Fully Decoupling Trajectory and Scene Encoding for Lightweight Heatmap-oriented Trajectory Prediction

[1]Renhao Huang, [2]Jingtao Ding, [1]Maurice Pagnucco, and [1]Yang Song

***Abstract*—Recently, heatmap-oriented approaches have demonstrated their state-of-the-art performance in pedestrian trajectory prediction by exploiting scene information from input images before running the encoder. To align the image and trajectory information, existing methods centre the scene images to agents' last observed locations or convert trajectory sequences into images. Such alignment processes cause repetitive executions of the scene encoder for each pedestrian in an input image while there are often many pedestrians in an image, thus leading to significant memory consumption. In this paper, we address this problem by fully decoupling scene and trajectory feature extractions so that the scene information is only encoded once for an input image regardless of the number of pedestrians in the image. To do this, we directly extract temporal information from trajectories in a global pixel coordinate system. Then, we propose a transformer-based heatmap decoder to model the complex interaction between high-level trajectory and image features via trajectory self-attention, trajectory-to-image cross-attention and image-to-trajectory cross-attention layers. We also introduce scene counterfactual learning to alleviate the over-focusing on the trajectory features and knowledge transfer from Segment Anything Model to simplify the training. Our experiments show that our framework shows highly competitive performance on multiple benchmarks, demonstrating scene-compliant predictions on complex terrains and much less memory consumption when handling multi-pedestrians. Code is publicly available at https://github.com/HRHLALALA/Decouple-Traj.**

***Index Terms*—Computer Vision for Automation; Semantic Scene Understanding; Deep Learning for Visual Perception**

I. Introduction

SCENE information is essential for accurate future path predictions of pedestrians [1], [2]. Pedestrians are more likely to walk on sidepaths and change their routes due to obstacles such as trees and benches. Therefore, an essential task in pedestrian trajectory prediction explores extracting useful scene features from images for *scene-compliant* predictions.

Scene-compliant pedestrian trajectory prediction methods typically aggregate image and trajectory features by using pooling or soft-attention and regress the coordinates of future trajectories [1]–[3]. More recent studies find that coordinate regression may suffer from the overfitting problem [4] and unexplainability for scene compliance [5]. Therefore, some methods regress heatmaps to enhance the scene understanding [6], [7]. With the development of *endpoint conditioned* trajectory prediction, some models [5], [8]–[11], categorised as *heatmap oriented trajectory prediction* methods, regress heatmaps to model the distribution of possible endpoints and sample them to generate multiple scene-compliant future trajectories, demonstrating leading performance in multiple scene-compliant trajectory prediction benchmarks [12], [13].

However, images and trajectories belong to two different data modalities. For heatmap-oriented methods, it is challenging to integrate temporal information from observed trajectories into scene features. To solve this problem, existing models fuse these two modalities at the input layer so that these two features can be integrated through image encoding. For example, some methods [14]–[16] center the image to the pedestrian's last observed position and rotate it to their facing direction, abbreviated as *Pedestrian-centric Alignment* [17], [18] as shown in Figure 1a. Other methods [5], [9], [10] directly convert the trajectories to distance maps or render them on images, abbreviated as *Map Rasterisation* [5], [19] in Figure 1b. However, different pedestrians require different inputs and thus the scene encoder needs to be executed repetitively for each pedestrian in a scene, resulting in significant memory consumption when the number of pedestrians increases.

To effectively integrate scene and trajectory information while minising memory consumption, our proposed solution is to *fully decouple trajectory and scene encoding* as shown in Figure 1c. Concretely, we create a global coordinate system that is shared between trajectories and the image space. During inference, we use observed trajectories based on *global pixel coordinates* and integrate them with scene features to regress heatmaps for endpoints. Therefore, the image encoder is only executed once, avoiding excessive memory consumption. However, such decoupling of images and trajectories also means features from the two modalities are extracted independently. Therefore, the main challenge is to model the complex interaction between scene and trajectory information and obtain a comprehensive representation useful to the endpoint heatmap prediction.

In this work, we address this challenge as follows. We first perform a coordinate encoding [20] on coordinates and then a positional encoding [21] on the trajectory. Then, we send them with image features into a lightweight *transformer-based heatmap decoder*, containing three kinds of multi-head attention mechanisms to model the interactions, where (1) *trajectory self-attention* further extracts the temporal information in trajectory features, (2) *trajectory-to-image cross attention* aims to query useful scene information from image features and update

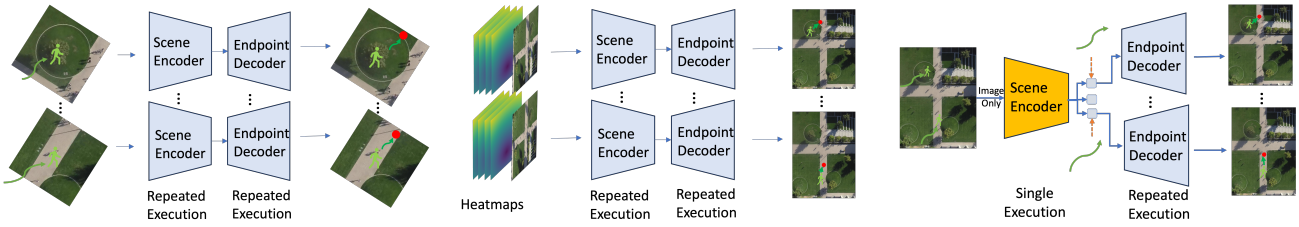
Manuscript received: March 4, 2024; Revised May, 23, 2024; Accepted June, 28, 2024.

This paper was recommended for publication by Editor A. Valada upon evaluation of the Associate Editor and Reviewers' comments.

[1] Renhao Huang, Maurice Pagnucco, Yang Song are with the School of Computer Science and Engineering, the Univerisity of New South Wales, Sydney, Australia {renhao.huang, morri, yang.song1}@unsw.edu.au

[2] Jingtao Ding is with the Future Intelligent Lab, Tsinghua University, Beijing, China dingjt15@tsinghua.org.cn

Digital Object Identifier (DOI): see top of this page.



