which is proposed to mainly focus on social interactions among agents.

Comparing Fig. 7 and Fig. 6, it can be seen that SocialCircle fails at capturing the state of this manual neighbor, especially how its trajectory is structured. For example, the spatial distribution of social modifications is almost the same across Fig. 7 (a1), (a2), and (a3), even though the trajectories of these manual neighbors are totally different. On the contrary, the proposed *Re* could better capture their differences through the resonance features, better seen by comparing Fig. 6 (b2) (b5) *v.s.* Fig. 7 (b1) (b3) that share the same prediction situation. It can be seen that the former pair present totally different social responses, especially to the left side of the ego agent, while the other

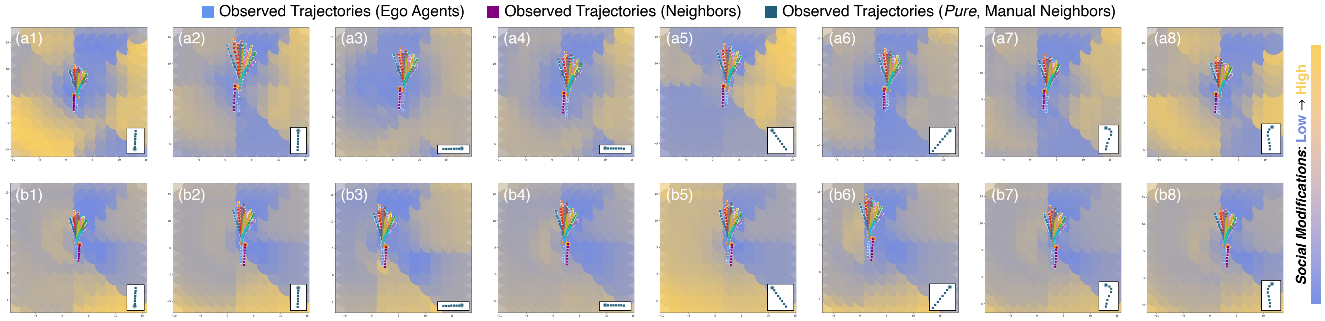

Figure 6. Spatial distribution $P(c(x, y|\mathbf{X}^m))$ of more structured of manual neighbors of the proposed *Re* model.

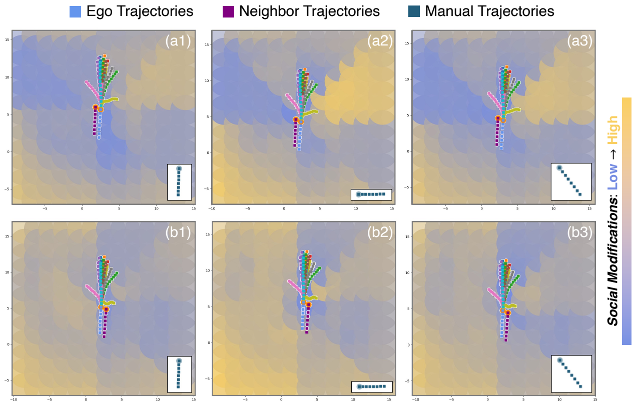

Figure 7. Spatial distribution $P(c(x, y|\mathbf{X}^m))$ of more structured of manual neighbors of SocialCircle[17] model.

pair shares almost the same spatial distribution. In addition, the differences of social modifications on different ego agents provided by the SocialCircle is also relatively small, while those *Re* ones could better describe such interaction differences, better seen by comparing Fig. 6 (a5) and (b5) that present an almost completely opposite spatial distribution. Therefore, we can conclude that the proposed *Re* could better capture the fine-level interaction differences among different egos and neighbors, thus forecasting better social-aware trajectories.

