A APPENDIX

A.1 ALGORITHM OF STLLM

Algorithm 1: The STLLM Learning Algorithm

Input: The spatial-temporal graph $\mathcal{G}$, the maximum epoch number E, the learning rate η;
Output: Regional Embedding $\mathbf{H}$
and trained parameters in $\mathbf{\Theta}$;

```
Initialize all parameters in Θ;
Design the spatial-temporal prompt P;
Obtain the Embedding matrix F via LLM and P;
Train the framework STLLM by Equation 6
for epoch = 1, 2, ..., E do
    Generate the subgraph G_1 via teh random walk algorithm;
    Send G_1 and the corresponding adjacent matrix to GCN Encoder;
    Obtain the embedding matrix H of N samples of subgraphs;
    Minimize the loss L by Equation 6 using gradient decent with learning rate η;
    for θ ∈ Θ do
        θ = θ − η · ∂L/∂θ
    end
end
Return H and all parameters Θ;
```

In this algorithm, the framework begins by designing a spatial-temporal prompt, which serves as input. The prompt is then passed into a Large Language Model (LLM) to generate an embedding matrix $\mathbf{F}$. Subsequently, a Graph Convolutional Network (GCN) encoder is trained to generate another embedding matrix $\mathbf{H}$ using the GCN equation 2 and optimizing it with the specified loss function 6. Steps 2 and 3 are repeated until convergence, ensuring that the resulting region embedding matrix is representative and captures the desired spatial-temporal information.

A.2 DETAILED ANALYSIS FOR STLLM

In this section, we provide a comprehensive theory analysis based on the work of Oord et al. (2018) on representation learning. This analysis forms the foundation for our approach of spatial-temporal graph contrastive learning using embeddings from a Large Language Model (LLM). The key concept is to leverage the principles outlined in Oord et al. (2018)'s work to enhance the effectiveness of our spatial-temporal graph contrastive learning framework.

$$
\begin{aligned}
&\mathbb{E}_{\tilde{\mathbf{H}}}[\log \frac{g(\mathbf{h},\mathbf{f})}{\sum_{\mathbf{h}_i\in\tilde{\mathbf{H}}} g(\mathbf{h}_i,\mathbf{f})}] \overset{g(\mathbf{h},\mathbf{f})=e^{G(\mathbf{h},\mathbf{f})}}{=} \mathbb{E}_{(\mathbf{h},\mathbf{f})}[G(\mathbf{h},\mathbf{f})] - \mathbb{E}_{(\mathbf{h},\mathbf{f})}[\log \sum_{\mathbf{h}_i\in\tilde{\mathbf{H}}} e^{G(\mathbf{h}_i,\mathbf{f})}] \\
&= \mathbb{E}_{(\mathbf{h},\mathbf{f})}[G(\mathbf{h},\mathbf{f})] - \mathbb{E}_{(\mathbf{h},\mathbf{f})}[\log(e^{G(\mathbf{h},\mathbf{f})}) + \sum_{\mathbf{h}_i\in\tilde{\mathbf{H}}_{\text{neg}}} e^{G(\mathbf{h}_i,\mathbf{f})}] \\
&\leq \mathbb{E}_{(\mathbf{h},\mathbf{f})}[G(\mathbf{h},\mathbf{f})] - \mathbb{E}_{(\mathbf{h},\mathbf{f})}[\log \sum_{\mathbf{h}_i\in\tilde{\mathbf{H}}_{\text{neg}}} e^{G(\mathbf{h}_i,\mathbf{f})}] \\
&= \mathbb{E}_{(\mathbf{h},\mathbf{f})}[G(\mathbf{h},\mathbf{f})] - \mathbb{E}_{\mathbf{h}}[\log \frac{1}{N-1} \sum_{\mathbf{h}_i\in\tilde{\mathbf{H}}_{\text{neg}}} e^{G(\mathbf{h}_i,\mathbf{f})} + \log(N-1)]
\end{aligned} \tag{7}
$$

A.3 DESCRIPTION OF BASELINES

We compare our model, STLLM, with baseline techniques from three research areas: graph representation, graph contrastive learning, and spatial-temporal region representation. This comprehensive analysis allows us to evaluate the strengths and advancements of our model in terms of graph representation, contrastive learning, and spatial-temporal information encoding.

