TrajPrompt: Aligning Color Trajectory with Vision-Language Representations

Li-Wu Tsao[1], Hao-Tang Tsui[1], Yu-Rou Tuan[1], Pei-Chi Chen[1], Kuan-Lin Wang[1], Jhih-Ciang Wu[2], Hong-Han Shuai[1], and Wen-Huang Cheng[2]

[1] National Yang Ming Chiao Tung University, Taiwan
{lwtsao.ee09,hhshuai}@nycu.edu.tw
[2] National Taiwan University, Taiwan
wenhuang@csie.ntu.edu.tw

Abstract. Cross-modal learning shows promising potential to overcome the limitations of single-modality tasks. However, without proper design for representation alignment between different data sources, the external modality cannot fully exhibit its value. For example, recent trajectory prediction approaches incorporate the Bird's-Eye-View (BEV) scene as an additional source but do not significantly improve performance compared to single-source strategies, indicating that the BEV scene and trajectory representations are not effectively combined. To overcome this problem, we propose TrajPrompt, a prompt-based approach that seamlessly incorporates trajectory representation into the vision-language framework, *i.e.* CLIP, for the BEV scene understanding and future forecasting. We discover that CLIP can attend to the local area of the BEV scene by utilizing our innovative design of text prompts and colored lines. Comprehensive results demonstrate TrajPrompt's effectiveness via outperforming the state-of-the-art trajectory predictors by a significant margin (over 35% improvement for ADE and FDE metrics on SDD and DroneCrowd dataset), using fewer learnable parameters than the previous trajectory modeling approaches with scene information included. Project page: https://trajprompt.github.io/

Keywords: Cross-Modal Learning · Vision-Language Understanding · Efficient Prompt Tuning · Bird's-Eye-View Scene · Trajectory Prediction

1 Introduction

With the rise of practical applications using Vision-Language Pre-trained models (VLP) [19, 21], various downstream tasks have shown remarkable results by leveraging VLP as the foundation model or feature extractor. However, a significant gap emerges when capturing visual knowledge from unseen domains, such as the Bird's-Eye-View (BEV) scene recorded from a drone's perspective. In our study, we highlight the concern of using VLP to capture details on BEV scene understanding and introduce the concept of color trajectory. Furthermore, by utilizing this strategy, VLP can address trajectory prediction challenge as well.

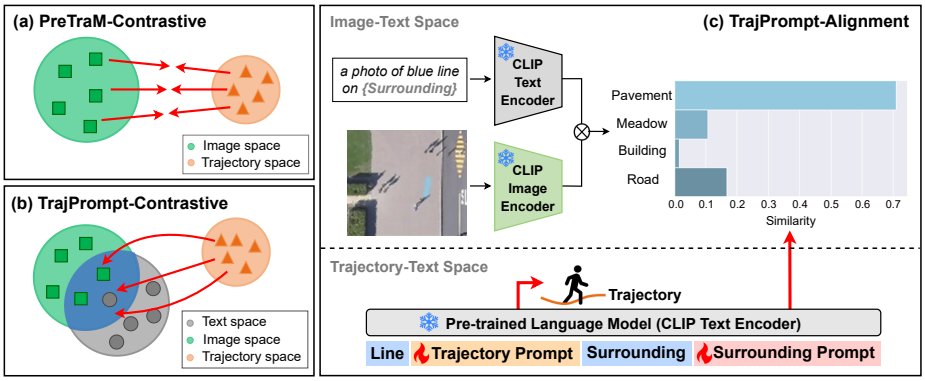


Fig. 1: (a) and (b) illustrate the concept of mapping representation spaces using different strategies for contrastive learning. (a) PreTraM [43] simultaneously update both image and trajectory encoders, which might destroy the original semantic of scene. (b) To preserve scene semantics, TrajPrompt only updates the learnable trajectory prompts to match the image-text space formed by pre-trained CLIP. (c) The vision-language understanding of CLIP on BEV scene is described by the distribution of similarity scores. We first introduce the concept of color trajectory before calculating the scores. Furthermore, by aligning surrounding prompts to the distribution of surrounding types, the trajectory prompt can learn to localize itself on the BEV scene.

Recently, a surge of interest has been in addressing trajectory prediction by adopting the BEV scene information [5–7, 12, 13, 23], driven by its promising practical applications across diverse fields, including drone surveillance [41, 42], robotics [9,33], and autonomous vehicles [3,8,13,23]. Specifically, by utilizing historical observations and the BEV scene to predict the future paths of objects, the agents can have the ability to see through objects without occlusion, and make safe decisions to prevent from colliding with others. Part of the studies [26, 37] try to learn the scene semantics from the pixel or patch labels to assist the understanding of the trajectory. However, obtaining the fully labeled semantic map is labor-consuming. On the other hand, several works [33, 45] encode the BEV scene directly to extract visual features. Obviously, without specific designed objectives to incorporate image and trajectory features, the improvements were marginal. As a result, PreTraM [43] developed the contrastive pre-training strategy on the BEV scene and trajectory to capture the interplay between visual semantics and motion dynamics. Nevertheless, this method encounters a significant challenge in pre-training the BEV encoder from scratch, where the amount of the BEV scene from the trajectory dataset is insufficient to support good visual representation learning. Thus, to mitigate the existing issue on the BEV representation and explore the cross-modal ability for trajectory prediction, we propose a prompt-based architecture named *TrajPrompt* to incorporate trajectory representation with the VLP framework.

As shown in Fig. 1, we demonstrate TrajPrompt with multi-modal contrastive learning and alignment strategy to handle the trajectory prediction problem on the BEV scene. A huge difference from Fig. 1a to Fig. 1b is that we utilize

the power of pre-trained CLIP [30], which can capture the general image-text understandings without additional cost of training. Furthermore, to reduce the number of trainable parameters on downstream trajectory prediction task, we design hard and soft prompts as the input to the frozen CLIP encoders. In our study, we discuss the selection of visual clues with the proper shape and color of trajectory line, which enhances the understanding of CLIP on the BEV scene. Also, by jointly optimizing the image-text contrastive loss, alignment loss, and trajectory modeling loss, the learnable soft prompts can map the understanding of trajectory with vision-language representations efficiently. To the best of our knowledge, TrajPrompt is the first work proposing trajectory alignment with the vision-language model. Our key contributions are highlighted as follows.

