GeoAggregator: An Efficient Transformer Model for Geospatial Tabular Data

Rui Deng[1], Ziqi Li[2] and Mingshu Wang[1]

1. University of Glasgow, UK
2. Florida State University, USA

- Deng, R., Li, Z., & Wang, M.* (2025). GeoAggregator: An efficient transformer model for geo-spatial tabular data. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 39, No. 11, pp. 11572–11580). AAAI Press.

- Deng, R., Li, Z., & Wang, M.* (2025). Improving the computational efficiency and explainability of GeoAggregator. In Proceedings of the 8th ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery (GeoAI ’25) (pp. 1–4). ACM.

GeoAggregator Paper **GA-sklearn Code**

Geospatial Tabular Data

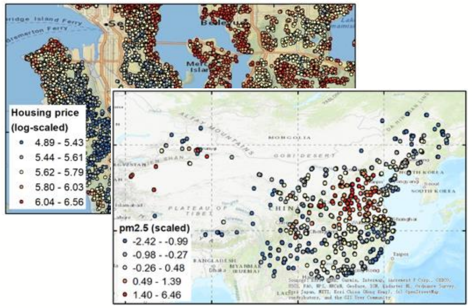
- GTD is used to organise **geospatial observations**:

GTD = geo-referenced rows (data points) × **columns** (features)

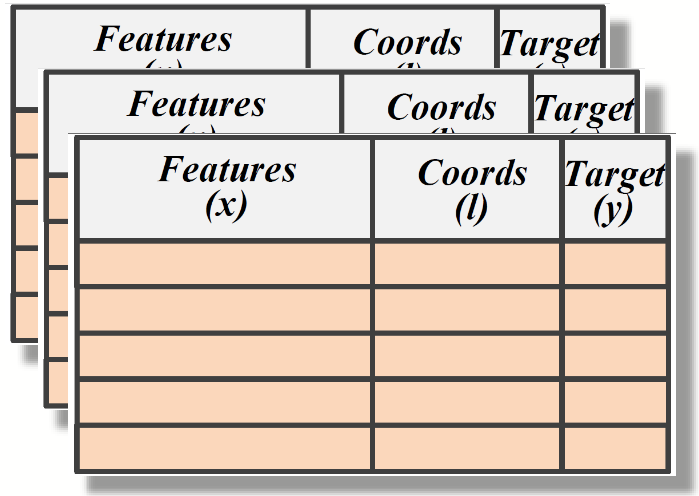
Features *(x)*	*Coords* *(l)*	*Target* *(y)*

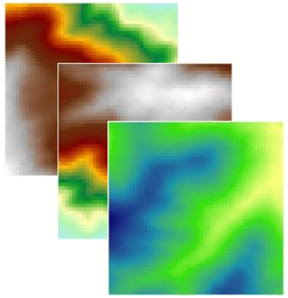
Geospatial Tabular Data

- A format that organises **various data** in a structured manner.



Irregularly sampled points
(GIS vector data)

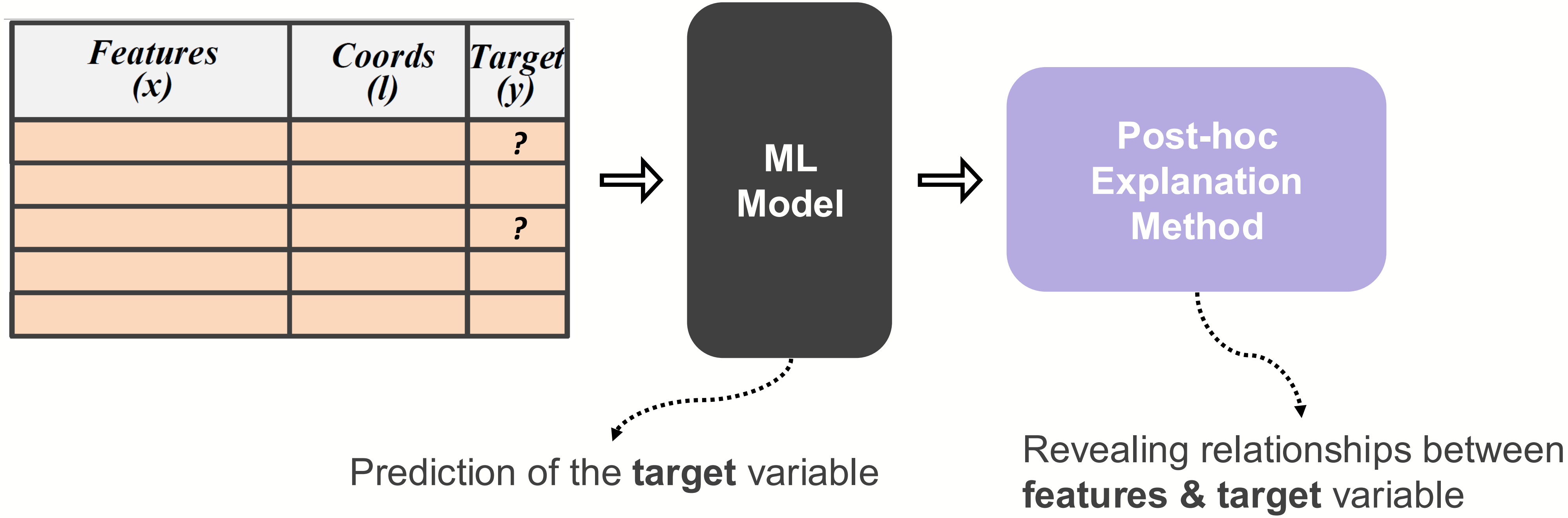




Regular grids
(rasters)