Table 3: Data Description of Experimented Datasets

Data	Census	Taxi Trips	Crime Data	POI Data	House Price
Description of Chicago data	234 regions	54,420	319,733	3,680,125 POIs (130 categories)	44,447
Description of NYC data	180 regions	128,566	60,002	20,659 POIs (50 categories)	22,540

Network Embedding/GNN Approaches. We contrast our model STLLM with a number of typical network embedding and graph neural network models in order to assess its performance. To create region embeddings, we apply these models to our region graph $\mathcal{G}$. Following is a description of each baseline's specifics: **Node2vec** Grover & Leskovec (2016): Using a Skip-gram algorithm based on random walks, it encodes network structure information. **GCN** Kipf & Welling (2017): It carries out the convolution-based message transmission along the edges between neighbor nodes for embedding refinement. It is a graph neural design that permits information aggregation from the sampled sub-graph structures, as stated in the text after the GraphSage Hamilton et al. (2017). Graph Auto-encoder encodes nodes into a latent embedding space with the input reconstruction aim across the graph structures, according to **GAE** Kipf & Welling (2016). By distinguishing the degrees of significance among nearby nodes, the Graph Attention Network improves the classification capabilities of GNNs. **GAT** Veličković et al. (2018): By varying the relevance levels among nearby nodes, the Graph Attention Network improves the capacity of GNNs to discriminate.

Graph Contrastive Learning Methods. We compare our model STLLM with two graph contrastive learning models, and in addition to the aforementioned graph representation and GNN-based models, namely, **GraphCL** You et al. (2020): Based on the maximizing of mutual knowledge, this strategy generates many contrastive viewpoints for augmentation. The goal is to ensure embedding consistency across various connected views. **RGCL** Li et al. (2022): This cutting-edge graph contrastive learning method augments the data based on the intended rationale generator.

Spatial-Temporal Region Representation Models. We contrast it with contemporary spatial-temporal representation techniques for region embedding as well. The following are these techniques: **HDGE** Wang & Li (2017): It creates a crowd flow graph using human trajectory data and embeds areas into latent vectors to maintain graph structural information. **ZE-Mob** Yao et al. (2018): This method uses region correlations to create embeddings while taking into account human movement and taxi moving traces. **MV-PN** Fu et al. (2019): To represent intra-regional and inter-regional correlations, an encoder-decoder network is used. **CGAL** Zhang et al. (2019): An adversarial learning technique that takes into account pairwise graph-structured relations to embed regions in latent space. **MVURE** Zhang et al. (2021): In order to simulate region correlations with inherent region properties and data on human mobility, it makes use of the graph attention mechanism. **MGFN** Wu et al. (2022): In order to aggregate information for both intra-pattern and inter-pattern patterns, it encodes region embeddings with multi-level cross-attention. **GraphST** Zhang et al. (2023b): A robust spatial-temporal graph augmentation is achieved using this adversarial contrastive learning paradigm, which automates the distillation of essential multi-view self-supervised data. By enabling GraphST to adaptively identify challenging samples for improved self-supervision, it improves the representation's resilience and discrimination capacity.

A.4 Category-Specific Crime Prediction Results

In the supplemental materials, we present the comprehensive evaluation findings on various criminal offense types for the cities of Chicago and New York in terms of the MAE and MAPE of 14 techniques. The foundational method for the other 14 methods is ST-SHN. The results in Table 4 demonstrate that our method STLLM consistently produces the best results on all crime categories for the two cities. This clearly demonstrates the substantial advantages that our region's embedding learning framework model brings. We credit the effectiveness of the spatial-temporal region graph's graph encoding in extracting useful regional features for region representation, as well as the various contrastive learning tasks, such as the contrastive learning paradigm for pulling close from the embedding matrix from LLM to that of GCN encoder. Besides, capturing global-view spatial-temporal graph knowledge via LLM is also beneficial to boosting the representation ability of our method.