- We introduce a novel vision-language prompt approach, which integrates the visual target of colored line and trajectory prompts into image and text.
- We propose three objectives for prompt learning, including context fusion, surrounding localization, and motion forecasting. Utilizing image-text contrastive loss, alignment loss, and trajectory modeling loss correspondingly.
- Our method achieves new art by significant improvements over other state-of-the-arts on drone datasets. Furthermore, TrajPrompt reduces a large number of learnable parameters compared to the previous trajectory models with image information included.

2 Related Work

2.1 Trajectory Models with BEV Scene

Trajectory prediction initially relied on past paths as the primary condition without considering critical environmental factors, such as adherence to road regulations and obstacle avoidance. In pursuit of exploring more sophisticated interactions across diverse modalities, recent works attempt to predict trajectories with BEV scene to comprehend the expected agent behaviors in various scenarios. For example, Trajectron++ [33] and AgentFormer [45] utilize a CNN-based encoder to extract map information and combine such features with past trajectories for prediction. Since those approaches are not specifically designed for better visual understanding, more discussions have emerged. Introvert [34] employs an attention mechanism to identify areas in the map that are relevant to the agent. Y-net [26] annotates the scenes with semantic segmentation maps and learns to predict the trajectory heatmaps with the uncertain distribution of walkable areas. PreTraM [43] employs a self-supervised scheme to construct the shared embedding space with cross-modal contrastive learning between trajectories and maps. TDOR [14] learns the trajectory distribution by minimizing symmetric cross-entropy on occupancy grid maps, resulting in a discrete set of trajectories. In contrast to previous works, we choose text as a pivot for aligning image and trajectory descriptions. This strategy stimulates the model to perform as a human-like understanding while forecasting the trajectory, offering a fresh perspective on the problem.

2.2 Prompt Representation in Vision-Language Model

The Vision-Language Model (VLM) has garnered significant attention owing to its remarkable generalization and flexibility of fine-tuning properties. For instance, CLIP [30] utilizes contrastive loss to construct the shared feature space for image-text understanding. To further seek the correlation between vision and language, BLIP [19] further introduces image-text matching loss and language modeling loss under an encoder-decoder structure, which captures the fine-grained alignment and generates textual descriptions, respectively. Unlike the end-to-end manner, BLIP-2 [18] employs a transformer with soft visual prompts to bridge the gap between image encoders and the Large Language Model (LLM). Since large modules like LLM or VLM are difficult to finetune, using prompts can reduce the number of learnable parameters and computation costs. The efforts can be categorized into hard prompt and soft prompt strategies. The former, such as CPT [44], uses co-referential color markers on bounding box and text for each individual object, also reformulates the target into a fill-in-the-color task for cross-modal prompt tuning. RedCircle [36] investigates drawing colored circles around objects as a visual clue, which can guide the model's attention to the local regions while maintaining global knowledge. For the works that utilize soft prompts, CoOp [48] expresses learnable context words as continuous vectors for the text encoder in VLM. VPT [16] proposes to prepend task-specific prompts as continuous embeddings into the patch sequences of pre-trained vision transformers [10, 24]. In our study, we investigate the hard prompts with colored lines on the BEV scene for cross-modal understanding, while a small number of soft prompts are introduced to excite the potential ability of pre-trained VLM in solving trajectory prediction task.

3 Method

We propose TrajPrompt, a prompt-based approach that bridges the understanding of trajectory and VLM. Although leveraging the pre-trained CLIP as our main structure of VLM, which has established a shared representative space for image and text, two main problems still make the alignment of trajectory on BEV scene challenging. (1) The image representation in CLIP is limited to the general feature on First-Person View, especially difficult to understand the description of local surroundings on BEV scene. (2) The text representation in CLIP cannot precisely describe the location of pixels in image, which makes it difficult to connect the relationship of trajectory points and BEV scene. To solve these challenges, we draw colored lines to represent the trajectory on the BEV scene and design the corresponding text to describe the semantics of the surroundings around the line.

Fig. 2 illustrates the architecture of TrajPrompt. We adopt the Table-to-Text (T2T) [22] format for the input of text encoder, which maintains the key context as attribute followed by the values. In our study, we introduce the idea of T2T sequence that combines textual attributes (blue blocks) followed by the trajectory tokens (orange blocks) or soft prompts (red blocks) as values. This

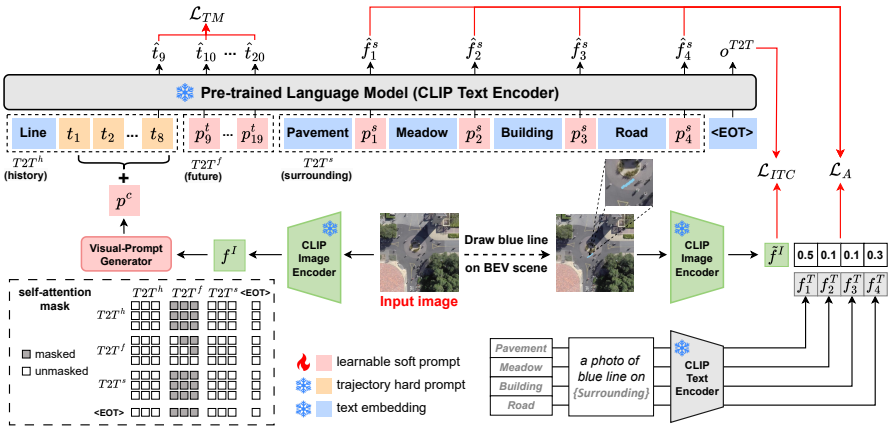


Fig. 2: An overview of architecture. TrajPrompt comprises the image and text encoder from pre-trained CLIP, along with an additional learnable visual-prompt generator. While the CLIP encoders remain frozen, the modification of self-attention masks better suits the autoregressive decoding process in the future timestamps. The input sequence, which we called Table-to-Text (T2T), consists of text-based word embeddings (blue blocks), trajectory hard prompts (orange blocks), and learnable soft prompts (red blocks). The alignment for trajectory and vision-language representation is accomplished via three objectives, including $\mathcal{L}_{ITC}$, $\mathcal{L}_A$, and $\mathcal{L}_{TM}$, to bridge the "blue line" (text), the "blue pixels" (image), and the "coordinates" (trajectory) into the shared representation space.

process particularly combines the understanding of trajectory and text in the bidirectional encoder, where the task-specific soft prompts are supervised via three objective functions. (1) Image-text contrastive loss induces the fusion of conditional image prompt p^c and trajectory by differentiating the global information of colored line on BEV scene. (2) Alignment loss provides the detailed surrounding clues for localizing the trajectory and realizing its nearby environments. (3) Trajectory modeling loss excites the potential ability of text encoder to generate the future coordinate embeddings. In the following sections, we first elaborate on the components of the input representations and then introduce the detailed TrajPrompt framework, objectives, and trajectory retrieval.