(a) Pedestrian-centric Alignment (b) Map Rasterisation (c) Ours

Fig. 1: Pedestrian-centric alignment, map rasterisation and our architecture. Existing methods (light blue) prepare the image inputs identical to each pedestrian and thus requires repeated execution of their scene encoders while our framework (yellow) only execute it once rightly after observing the scene image.

trajectory features and finally, (3) *image-to-trajectory cross attention* fuses useful trajectory features into image features. After stacking these attentions several times, the updated image features would contain rich trajectory information and we send them into a scale-up convolution to generate high-resolution heatmaps for endpoints. Furthermore, we find that our model can share a similar coordinate-image interaction as used in the Segment Anything Model (SAM) [22] and hence, we explore the potential knowledge transfer from SAM to our model to improve the training. Finally, we propose the *scene counterfactual learning* to let the model focus more on scene features. We consider the scenario when no scene feature is provided and simply perform a subtraction on the predictions from normal and counterfactual cases to obtain our final predictions.

In summary, our contributions are as follows. **First**, we design a novel framework that decouples the scene and trajectory feature extraction to avoid excessive memory consumption due to repeated image encoding for multiple pedestrians. **Second**, we propose a transformer-based heatmap decoder that fuses the scene and image features and generates heatmaps via trajectory self-attention, trajectory-to-image and image-to-trajectory cross-attentions. **Third**, we further propose scene counterfactual learning that makes the model focus more on scene features and knowledge transfer from SAM, enhancing the model performance. **Finally**, our experiments demonstrate that our model shows its scene-compliance from the endpoint distribution, requires less memory consumption when the number of pedestrians increases and maintains competitive performance with the state-of-the-art approaches for both short- and long-term predictions on multiple benchmarks.

II. Related Work

Pedestrian trajectory prediction tasks usually focus on the social and scene (physical [1]) interactions, where their corresponding tasks are socially-aware trajectory prediction and scene-compliant trajectory prediction [1], [6], [9]. In this paper, we focus on scene-compliant trajectory prediction.

Classical data-driven methods [1], [23] directly use the pretrained image backbones to extract scene features and perform soft attention on them to obtain important scene features for trajectories. SS-LSTM [2] directly sends the image features to recurrent networks together with trajectory features. PITF [24] directly selects the scene feature at the agent's last observed location and concatenate it with trajectory features. However, these methods all regress coordinates and thus scene-compliance cannot be guaranteed. Then, Multiverse [7] and ST-Grids [6] convert the trajectories into distance maps and send them together with scene segmentation maps into convolutional blocks and output heatmaps of future trajectories. They illustrate the scene-compliance through the heatmaps in their experiments. Then, endpoint-conditioned models [5], [8], [9], [25] are proposed, suggesting that the heatmaps can be used to indicate the distribution of endpoints. GoalSAR [10] illustrates that the waypoints can also be scene-compliant if the conditioned endpoints are scene-compliant. P2T [15] considers endpoint heatmaps as reward maps and performs inverse reinforcement learning on them. NSP [11] directly uses the scene-compliant endpoints sampled from the endpoint heatmap predicted by [9]. Therefore, heatmap-oriented trajectory prediction has strongly illustrated its effectiveness of scene-compliance in complex terrains. However, they are memory-consuming and thus cannot handle large number of pedestrians in real time. HyerTraj [26] suggests that the repetitive upsampling stages cause huge memory consumption during waypoints decoding. Therefore, they generate convolution kernels for each sampled endpoint and use them to render heatmaps for waypoints. Our model successfully addresses the memory consumption from repetitive execution of scene encoding without a performance drop.

In vehicle trajectory prediction, scene representation is unified as high-definition (HD) maps. Early methods converted into the rastered images [19], centered at the observed locations of ego vehicles. Then, vector representation [27] is proposed which compresses the geometric information into polylines. This representation is more efficient and less noisy than raster images [28], [29] and thus becomes the default setting in current vehicle trajectory prediction studies. Our method focuses on the pedestrian trajectory prediction, where RGB images as the scene representations in default. Therefore, vehicle trajectory prediction is not in-scope for this study.

III. Methodology

A. Overview

An overview of our framework can be seen in Figure 2. Our framework includes a ViT-based scene encoder $\mathcal{F}_{scene}$ [30], a trajectory encoder $\mathcal{F}_{traj}$ and a heatmap decoder $\mathcal{F}_{dec}$.

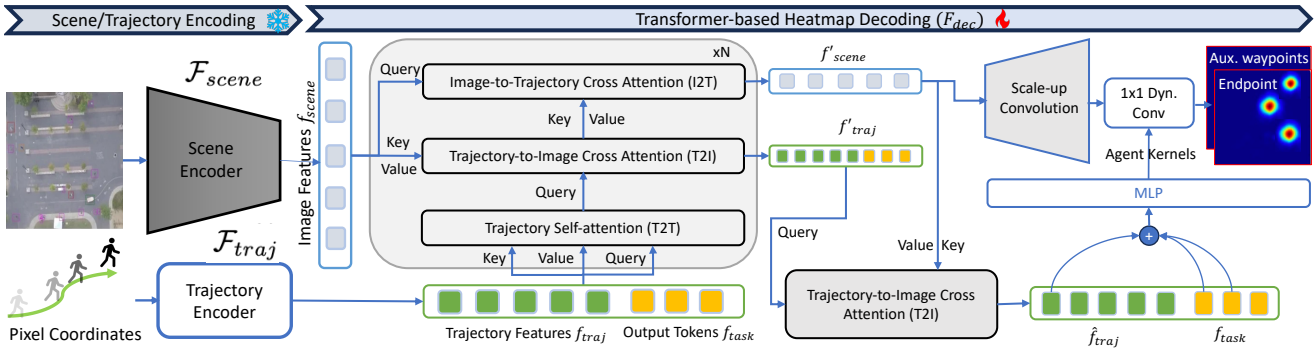


Fig. 2: An overview of our framework for the endpoint prediction, which includes a scene encoder, a trajectory encoder and a transformer-based heatmap decoder, with 89M, 0.4M and 4.1M trainable weights. Our trajectory encoder contains a coordinate encoder to encode pixel coordinates and a position encoding for temporal information. Both scene and trajectory encoders are frozen during the training. Our scene encoder does not require the trajectory information and can be executed once rightly after observing the scene image.

