GRAPHTSA: ADAPTIVE TOKEN COMMUNICATION FOR LOCAL HETEROGENEITY AND GLOBAL SYNCHRONIZATION IN MEDICAL TIME SERIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Medical time series (MedTS) exhibit a structural duality: measurements retain **local heterogeneity** from distributed sources and measurement geometry, while sharing **global synchronization** from common physiological activity. Dense pairwise attention captures heterogeneous interactions but lacks explicit global coordination; CoTAR efficiently aggregates global information but broadcasts a shared core across tokens, limiting heterogeneous communication. We propose **Graph Token Structural Attention (GraphTSA)**, an MLP-based module that preserves CoTAR's token-message construction but replaces shared broadcast with input-adaptive, directional structural mixing without a separate value projection. A low-dimensional query–key factorization controls affinity-logit capacity. Across seven public EEG, ECG, and human-activity-recognition datasets, GraphTSA improves upon TeCh with CoTAR on six. Rank and topology analyses reveal dataset-dependent rank sensitivity, richer heterogeneous communication, and a persistent dominant spectral mode. Despite replacing TeCh's linear token mixing with adaptive pairwise communication, GraphTSA incurs **less than 1% parameter and memory overhead on PTB at its best-performing rank**. Code and training scripts will be released upon acceptance.

1 INTRODUCTION

Medical time series (MedTS), such as electroencephalogram (EEG) and electrocardiogram (ECG), support applications from neurological monitoring to cardiovascular diagnosis (Badr et al., 2024; Liu et al., 2021). Their analysis must capture a structural duality (Fig. 1): distributed physiological sources, tissue properties, and sensor geometry produce heterogeneous measurements, while common underlying activity induces shared structure across signals (Stam, 2005; Graña & Morais-Quilez, 2023; Müller et al., 2026). We refer to these complementary properties as *local heterogeneity* and *global synchronization*, respectively.

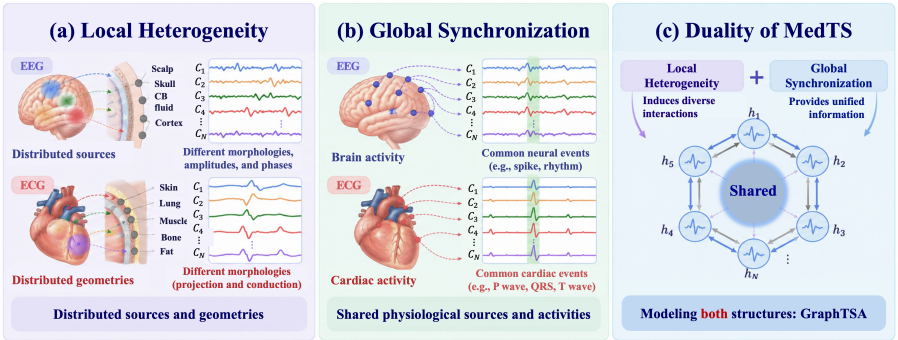

Figure 1: **Structural duality of MedTS. (a)** Distributed physiological sources and measurement geometries produce local heterogeneity. **(b)** Shared physiological activity induces global synchronization. **(c)** Effective modeling should capture both structures.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Early MedTS methods used handcrafted features (Berkaya et al., 2018; Malakouti et al., 2025), followed by CNNs and RNNs (Lawhern et al., 2018; Roy et al., 2019). Transformers provide dense pairwise communication (Vaswani et al., 2017), while graph models explicitly learn structured dependencies (Fan et al., 2025). In contrast, CoTAR replaces pairwise attention with efficient shared core aggregation–redistribution, biasing communication toward global coordination (Guoqi et al., 2026). These mechanisms represent two communication extremes: pairwise interaction flexibly captures heterogeneous relationships but lacks an explicit global coordination bias, whereas shared aggregation promotes global coordination but can underrepresent differentiated token relationships. Since MedTS exhibit both structures, neither extreme should be assumed sufficient across datasets.

We propose **Graph Token Structural Attention (GraphTSA)**, an MLP-based module that preserves CoTAR's token-message construction while replacing shared broadcast with input-adaptive, directional structural mixing. Rather than using low rank merely to approximate attention efficiently, GraphTSA uses a low-dimensional query–key factorization to control affinity-logit capacity, providing a tunable inductive bias for heterogeneous communication while retaining global access. Integrated into TeCh's temporal and channel branches, GraphTSA operates at both token granularities. We further analyze how communication capacity relates to data structure, performance, learned topology, and its evolution during training.