3.1 Input Representations

Here, we present each material integrated into TrajPrompt in detail, including the representations for trajectory, prompt, vision-language, and Table-to-Text.

Trajectory Representation. Recent studies [11, 38] projected the trajectory coordinates into a high-dimensional vector with 1D positional embedding to represent the sequence timestamp. Unlike the previous strategy, we adopt a more generalized representation that employs 2D positional embeddings [40] for a better match to the 2D image coordinates. Based on the setting of the resized image used in previous work [14], we construct the fixed number of token *index* to represent each pixel in dictionary $\mathcal{D}$. Specifically, $\mathcal{D}$ includes the transformation of

all the pixel coordinates (w, h) into the format of *pixel embedding* $\phi(w, h) \in \mathbb{R}^d$, where $\phi(\cdot)$ is the 2D positional embedding function. These distinct pixels act as (*index*, *embedding*) token pairs in $\mathcal{D}$, where the number of tokens can be easily extend to different image scales. We describe the given trajectory as an embedding set $T = T^h \cup T^f$, which contains history and future pixel embeddings, respectively. The pixel embeddings of history $t_i \in T^h$ and future $t_j \in T^f$ trajectory are based on the representation in $\mathcal{D}$, where $1 \leq i \leq \left|T^h\right|$ and $\left|T^h\right| < j \leq |T|$.

Prompt Representation. There are two types of prompt representations in TrajPrompt, *i.e.*, hard and soft prompts. The hard prompts, which represent the given histories, offer clues about the location of the trajectory on the map. The soft prompts, which are learnable, involve trajectory prompts $P^t = \{p_j^t \in \mathbb{R}^d\}$, surrounding prompts $P^s = \{p_k^s \in \mathbb{R}^d \mid k = 1, 2, 3, 4\}$, and conditional image prompt p^c, where p^c is derivated from visual-prompt generator $g(\cdot)$.

Vision-Language Representation. We leverage the pre-trained CLIP model as semantic feature extractors for both the images and text. We introduce the visual target by drawing a colored line on BEV scene directly. The map with a colored line can be encoded as prompted image feature $\tilde{f}^I$. Also, we employ text prompt "a photo of *color* line on $\{Surrounding\}$" to describe the BEV scene with common types of surroundings $\Omega = \{pavement, meadow, building, road\}$. The selection of types are based on the definition of semantic labels from Y-net [26]. However, different from pixel annotations in Y-net, we replace the semantic categories with concrete text prompts for CLIP, which is a simple yet efficient approach to extract surrounding feature $F^T = \{f_k^T \in \mathbb{R}^d \mid k = 1, 2, 3, 4\}$ without human labeling. On the other hand, in order to enhance the localization capability of the trajectory prompt, we introduce a visual-prompt generator that takes the raw image feature f^I as a condition, which is inspired by CoCoOp [47]. The collaboration of the conditional image prompt p^c achieved through the visual-prompt generator $g(f^I)$ and the visual target $\tilde{f}^I$ promotes a comprehensive understanding of the BEV scene with colored trajectory line.

Table-to-Text Representation. For each agent, we characterize a sequence that simultaneously describes the trajectory and surroundings according to T2T format. In addition to the previously mentioned prompts, we incorporate attributes such as *line* and *surroundings* in Ω encoded by text embedding. The overall representative attributes and prompt values are combined into the T2T sequence with clear illustrations shown in Fig. 2.

3.2 TrajPrompt Framework

The principal goal of TrajPrompt is to predict the trajectory embeddings $\hat{T}^f = \{\hat{t}_j \in \mathbb{R}^d\}$ by the given sequence via a pre-trained language model, which we use the text encoder of CLIP. Compared to the conventional CLIP text encoder, we adopt special arrangement on self-attention mask (see Fig. 2) to adapt the understanding of T2T sequence. Based on the design, the output embeddings can condition on features from history $T2T^h$, surrounding $T2T^s$, and global sequence understanding <EOT>. With such knowledge, TrajPrompt can forecast the trajectories that follow human-like behaviors by maintaining the movement

on the *pavement* or *road*, reducing the possibility of walking across the *meadow* and avoiding pass through the *building*.

3.3 Objective Functions

We present three objective losses to bridge the gap between trajectory and vision-language representation: (1) image-text contrastive loss, (2) alignment loss, and (3) trajectory modeling loss. In the following, we introduce the detailed concept of these objectives with the use of frozen CLIP.