Other panel data…

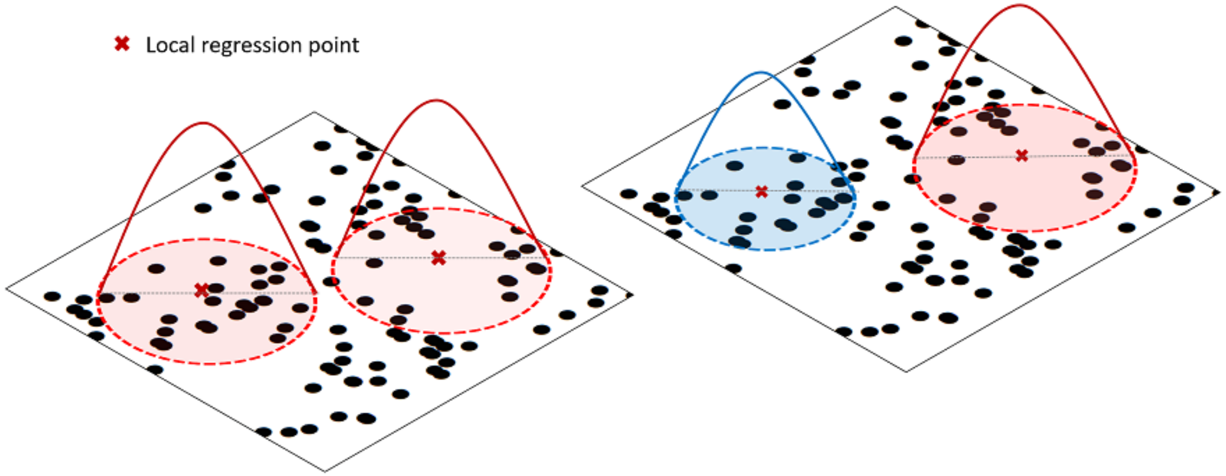
How Machine Learning Models Understand a GTD?



- A **better-performing ML model** benefits the understanding of such relationships.

Related Work (1/3): Statistical Models

- **(Multi-scale) Geographically Weighted Regression (GWR, MGWR)**
  - Strength: Efficient, geospatially explicit and explainable.
  - Limitation 1: Strong assumptions.
  - Limitation 2: Weaker on large/non-linear datasets.



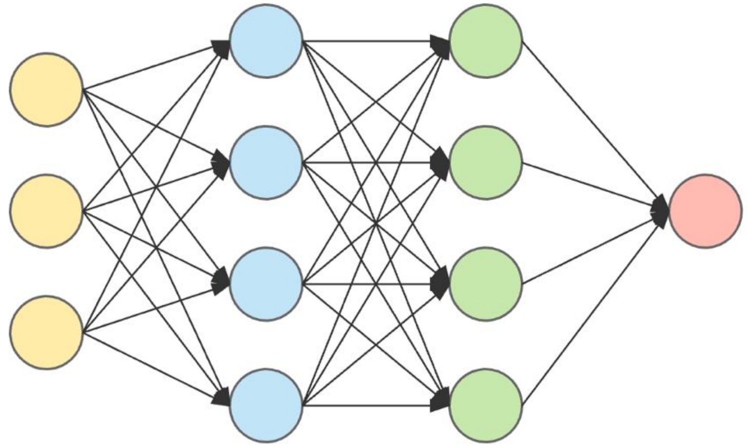
Related Work (2/3): Tree Ensemble Models

- **Gradient Boosting Decision Trees (GBDTs)**
  - Strength 1: Fast to train & explain.
  - Strength 2: Fairly explainable.
  - Limitation: Most are not spatially explicit.

XGBoost   CatBoost   Google’s AutoML

Related Work (3/3): Deep Learning Models

- Traditional **Supervised/semi-supervised** Models
  - Strength: Highly effective.
  - Limitation 1: Slow to train & explain.
  - Limitation 2: Less effective on small datasets.



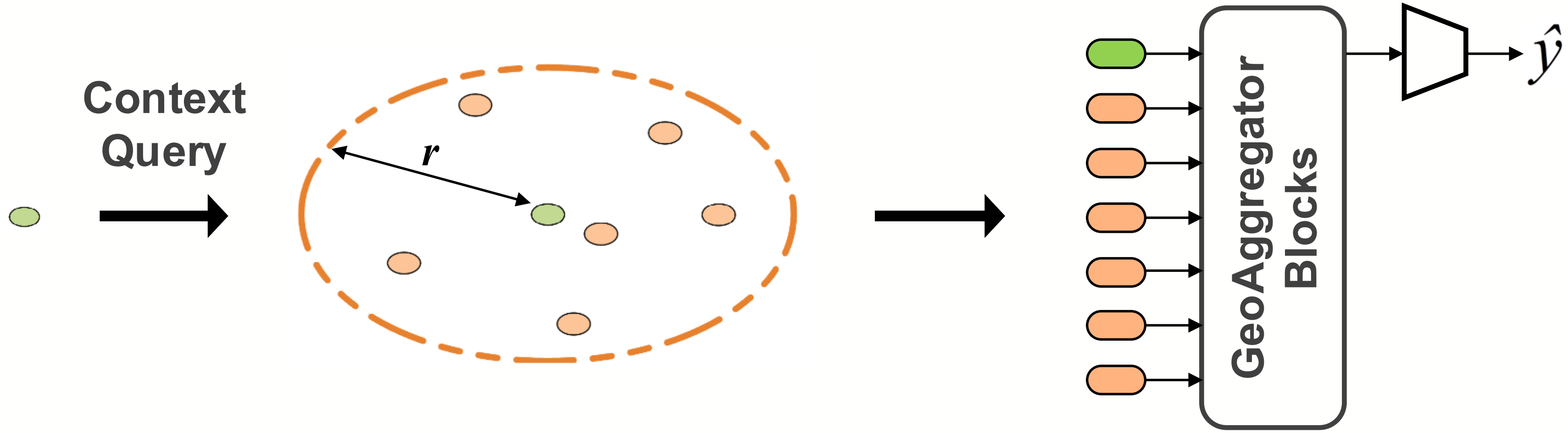
Key challenges in understanding GTD with ML models

- Geospatial explicitness is essential yet <u>underestimated</u>.