Our contributions are fourfold:

- **A structural view of MedTS.** We motivate MedTS modeling through the duality of local heterogeneity and global synchronization, calling for communication beyond either dense pairwise interaction or shared global aggregation alone.
- **Adaptive structural communication.** We propose GraphTSA, which combines CoTAR-style token messages with input-adaptive, directional structural mixing, using affinity-logit rank to explicitly control communication capacity.
- **Dataset-dependent communication capacity.** We investigate how affinity-logit rank relates to data structure, classification performance, and learned topology. Contrasting datasets exhibit different rank sensitivity, suggesting that useful communication capacity is dataset dependent rather than universally fixed and motivating rank as a controllable structural capacity parameter.
- **Empirical gains with structural evidence.** Across seven datasets, GraphTSA improves upon TeCh with CoTAR on six. Rank, topology, and training-dynamics analyses characterize how learned communication varies across capacity, inputs, and optimization.

2 Related Work

MedTS Modeling. MedTS exhibit temporal and cross-measurement dependencies arising from shared, spatially distributed physiological activity (da Silva & Niedermeyer, 2005; Goldberger et al., 2000; Stam, 2005; Macfarlane et al., 2005). Early approaches relied on handcrafted features, while CNNs and RNNs enabled representation learning directly from signals (Roy et al., 2019; Lawhern et al., 2018). Recent methods increasingly model cross-channel and long-range dependencies.

Token Communication for MedTS. Transformers provide dense pairwise communication through self-attention (Vaswani et al., 2017); Medformer adapts this paradigm through cross-channel patching and multi-granularity attention (Wang et al., 2024b). Graph-based methods explicitly represent relational structure (Veličković et al., 2017; Wu et al., 2020); MedGNN, for example, learns adaptive multi-resolution spatiotemporal graphs (Fan et al., 2025). At the other communication extreme, TeCh replaces pairwise attention with CoTAR, which aggregates information into a shared core and redistributes it to all tokens, providing an efficient bias toward global coordination (Guoqi et al., 2026). GraphTSA bridges these communication biases by retaining CoTAR-style token messages while replacing shared broadcast with input-adaptive, directional communication graph.

Rank Structure and Low-Rank Time-Series Modeling. Low-rank structure has been used to constrain or compress time-series representations and attention. JTFT uses low-rank attention for cross-channel dependencies (Chen et al., 2024), while SVD-based low-rank attention has been applied to ECG representation learning (Zhang et al., 2024a). ALoRa-T exploits attention-rank structure for time-series anomaly detection (Shimillas et al., 2026), and Yu et al. (2026) connect low-rank time-

series representations to Transformer compressibility. Unlike approaches that use low rank primarily to structure or approximate attention, GraphTSA uses affinity-logit rank as a controllable communication capacity within an MLP-based token-mixing mechanism, and studies its relationship with MedTS structure, performance, and learned communication topology.

3 METHOD

3.1 GRAPH TOKEN STRUCTURAL ATTENTION

Given N tokens, we model token communication as a directed, weighted graph whose edge weights determine information flow between tokens. Fig. 2 contrasts three communication mechanisms: Transformer uses dense pairwise attention, CoTAR aggregates information into a shared core and redistributes it to all tokens, whereas GraphTSA retains CoTAR-style token messages but learns an input-adaptive communication graph for their aggregation.

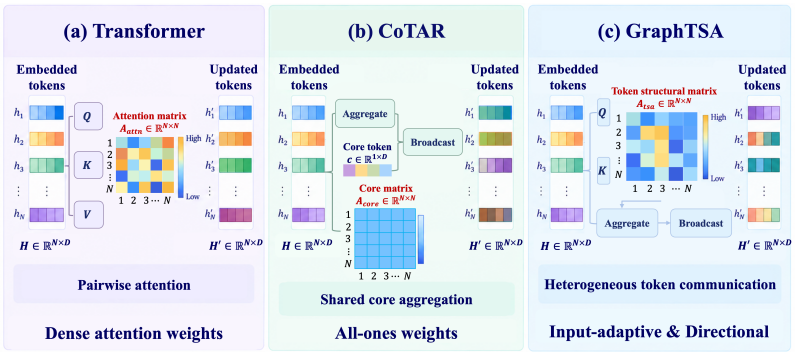

Figure 2: **Comparison of token communication mechanisms.** (a) Transformer uses dense pairwise attention. (b) CoTAR aggregates information into a shared core and redistributes it to all tokens. (c) GraphTSA retains CoTAR-style token messages while replacing shared broadcast with an input-adaptive, directional communication graph.