A.5 Spatio-Temporal Prompt Example

In this section, we provide an illustrative example of a spatio-temporal prompt, as depicted in Figure 7. This example highlights the effectiveness of incorporating spatial information in improving the

Table 4: Overall performance comparison in crime prediction on both Chicago and NYC datasets.

Model	Chicago								New York City							
	Theft		Battery		Assault		Damage		Burglary		Larceny		Robbery		Assault	
	MAE	MPAE	MAE	MPAE	MAE	MPAE	MAE	MPAE	MAE	MPAE	MAE	MPAE	MAE	MPAE	MAE	MPAE
Node2vec	1.1472	0.9871	1.7701	0.8945	1.9781	0.9764	1.9103	0.9712	4.8328	0.8572	0.6697	0.3974	1.1272	0.9566	0.9753	0.7020
GCN	1.1143	0.9675	1.3057	0.8123	1.5578	0.8126	1.5144	0.8173	4.7211	0.8428	0.6288	0.3470	1.0213	0.8626	0.7564	0.6637
GAT	1.1204	0.9708	1.3214	0.8408	1.5942	0.8231	1.5317	0.8188	4.7801	0.8215	0.6301	0.3518	0.9301	0.9293	0.7549	0.6329
GraphSage	1.1241	0.9765	1.3653	0.8609	1.6133	0.8643	1.5801	0.8506	4.7930	0.8448	0.6587	0.3952	0.9673	0.9056	0.7346	0.6423
GAE	1.1134	0.9675	1.3188	0.8193	1.5413	0.7998	1.4997	0.8066	4.7875	0.8395	0.6226	0.3504	0.9492	0.8643	0.7502	0.6308
GraphCL	1.0893	0.9012	1.0628	0.8419	1.3021	0.5261	1.2783	0.6429	4.3819	0.6528	0.6328	0.3562	0.7018	0.4312	0.6189	0.5503
RGCL	1.0790	0.8990	1.0567	0.8312	1.2078	0.5672	1.2084	0.6214	4.3792	0.6458	0.6450	0.3561	0.6901	0.4284	0.6184	0.5497
HDGE	1.0965	0.9123	1.0976	0.8005	1.3987	0.7304	1.3780	0.7367	4.5311	0.7582	0.6655	0.3916	0.8061	0.7049	0.7564	0.6637
ZE-Mob	1.1022	0.9604	1.3246	0.8309	1.5367	0.8201	1.5176	0.8284	4.5414	0.7523	0.6542	0.3870	0.7314	0.6944	0.7355	0.6401
MV-PN	1.0878	0.9201	1.1082	0.7906	1.4032	0.7405	1.3606	0.7245	4.4832	0.7360	0.6518	0.3831	0.7028	0.6871	0.7362	0.6399
CGAL	1.0896	0.9112	1.0876	0.7912	1.3986	0.7351	1.3607	0.7233	4.4935	0.7446	0.6564	0.3898	0.6958	0.5078	0.6572	0.6034
MVURE	1.0863	0.8932	1.0578	0.7983	1.3655	0.6382	1.2985	0.6607	4.4068	0.6663	0.6390	0.3708	0.6813	0.4677	0.6324	0.5882
MGFN	1.0824	0.8953	1.0765	0.7904	1.2943	0.5986	1.2507	0.6299	4.3767	0.6494	0.6595	0.3689	0.6901	0.4530	0.6278	0.5586
GraphST	1.0722	0.4764	0.8933	0.8424	1.1796	0.4387	0.9044	0.4714	4.3564	0.6455	0.6362	0.3430	0.6802	0.4035	0.6083	0.5337
STLLM	**1.0717**	**0.4697**	**0.8392**	**0.7892**	**1.1651**	**0.4217**	**0.8940**	**0.4508**	**4.3430**	**0.6402**	**0.6213**	**0.3261**	**0.6766**	**0.3848**	**0.6028**	**0.5075**