- Geospatial explicitness often causes <u>computational overhead</u>.
  - ***Scalability*** of most models (to large datasets) is limited.
  - Construction of graphs or proxy grids are computationally ***expensive.***

Overview of GeoAggregator

- GeoAggregator takes a **target point** & an arbitrarily long sequence of **contextual points** as the input…

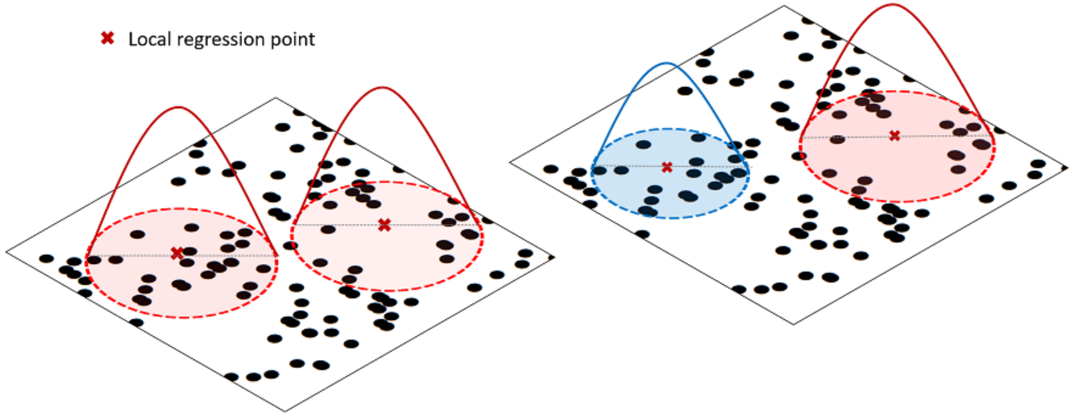


- … and **directly** learns interactions between target and contextual points to account for spatial effects.

Deng, R., Li, Z. and Wang, M. (2025) "GeoAggregator: An Efficient Transformer Model for Geo-Spatial Tabular Data", Proceedings of the AAAI Conference on Artificial Intelligence, 39(11), pp. 11572-11580. doi: 10.1609/aaai.v39i11.33259.

Key Design 1: Spatial Explicitness

- Considering a local **prior** inspired by the First law of Geography & GWR…

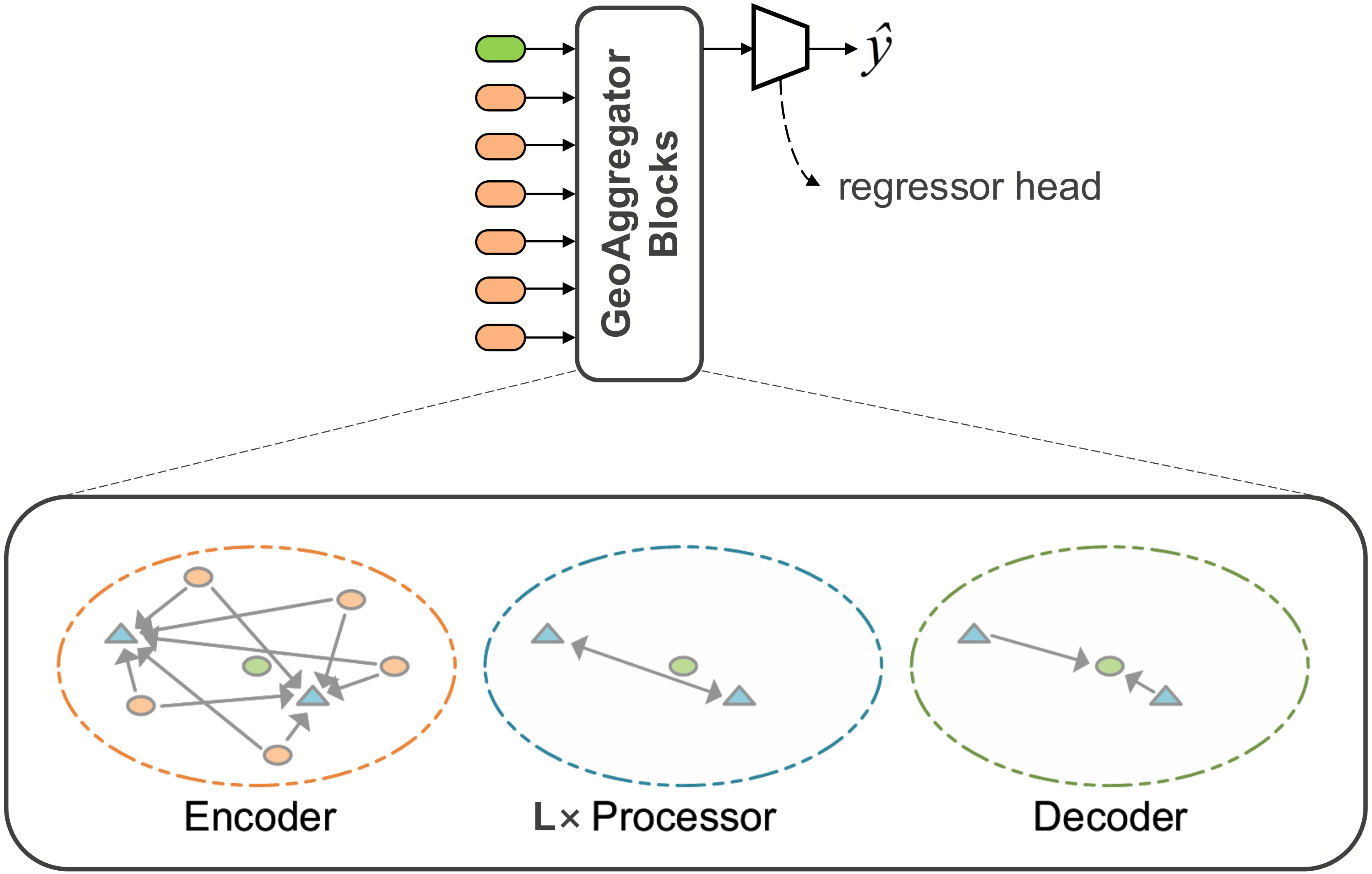


- …we modify the attention scores between point pairs with a **Gaussian distance-decay bias**, so that:

Model generally attends more to near points than distant ones.