Given $Z \in \mathbb{R}^{N\times D}$, GraphTSA first constructs CoTAR-style transformed and re-weighted token messages,

$$R = \sigma(ZW_1)W_2, \qquad O = R \odot \mathrm{Softmax}_N(R), \tag{1}$$

where $W_1 \in \mathbb{R}^{D\times D}$, $W_2 \in \mathbb{R}^{D\times d_c}$, $R, O \in \mathbb{R}^{N\times d_c}$, $\sigma(\cdot)$ denotes GELU, and $\mathrm{Softmax}_N$ is applied across tokens.

Rather than aggregating O into a shared core, GraphTSA constructs low-dimensional query–key projections

$$Q = ZW_Q, \qquad K = ZW_K, \qquad W_Q, W_K \in \mathbb{R}^{D\times k}, \quad Q, K \in \mathbb{R}^{N\times k}, \tag{2}$$

and defines the communication graph as

$$S = \frac{QK^\top}{\sqrt{D}}, \qquad A = \mathrm{Softmax}_{\mathrm{row}}(S), \qquad S, A \in \mathbb{R}^{N\times N}. \tag{3}$$

Here, A is input-adaptive and generally asymmetric, enabling directional token communication. Since

$$\mathrm{rank}(S) = \mathrm{rank}(QK^\top) \le k, \tag{4}$$

k controls the capacity of the pre-softmax affinity logits; the nonlinear softmax does not impose the same rank bound on A.

GraphTSA then aggregates the token messages through the learned graph,

$$M = AO, \qquad M \in \mathbb{R}^{N\times d_c}, \tag{5}$$

so each row of A defines a token-specific aggregation pattern. The aggregated message is concatenated with the original token and transformed:

$$G(Z) = \sigma([Z; M]W_3)\, W_4, \tag{6}$$

where $W_3 \in \mathbb{R}^{(D+d_c)\times D}$ and $W_4 \in \mathbb{R}^{D\times D}$. Thus, Q and K construct the communication graph, while the graph mixes CoTAR-style token messages without a separate value projection. The pairwise operations require $\mathcal{O}(N^2(k+d_c))$ computation; full complexity and empirical efficiency analyses are provided in Appendix H.

3.2 GraphTSA Encoder

GraphTSA replaces CoTAR as the token-mixing sublayer of the TeCh encoder and includes an internal residual,

$$\widetilde{Z} = G(Z) + Z, \tag{7}$$

followed by TeCh's existing encoder residual,

$$U' = \widetilde{Z} + Z = G(Z) + 2Z, \quad U = \mathrm{LayerNorm}(U'), \tag{8}$$

and feed-forward sublayer,

$$Z' = \mathrm{LayerNorm}(\mathrm{FFN}(U) + U). \tag{9}$$

The two residual paths therefore yield an effective identity-path weight $\alpha = 2$. Since residual scaling can affect Transformer optimization and signal propagation (Wang et al., 2024a), we ablate this design together with affinity scaling in Section 4.3.

3.3 Dual-Branch Architecture

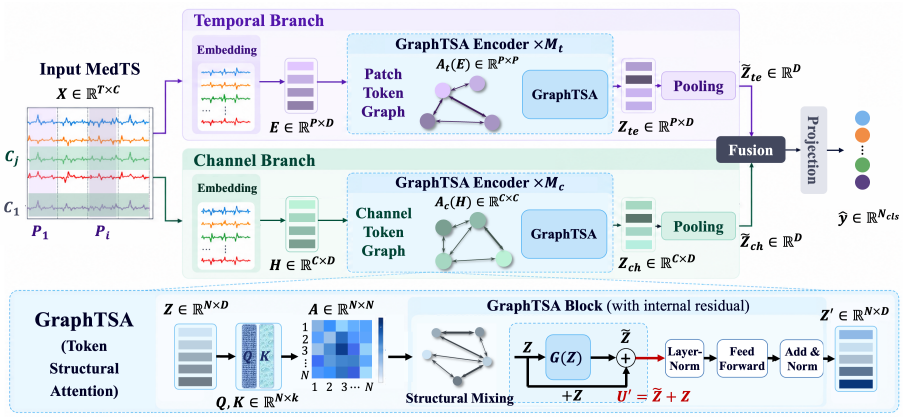

Figure 3: **GraphTSA dual-branch architecture.** Following TeCh, the input is represented as temporal-patch and channel tokens and processed by independent encoder branches. GraphTSA replaces CoTAR within each encoder, providing input-adaptive token communication at both granularities. The resulting branch representations are pooled, fused, and projected for classification.