# E. Additional Ablation Studies

In the main manuscript, we have made ablation variations to verify the effect of different trajectory biases in the proposed *Re*. However, due to the page limitations, some other hyperparameters, like the number of waypoints $N_{way}$, the number of angle partitions $N_\theta$, or the transform $\mathcal{T}$, have not been validated, since they are set following the experimental results of former researches [16, 17, 19]. Here, we conduct several ablation variations and report their corresponding ablation results. Please note that these validations are only to determine the parameter choices since these parts are not the main contributions of the proposed *Re*.

## E.1. Ablation: Spectrums and Transform Types

Our core idea is to forecast trajectories as different vibration portions and regard that social interactions are associated with the spectral properties of agents' trajectories. Naturally, we need to use some transform to get trajectory spectrums. The Fourier transform and its variations have been used in a wide variety of fields. E-V$^2$-Net [19] applies discrete Fourier transform (DFT) and discrete Haar transform to obtain trajectory spectrums so that the trajectories can be predicted hierarchically via different spectral components.

According to their experimental results, Haar transform has better time efficiency than DFT, and it could also better describe signals with rapidly changing characteristics (with a relatively lower *Vanishing Moment*). Thus, we choose Haar transform to obtain trajectory spectrums in this manuscript. We have also conducted several ablation experiments to verify its usefulness compared to the non-transform ones and the DFT, whose results are reported in Tab. 2. We can see that the Haar variation b2 obtains the best performance across these ETH-UCY sets, compared to whether DFT variation b1 or the non-transform variation b3. Especially, such performance enhancements appear to be more effective on FDE, with about 6.5% better average FDE than variation b1, and about 5.4% better than variation b3. Such experimental results verify the usefulness of the Haar transform. Since we are not the first to work on forecasting trajectories with Haar transforms, we think this part of the analysis is not the necessary for the main manuscript.

## E.2. Ablation: the Number of Waypoints $N_{way}$

We regard self-bias as roughly reflecting agents' intention changes or random behaviors on limited waypoints rather than the whole prediction period. Our idea is to reduce the computational load by such an operation and improve the network's generalization to prevent it from overfitting to some training samples when forecasting agents' random behaviors. According to previous research [12, 16], a too-low waypoint setting may lead to the loss of the accuracy of

| ID | $\mathcal{T}$ | eth | hotel | univ | zara1 | zara2 |
|---|---|---|---|---|---|---|
| b1 | DFT | 0.240/0.359 | 0.112/0.174 | 0.258/0.459 | 0.181/0.306 | 0.137/0.237 |
| b2 | Haar (default) | 0.232/0.354 | 0.103/0.152 | 0.242/0.414 | 0.172/0.290 | 0.131/0.225 |
| b3 | None | 0.238/0.364 | 0.111/0.178 | 0.248/0.436 | 0.177/0.302 | 0.138/0.237 |

Table 2. Ablation studies on transforms used to obtain trajectory spectrums, including Haar (default), DFT, and none transform.

| ID | $N_{way}$ | eth | hotel | univ | zara1 | zara2 |
|---|---|---|---|---|---|---|
| c1 | 2 | 0.249/0.384 | 0.104/0.153 | 0.276/0.475 | 0.181/0.301 | 0.137/0.234 |
| c2 | 3 | 0.236/0.362 | 0.102/0.150 | 0.253/0.446 | 0.178/0.299 | 0.131/0.223 |
| c3 | 4 (default) | 0.232/0.354 | 0.103/0.152 | 0.242/0.414 | 0.172/0.290 | 0.131/0.225 |
| c4 | 6 | 0.229/0.356 | 0.102/0.154 | 0.250/0.441 | 0.182/0.310 | 0.135/0.229 |
| c5 | 12 | 0.237/0.378 | 0.104/0.155 | 0.245/0.437 | 0.180/0.307 | 0.134/0.232 |

Table 3. Ablation studies on the number of waypoints $N_{way}$ when forecasting self-biases.

intention prediction, and conversely, a too-high setting may limit the ability of prediction models to forecast stochastic trajectories. Tab. 3 reports the ablation results on changing the $N_{way}$. It can be seen that *Re* obtains the best average performance when setting $N_{way}$ to 4. Also, the usage of waypoints is not our main contribution. Thus, this ablation table is not included in the main manuscript.