Image-text contrastive loss. The image-text contrastive loss is inspired by PreTraM [43], which introduces the map-trajectory contrastive loss. The main difference comes from the use of semantic words. In PreTraM, the agent sequence only describes the position of each timestamp, while in TrajPrompt, we use T2T sequence to enforce the trajectory embeddings attending the text-describable surroundings. The presented objective preserves the property of the BEV scene and trajectory through additional text clues. In general, for each draw-line image feature $\tilde{f}^I$ and T2T global sequence embedding o^{T2T}, we calculate the softmax-normalized score based on the cosine similarity function $d(\tilde{f}^I, o^{T2T})$, using B samples in a batch to construct positive and negative pairs. The overall image-text contrastive loss includes M positive pairs with a learnable temperature τ, which is represented as,

$$\mathcal{L}_{i2o} = -\frac{1}{M}\sum_{m=1}^{M}\log\frac{exp\left(d(\tilde{f}_m^I, o_m^{T2T})/\tau\right)}{\sum_{b=1}^{B} exp\left(d(\tilde{f}_m^I, o_b^{T2T})/\tau\right)}. \tag{1}$$

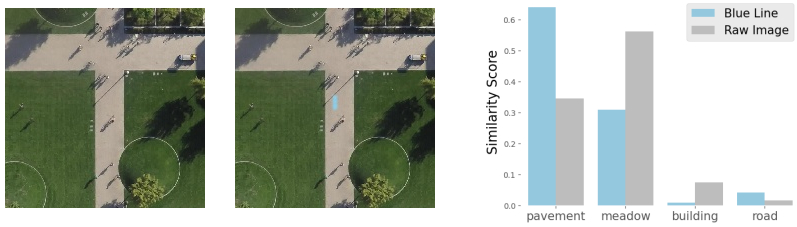
while $\mathcal{L}_{o2i}$ is a symmetric form that can be formulated by switching $\tilde{f}^I$ and o^{T2T} in (1). The complete image-text contrastive loss can be represented by

$$\mathcal{L}_{ITC} = \frac{1}{2}\left(\mathcal{L}_{i2o} + \mathcal{L}_{o2i}\right). \tag{2}$$

Alignment loss. Unlike the conventional technique, which employs cross-entropy loss to align the cross-modal feature for image-text matching [4, 20], the BEV scene in trajectory prediction benchmark do not have the exact caption of text descriptions. Thus, we design the surrounding text prompts F^T to estimate the similarity scores with the feature of colored line BEV scene $\tilde{f}^I$ using CLIP, as shown in Fig. 3. Given that the surroundings are challenging to segment precisely within BEV scene, we aim to learn a representative distribution of the compositional proportion types. To achieve this, we model the surrounding prompts $\hat{F}^S = \{\hat{f}_k^S \in \mathbb{R}^d \mid k = 1, 2, 3, 4\}$ to match the relative scores in scene type distribution using KL divergence, exploring the mutual information across all surroundings. The alignment loss is formulated as

$$\mathcal{L}_A = KL(F^T(\tilde{f}^I)^\intercal \parallel \hat{F}^S). \tag{3}$$

Trajectory modeling loss. Similar to language modeling that generates word vectors for sentence completion, we introduce a way to apply trajectory modeling loss on the text encoder. The basic idea of text generation is to compare



(a) Raw image **(b)** Blue line **(c)** Scene type distribution

Fig. 3: Demonstrating the zero-shot ability of CLIP by scene-surrounding similarity scores. The BEV scene with blue line shows better understanding on the local surroundings, while the raw image considers more on the global view. In this study, drawing color trajectory can significantly influence CLIP's attention to a specific area, which is effective for capturing details of road structure.

the correctness of semantics on word embeddings, which has a pre-defined dictionary to collect all the possibilities of word tokens. In TrajPrompt, we follow the idea of generating the trajectory embeddings $\hat{T}^f$, and retrieve the nearest neighbor from the pre-defined pixel dictionary space $\mathcal{D}$. By doing this, we do not need additional trainable parameters to project the high-dimensional trajectory embeddings into the (w, h) pixel coordinate directly, also ensure the semantics of trajectory embeddings remain the same. The trajectory modeling loss is calculated by regressing the high-dimensional ground truth feature T^f for the future timestamps j, which is performed as

$$\mathcal{L}_{TM} = \sum_{j} \|\hat{t}_j - t_j\|_2^2, \tag{4}$$

and the overall objective for our prompt learning can be represented as

$$\mathcal{L} = \mathcal{L}_A + \mathcal{L}_{ITC} + \mathcal{L}_{TM}. \tag{5}$$

3.4 Trajectory Retrieval

To retrieve the precise location of a pixel from the prediction of trajectory embedding $\hat{t}_j$, where j is defined as future timestamp in Sec. 3.1, we select the top-k closest candidates from the pixel dictionary $\mathcal{D}$. Specifically, these candidates are selected according to the Euclidean distance between $\hat{t}_j$ and the overall pixel embeddings. In addition, the incorporation of probabilistic sampling among candidates allows us to assess trajectory diversity, where a shorter distance indicates a higher probability, and vice versa. Finally, the model can retrieve the corresponding pixel embedding t_j^r from $\mathcal{D}$ by sampling a single future timestamp j. During training, we use the teacher forcing technique to guide the learning of trajectory prompt p_j^t as the residual term in ground truth pixel embedding. Therefore, during inference, the model can follow the comprehensible format to reformulate the input sequence for $j+1$ timestamp with the previously retrieved pixel embedding as $t_j^r + p_j^t$. More discussion details can be found in Sec. 4.7.

4 Experiments

4.1 Experimental Setup

Evaluation Metrics. We follow previous works [1, 29] using Average Displacement Error (ADE) and Final Displacement Error (FDE) as primary metrics for trajectory forecasting. These metrics measure the l_2 distance between predictions and the ground truth over the entire trajectory and the destination point, respectively. To explore the stochastic analysis and compare fairly, we follow earlier works [2, 15] that generate N possible future paths and select the best one to assess the model's performance.

Dataset. Our experiments are measured in real-world BEV scenes, including Stanford Drone Dataset (SDD) [31] and DroneCrowd dataset [41]. These datasets provide large-scale agent trajectories with distinct scenarios. SDD consists the day scene of a university campus with heterogeneous agents, *i.e.* pedestrians, bicyclists, and vehicles, while DroneCrowd is collected from the city street with pedestrians on diverse illumination conditions, *i.e.* cloudy, sunny, and night. Since the trajectory datasets do not include captions to describe the scene, we introduce meaningful textual prompts to connect the understanding of image and trajectory via CLIP. We follow the conventional train-test split to demonstrate the effectiveness of the proposed TrajPrompt against SOTAs.