Key Design 2: Keep the Model Lightweight

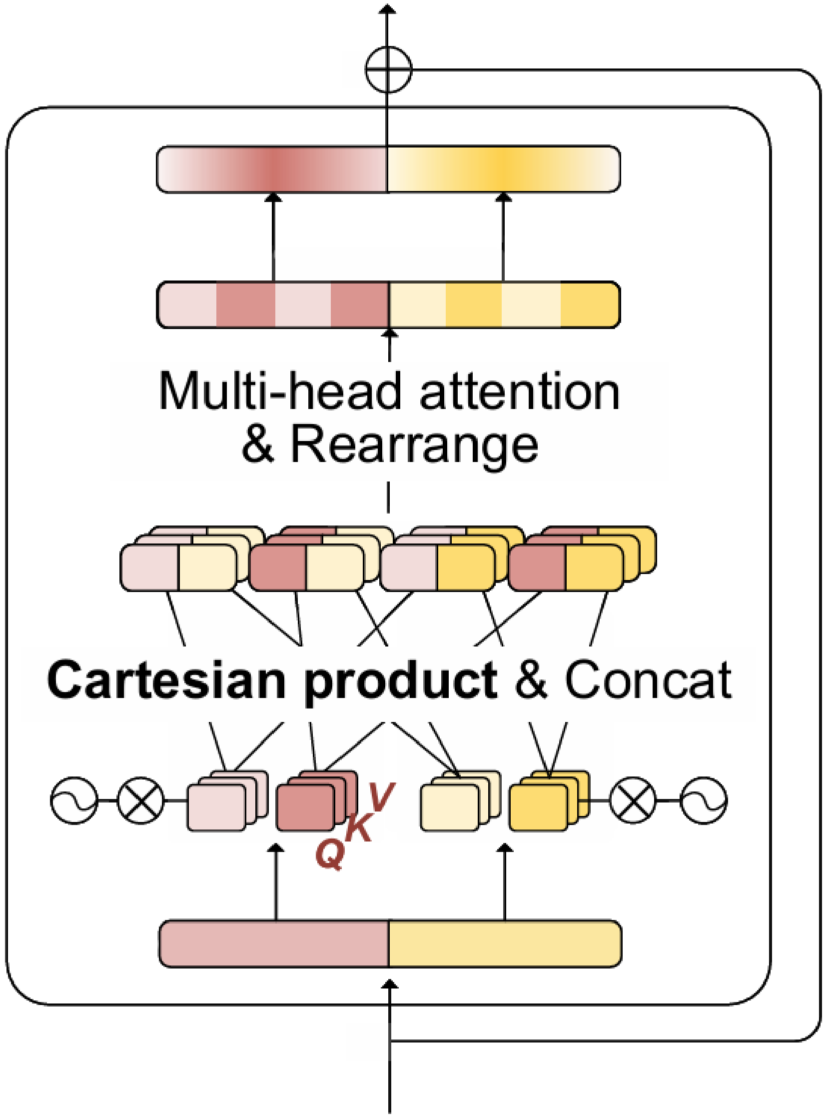
- **Encoder** aggregates contextual points to latent points.
- **Processor** learns interaction between latent points.
- **Decoder** aggregates latent points into an embedding vector.



Key Design 2: Keep the Model Lightweight

- **Masked Cartesian Product Attention (MCPA)** reduces learnable parameters in attention mechanism.

- Compared to Vanilla Attention, MCPA lowers learnable parameters **by ~40%** per layer for our architecture.



Evaluation 1: How Well Does GeoAggregator Handle Various Spatial Processes?

- We compare GeoAggregator against **6 baseline models**, on **8 synthetic datasets**, covering 4 Data Generating Processes and 2 types of covariates:

Type	Model Name
Spatial Statistical Model	GWR
GBDT	XGBoost
Graph Neural-Net	SRGCNN
Convolutional Neural-Net	GCNNWR
Hybrid Ensemble	GSH-EL
GeoAgregator (without MCPA)	Vanilla
GeoAggregator (with MCPA)	GA

Data Generating Process	Spatial Lag of X	Spatial Lag of y
GWR		
Spatial Lagged X	✓	
Spatial Lag		✓
Spatial Durbin	✓	✓

Evaluation 1: How Well Does GeoAggregator Handle Various Spatial Processes?

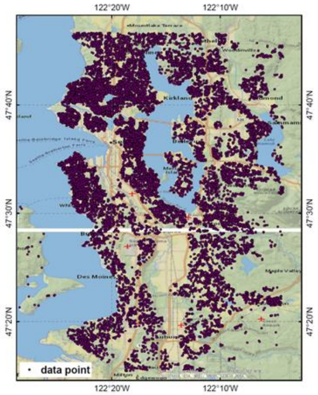
- GeoAggregators show strong performance in modeling **Spatial Lag Effects** and **Spatial Heterogeneity**.

