Thank you very much for your recognition of the main contributions of our work and the experimental results, as well as your positive feedback on the organization and expression of our paper. Below is our detailed response to your comments and questions. If any of our responses fail to adequately address your concerns, please let us know, and we will promptly follow up.

> **[W1]** Figure 2 presents difficulties in terms of clarity and comprehension. It may be beneficial to reconsider the inclusion of the Equation within the framework.

Thank you for this constructive suggestion. We will reconsider the organization of Figure 2 and remove the Equation from the framework in the revision.

> **[W2]** Writing inconsistencies. For example, Equation (11) misses a comma and Equation (6) lacks an explanation of μ.

Thanks for pointing out the writing issues. μ denotes the temperature parameter which controls the range of the logits in the softmax [1]. We will supplement the missing comma and the explanation for μ in the revision.

> **[W3]** The experiment does not include comparisons of inference time with the main baselines.

Thanks for your reminder. We have supplemented relevant experiments on both training and inference speed. We conducted experiments on the Beijing dataset with 16-shot setting to test the training and inference time of ST-CLIP and other baseline models using two visual backbone networks: **ResNet-50** and **ViT-B/32**. We conducted a comparison using the official code of baseline models, and the results are presented in the table below:

Model	Training Time (ResNet-50)	Inference Speed (ResNet-50)	Training Time (ViT-B/32)	Inference Speed (ViT-B/32)
CLIP$_{\text{ZS}}$	\	387.8 item/s	\	387.8 item/s
CoOp	10min16s	19.2 item/s	10min40s	19.0 item/s
CoCoOp	12min38s	14.4 item/s	13min11s	14.2 item/s
CLIP-Adapter	6min21s	19.0 item/s	7min14s	19.4 item/s
Tip-Adapter	6min24s	375.4 item/s	6min40s	375.1 item/s
Tip-Adapter-F	21min46s	373.2 item/s	22min4s	372.5 item/s
ST-CLIP	11min5s	35.3 item/s	11min33s	34.4 item/s

- The training time of ST-CLIP is moderate. Compared to the optimal baseline model Tip-Adapter-F, it significantly reduces training time while achieving better experimental results. This may be attributed to the fact that Tip-Adapter-F requires initializing parameters with features from the training dataset before training, and additionally involves an extra step of hyperparameter search after training.
- Regarding inference speed, ST-CLIP constructs unique text features for each input image, requiring the use of the text encoder each time. In contrast, for Tip-Adapter-F, the text features for all images are the same and fixed, so they can be pre-computed and reused, resulting in faster inference speed. Compared to other baseline models, ST-CLIP demonstrates faster inference speed due to its ability to simultaneously address all aspects without the need for one-by-one processing.

> **[Q1]** Could you elaborate on how the proposed ST-CLIP utilizes real-time traffic flow or traffic state data?

Thank you for giving us the opportunity to clarify this point. ST-CLIP utilizes real-time traffic flow or traffic state data by incorporating the **Time-varying Properties** introduced in **Section 3.1.2.** Specifically, we divide the day into evenly spaced time slice. For each time slice, we count the number of trajectories passing through a road segment as its traffic flow and calculate the medium speed of the trajectories. We utilize trajectory count (TC) and medium speed (MS) as properties of road segments, which fluctuate over time and are thus referred to as Time-varying Properties. For a given image which is taken on a road segment at the time slice t, we use the real-time time-varying properties of the segment at t to calculate the ST context representation (see Eq. (8-12)), and use the ST context representation as the inputs of the prompt learning module, *i.e.*, SCAMP, of ST-CLIP. In this way, the real-time traffic flow or traffic state data are utilized by ST-CLIP. The real-time traffic state information can assist ST-CLIP in gaining a better understanding of the traffic environment, as traffic conditions are closely linked to certain aspects of traffic scenes, such as [Width] and [Accessibility]. For instance, wider and smoother roads typically experience higher traffic flow and faster average speeds compared to narrower and less well-maintained roads.

> **[Q2]** Since taxi trajectory datasets in the DiDi platform are no longer open-source. How to reproduce your experimental results?

Thanks for your thoughtful concern. To improve the reproducibility of our work, we provide relevant code of our work in the link.

> **[Q3]** Compared with selected baselines, does ST-CLIP have any advantages regarding time and memory usage?

- **Time usage**

  The introduction to the computational complexity of the model is provided in **Appendix C**, following Algorithm 1. Here, we briefly summarize it.

  Since the image and text encoders of ST-CLIP are kept frozen, the primary sources of the model's time complexity lie in two aspects: the **spatio-temporal context representation learning** and **ST-aware multi-aspect prompts learning**. The time complexity of the former one can be approximated as $\mathcal{O}(D^2)$, where D denotes the dimension of the ST-context representation vector. The time complexity of the latter one can be roughly given as $\mathcal{O}((M \times P)^2 D)$, where M represents the length of

learnable prompt and P is number of aspects. In conclusion, the overall time complexity of ST-CLIP is $\mathcal{O}(D^2 + (M \times P)^2 D)$.

Besides, we have also supplemented relevant experiments on both training and inference speed in the answer to **[W3]**.

- **Memory usage**

  Due to the additional trajectory encoder and multi-aspect attention mechanism, the parameter of ST-CLIP is slightly higher compared to the baseline models. However, since the encoders of CLIP are frozen, the number of updatable parameters is limited. Therefore, it is sufficient to run on a NVIDIA Tesla P40 GPU with 8GB of VRAM, as described in **Appendix B**.

References

[1] Radford A, Kim J W, Hallacy C, et al. Learning transferable visual models from natural language supervision[C]//International conference on machine learning. PMLR, 2021: 8748-8763.