Before the prediction, we send the RGB scene image I into the scene encoder to obtain image features f_{scene}. During the prediction phase, the observed trajectory of target pedestrian i is received as X_i^t, where $X_i^t = \{(x_i^t, y_i^t)|t = 1, 2, \cdots, t_{obs}\}$, and the trajectory encoder extracts temporal embeddings f_{traj}. The heatmap decoder first fuses these two embeddings using multiple attention layers in the fusion module and a dynamic scale-up convolution is used to generate endpoint heatmaps $\mathbb{Y}_i^{t_{pred}}$. Finally, an endpoint $\hat{Y}_i^{t_{pred}}$ is sampled and a waypoint decoder [10], [11] is used to complete the middle trajectory towards the sampled endpoint, named *waypoints* and denoted as $\hat{Y}_i^t = \{(\hat{x}_i^t, \hat{y}_i^t)|t = t_{obs} + 1, ..., t_{pred} - 1\}$. The predicted sequence is evaluated against the ground truth $Y_i^t = \{(x_i^t, y_i^t)|t = t_{obs+1}, ..., t_{pred}\}$. In this paper, we mainly focus on the endpoint prediction.

B. Scene and Trajectory Encoder

Scene Encoder. A strong image encoder is needed to extract rich and high quality image features from the input image I. In this work, we select a Vision Transformer (ViT) [30], a high-performance transformer-based image encoder, to obtain the image features f_{scene}. We also use the weights pretrained on SA-1B [22], the largest segmentation dataset in the world. We follow the default preprocessing step in [30] to prepare the image by rescaling and padding them to a resolution of 1024×1024 and adapt the trajectories to fit this resolution. Formally, we summarise this part as follows:

$$f_{scene} = \mathcal{F}_{scene}(I) = \text{ViT}(I) \tag{1}$$

Trajectory Encoder. We first map X_i^t into a *global pixel coordinate system*, originating at the top left corner of the scene image. These coordinates are further normalised by the size of the scene image $X_i^t = (\frac{x_i^t}{W}, \frac{y_i^t}{H})$. We then use the coordinate encoder to project the coordinates into a high dimension Fourier features as $\gamma_i^t = (\sin(2^0\pi X_i^t), \cos(2^0\pi X_i^t), \cdots, \cos(2^{Dim-1}\pi X_i^t))$, where Dim is a hyperparameter of the dimension of encoded coordinates. As mentioned in [20], this encoding scheme can help the model more easily approximate a higher frequency function and distinguish different coordinates. We then send γ_i^t into a Gated Recurrent Unit (GRU) [21] as positional encoding to further extract the temporal features f_{traj}:

$$f_{traj} = \mathcal{F}_{traj}(X_i^t) = GRU(\gamma_i^t). \tag{2}$$

We believe that a single GRU is more effective than the positional encoding in [31] since the observed trajectory is relatively short and simple and recurrent neural networks can also be a strong positional encoding for transformers [32].

C. Transformer-based Heatmap Decoder

Our heatmap decoder aims to (1) model the interactions between the f_{traj} and f_{scene} extracted from different data modalities and (2) regress heatmaps for endpoints. Taking the inspiration of Transformer-based segmentation models [22], [33], we use a stacked Transformer decoder to model three kinds of interactions: the trajectory self-attention, trajectory-to-image cross-attention and image-to-trajectory cross-attention, each one is a multi-head attention (MHA) block [31] as follows:

$$MHA(Q, K, V) = LN(Q + MLP(\sigma(\frac{QK^T}{\sqrt{Dim_K}}) \cdot V)) \tag{3}$$

where Q, K, V are queries, keys and values respectively while σ and LN are the SoftMax and LayerNorm operators. Finally, we use a scale-up convolution to generate high resolution heatmaps for endpoints.

Trajectory Self-attention (T2T). T2T aims to explore the temporal relationship among trajectory features and integrate them with output embeddings, formulated as $f'_{traj} = MHA(f'_{traj}, f'_{traj}, f'_{traj})$, where $f'_{traj} = (f_{task}||f_{traj})$ initially as the input at the first layer. Here, we introduce output embeddings f_{task}, which are learnable embeddings used as prompts for different heatmap generation tasks, e.g., heatmaps for required waypoints in long-term prediction, and can be extended as other conditional variables for multimodal predictions [25] in future works.

Trajectory-to-Image Cross-attention (T2I). Secondly, we filter out useful scene features based on trajectory features. Therefore, we use f'_{traj} as the queries and f'_{scene}

as keys and values to perform the cross attention $f'_{traj} = MHA(f'_{traj}, f'_{scene}, f'_{scene})$, where $f'_{scene} = f_{scene}$ initially. Note that this process is similar to the Soft Attention in [1] where the embeddings in f'_{traj} contain rich scene interaction information and can be used for coordinate regression.

Image-to-Trajectory Cross-Attention (I2T). Since our model aims to regress heatmaps to indicate the endpoint distribution, it is essential to know the relationship between pixels and trajectory features. To model this, we use f'_{scene} as queries while f'_{traj} as keys and values to generate a new f'_{scene} containing the trajectory information $f'_{scene} = MHA(f'_{scene}, f'_{traj}, f'_{traj})$. Since f'_{scene} is still two dimensional and has the same size as f_{scene}, we can simply convert f'_{scene} to heatmaps.