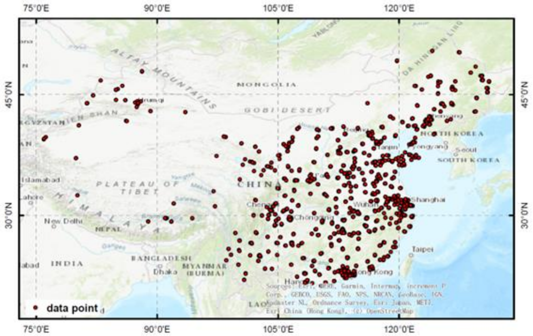
Model	Lin-r		SL-r		SLX-r		Durbin-r		Lin-d		SL-d		SLX-d		Durbin-d		#best
	MAE	R^2	MAE	R^2	MAE	R^2	MAE	R^2	MAE	R^2	MAE	R^2	MAE	R^2	MAE	R^2	
GWR	0.873	0.906	1.445	0.860	0.864	0.697	1.338	0.705	0.830	0.802	**0.860**	**0.991**	0.807	0.607	0.869	0.980	2
XGBoost	0.957	0.895	1.719	0.820	0.891	0.746	1.642	0.649	0.840	0.827	0.880	0.991	0.822	0.720	0.912	0.979	0
SRGCNN	2.805	-0.271	2.904	0.374	1.78	-0.184	1.754	0.611	1.013	0.691	0.982	0.984	1.066	0.499	0.900	0.977	0
GCNNWR	0.897	0.907	**1.133**	**0.929**	0.898	0.768	1.013	0.858	0.818	0.827	0.868	0.990	0.864	0.713	**0.857**	**0.983**	4
GSH-EL	1.382	0.720	2.884	0.157	0.961	0.647	2.178	-0.178	1.115	0.547	2.938	0.828	0.917	0.558	1.791	0.921	0
Vanilla-mini	0.887	0.909	1.213	0.919	**0.839**	0.753	0.929	0.884	0.913	0.800	1.124	0.984	0.836	0.709	0.995	0.978	1
Vanilla-small	0.892	0.908	2.351	0.694	0.844	0.751	**0.897**	**0.892**	0.878	0.817	0.980	0.988	0.833	0.710	0.924	0.981	2
Vanilla-large	0.880	0.912	2.555	0.642	0.848	0.749	1.554	0.666	0.883	0.816	1.223	0.982	0.830	0.713	1.143	0.970	0
GA-mini	**0.870**	**0.920**	1.269	0.911	0.881	0.751	0.946	0.876	**0.818**	0.810	1.039	0.985	**0.804**	0.710	1.054	0.971	4
GA-small	0.905	0.910	1.530	0.840	0.872	**0.771**	0.920	0.880	0.851	**0.827**	1.000	0.988	0.819	0.736	1.280	0.964	2
GA-large	0.905	0.911	2.383	0.646	0.870	0.771	1.847	0.547	0.867	0.823	1.169	0.984	0.820	**0.737**	1.165	0.970	1

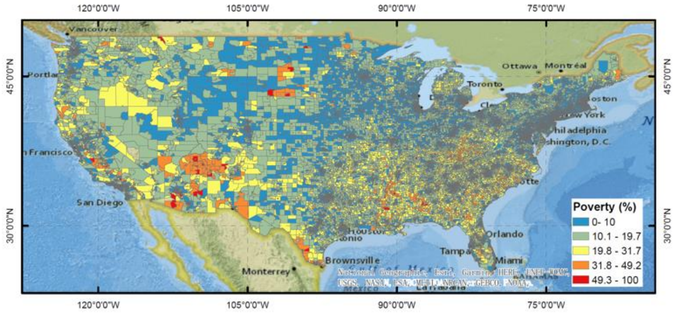
Evaluation 2: Is GeoAggregator Competitive in Real-World Scenarios?

- Yes - We evaluate GeoAggregator on **4 real-world** datasets, GeoAggregator performs well especially on **larger datasets**.

Dataset	# Data Points	# Covariates (X)
PM2.5	1,457	7
Election	3,108	14
Housing	16,580	8
Poverty	71,900	14







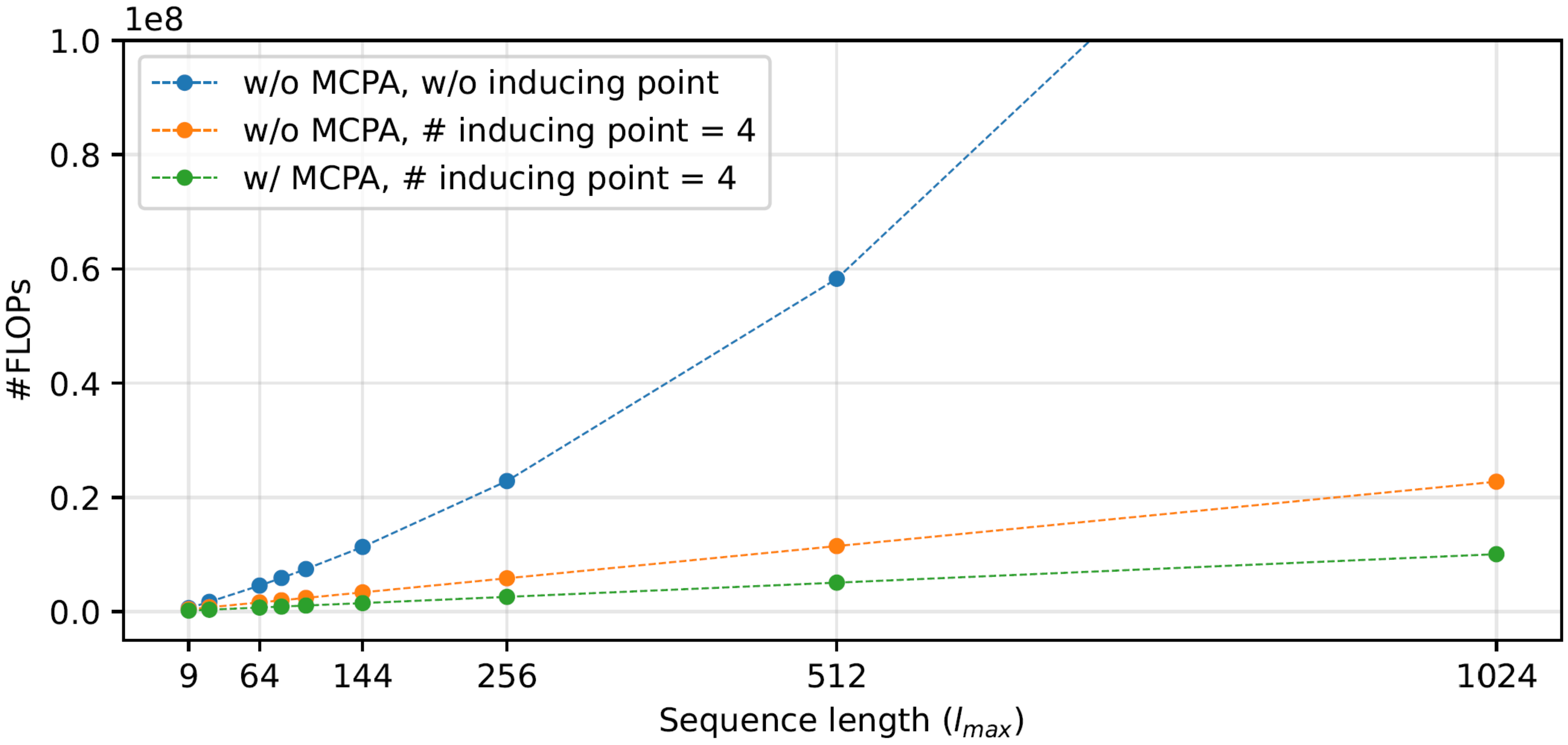
Evaluation 3: Is GeoAggregator Computationally Efficient?