Implementation Details. The experiments are implemented using Pytorch on a single RTX 3090 GPU. We use ViT-B/32 as the CLIP backbone for the VLM, while the visual-prompt generator adopts a lightweight two linear layers that are the same as [47]. The input BEV scene is cropped to 200×200, which is smaller than the original scene provided by the dataset. However, it retains sufficient information as illustrated in Fig. 6, with the image center corresponding to the agent's current position. The dimension d for embedded vision, language, and trajectory representation is 512. The learnable prompts are trained for 10 epochs using Adam [17] optimizer with the learning rate set to 10^{-3}. During inference, the retrieval of top-k closest embedding is set to $k = 1$ in the deterministic setting, while using $k = 5$ for the stochastic setting with sample diversity.

4.2 Comparison with SOTA Methods

SDD Dataset. Table 1 compares existing methods and TrajPrompt on SDD dataset. The results demonstrate that TrajPrompt outperforms the other SOTAs, achieving 42% and 35% improvements on ADE and FDE, respectively. Besides, TrajPrompt shows the benefit of parameter efficiency when utilizing BEV scene and text as additional sources. Since the pre-trained encoders from CLIP are frozen during prompt learning, the remaining learnable parameters are the soft prompts. Hence, TrajPrompt brings another advantage on 4x fewer learnable parameters than other methods using image, such as TDOR and Y-net. The fewest learnable parameters model is TUTR [35], which uses a relatively low-dimensional trajectory feature to train the transformer-based model. In contrast, although our approach uses high-dimensional features for tokenization with frozen CLIP, it still results in a competitive number of trainable parameters.

Table 1: Performance comparisons on SDD dataset (Evaluated in pixel). All the results are evaluated using 20 samplings, and Params stands for the amount of trainable parameters. The red arrows show the improvement of TrajPrompt compared to the best among SOTAs (highlighted as underline).

Method	Models *without* image information						Models *with* image information			
	Social-GAN [15] CVPR'18	PECNet [27] ECCV'20	NSP-SFM [46] ECCV'22	Leapfrog [28] CVPR'23	FlowChain [25] ICCV'23	TUTR [35] ICCV'23	SOPHIE [32] CVPR'19	Y-net [26] ICCV'21	TDOR [14] CVPR'22	**TrajPrompt (ours)**
ADE (↓)	27.33	9.96	6.52	8.48	9.93	7.76	16.27	7.85	6.77	**3.78** (↓ 42%)
FDE (↓)	41.44	15.88	10.61	11.66	17.17	12.69	29.38	11.85	10.46	**6.81** (↓ 35%)
Params	1.75M	2.10M	0.64M	11.20M	1.63M	0.11M	26.23M	1.64M	0.85M	0.17M

Table 2: Performance comparisons on DroneCrowd dataset.

Method	ADE (↓)	FDE (↓)
PreTraM [43]	7.27	13.15
TDOR [14]	6.92	11.23
TUTR [35]	7.68	13.52
TrajPrompt	**3.72**	**7.04**

Table 3: Comparison of computation cost and inference time.

Method	GFLOPs (↓)	Time (ms)
Y-net [26]	131.36	453.38
Leapfrog [28]	27.42	328.41
FlowChain [25]	10.68	47.25
TrajPrompt	**1.87**	**15.24**

DroneCrowd Dataset. To evaluate the effectiveness of TrajPrompt on diverse BEV scene outside the campus, we notice a potential dataset DroneCrowd, which originally benchmarks pedestrian tracking and offers another avenue for trajectory forecasting assessment. Table 2 shows that the proposed TrajPrompt outperforms the other SOTAs on DroneCrowd dataset, achieving 46% and 37% improvements on ADE and FDE, respectively. Compare to other works that needs to realize the BEV scenario from scratch, our prompt learning approach can be more efficient to transfer the understanding of CLIP and handle the problem well in the diverse city scenes.

Computation Cost. It is worth noting that the computation cost and inference speed are also superior for TrajPrompt. Table 3 compares TrajPrompt with various generative models for trajectory prediction. Specifically, Y-net [26] retains more computation cost due to the construction of the trajectory heatmap and semantic map within all the pixels, leading to a 30x lower inference speed than ours. Leapfrog [28] executes in long inference time because of the diffusion process. Recently, FlowChain [25] provides efficient computation by adopting flow-based structure. Compared to above methods, TrajPrompt outperforms FlowChain by 3x inference speed while using less computation cost by evaluating GFLOPs.

4.3 Effect of Colored Line

Since the colored lines bring benefits in TrajPrompt, we raise two intuitive questions. *What type of line is most effective? What color is most suitable for the line?* To answer these questions, we conduct experiments by analyzing colored line configurations from two perspectives, which is (1) the shape of trajectory points and (2) the color of the drawing line.

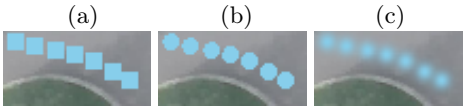
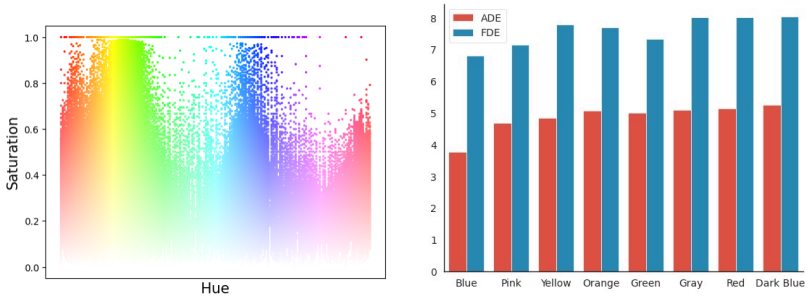


Fig. 4: Configuration of trajectory points.

Table 4: Configuration results.

Configuration	ADE	FDE
(a) Square	7.33	12.67
(b) Circle	4.28	7.60
(c) Gaussian	**3.78**	**6.81**



(a) Color distribution **(b)** Comparing different colors

Fig. 5: In previous study [36], using less frequent colors as visual clues can better excite the model's attention on specific area. Since it is impossible to observe sky or ocean on the BEV scene of campus, we select blue as our main color to draw the line. By demonstrating the pixel-wise color statistic on the overall scene of SDD dataset, it verifies our assumption that blue is exact one of the less frequent colors.