As illustrated in Fig. 3, we integrate GraphTSA into TeCh's dual-branch architecture while retaining its tokenization scheme (Guoqi et al., 2026). Given $X \in \mathbb{R}^{T\times C}$, the temporal branch follows TeCh's cross-channel patching: after right replication padding by L samples, length-L windows spanning all C channels are projected to D-dimensional tokens, yielding $E \in \mathbb{R}^{P\times D}$ with $P = \lfloor T/L \rfloor + 1$.

Following the inverted tokenization strategy (Liu et al., 2024b), the channel branch treats the complete time series of each channel as one token and projects it to D dimensions, yielding $H \in \mathbb{R}^{C\times D}$.

GraphTSA operates without changing its formulation, with $N = P$ for temporal-patch tokens and $N = C$ for channel tokens. Independent stacks of M_t and M_c GraphTSA encoders produce $Z_{te} \in \mathbb{R}^{P\times D}$ and $Z_{ch} \in \mathbb{R}^{C\times D}$, which are mean-pooled to $\widetilde{Z}_{te}, \widetilde{Z}_{ch} \in \mathbb{R}^{D}$. Following TeCh, the representations are fused by addition and projected to N_{cls} class logits,

$$\hat{y} = \left(\widetilde{Z}_{te} + \widetilde{Z}_{ch}\right) W_y + b_y, \qquad W_y \in \mathbb{R}^{D\times N_{\mathrm{cls}}}. \tag{10}$$

Thus, the two branches model communication among temporal-patch and channel tokens, respectively, using the same GraphTSA mechanism at complementary token granularities.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We evaluate GraphTSA on five MedTS datasets (three EEG and two ECG) and two HAR datasets following the dataset-specific evaluation protocols of TeCh and Medformer (Guoqi et al., 2026; Wang et al., 2024b). Dataset statistics are given in Table 1; detailed preprocessing, augmentation, baseline configurations, and implementation details are provided in Appendix B–D.

Table 1: **Statistics of the processed datasets**, including modality, number of subjects, samples, classes, channels, time points, and sampling rate.

Datasets	Modality	No. of Subjects	No. of Samples	No. of Classes	No. of Channels	No. of Time Points	Sampling Rate
ADFTD	EEG	88	69,752	3	19	256	256 Hz
APAVA	EEG	23	5,967	2	16	256	256 Hz
TDBrain	EEG	50	6,240	2	33	256	256 Hz
PTB	ECG	198	64,356	2	15	300	250 Hz
PTB-XL	ECG	17,596	191,400	5	12	250	250 Hz
UCI-HAR	HAR	30	10,299	6	9	128	50 Hz
FLAAP	HAR	8	13,123	10	6	100	100 Hz

Datasets and baselines. We evaluate GraphTSA on seven public datasets spanning EEG (**ADFTD**, **APAVA**, **TDBrain**), ECG (**PTB**, **PTB-XL**), and HAR (**UCI-HAR**, **FLAAP**). We compare against 15 baselines: 10 Transformer-based models, four graph-based models, and **TeCh with CoTAR**, our direct controlled baseline (Guoqi et al., 2026). For controlled comparison, GraphTSA follows the dataset-specific preprocessing and evaluation protocols used by TeCh. Dataset statistics, baseline implementations, and evaluation protocols are detailed in Appendix C.

Implementation. We evaluate accuracy, precision, recall, F1, AUROC, and AUPRC over five seeds (42–46), reporting mean and standard deviation and selecting checkpoints by validation F1. Transformer results follow the TeCh benchmark (Guoqi et al., 2026), while graph-baseline results on the five MedTS datasets are taken from MedGNN (Fan et al., 2025). We reproduce MedGNN on the two HAR datasets using its official implementation. AUPRC is unavailable for the reported graph-baseline results. Further implementation details are provided in Appendix D.