- Yes – GeoAggregators show **low computational complexity**.

Model	Synthetic		PM25		Housing		Poverty	
	#Params	#FLOPs	#Params	#FLOPs	#Params	#FLOPs	#Params	#FLOPs
SRGCNN			224.4K	670.9M	2820K	8.5M	-	-
GCNNWR	1930K	474.6M	1210K	231.2M	8910K	3281.8M	-	-
GSH-EL	460K	29.3M	220K	14.0M	2980K	190.1M	-	-
GA-mini	**4.3K**	**0.7M**	**4.6K**	**1.6M**	**4.6K**	**1.6M**	**4.9K**	**1.7M**
GA-small	6.3K	1.4M	6.5K	2.9M	6.6K	3.0M	6.8K	3.2M
GA-large	8.2K	2.7M	8.5K	5.7M	8.5K	5.8M	8.8K	6.2M

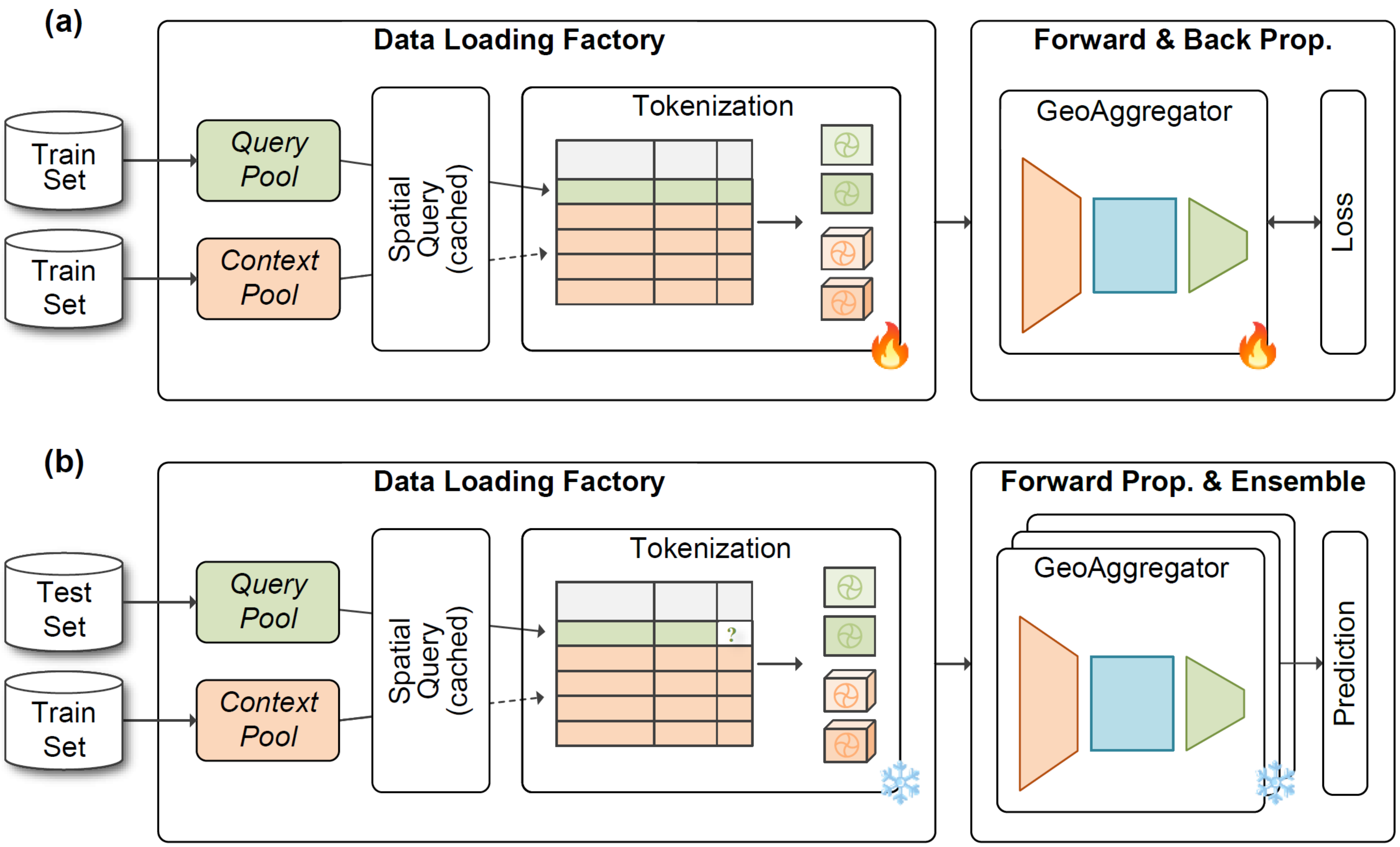
Evaluation 3: Is GeoAggregator Computationally Efficient?

- Yes – GeoAggregators show **low computational complexity**.



Our Fast and Easy-to-use Implementation: GA-Sklearn

- ✓ Supervised model **fitting**.
- ✓ **Fast inference** with post-hoc ensemble.
- ✓ **Explainability** function (SHAP/GeoShapley).