Scale-up Convolution. Generating heatmaps with higher resolution brings more accurate endpoint estimation. Therefore, we upsample f'_{scene} with 4x of resolution via multiple layers of transpose convolutions. In addition, since we use output embeddings as prompts to generate heatmaps at different future steps, we perform an extra T2I layer rightly after the last I2T to generate $(\hat{f}_{task}||\hat{f}_{traj})$ as suggested in [22] and aggregate the output embeddings with the attended pedestrians' features. Then, we sent them into 3-layer MLPs to generate the kernels for 1×1 dynamic convolution to generate different heatmaps. Mathematically, we can use the following equation to generate the heatmap at the last prediction step:

$$\hat{\mathbb{Y}}_i^{t_{pred}} = Conv(f'_{scene}) * \text{MLP}_{task}(\hat{f}_{traj}^{i,t_{obs}} + \hat{f}_{task}^{t_{pred}}) \quad (4)$$

where $\hat{f}_{task}^{t_{pred}}$ denotes the $\hat{f}_{task}$ for generating the heatmap at the last prediction step. Note that aggregating the updated output tokens and trajectory embeddings allows our model to be extended to multi-pedestrians in future works to reduce the consumptions from scale-up blocks.

D. Training and Inference

Scene Counterfactual Learning. Ideally, future trajectories are predicted jointly with the scene and observed trajectories. However, we observe that the model can become over-fitted and learn a direct relationship between observed and future trajectories while ignoring the scenes. For example, most pedestrians' trajectories are straight due to the straight sidepath. In some case, the observed trajectory follows a smooth, slow-changing path that can directly lead to the endpoint even without seeing the scenes. In these cases, the model would find its shortcut to directly estimate the endpoint distribution based on observed trajectories, which would however cause problems for cases with more complex paths.

To address this problem, we ask: how do the pedestrians move without seeing the scene? Inspired by [34], we propose to utilise counterfactual analysis, a useful technique to infer the causality between two variables, which are image features and endpoints in our method. Concretely, we perform intervention by substituting image features with counterfactual values f_{scene}^{cf} agnostic to the scene image, e.g., zero tensors. Then, we compute the difference of predictions with and without using counterfactual values. Formally, we formulate this part as follows:

$$\hat{\mathbb{Y}}_{i,cf}^t = \hat{\mathbb{Y}}_i^t - \mathcal{F}_{dec}(f_{scene}^{cf}, f_{traj}). \quad (5)$$

This strategy forces the model to make a prediction after seeing the image features. To enhance such causality, we use $\hat{\mathbb{Y}}_{i,cf}^t$ as our final results for both training and testing.

Knowledge Transfer from Segment Anything for Effective Training. We notice that training our model from scratch is hard due to the complex interactions between trajectory and image features. Therefore, we propose to perform a knowledge transfer from a promptable segmentation model, SAM. This model uses coordinates as prompts, integrates them with image embeddings using attention and generates heatmaps of target objects, which finally become the segmentation masks after thresholding. We adopt a similar idea in our approach so that prompt-image interaction in SAM can help our model better capture the scene-trajectory interactions. In particular, we adopt the pretrained prompt encoder and mask decoder in SAM as our coordinate encoder and heatmap decoder. We also note that SAM has four output tokens for segmentations with different granularities. We use one of them to generate endpoint heatmap and the others to generate auxiliary waypoints. We then fine-tune them using our ground truth Gaussian heatmaps converted from 2D coordinates of endpoints. Finally, we use the Binary Cross Entropy (BCE) loss to optimise the model for generating endpoint heatmaps: $L = BCE(\hat{\mathbb{Y}}_{i,cf}^t, Y_i^t)$.

Inference. After the endpoint heatmap is predicted, we first sample the endpoints and conditioned waypoints (for long-term predictions) using the Test-Time-Sampling-Trick and Conditional Waypoints Sampling proposed in [9]. We then input the endpoints (as well as waypoints for long-term prediction) into a waypoints decoder to predict the middle waypoints. The design of the waypoints decoder can be alternative in [5], [9]–[11]. Since the waypoint decoder is not our focus in this paper, we simply choose a four-layer MLP for short-term prediction and a scale-up convolution for long-term predictions, extremely lightweight for waypoint generation.

IV. Experiments

A. Experimental Setup

Datasets. Our experiments are conducted on the SDD [12] and InD [13] datasets. We directly use the preprocessed train-test splits from [9] named **SDD-TrajNet**, **SDD Long-term** and **InD Long-term**. The trajectories are downsampled to 2.5Hz and 1Hz for short-term and long-term predictions, respectively, and the non-pedestrian trajectories are filtered. We also build a **SDD-Full** as in [15] to include road users other than pedestrians to match the experimental setting in [1], [14], [15]. For short-term prediction, we use 8 and 12 steps for observation and prediction, respectively, and only use the last endpoint as the condition for waypoint prediction. For long-term prediction, we use 5 and 30 steps for observation and prediction and the trajectory completion is conditioned on the endpoint and waypoint at the 15^{th} prediction timestep. All these settings are consistent with those in [9], [10], [15].

Evaluation Metrics. Following the commonly used evaluation metrics in trajectory prediction, we use Average Displacement Error (ADE) and Final Displacement Error (FDE), which are all L_2 errors. For ADE, we compare the entire trajectory generated from the waypoint decoder while for FDE, we compare the

		Short-term Prediction		Long-term Prediction		Off-road Rate
Model	Year	SDD-TrajNet	SDD-Full	SDD Long-term	InD Long-term	SDD-TrajNet
S-GAN [35]	2018	N/A	27.25/41.44	155.32/307.88	38.57/84.61	N/A
SoPhie [1]	2019	N/A	16.27/29.38	N/A	N/A	N/A
GoalGAN [14]	2020	N/A	12.20/22.10	N/A	N/A	N/A
PECNet [36]	2020	9.96/15.88	N/A	72.22/118.13	20.25/32.95	0.071
P2T [15]*	2020	12.81/14.08	15.90/18.40	N/A	N/A	0.058
YNet [9]*	2021	7.85/11.85	**12.03**/17.03	**47.94**/66.71	**14.99/21.13**	0.048
TDOR [16]*	2022	7.64/12.12	N/A	N/A	N/A	0.06
SocialVAE [37]*	2022	8.10/11.72	N/A	N/A	N/A	N/A
TUTR [38]*	2023	7.76/12.69	N/A	N/A	N/A	N/A
Ours*	2024	**7.44**±0.02/**11.63**±0.08	12.37±0.03/**15.14**±0.06	51.32±0.34/**64.85**±**1.41**	17.97±0.55/22.88±1.21	**0.043**

TABLE I: Overall minADE_{20}/minFDE_{20} (↓) and Off-road Rate (↓) results in pixels on the SDD and InD datasets for short-term and long-term prediction. Our experiments were conducted five times with random restart. The best and the second best scores are **bolded** and underlined. "*" denotes the methods using test-time sampling trick in [9].

sampled endpoints with the last location of the ground truth trajectory. Since our model focuses on the endpoint prediction as [9], *FDE is the most representative evaluation metric for our method.* We also follow the evaluation protocol in stochastic trajectory prediction which uses *minA(F)DE*$_K$ to denote the best measurement among K different proposals, where we use $K = 20$ samples by default. To measure the scene compliance, we follow [15], [16] to compute the *Off-road Rate*, the rate of predictions outside the walkable places. To mitigate the randomisation from endpoint sampling, we follow [9], [10] to repeat the evaluation process five times using different random seeds and use the average values as our final results.