Shape of Trajectory Points. We compare the performance using different ways to draw lines on the BEV scene, as seen in Fig. 4 and Table 4. The square coloring obtains the worst results due to the sharp corner, which will sometimes produce the jagged line when the trajectory points are too close to each other. Coloring with the circle brings a smooth connection of points into line while taking turns or walking in a skewed direction, resulting in a better impact than the former. Eventually, coloring the points by Gaussian gets the best performance because the color of the original background is partially preserved after the Gaussian process. The rich semantics under the line can give more precise clues to make correct decisions.

Color Analysis. Inspired by prior research [36], which found that red circles in images can effectively enhance CLIP's attention to a specific area. From their color analysis on the CUB dataset [39], we observe that the dataset contains the object of bird images against blue sky or green leaves as backgrounds. According to the observation, the infrequent use of red causes it to stand out prominently. Following the idea, we estimate the frequency for each color as shown in Fig. 5a, and assume that choosing a relatively rare color can mark the differences between the *line* and *surroundings* on the BEV scene. Build upon this assumption, we experiment to modify the color of lines in the image and the corresponding text prompt. Our results in Fig. 5b confirm that blue or pink are effective while also being the low-frequency colors within SDD dataset.

Table 5: Ablation study between different combinations of the loss function. Trajectory modeling loss $\mathcal{L}_{TM}$ is added for all the cases to solve the trajectory prediction task.

Method	**TrajPrompt (TP)**	**TP-CL**	**TP-A**	**TP-T**	**TP-IC**	**TP**
Description		*No Contrastive Loss*	*No Alignment Loss*	*No Text Embeddings*	*No Image Condition*	*All*
Strategy	$\mathcal{L}_{ITC}$		✓	✓	✓	✓
	$\mathcal{L}_{A}$	✓			✓	✓
	p^c	✓	✓	✓		✓
ADE / FDE		10.15 / 15.76	5.12 / 9.46	5.21 / 10.22	4.30 / 7.63	**3.78 / 6.81**

4.4 Ablation Studies

Table 5 presents a comparative analysis of the results when specific loss or image condition are excluded during the training process. In the ablation TP-CL, we do not perform contrastive learning, where the trajectory representation cannot map to the image-text space well. Specifically, the TP-CL model lacks a well-constructed understanding of T2T sequence, including *image condition, trajectory and surrounding text*, due to the removal of element $\mathcal{L}_{ITC}$. Based on the analysis, we can conclude that contrastive loss is necessary to emphasize the correlation of global sequence representation with our colored line image. However, different strategies on the contrastive loss also affect the construction of representation space, we leave the detailed discussion in Sec. 4.6.

Comparing the ablation TP-A and TP-T, we demonstrate the effectiveness of adding text embeddings into the specific design of T2T sequence. In these settings, we remove alignment loss $\mathcal{L}_A$ for a clear discussion on the impact of $\mathcal{L}_{ITC}$ and text embeddings. Since the <EOT> token in CLIP gathers information from the entire sequence, any context within the sequence can be informative. Our study shows that by incorporating text prompts regarding surrounding elements in TP-A, the ability to pay attention to these patterns in the BEV scene can be improved, which is useful in making precise decisions on trajectory prediction.

For the overall study, we observe that the simultaneous use of $\mathcal{L}_{ITC}$ and $\mathcal{L}_A$ gain benefits on the trajectory and image-text feature alignment. Also, $\mathcal{L}_A$ can give the information constraint about scene type distribution, which facilitates the localization of trajectory based on the image condition p^c.

4.5 Qualitative Results

We compare TrajPrompt with the baseline TDOR by visualizing trajectory prediction results on SDD and DroneCrowd datasets. We first discuss Fig. 6a and Fig. 6b, which is evaluated from SDD dataset. As shown in Fig. 6a, TrajPrompt-S predicts trajectories with high diversity in speed and direction, while TDOR is constrained in a fixed pattern of moving straight forward, which makes too conservative decisions at the *road* intersection on campus. Fig. 6b shows that TrajPrompt accurately describes a person standing in place or with small movements, especially the deterministic approach in TrajPrompt-D, while TDOR predicts radioactively paths in random directions, which does not consider the structure of *pavement* and *meadow*. Fig. 6c and Fig. 6d are the city street scenarios from DroneCrowd dataset, which is different from the predictions made on campus. Since it is more dangerous on the city street, the decision of the

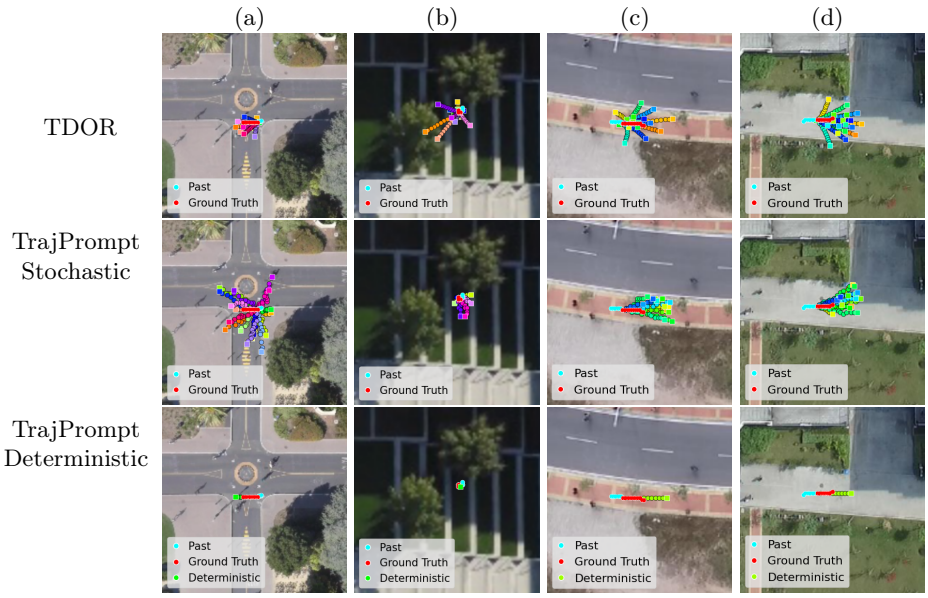


Fig. 6: Comparing the visualization results with baseline. TDOR is based on the stochastic process, thus sampling multiple trajectories is a proper way to observe their overall performance. TrajPrompt is flexible to provide both the stochastic and deterministic results, represented as TrajPrompt-S ($k = 5$) and TrajPrompt-D ($k = 1$).

agents should be more cautious. As shown in Fig. 6c and Fig. 6d, TrajPrompt-S carefully predict the trajectories with lower diversity, while TDOR does not care about the scenarios and make casual decisions just like walking on campus. In conclusion, TrajPrompt better understands the agent behavior based on the surroundings, exhibiting diverse trajectory patterns rather than consistently following the arc-shaped prediction like TDOR.