4.2 MAIN RESULTS

Overall performance. Tables 2 and 3 report results on five MedTS and two HAR datasets under their respective evaluation protocols. Compared with TeCh with CoTAR, GraphTSA improves all six metrics on ADFTD, TDBrain, PTB-XL, and UCI-HAR, five of six on APAVA, and four of six on PTB. These results show that input-adaptive token communication generally improves upon shared core-based communication across the evaluated tasks.

Substantial gains on PTB. The most pronounced improvement is observed on PTB, where GraphTSA improves accuracy by 3.15 percentage points and F1 by 4.56 points over TeCh, resulting in the largest overall gain among the evaluated MedTS datasets. While AUROC and AUPRC decrease by less than one point, the gains are concentrated in accuracy, precision, recall, and F1 rather than ranking metrics. We further investigate the particularly strong performance on PTB through the rank analysis in Section 4.3.

Understanding the FLAAP exception. FLAAP is the only dataset where GraphTSA consistently underperforms TeCh. As analyzed in Section 4.3, FLAAP also differs from PTB in two relevant observations: its channel representation shows little redundancy under our PCA criterion ($n_{95}/C = 1.0$), and its performance is relatively insensitive to rank k. FLAAP additionally contains only eight subjects. Together, these observations suggest that the benefit of adaptive structural communication is dataset dependent, motivating the rank and structural analyses that follow.

Table 2: **Results on five MedTS datasets using subject-independent splits.** Best results are **bolded** and second-best results are underlined.