Implementation Details. To satisfy the required image resolution of 1024×1024 in ViT [30], we scale the scene images and pad the shorter side with zeros. In addition, we directly adopt the prompt encoder and the mask decoder in [22] as our coordinate encoder and heatmap decoder as mentioned in Section III-D. During the training, the ground truth heatmaps are created with a σ as 8. We train our model with an Adam optimiser with a learning rate of 0.0001 and batch size of 8. For short-term predictions, we train the model for 200 epochs and optimise predicted heatmaps at $\{6, 8, 10, 12\}^{th}$ future time steps. For long-term predictions, we train the model for 300 epochs and optimise $\{11, 15, 21, 30\}^{th}$ future time steps. To reduce the rotation and translation variance, we follow [9], [10] to randomly combine transpose, flipping, translation, affine and 10° of rotations as data augmentation.

Baselines. We compare our model with existing pedestrian trajectory prediction models [1], [9], [14]–[16], [35]–[38]. Among these, GoalGAN [14] and P2T [15] are two heatmap-oriented models that use pedestrian-centric alignment while [9] rasterises trajectories into distance maps and generates heatmaps for endpoints and middle waypoints. For TDOR [16], we record the performance using the same sampling trick as [9]. We mainly consider YNet [9] as our baseline approach because since 2021, it has the leading performance among all heatmap-oriented methods on SDD and InD datasets for short-term and long-term predictions. Other highly-ranked methods on the leaderboard including GoalSAR [10] and NSP [11] follow the same overall model as YNet but design more advanced waypoint decoders. Since our method uses only simple waypoint decoders as in YNet, but replaces the overall architecture for the endpoint prediction, our method is more directly comparable with YNet, but not GoalSAR or NSP.

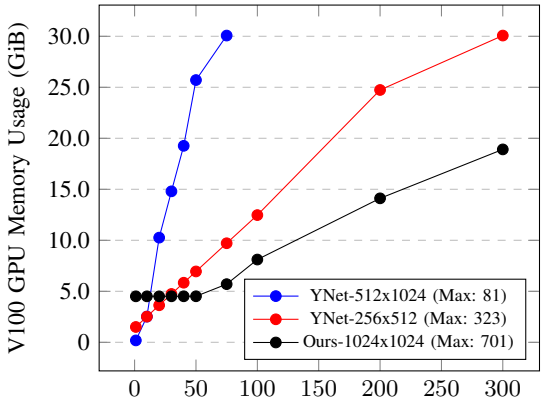


Fig. 3: Memory consumption benchmark of our model comparing with YNet using images of 256×512, 512×1024 pixels using the short-term prediction protocol. The x-axis is the number of pedestrians. All models are tested on on a 32GB Telsa V100 and 64GB memory.

B. Quantitative Results

Overall Performance on SDD and InD Datasets. Table I shows the performance on SDD and InD datasets on short-term and long-term prediction. Our performance on SDD has a better FDE performance than our baselines on both long-term and short-term predictions and becomes the new state-of-the-art on this benchmark. For the InD dataset, our FDE performance is slightly lower than YNet by only one pixel, which is still highly competitive. Therefore, all methods indicate that our model has an excellent performance for the endpoint prediction, which strongly proves the success of our framework. In addition, YNet uses segmentation masks as the scene inputs, which filters much noisy information from the RGB image. We directly send into the model the RGB images, largely reducing the labelling effort and still having competitive performance. Finally, our model has the fewest predictions outside the walkable region on SDD-TrajNet, proving that our model is scene-compliant.

Our ADE results are slightly worse than our baselines mainly due to the selection of extremely lightweight waypoint decoders to simplify the experiments. In our future work, we will explore more advanced techniques [9]–[11] for waypoint decoding.

Memory Consumption Benchmark. To measure memory consumption, we test our model and our baseline YNet on a

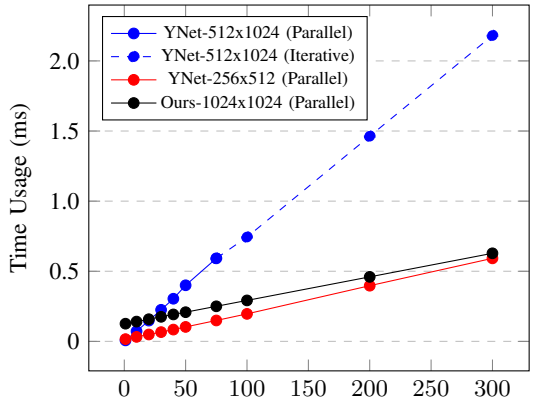


Fig. 4: Time consumption benchmark of our model comparing with YNet using images of 256×512, 512×1024 pixels. The X-axis is the number of agents. We use the same setting as Memory Consumption Benchmark. YNet-512×1024 switches to an iterative prediction when there are more than 81 agents.