4.6 Effect of Image-Text Contrastive Loss

Since we introduce the use of text prompts and colored line image in TrajPrompt, we differentiate our contrastive loss $\mathcal{L}_{ITC}$ from the previous $\mathcal{L}_{MTCL}$ [43] as shown in Table 6. Based on the study, we verify that using a simple map-trajectory contrastive loss $\mathcal{L}_{MTCL}$ is not enough to bridge the gap between the BEV scene and trajectory, while adding more proper clues within the input sources can be a better way to excite the potential of contrastive learning.

Table 6: Comparing the difference on contrastive loss.

Method	Strategy	ADE	FDE
PreTraM [43]	$\mathcal{L}_{MTCL}$	7.90	15.37
TrajPrompt	$\mathcal{L}_{ITC}$	**4.69**	**8.29**

Table 7: Comparing the difference on embedding prediction and retrieval.

Decoding method	ADE	FDE
Embedding prediction	17.03	32.61
Top-k retrieval ($k = 1$)	**7.73**	**14.38**

4.7 Effect of Trajectory Retrieval

In Table 7, we demonstrate different decoding methods on output embeddings to obtain the pixel location during inference. The predicted trajectory embedding cannot exactly match any representation within the pixel dictionary $\mathcal{D}$. Therefore, if this unrecognized prediction embedding is passed to the next step of the autoregressive process, it can cause confusion in the model and lead to poor performance on the iterative steps. In comparison, retrieving the closest embedding from $\mathcal{D}$ for each decoding step can ensure that the following trajectory input is a recognizable representation for TrajPrompt. The decoding method used by TrajPrompt leverages the retrieval process and achieves great success, which reduces a significant gap compared to the direct prediction of trajectory embeddings.

5 Limitation and Discussion

The frozen CLIP model has difficulty in computing fully labeled image or synthesized images, because most of the pre-trained data for CLIP are natural scene. However, general BEV scene provided by the autonomous driving datasets are constructed by unnatural formats. For example, the BEV scene provided by nuScenes [5] is fully assigned by pixel-wise labels as the semantic content. In other words, the texture and structure of surroundings in the semantic map may differ from the natural concepts of visual appearance, making it difficult for the CLIP model to understand. Argoverse [7] provides the dataset with plain attributes on road and separable lanes, using a vector graph to represent the overall scenario, without any other semantics. This is even challenging for CLIP since the only thing that the model can observe is the walkable area.

6 Conclusion

In this work, we propose TrajPrompt, which employs a novel vision-language prompt approach and uses a significantly low amount of learnable parameters to integrate the understanding of colored trajectory lines into frozen CLIP encoders. By designing the Table-to-Text sequence along with three objectives to enhance the ability of trajectory alignment in the image-text space, TrajPrompt outperforms the SOTAs in the trajectory prediction task. The experiment demonstrates the benefit of incorporating colored lines on the BEV scene and corresponding text descriptions about the surroundings. This approach has achieved significant success in establishing the relationship between trajectories and images. In the future, we plan to employ the potential for broad application in domains necessitating precise position guidance with unified visual semantics on BEV scene, especially the integration of line on the data sources. We hope our paper inspires future research in developing pre-trained vision-language models for advancing topic such as localization and tracking using drones.

Acknowledgment

This work is partially supported by the National Science and Technology Council, Taiwan, under Grants: NSTC-112-2628-E-002-033-MY4, NSTC-112-2634-F-002-002-MBK, NSTC-112-2221-E-A49-059-MY3 and NSTC-112-2221-E-A49-094-MY3, and was financially supported in part by the Center of Data Intelligence: Technologies, Applications, and Systems, National Taiwan University (Grants: 113L900901/113L900902/113L900903), from the Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education, Taiwan.

References

1. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social lstm: Human trajectory prediction in crowded spaces. In: CVPR. pp. 961–971 (2016)
2. Amirian, J., Hayet, J.B., Pettré, J.: Social ways: Learning multi-modal distributions of pedestrian trajectories with gans. In: CVPRW (2019)
3. Aydemir, G., Akan, A.K., Güney, F.: Adapt: Efficient multi-agent trajectory prediction with adaptation. In: ICCV. pp. 8295–8305 (2023)
4. Bao, H., Wang, W., Dong, L., Liu, Q., Mohammed, O.K., Aggarwal, K., Som, S., Piao, S., Wei, F.: Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. In: NeurIPS. vol. 35, pp. 32897–32912 (2022)
5. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: CVPR. pp. 11621–11631 (2020)
6. Cao, Y., Xiao, C., Anandkumar, A., Xu, D., Pavone, M.: Advdo: Realistic adversarial attacks for trajectory prediction. In: ECCV. pp. 36–52 (2022)
7. Chang, M.F., Lambert, J., Sangkloy, P., Singh, J., Bak, S., Hartnett, A., Wang, D., Carr, P., Lucey, S., Ramanan, D., et al.: Argoverse: 3d tracking and forecasting with rich maps. In: CVPR. pp. 8748–8757 (2019)
8. Chen, H., Wang, J., Shao, K., Liu, F., Hao, J., Guan, C., Chen, G., Heng, P.A.: Traj-mae: Masked autoencoders for trajectory prediction. In: ICCV (2023)
9. Dharmadhikari, M., Dang, T., Solanka, L., Loje, J., Nguyen, H., Khedekar, N., Alexis, K.: Motion primitives-based path planning for fast and agile exploration using aerial robots. In: ICRA. pp. 179–185 (2020)
10. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2020)
11. Giuliari, F., Hasan, I., Cristani, M., Galasso, F.: Transformer networks for trajectory forecasting. In: ICPR. pp. 10335–10342 (2021)
12. Gu, J., Hu, C., Zhang, T., Chen, X., Wang, Y., Wang, Y., Zhao, H.: Vip3d: End-to-end visual trajectory prediction via 3d agent queries. In: CVPR. pp. 5496–5506 (2023)
13. Gu, J., Sun, C., Zhao, H.: Densetnt: End-to-end trajectory prediction from dense goal sets. In: ICCV. pp. 15303–15312 (2021)
14. Guo, K., Liu, W., Pan, J.: End-to-end trajectory distribution prediction based on occupancy grid maps. In: CVPR. pp. 2242–2251 (2022)

15. Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A.: Social gan: Socially acceptable trajectories with generative adversarial networks. In: CVPR. pp. 2255–2264 (2018)
16. Jia, M., Tang, L., Chen, B.C., Cardie, C., Belongie, S., Hariharan, B., Lim, S.N.: Visual prompt tuning. In: ECCV. pp. 709–727 (2022)
17. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)
18. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In: ICML (2023)
19. Li, J., Li, D., Xiong, C., Hoi, S.: Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: ICML. pp. 12888–12900 (2022)
20. Li, J., Selvaraju, R., Gotmare, A., Joty, S., Xiong, C., Hoi, S.C.H.: Align before fuse: Vision and language representation learning with momentum distillation. In: NeurIPS. vol. 34, pp. 9694–9705 (2021)
21. Li, L.H., Yatskar, M., Yin, D., Hsieh, C.J., Chang, K.W.: What does bert with vision look at? In: ACL. pp. 5265–5275 (2020)
22. Li, X.L., Liang, P.: Prefix-tuning: Optimizing continuous prompts for generation. In: ACL. pp. 4582–4597 (2021)
23. Liu, Y., Zhang, J., Fang, L., Jiang, Q., Zhou, B.: Multimodal motion prediction with stacked transformers. In: CVPR. pp. 7577–7586 (2021)
24. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV. pp. 10012–10022 (2021)
25. Maeda, T., Ukita, N.: Fast inference and update of probabilistic density estimation on trajectory prediction. In: ICCV. pp. 9795–9805 (2023)
26. Mangalam, K., An, Y., Girase, H., Malik, J.: From goals, waypoints & paths to long term human trajectory forecasting. In: ICCV. pp. 15233–15242 (2021)
27. Mangalam, K., Girase, H., Agarwal, S., Lee, K.H., Adeli, E., Malik, J., Gaidon, A.: It is not the journey but the destination: Endpoint conditioned trajectory prediction. In: ECCV. pp. 759–776 (2020)
28. Mao, W., Xu, C., Zhu, Q., Chen, S., Wang, Y.: Leapfrog diffusion model for stochastic trajectory prediction. In: CVPR. pp. 5517–5526 (2023)
29. Pellegrini, S., Ess, A., Schindler, K., Van Gool, L.: You'll never walk alone: Modeling social behavior for multi-target tracking. In: ICCV. pp. 261–268 (2009)
30. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML. pp. 8748–8763 (2021)
31. Robicquet, A., Sadeghian, A., Alahi, A., Savarese, S.: Learning social etiquette: Human trajectory understanding in crowded scenes. In: ECCV. pp. 549–565 (2016)
32. Sadeghian, A., Kosaraju, V., Sadeghian, A., Hirose, N., Rezatofighi, H., Savarese, S.: Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In: CVPR. pp. 1349–1358 (2019)
33. Salzmann, T., Ivanovic, B., Chakravarty, P., Pavone, M.: Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In: ECCV. pp. 683–700 (2020)
34. Shafiee, N., Padir, T., Elhamifar, E.: Introvert: Human trajectory prediction via conditional 3d attention. In: CVPR. pp. 16815–16825 (2021)
35. Shi, L., Wang, L., Zhou, S., Hua, G.: Trajectory unified transformer for pedestrian trajectory prediction. In: ICCV. pp. 9675–9684 (2023)

36. Shtedritski, A., Rupprecht, C., Vedaldi, A.: What does clip knows about a red circle? visual prompt engineering for vlms. In: ICCV (2023)
37. Sun, J., Li, Y., Chai, L., Fang, H.S., Li, Y.L., Lu, C.: Human trajectory prediction with momentary observation. In: CVPR. pp. 6467–6476 (2022)
38. Tsao, L.W., Wang, Y.K., Lin, H.S., Shuai, H.H., Wong, L.K., Cheng, W.H.: Social-ssl: Self-supervised cross-sequence representation learning based on transformers for multi-agent trajectory prediction. In: ECCV. pp. 234–250 (2022)
39. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset. Tech. Rep. CNS-TR-2011-001, California Institute of Technology (2011)
40. Wang, Z., Liu, J.C.: Translating math formula images to latex sequences using deep neural networks with sequence-level training. International Journal on Document Analysis and Recognition **24**(1-2), 63–75 (2021)
41. Wen, L., et al.: Detection, tracking, and counting meets drones in crowds: A benchmark. In: CVPR (2021)
42. Wong, C., Xia, B., Hong, Z., Peng, Q., Yuan, W., Cao, Q., Yang, Y., You, X.: View vertically: A hierarchical network for trajectory prediction via fourier spectrums. In: ECCV. pp. 682–700 (2022)
43. Xu, C., Li, T., Tang, C., Sun, L., Keutzer, K., Tomizuka, M., Fathi, A., Zhan, W.: Pretram: Self-supervised pre-training via connecting trajectory and map. In: ECCV. pp. 34–50 (2022)
44. Yao, Y., Zhang, A., Zhang, Z., Liu, Z., Chua, T.S., Sun, M.: Cpt: Colorful prompt tuning for pre-trained vision-language models. arXiv preprint arXiv:2109.11797 (2021)
45. Yuan, Y., Weng, X., Ou, Y., Kitani, K.M.: Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In: ICCV. pp. 9813–9823 (2021)
46. Yue, J., Manocha, D., Wang, H.: Human trajectory prediction via neural social physics. In: ECCV. pp. 376–394 (2022)
47. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Conditional prompt learning for vision-language models. In: CVPR. pp. 16816–16825 (2022)
48. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. IJCV **130**(9), 2337–2348 (2022)