Datasets	Models	Accuracy	Precision	Recall	F1	AUROC	AUPRC
ADFTD (3 Classes)	**Autoformer**	45.25±1.48	43.67±1.94	42.96±2.03	42.59±1.85	61.02±1.82	43.10±2.30
	FEDformer	46.30±0.59	46.05±0.76	44.22±1.38	43.91±1.37	62.62±1.75	46.11±1.44
	Informer	48.45±1.96	46.54±1.68	46.06±1.84	45.74±1.38	65.87±1.27	47.60±1.30
	iTransformer	52.60±1.59	46.79±1.27	47.28±1.29	46.79±1.13	67.26±1.16	49.53±1.21
	MTST	45.60±2.03	44.70±1.33	45.05±1.30	44.31±1.74	62.50±0.81	45.16±0.85
	Nonformer	49.95±1.05	47.71±0.97	47.46±1.50	46.96±1.35	66.23±1.37	47.33±1.78
	PatchTST	44.37±0.95	42.40±1.13	42.06±1.48	41.97±1.37	60.08±1.50	42.49±1.79
	Reformer	50.78±1.17	49.64±1.49	49.89±1.67	47.94±0.69	69.17±1.58	**51.73**±1.94
	Transformer	50.47±2.14	49.13±1.83	48.01±1.53	48.09±1.59	67.93±1.59	48.93±2.02
	Medformer	53.27±1.54	51.02±1.57	50.71±1.55	50.65±1.51	70.93±1.19	51.21±1.32
	FourierGNN	52.00±1.33	47.20±0.87	48.21±0.95	47.30±0.75	66.18±0.94	-
	SimTSC	49.26±2.58	47.04±0.05	42.75±0.30	41.79±1.25	65.10±0.58	-
	TodyNet	42.34±2.48	42.99±2.51	42.48±2.57	41.05±2.03	60.69±2.63	-
	MedGNN	**56.12**±0.11	**55.07**±0.09	**55.47**±0.34	**55.00**±0.24	**74.68**±0.33	-
	TeCh (CoTAR)	54.54±0.70	53.02±0.87	49.25±1.01	48.84±1.72	68.67±1.05	50.62±1.26
	GraphTSA (Ours)	55.42±0.21	53.22±0.84	50.08±0.81	49.88±1.17	69.07±0.61	51.52±0.79
APAVA (2 Classes)	**Autoformer**	68.64±1.82	68.48±2.10	68.77±2.27	68.06±1.94	75.94±3.61	74.38±4.05
	FEDformer	74.94±2.15	74.59±1.50	73.56±3.55	73.51±3.39	83.72±1.97	82.94±2.37
	Informer	73.11±4.40	75.17±6.06	69.17±4.56	69.47±5.06	70.46±4.91	70.75±5.27
	iTransformer	74.55±1.66	74.77±2.10	71.76±1.72	72.30±1.79	85.59±1.55	84.39±1.57
	MTST	71.14±1.59	79.30±0.97	65.27±2.28	64.01±3.16	68.87±2.34	71.06±1.60
	Nonformer	71.89±3.81	71.80±4.58	69.44±3.56	69.74±3.84	70.55±2.96	70.78±4.08
	PatchTST	67.03±1.65	78.76±1.28	59.91±2.02	55.97±3.10	65.65±0.28	67.99±0.76
	Reformer	78.70±2.00	82.50±3.95	75.00±1.61	75.93±1.82	73.94±1.40	76.04±1.14
	Transformer	76.30±4.72	77.64±5.95	73.09±5.01	73.75±5.38	72.50±6.60	73.23±7.60
	Medformer	78.74±0.64	81.11±0.84	75.40±0.66	76.31±0.71	83.20±0.91	83.66±0.92
	FourierGNN	68.47±7.86	67.60±8.76	66.10±7.94	66.36±8.13	75.03±11.62	-
	SimTSC	82.35±0.77	85.49±1.68	78.69±0.70	79.90±0.77	82.37±2.79	-
	TodyNet	72.66±5.70	74.74±7.33	71.25±4.61	70.91±5.23	78.85±4.70	-
	MedGNN	82.60±0.35	87.70±0.22	78.93±0.09	80.25±0.16	85.93±0.26	-
	TeCh (CoTAR)	86.86±1.09	86.85±1.29	**86.10**±1.00	86.30±1.06	94.02±0.52	93.79±0.56
	GraphTSA (Ours)	**87.38**±0.57	**89.06**±0.55	85.32±0.69	**86.40**±0.65	**95.69**±0.33	**95.69**±0.32
TDBrain (2 Classes)	**Autoformer**	87.33±3.79	88.06±3.56	87.33±3.79	87.26±3.84	93.81±2.26	93.32±2.42
	FEDformer	78.13±1.98	78.52±1.91	78.13±1.98	78.04±2.01	86.56±1.86	86.48±1.99
	Informer	89.02±2.50	89.43±2.14	89.02±2.50	88.98±2.54	96.64±0.68	96.75±0.63
	iTransformer	74.67±1.06	74.71±1.06	74.67±1.06	74.65±1.06	83.37±1.14	83.73±1.27
	MTST	76.96±3.76	77.24±3.59	76.96±3.76	76.88±3.83	85.27±4.46	82.81±5.64
	Nonformer	87.88±2.48	88.86±1.84	87.88±2.48	87.78±2.56	97.05±0.68	96.99±0.68
	PatchTST	79.25±3.79	79.60±4.09	79.25±3.79	79.20±3.77	87.95±4.96	86.36±6.67
	Reformer	87.92±2.01	88.64±1.40	87.92±2.01	87.85±2.08	96.30±0.54	96.40±0.45
	Transformer	87.17±1.67	87.99±1.68	87.17±1.67	87.10±1.68	96.28±0.92	96.34±0.81