32GB Telsa V100 and 64GB memory. The measurement begins with sending scene images and trajectories to the scene encoder and finishes when the heatmaps are produced. We conduct the benchmark on images with resolutions of both 256×512 or 512×1024 pixels. Note that our model automatically resizes the scene image to a resolution of 1024×1024. For simplicity, we directly use the random vectors as images and trajectory inputs. Figure 3 shows that the memory consumption largely scales in YNet with the resolution increases while ours is even much lower than YNet-256×512, mainly because we only execute image encoding once while YNet needs repeated executions for different pedestrians. In addition, YNet can handle a maximum of 323 and 81 pedestrians on these two resolutions. Our model accepts a higher resolution than YNet but can predict 701 pedestrians, twice and eight times more than the baselines on images with low and high resolutions, which is suitable for almost all scenarios. Therefore, our method provides the opportunity to accept dense scenarios, which will be further evaluated in our future studies.

Time Consumption Benchmark. We also measure the time consumption of YNet [9] and our model using the same environment of the memory consumption benchmark. Figure 4 shows when using images with resolution of 1024, the time consumption of our model is much lower than YNet-512×1024, but slightly higher than YNet-256×512. In addition, our time consumption scales more slowly than YNet-512×1024 and YNet-256×512, proving the advantage of fusing the features in the latent space to latency. The latency bottleneck from the ViT encoder and attention opterations. In future studies, we can distil a lighter version as [39] or choose optimised attention operations [40] for acceleration.

C. Ablation Study

Trajectory Encoding. We compare different alternatives for our coordinate encoder and positional encoder. Firstly, we simply replace the coordinate encoder with a three-layer MLP. Table II shows that using the MLP leads to a worse performance which proves that the high-frequency encoding is needed for global pixel coordinates and can be a strong training-less encoding scheme. Then, we compare the performance of using the positional encoding in [31] and GRU. Table II also shows that using GRU has a better performance than the position encoding in this case, illustrating that GRU is a powerful position encoding strategy.

Coordinate encoder	**minADE$_{20}$↓**	**minFDE$_{20}$↓**
MLP	8.06	13.34
High Frequency [20]	**7.60**	**12.05**
Positional Encoder	**minADE$_{20}$↓**	**minFDE$_{20}$↓**
Wave Encoding [31]	8.03	13.03
GRU	**7.60**	**12.05**

TABLE II: Performance on SDD-TrajNet of different coordinate encoding (upper half) to encode absolute pixel coordinates and positional encoding (lower half) to inject temporal information.

$\mathcal{F}_{img}$	$\mathcal{F}_{traj}$	$\mathcal{F}_{dec}$	minADE$_{20}$	minFDE$_{20}$
✓			8.37	14.12
✓	✓		8.61	14.49
✓		✓	8.12	13.54
✓	✓	✓	**7.60**	**12.05**

TABLE III: Performance on SDD-TrajNet before and after loading pretrained weights from SAM to our trajectory encoder and heatmap decoder.

Knowledge Transfer from Segment Anything. As described, we can transfer knowledge from SAM by using the pretrained components from it. As shown in Table III, using a pretrained prompt encoder only may even lead to a worse performance. A plausible explanation is that these pretrained embeddings cannot provide useful information without a pretrained heatmap decoder or even provide a worse initialisation. Then, using a pretrained weights from mask decoder largely improve the results. Finally, using the entire pretrained SAM performs the best, suggesting that the correlation between prompt encoder and mask decoder also provides important knowledge to accelerate the optimisation of our model.

Importance of Scene Information. We first experiment to explore the performance without scene information by replacing

		SDD-TrajNet		SDD-Longterm	
	Values	**minADE$_{20}$**	**minFDE$_{20}$**	**minADE$_{20}$**	**minFDE$_{20}$**
f_{scene}=	Zeros	7.71	12.68	51.81	73.80
	Base	**7.60**	**12.05**	**51.52**	**65.86**
f^{cf}_{scene}=	Random	7.69	12.30	51.67	65.91
	Empty	7.70	12.38	52.94	68.76
	Zeros	**7.44**	**11.63**	**51.32**	**64.85**

TABLE IV: Performance using different scene features (upper) and scene counterfactual values (lower) on SDD-TrajNet and SDD-Longterm datasets. **Base** denotes the scene features from the original image. **Random** and **Zeros** are tensors with random numbers between [-0.1,0.1] and zeros respectively. **Empty** is the scene features from a blank image.

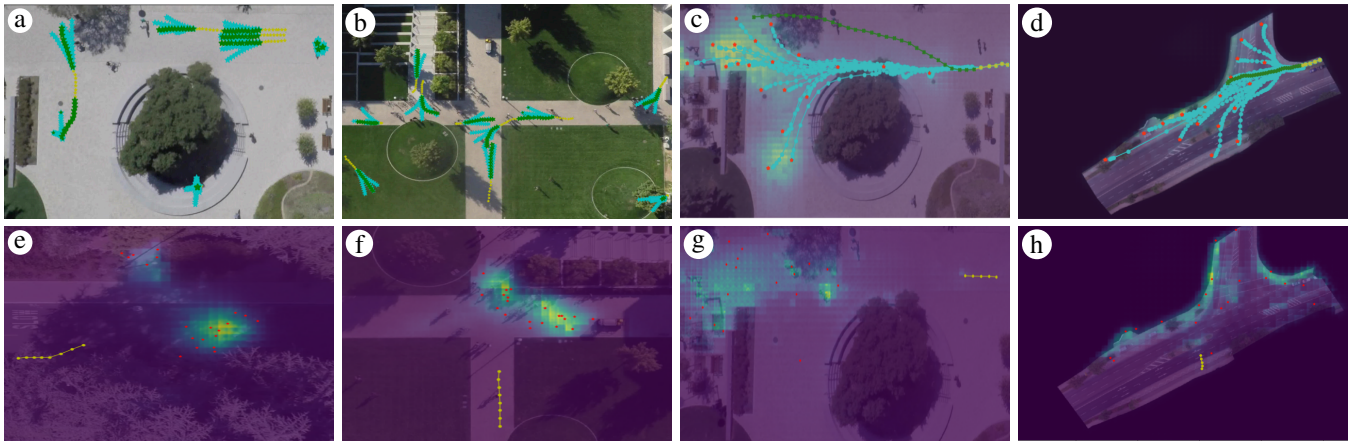


Fig. 5: Visualisations of short-term (Fig.4a/b/e/f) and long-term (Fig.4c/d/g/h) predictions with observed trajectories (yellow), sampled endpoints (red), ground truth endpoint/trajectories (green) and predicted trajectories (cyan). Fig. e/f/g.h are heatmaps for predicted endpoints. These visualisations illustrate that our model (1) successfully fuses the high-level scene and trajectory features and (2) has strong capability of scene compliance.

the scene features f_{scene} with zero tensors during the training. The upper part of Table IV indicates that the performance drops when the model cannot see the scene image, which further suggests that scene information is essential to our model. Then, we further perform the counterfactual learning using different counterfactual values f^{cf}_{scene} for scenes such as zeros (Zeros), random values (Random) and embeddings from an empty image (Empty). The lower part of Table IV shows that using the zero tenors provides the largest improvement among these three choices, which is consistent with the suggestions in [34].