### E.3. Ablation: Social-Interaction-Representation

One significant concern is representing agents' interaction context when making predictions. Its core is to properly share motion status with all neighboring agents. In *Re*, we use the resonance feature $\mathbf{f}^{i \leftarrow j}$ to represent the single-directional relationship caused by a neighbor $j$ onto the ego agent $i$. Then, the angle-based resonance gathering is used to gather features into the ego agent as the final social interaction representation, *i.e.*, the resonance matrix $\mathbf{F}^i_{\mathcal{R}}$. This information-gathering approach is inspired by the previous work SocialCircle [17]. We have also conducted ablation studies to verify its effectiveness to other social-interaction-representation approaches, including the Social Pooling [1], graph convolutional network [7], and the vanilla SocialCircle. Results are reported in Tab. 4.

### E.4. Ablation: the Number of Angle Partitions $N_\theta$

Like SocialCircle [17], we use an angle-based resonance gathering method to gather all neighbors' resonance features in each angle-based partition. According to their experimental results, setting $N_\theta = t_h$ may obtain the best prediction performance. Here, we conduct ablation variations to verify how the number of partitions $N_\theta$ affects model performance. Results are reported in Tab. 5. We observe that *Re* obtains the best performance when setting $N_\theta = 8$ $(= t_h)$, indicating similar conclusions of angle-based partitions in SocialCircle. Also, the angle-based partitioning is only used to gather resonance features, which is not the main concern of this manuscript. Thus, these results are only used as a reference for parameter selection.

## F. Further Discussions on the Vehicle Dataset

The proposed *Re* model is designed to forecast pedestrian trajectories. Furthermore, we now discuss and analyze its prediction performance quantitatively and qualitatively on the large-scale vehicle dataset **nuScenes** [2, 3] collected in urban cities. We split 550/150/150 scenes to train/test/val only on vehicles, under $(t_h, t_f, \Delta t) = (4, 12, 0.5)$ [8, 13]. This section further discusses how *Re* works to forecast vehicle trajectories, especially the usefulness of the *vibration-like* trajectory prediction strategy.

### F.1. Quantitative Analyses

As shown in Tab. 6, we observe that *Re* achieves an impressive performance compared to the state-of-the-art methods. Especially, *Re* outperforms the newly published model SoperModel by 42.9%/40.6% ADE/FDE when generating 5 trajectories while the improvement in FDE even reaches as high as 60.3% when generating 10 trajectories. In addition, compared to the pedestrian-focused SocialCircle, *Re* obtains about 8.8%/7.8% ADE/FDE improvements when generating 10 trajectories. Nevertheless, *Re* loses about 1.0% FDE compared to the state-of-the-art vehicle-centric MUSE-VAE, but it obtains better performance when only generating 5 fewer trajectories for each ego agent, with about 8.7%/6.2% better performance. Also, in Tab. 7, variations present similar trends like those on NBA, reported the original Tab. 4, that *Re* perform even worse when adding linear bases in the final predictions. Variations a7 to a9 exhibit performance drops of up to 7.6%/12.9%, which are rarely seen in ETH-UCY and SDD. Nevertheless, vibrations (self-biases and re-biases) could still help to improve the performance (variation a6), especially compared to the SocialCircle variation a5. Similar to our conclusion in the main manuscript, though the assumption that linear base serve as the final reference points may not apply to the ve-

| ID | Interaction Representation | eth | hotel | univ | zara1 | zara2 |
|---|---|---|---|---|---|---|
| d1 | Social Pooling | 0.243/0.376 | 0.107/0.166 | 0.256/0.461 | 0.183/0.312 | 0.138/0.242 |
| d2 | GCN | 0.240/0.272 | 0.107/0.164 | 0.253/0.448 | 0.179/0.311 | 0.133/0.230 |
| d3 | SocialCircle | 0.238/0.370 | 0.112/0.181 | 0.252/0.444 | 0.180/0.303 | 0.139/0.241 |
| d4 | Resonance Gathering (Ours) | 0.232/0.354 | 0.102/0.152 | 0.242/0.414 | 0.172/0.290 | 0.131/0.225 |