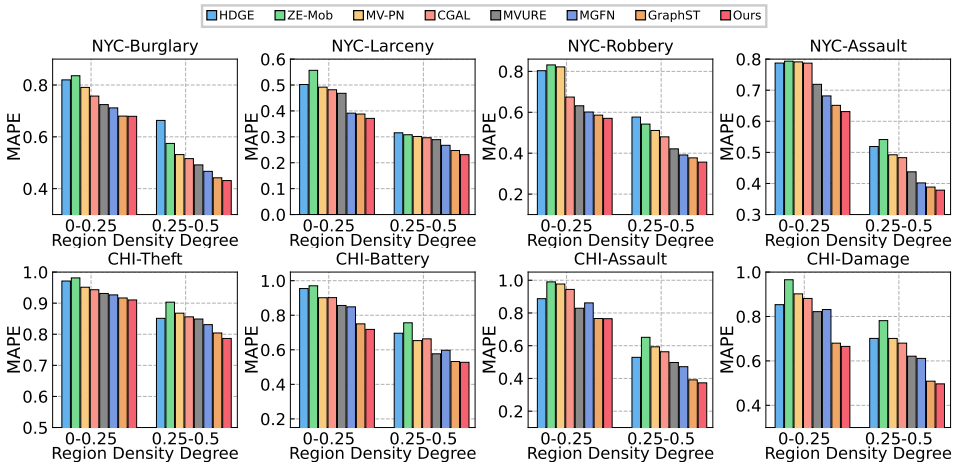

Figure 6: Results on NYC and CHI crime for four crime types *w.r.t* different data density degrees.

performance of summary generation. Additionally, we demonstrate how ChatGPT can generate summaries specifically tailored to the functions of different regions. **In the first case**, we showcase the impact of spatial information on summary generation. By considering the spatial context of the regions, we observe improved summarization results. Furthermore, ChatGPT leverages this spatial information to generate more informative summaries that capture the unique characteristics and functionalities of each region. **In the second case**, we investigate the influence of temporal information on the summarization process. By incorporating temporal dynamics, such as traffic flows and temporal semantics, ChatGPT is able to deduce additional connections and relationships among regions. This leads to more comprehensive and nuanced summaries that reflect the temporal aspects of the regions. **In summary**, by examining these cases, we highlight the significance of both spatial and temporal information in spatio-temporal prompt generation. The integration of these factors enhances the summarization capabilities of ChatGPT, allowing for more contextually-aware and informative summaries tailored to specific regions and their temporal dynamics.

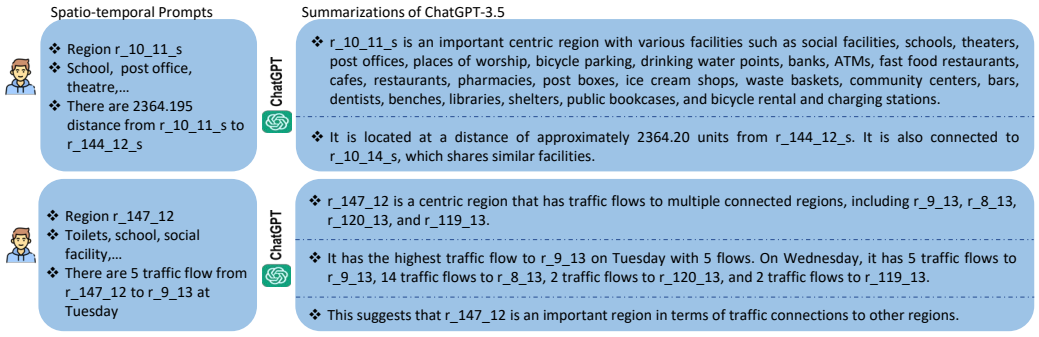

Figure 7: The spatio-temporal prompt example of STLLM.