Qualitative Results. To further demonstrate that our fusion of scene and trajectory features is successful, we visualise in Figure 5 the scenarios from short-term and long-term predictions, where Figure 5a and Figure 5b are two randomly selected scenarios to show the overall accuracy while the rest are selected scenarios that mostly demonstrate the scene compliance. For short-term predictions, we show in Figure 5a and Figure 5b predictions for all pedestrians in the scene, where all predictions have consistent speeds and directions as the ground truth, illustrating that our model correctly considers trajectory features. Furthermore, Figure 5e and Figure 5f visualise the predicted endpoint heatmaps of two additional scenarios, where regions with high probabilities are mostly walkable places (roads), demonstrating that our predictions are scene compliant. For long-term predictions, Figure 5c shows a clear collision avoidance with trees and the distribution in Figure 5g excludes the location of the flower bed. Furthermore, we can see in Figure 5d and Figure 5h that the predictions consider the road geometry, providing high probabilities to regions along the roadside. In summary, all these visualisations prove that (1) our model successfully integrates the trajectory and image features even though they belong to two different modalities and (2) provide scene-compliant predictions.

D. Limitation and Future Work

Firstly, we can explore methods to handle abnormal behaviours. For example, in Figure 5c, the pedestrian faces the tree at the last observed step and thus our model gives high probability to regions around the tree for conditioned waypoints, while the ground truth changes the direction suddenly after a few timesteps. To mitigate this problem, we can either explore more diverse datasets or explore advanced conditions to control the predictions. **Secondly**, we do not consider social interaction because crowds in current datasets are too sparse to train a social interaction module while other datasets such as ETH/UCY do not provide diverse scenes to train the scene interaction. In our future studies, we will find a better training dataset and extend our model to perform the social interaction, such as extending our scene encoder to support videos [41], accepting social interactions in trajectory encoder [42] or waypoint decoding [11], [36]. **Finally**, our current method is designed for image-based scene representations for pedestrian trajectory prediction. Although several vehicle trajectory prediction studies also convert HD maps to images [19], the polyline-based representation [27] is preferred as mentioned in Section II. In our future work, our method can be extended for such different applications after finding a unified scene representation.

V. Conclusions

We propose a new lightweight heatmap-oriented trajectory prediction framework that avoids repeated execution of scene encoder for different pedestrians. We directly uses a scene image without any alignment and 2D coordinates to generate heatmaps for endpoints. We successfully use global pixel coordinates for trajectory features and fuse them with scene features via a multilayer attention module in our heatmap decoder, each layer containing multihead trajectory self-attention, trajectory-to-image cross-attention and image-to-trajectory cross-attention. Furthermore, due to the analogy between our model and SAM, we transfer the knowledge from SAM to enhance the training. Finally, to further enhance the correlation between scene features and endpoints, we propose to use scene counterfactual learning by considering the counterfactual cases agnostic to the

scene image. Experiments show that our model has competitive FDE performance on endpoint predictions, can handle more pedestrians in parallel and illustrates its scene-compliance in scenarios with complex terrains.

REFERENCES

[1] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese, "SoPhie: An Attentive GAN for Predicting Paths Compliant to Social and Physical Constraints," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1349–1358.

[2] H. Xue, D. Q. Huynh, and M. Reynolds, "SS-LSTM: A Hierarchical LSTM Model for Pedestrian Trajectory Prediction," in *Winter Conference on Applications of Computer Vision*, 2018, pp. 1186–1194.

[3] P. Dendorfer, S. Elflein, and L. Leal-Taixé, "MG-GAN: A Multi-Generator Model Preventing Out-of-Distribution Samples in Pedestrian Trajectory Prediction," in *IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 158–13 167.

[4] H. Jin, S. Liao, and L. Shao, "Pixel-in-Pixel Net: Towards Efficient Facial Landmark Detection in the Wild," *International Journal of Computer Vision*, 2021.

[5] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, and F. Moutarde, "HOME: Heatmap Output for Future Motion Estimation," in *IEEE International Conference on Intelligent Transportation Systems*, 2021, pp. 500–507.

[6] D. A. Ridel, N. Deo, D. F. Wolf, and M. M. Trivedi, "Scene Compliant Trajectory Forecast With Agent-Centric Spatio-Temporal Grids," *IEEE Robotics and Automation Letters*, pp. 2816–2823, 2019.

[7] J. Liang, L. Jiang, K. Murphy, T. Yu, and A. Hauptmann, "The Garden of Forking Paths: Towards Multi-Future Trajectory Prediction," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 508–10 518.

[8] H. Zhao, J. Gao, T. Lan, C. Sun, B. Sapp, B. Varadarajan, Y. Shen *et al.*, "TNT: Target-driven Trajectory Prediction," in *Conference on Robot Learning*, 2021, pp. 895–904.

[9] K. Mangalam, Y. An, H. Girase, and J. Malik, "From Goals, Waypoints & Paths to Long Term Human Trajectory Forecasting," in *IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 233–15 242.

[10] L. F. Chiara, P. Coscia, S. Das, S. Calderara, R. Cucchiara, and L. Ballan, "Goal-Driven Self-Attentive Recurrent Networks for Trajectory Prediction," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2022, pp. 2518–2527.

[11] J. Yue, D. Manocha, and H. Wang, "Human Trajectory Prediction via Neural Social Physics," in *European Conference on Computer Vision*, 2022, pp. 376–394.