Table 4. Ablation studies on the usages of different social interaction representations when forecasting re-biases.

| ID | $N_\theta$ | eth | hotel | univ | zara1 | zara2 |
|---|---|---|---|---|---|---|
| e1 | 1 | 0.237/0.363 | 0.102/0.153 | 0.246/0.419 | 0.181/0.313 | 0.137/0.238 |
| e2 | 2 | 0.222/0.391 | 0.103/0.156 | 0.246/0.419 | 0.181/0.310 | 0.144/0.248 |
| e3 | 4 | 0.232/0.350 | 0.101/0.154 | 0.250/0.416 | 0.179/0.303 | 0.139/0.235 |
| e4 | 8 (default) | 0.232/0.354 | 0.102/0.152 | 0.242/0.414 | 0.172/0.290 | 0.131/0.225 |
| e5 | 12 | 0.249/0.390 | 0.103/0.159 | 0.254/0.426 | 0.178/0.299 | 0.145/0.250 |
| e6 | 16 | 0.253/0.391 | 0.104/0.160 | 0.252/0.429 | 0.178/0.295 | 0.146/0.247 |

Table 5. Ablation studies on the number of angle-based partitions $N_\theta$ when forecasting re-biases.

| Method (nuScenes) | *best-of-5* ↓ | *best-of-10* ↓ |
|---|---|---|
| Trajectron++[14] (2020) | 3.14/7.45 | 2.46/5.65 |
| Y-net[12] (2020) | 2.46/5.15 | 1.88/3.47 |
| SoperModel[22] (2025) | 2.21/4.58 | 1.79/3.40 |
| AgentFormer[23] (2021) | 1.86/3.89 | 1.45/2.86 |
| Dice[4] (2024) | 1.76/3.70 | 1.44/2.67 |
| AgentFormer-FLN[21] (2024) | 1.83/3.78 | 1.32/2.73 |
| E-V$^2$-Net[19] (2023) | 1.46/3.18 | 1.15/2.37 |
| SocialCircle[17] (2024) | 1.44/3.10 | 1.13/2.30 |
| MUSE-VAE[8] (2022) | 1.38/2.90 | 1.09/2.10 |
| *Re* (Ours) | 1.26/2.72 | 1.03/2.12 |

Table 6. Comparisons to other state-of-the-art methods on nuScenes. Metrics reported are "ADE/FDE" in meters under *best-of-5* and *best-of-10*.

hicle scenes, vibrations and their sampled trajectory-biases still work continuously. These results validate the competitiveness of *Re*, even in vehicle scenarios.

## F.2. Qualitative Analyses

According to results in the original Figs. 7 and 8 in the main manuscript, or the above Fig. 4, we conclude that self-biases are better at capturing intention changes or random path choices, represented as an additional vibration vertical to the direction of motion, while re-biases capturing social modifications as a vibration alongside the motion direction. As shown in Fig. 8, we observe that self-biases and re-biases vibrate in a different way when forecasting vehicle trajectories from those presented in pedestrian datasets. It can be seen that self-biases now describe how ego vehicles' velocities would change, while re-biases describe whether they would turn in the future. For example, Fig. 8 (a1) to (a5) indicate that different forecasted self-biases own different velocities (among $K = 10$ random predictions), while they are almost distributed in the same direction, different from those in pedestrian cases. Also, the forecasted re-biases indicate whether the vehicle will change its direction in the future (like to make a turn at an intersection in Fig. 8 (b2) and (b4)). This phenomenon demonstrates the adaptability and effectiveness of the proposed vibration-like prediction strategy in capturing and predicting vehicle trajectories. In particular, the difference in trajectory biases between pedestrian data and vehicle data further illustrates the adaptability of the proposed *Re* to learn to fit trajectories according to the properties of ego agents.