Our Fast and Easy-to-use Implementation: GA-Sklearn

Pseudo Python code 1. Initiate your model

```python
from model.estimator import GARegressor


# Set hyperparameters for your model
hps = {
    'model_variant': 'mini',
    'seq_len': 144,
    'epochs': 21,
    'lr': 5e-3,
    'verbose': True,
    ...
}
model = GARegressor(**hps)

```

GA-sklearn Code

Our Fast and Easy-to-use Implementation: GA-Sklearn

Pseudo Python code 2. Train your model

```
# Suppose you have loaded a GTD
# Pass covariates, coords and target variable separately
model.fit(
    X=X_trn, l=l_trn, y=y_trn
)

```

GA-sklearn Code

Our Fast and Easy-to-use Implementation: GA-Sklearn

Pseudo Python code 3. Make inference

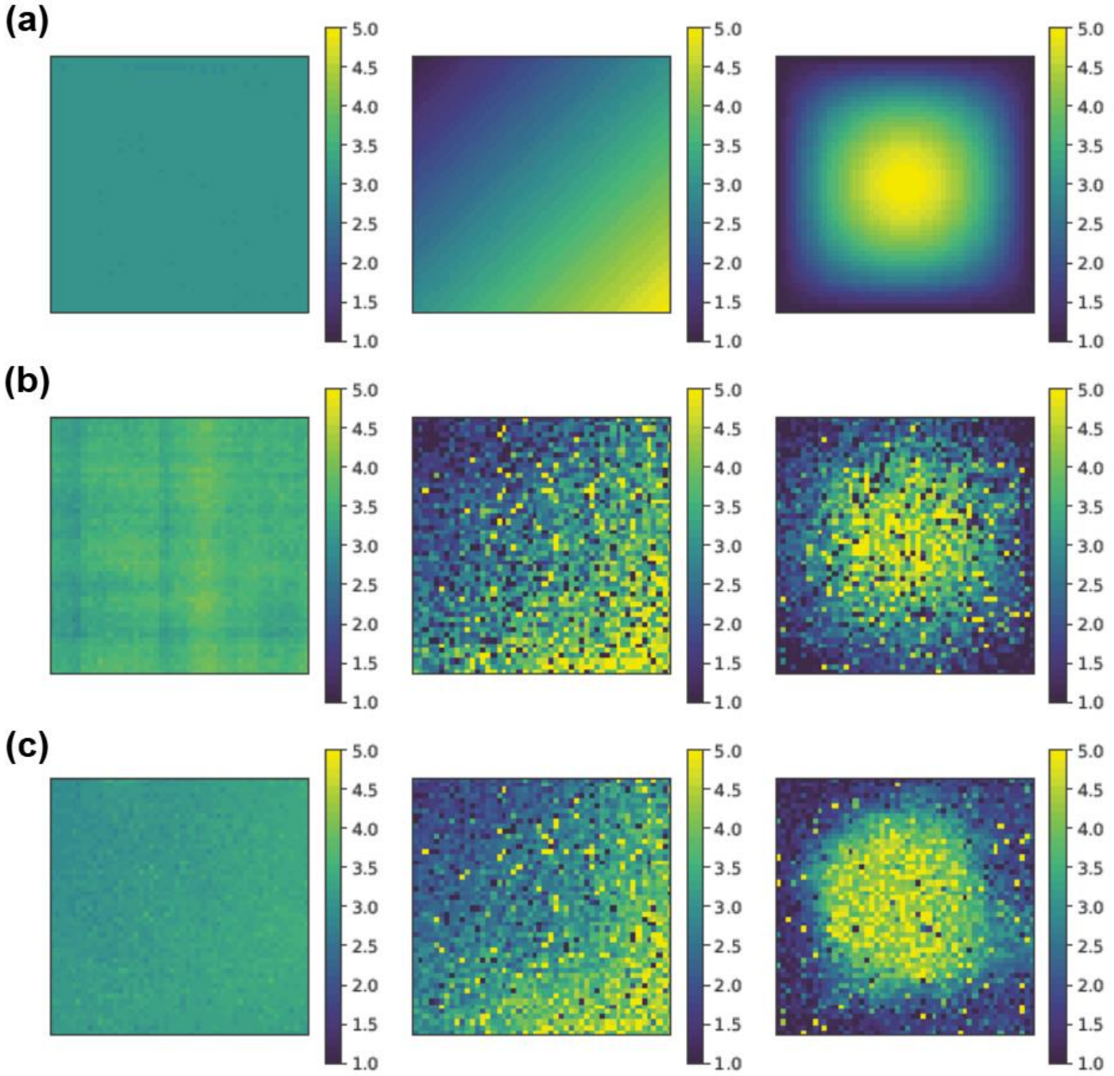
```
# Suppose you have trained your model
y_pred, y_pred_std = model.predict(
    X=X_tst, l=l_tst,
    n_estimate=8,
    get_std=True
)

```

GA-sklearn Code

Our Fast and Easy-to-use Implementation: GA-Sklearn

Explainability Results



Real

Retrieved by **XGBoost**

Retrieved by **GeoAggregator**

Conclusion and Key Takeaways

- GeoAggregator is a **spatially explicit and efficient** transformer model designed specifically for understanding GTD.

Deng, R., Li, Z., & Wang, M.* (2025). GeoAggregator: An efficient transformer model for geo-spatial tabular data. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 39, No. 11, pp. 11572–11580). AAAI Press.

Deng, R., Li, Z., & Wang, M.* (2025). Improving the computational efficiency and explainability of GeoAggregator. In Proceedings of the 8th ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery (GeoAI ’25) (pp. 1–4). ACM.