[12] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, "Learning Social Etiquette: Human Trajectory Understanding in Crowded Scenes," in *European Conference on Computer Vision*, 2016, pp. 549–565.

[13] J. Bock, R. Krajewski, T. Moers, S. Runde, L. Vater, and L. Eckstein, "The inD Dataset: A Drone Dataset of Naturalistic Road User Trajectories at German Intersections," in *IEEE Intelligent Vehicles Symposium*, 2020, pp. 1929–1934.

[14] P. Dendorfer, A. Osep, and L. Leal-Taixe, "Goal-GAN: Multimodal Trajectory Prediction Based on Goal Position Estimation," in *Asian Conference on Computer Vision*, 2020.

[15] N. Deo and M. M. Trivedi, "Trajectory Forecasts in Unknown Environments Conditioned on Grid-Based Plans," *ArXiv*, vol. abs/2001.00735, 2020.

[16] K. Guo, W. Liu, and J. Pan, "End-to-End Trajectory Distribution Prediction Based on Occupancy Grid Maps," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2242–2251.

[17] G. Aydemir, A. K. Akan, and F. Güney, "ADAPT: Efficient Multi-Agent Trajectory Prediction with Adaptation," in *IEEE/CVF International Conference on Computer Vision*, 2023.

[18] J. Ngiam, B. Caine, V. Vasudevan, Z. Zhang, H.-T. L. Chiang, J. Ling, R. Roelofs, A. Bewley, C. Liu, A. Venugopal *et al.*, "Scene Transformer: A Unified Architecture for Predicting Multiple Agent Trajectories," in *International Conference on Learning Representations*, 2022.

[19] H. Cui, V. Radosavljevic, F.-C. Chou, T.-H. Lin, T. Nguyen, T.-K. Huang, J. Schneider, and N. Djuric, "Multimodal Trajectory Predictions for Autonomous Driving using Deep Convolutional Networks," in *International Conference on Robotics and Automation*, 2019, pp. 2090–2096.

[20] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing Scenes as Neural Radiance Fields for View Synthesis," *Communications of the ACM*, pp. 99–106, 2021.

[21] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," in *Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1724–1734.

[22] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment Anything," in *IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4015–4026.

[23] A. Sadeghian, F. Legros, M. Voisin, R. Vesel, A. Alahi, and S. Savarese, "CAR-Net: Clairvoyant Attentive Recurrent Network," in *European Conference on Computer Vision*, 2018, pp. 162–180.

[24] J. Liang, L. Jiang, J. C. Niebles, A. G. Hauptmann, and L. Fei-Fei, "Peeking Into the Future: Predicting Future Person Activities and Locations in Videos," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5718–5727.

[25] R. Huang, H. Xue, M. Pagnucco, F. Salim, and Y. Song, "Multimodal Trajectory Prediction: A Survey," *arXiv preprint arXiv:2302.10463*, 2023.

[26] R. Huang, M. Pagnucco, and Y. Song, "HyperTraj: Towards Simple and Fast Scene-Compliant Endpoint Conditioned Trajectory Prediction," in *International Conference on Intelligent Robots and Systems*, 2021, pp. 7977–7984.

[27] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov *et al.*, "VectorNet: Encoding HD maps and Agent Dynamics from Vectorized Representation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 525–11 533.

[28] M. Liang, B. Yang, R. Hu, Y. Chen, R. Liao, S. Feng, and R. Urtasun, "Learning Lane Graph Representations for Motion Forecasting," in *European Conference on Computer Vision*, 2020, pp. 541–556.

[29] Z. Zhou, L. Ye, J. Wang, K. Wu, and K. Lu, "HiVT: Hierarchical Vector Transformer for Multi-Agent Motion Prediction," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8823–8833.

[30] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in *International Conference on Learning Representations*, 2021.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Advances in Conference on Neural Information Processing Systems*, 2017.

[32] Z. Dai, Z. Yang, Y. Yang, J. G. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-xl: Attentive Language Models Beyond a Fixed-length Context," in *Annual Meeting of the Association for Computational Linguistics*, 2019.

[33] B. Cheng, A. Schwing, and A. Kirillov, "Per-Pixel Classification is Not All You Need for Semantic Segmentation," in *Advances in Neural Information Processing Systems*, 2021, pp. 17 864–17 875.

[34] G. Chen, J. Li, J. Lu, and J. Zhou, "Human Trajectory Prediction via Counterfactual Analysis," in *IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9804–9813.

[35] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2255–2264.

[36] K. Mangalam, H. Girase, S. Agarwal, K.-H. Lee, E. Adeli *et al.*, "It is Not the Journey but the Destination: Endpoint Conditioned Trajectory Prediction," in *Europian Conference on Computer Vision*, 2020, pp. 759–776.

[37] P. Xu, J.-B. Hayet, and I. Karamouzas, "SocialVAE: Human Trajectory Prediction using Timewise Latents," in *European Conference on Computer Vision*, 2022, pp. 511–528.

[38] L. Shi, L. Wang, S. Zhou, and G. Hua, "Trajectory Unified Transformer for Pedestrian Trajectory Prediction," in *IEEE/CVF International Conference on Computer Vision*, 2023, pp. 9675–9684.

[39] C. Zhang, D. Han, Y. Qiao, J. U. Kim, S.-H. Bae, S. Lee, and C. S. Hong, "Faster Segment Anything: Towards Lightweight SAM for Mobile Applications," *arXiv preprint arXiv:2306.14289*, 2023.

[40] T. Dao, D. Fu, S. Ermon, A. Rudra, and C. Ré, "Flashattention: Fast and Memory-efficient Exact Attention with Io-awareness," *Advances in Neural Information Processing Systems*, pp. 16 344–16 359, 2022.

[41] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "ViViT: A Video Vision Transformer," in *IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6836–6846.

[42] Y. Yuan, X. Weng, Y. Ou, and K. Kitani, "AgentFormer: Agent-Aware Transformers for Socio-Temporal Multi-Agent Forecasting," in *IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9813–9823.