Please note that the proposed *Re* currently uses only observed trajectories (ego and neighbors) to predict future trajectories and does not include other kinds or modals of observations, which leads to its shortcomings in forecasting vehicle trajectories, like not being able to observe lane positions (which are already widely used by existing vehicle trajectory prediction methods [15]), etc. Therefore, although it shows the potential for vehicle trajectory prediction, the current approach is more applicable to pedestrian agents. We will try to adapt it in the future to better suit vehicle prediction scenarios.

## References

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 961–971, 2016.

[2] 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. *arXiv preprint arXiv:1903.11027*, 2019.

[3] 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 multi-

| ID | $l$ | $\Delta_s$ | $\Delta_r$ | nuScenes ↓ | ID | $l$ | $\Delta_s$ | $\Delta_r$ | nuScenes ↓ |
|---|---|---|---|---|---|---|---|---|---|
| a1 | ✓ | × | × | 3.48/7.93 | a6 | × | ✓ | Re | 1.03/2.12 |
| a2 | ✓ | ✓ | × | 1.26/2.36 | a7 | ✓ | × | SC | 1.13/2.45 |
| a3 | × | × | SC | 1.09/2.30 | a8 | ✓ | × | Re | 1.13/2.44 |
| a4 | × | × | Re | 1.04/2.16 | a9 | ✓ | ✓ | SC | 1.06/2.23 |
| a5 | × | ✓ | SC | 1.06/2.24 | a0 (default) | ✓ | ✓ | Re | 1.05/2.17 |

Table 7. Ablation studies on nuScenes under *best-of-10*. "$l$", "$\Delta_s$", "$\Delta_r$" represent whether $\hat{\mathbf{Y}}_l^i$, $\Delta\hat{\mathbf{Y}}_s^i$, and $\Delta\hat{\mathbf{Y}}_r^i$ are superposed when training. "Re" and "SC" denote resonance gathering and SocialCircle [18] when learning $\mathbf{V}_r^i$. Results marked in Red denote worse ones than the best variation.

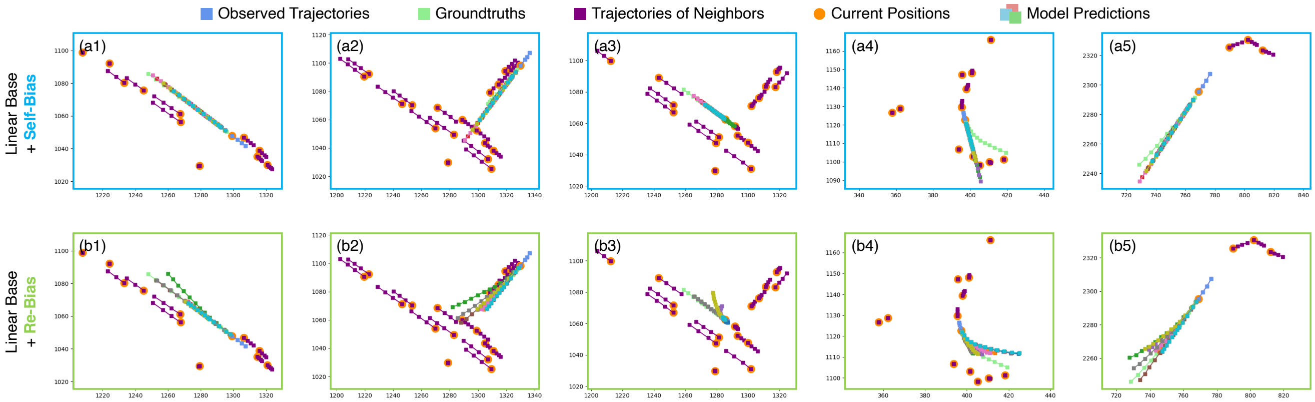

Figure 8. Visualized trajectory biases ($K = 10$) on nuScenes.

modal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[4] Younwoo Choi, Ray Coden Mercurius, Soheil Mohamad Alizadeh Shabestary, and Amir Rasouli. Dice: Diverse diffusion model with scoring for trajectory prediction. In *2024 IEEE Intelligent Vehicles Symposium (IV)*, pages 3023–3029. IEEE, 2024.