GeoAggregator Paper **GA-sklearn Code**

More About Tabular Foundation Model: A Spatially Enhanced TabPFN

nature

Explore content About the journal Publish with us

nature articles article

Article | Open access | Published: 08 January 2025

Accurate predictions on small data with a tabular foundation model

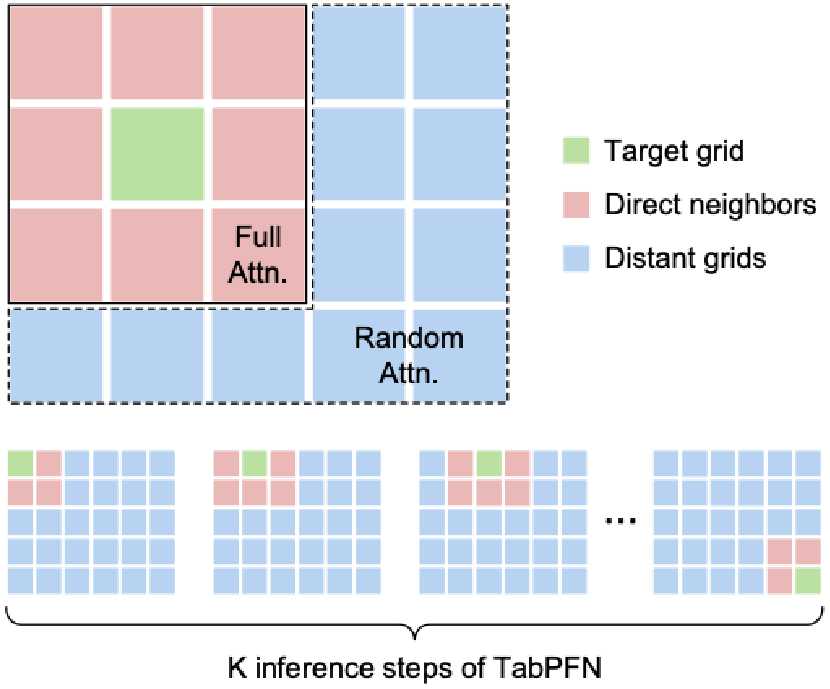
Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister & Frank Hutter

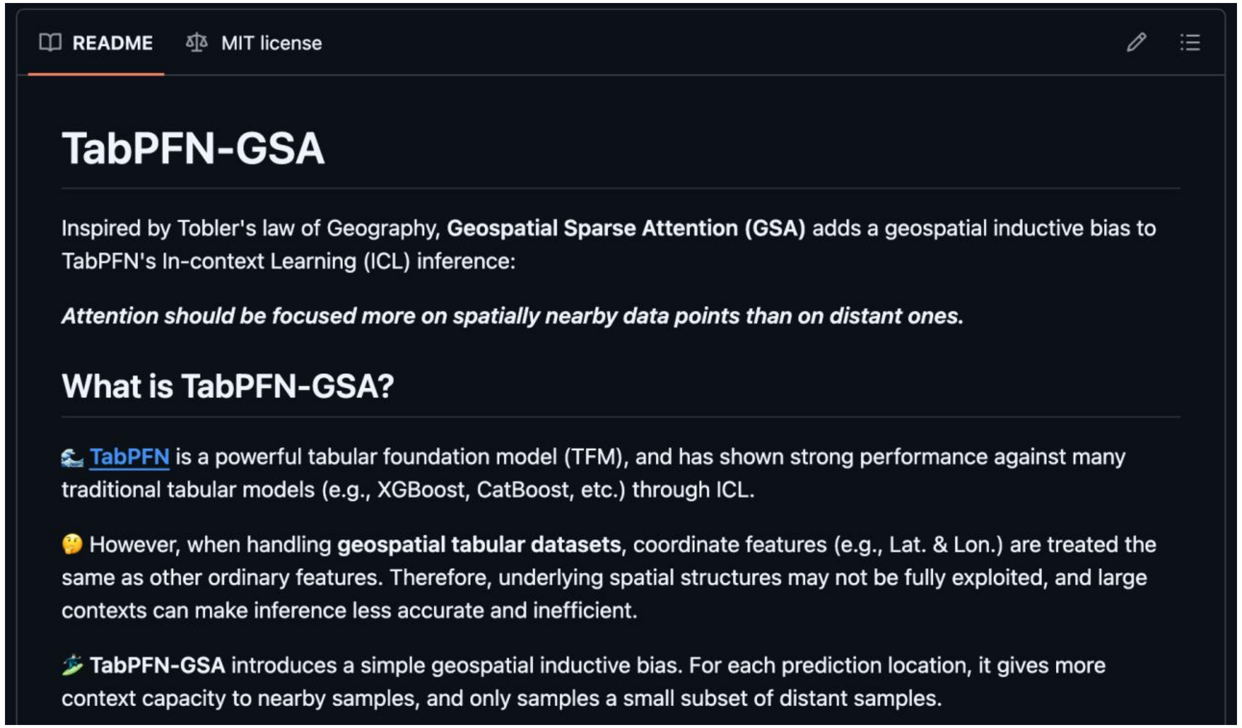
Nature **637**, 319–326 (2025) | Cite this article

513k Accesses | **753** Citations | **518** Altmetric | Metrics

More About Tabular Foundation Model: A Spatially Enhanced TabPFN

- We **prune** redundant attention calculations, providing TabPFN with localised contexts…





README MIT license

TabPFN-GSA

Inspired by Tobler's law of Geography, **Geospatial Sparse Attention (GSA)** adds a geospatial inductive bias to TabPFN's In-context Learning (ICL) inference:

Attention should be focused more on spatially nearby data points than on distant ones.

What is TabPFN-GSA?

TabPFN is a powerful tabular foundation model (TFM), and has shown strong performance against many traditional tabular models (e.g., XGBoost, CatBoost, etc.) through ICL.

However, when handling **geospatial tabular datasets**, coordinate features (e.g., Lat. & Lon.) are treated the same as other ordinary features. Therefore, underlying spatial structures may not be fully exploited, and large contexts can make inference less accurate and inefficient.

TabPFN-GSA introduces a simple geospatial inductive bias. For each prediction location, it gives more context capacity to nearby samples, and only samples a small subset of distant samples.

- … the resulting **TabPFN-GSA model** outperforms TabPFN (without PHE).

Thank you for your attention!

GeoAggregator Paper

GA-sklearn Code

TabPFN-GSA Code

Rui Deng

rui.deng@glasgow.ac.uk