[5] Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society*, pages 424–438, 1969.

[6] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2255–2264, 2018.

[7] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[8] Mihee Lee, Samuel S Sohn, Seonghyeon Moon, Sejong Yoon, Mubbasir Kapadia, and Vladimir Pavlovic. Muse-vae: Multi-scale vae for environment-aware long term trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2221–2230, 2022.

[9] Shijie Li, Yanying Zhou, Jinhui Yi, and Juergen Gall. Spatial-temporal consistency network for low-latency trajectory forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1940–1949, 2021.

[10] Yuexin Ma, Xinge Zhu, Sibo Zhang, Ruigang Yang, Wenping Wang, and Dinesh Manocha. Trafficpredict: Trajectory prediction for heterogeneous traffic-agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6120–6127, 2019.

[11] Osama Makansi, Julius Von Kügelgen, Francesco Locatello, Peter Gehler, Dominik Janzing, Thomas Brox, and Bernhard Schölkopf. You mostly walk alone: Analyzing feature attribution in trajectory prediction. *arXiv preprint arXiv:2110.05304*, 2021.

[12] Karttikeya Mangalam, Yang An, Harshayu Girase, and Jitendra Malik. From goals, waypoints & paths to long term human trajectory forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15233–15242, 2021.

[13] Saeed Saadatnejad, Yi Zhou Ju, and Alexandre Alahi. Pedestrian 3d bounding box prediction. *arXiv preprint arXiv:2206.14195*, 2022.

[14] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Proceedings of the European conference on computer vision (ECCV)*, pages 683–700. Springer, 2020.

[15] Jingke Wang, Tengju Ye, Ziqing Gu, and Junbo Chen. Ltp: Lane-based trajectory prediction for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17134–17142, 2022.

[16] Conghao Wong, Beihao Xia, Ziming Hong, Qinmu Peng, Wei Yuan, Qiong Cao, Yibo Yang, and Xinge You. View vertically: A hierarchical network for trajectory prediction via fourier spectrums. In *European Conference on Computer Vision*, pages 682–700. Springer, 2022.

[17] Conghao Wong, Beihao Xia, Ziqian Zou, Yulong Wang, and Xinge You. Socialcircle: Learning the angle-based social interaction representation for pedestrian trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19005–19015, 2024.

[18] Conghao Wong, Beihao Xia, Ziqian Zou, and Xinge You. Socialcircle+: Learning the angle-based conditioned interaction representation for pedestrian trajectory prediction. *arXiv preprint arXiv:2409.14984*, 2024.

[19] Beihao Xia, Conghao Wong, Duanquan Xu, Qinmu Peng, and Xinge You. Another vertical view: A hierarchical network for heterogeneous trajectory prediction via spectrums. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[20] Chenxin Xu, Maosen Li, Zhenyang Ni, Ya Zhang, and Siheng Chen. Groupnet: Multiscale hypergraph neural networks for trajectory prediction with relational reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6498–6507, 2022.

[21] Yi Xu and Yun Fu. Adapting to length shift: Flexilength network for trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15226–15237, 2024.

[22] Heming Yang, Yu Tian, Changyuan Tian, Hongfeng Yu, Wanxuan Lu, Chubo Deng, and Xian Sun. Sopermodel: Leveraging social perception for multi-agent trajectory prediction. *IEEE Transactions on Geoscience and Remote Sensing*, 2025.

[23] 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 (ICCV)*, pages 9813–9823, 2021.

[24] Pu Zhang, Wanli Ouyang, Pengfei Zhang, Jianru Xue, and Nanning Zheng. Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12085–12094, 2019.