	Medformer	89.62±0.81	89.68±0.78	89.62±0.81	89.62±0.81	96.41±0.35	96.51±0.33
	FourierGNN	76.04±3.14	76.22±3.22	76.04±3.14	76.00±3.14	84.11±2.83	-
	SimTSC	89.06±1.67	89.41±0.86	89.06±0.50	89.04±1.01	97.28±1.69	-
	TodyNet	89.58±2.15	90.21±1.39	89.58±3.87	89.54±4.69	97.41±1.00	-
	MedGNN	91.04±0.09	91.15±0.12	91.04±0.20	91.04±0.08	96.74±0.04	-
	TeCh (CoTAR)	93.21±0.61	93.39±0.58	93.21±0.61	93.20±0.61	98.68±0.19	98.72±0.17
	GraphTSA (Ours)	**93.71**±0.37	**93.77**±0.35	**93.71**±0.37	**93.71**±0.37	**98.71**±0.15	**98.75**±0.14
PTB (2 Classes)	**Autoformer**	73.35±2.10	72.11±2.89	63.24±3.17	63.69±3.84	78.54±3.48	74.25±3.53
	FEDformer	76.05±2.54	77.58±3.61	66.10±3.55	67.14±4.37	85.93±4.31	82.59±5.42
	Informer	78.69±1.68	82.87±1.02	69.19±2.90	70.84±3.47	92.09±0.53	90.02±0.60
	iTransformer	83.89±0.71	88.25±1.18	76.39±1.01	79.06±1.06	91.18±1.16	90.93±0.98
	MTST	76.59±1.90	79.88±1.90	66.31±2.95	67.38±3.71	86.86±2.75	83.75±2.84
	Nonformer	78.66±0.49	82.77±0.86	69.12±0.87	70.90±1.00	89.37±2.51	86.67±2.38
	PatchTST	74.74±1.62	76.94±1.51	63.89±2.71	64.36±3.38	88.79±0.91	83.39±0.96
	Reformer	77.96±2.13	81.72±1.61	68.20±3.35	69.65±3.88	91.13±0.74	88.42±1.30
	Transformer	77.37±1.02	81.84±0.66	67.14±1.80	68.47±2.19	90.08±1.76	87.22±1.68
	Medformer	83.50±2.01	85.19±0.94	77.11±3.39	79.18±3.31	92.81±1.48	90.32±1.54
	FourierGNN	79.70±3.30	81.46±4.58	71.53±4.55	73.33±5.19	84.93±3.44	-
	SimTSC	76.99±0.98	80.61±1.70	66.92±1.82	68.19±2.21	88.60±0.66	-
	TodyNet	77.41±3.65	82.50±2.10	67.12±6.09	67.81±7.62	91.72±1.55	-
	MedGNN	84.53±0.28	87.35±0.45	77.90±0.66	80.40±0.62	93.31±0.46	-
	TeCh (CoTAR)	85.96±2.52	89.92±0.74	79.43±4.13	81.97±4.07	**94.57**±0.70	**94.36**±0.66
	GraphTSA (Ours)	**89.11**±1.16	**92.16**±0.74	**83.96**±1.82	**86.53**±1.64	93.90±1.19	94.15±0.93
PTB-XL (5 Classes)	**Autoformer**	61.68±2.72	51.60±1.64	49.10±1.52	48.85±2.27	82.04±1.44	51.93±1.71
	FEDformer	57.20±9.47	52.38±6.09	49.04±7.26	47.89±8.44	82.13±4.17	52.31±7.03
	Informer	71.43±0.32	62.64±0.60	59.12±0.47	60.44±0.43	88.65±0.09	64.76±0.17
	iTransformer	69.28±0.22	59.59±0.45	54.62±0.18	56.20±0.19	86.71±0.10	60.27±0.21
	MTST	72.14±0.27	63.84±0.72	60.01±0.81	61.43±0.38	88.97±0.33	65.83±0.51
	Nonformer	70.56±0.55	61.57±0.66	57.75±0.72	59.10±0.66	88.32±0.36	63.40±0.79
	PatchTST	73.23±0.25	65.70±0.64	60.82±0.76	62.61±0.34	89.74±0.19	67.32±0.22
	Reformer	71.72±0.43	63.12±1.02	59.20±0.75	60.69±0.18	88.80±0.24	64.72±0.47
	Transformer	70.59±0.44	61.57±0.65	57.62±0.35	59.05±0.25	88.21±0.16	63.36±0.29
	Medformer	72.87±0.23	64.14±0.42	60.60±0.46	62.02±0.37	89.66±0.13	66.39±0.22
	FourierGNN	63.49±0.88	52.05±1.21	47.85±1.13	48.92±1.19	82.45±0.86	-
	SimTSC	71.87±0.91	63.61±1.73	58.86±0.99	60.19±0.60	88.19±0.78	-
	TodyNet	72.21±0.65	64.30±0.68	57.12±1.66	58.94±1.75	89.14±0.44	-
	MedGNN	73.87±0.18	66.26±0.29	61.13±0.23	62.54±0.20	90.21±0.15	-
	TeCh (CoTAR)	73.53±0.07	65.92±0.52	60.61±0.59	62.44±0.27	90.03±0.12	67.19±0.25
	GraphTSA (Ours)	**73.91**±0.20	**66.61**±0.70	**61.19**±0.37	**62.87**±0.15	**90.22**±0.11	**67.62**±0.42

Table 3: **Results on two HAR datasets for evaluation beyond MedTS.** Best results are **bolded** and second-best results are underlined.

Datasets	Models	Accuracy	Precision	Recall	F1	AUROC	AUPRC
FLAAP (10 Classes)	**Autoformer**	38.93±1.01	38.22±1.31	37.40±1.17	33.51±1.14	74.12±0.35	35.77±0.91
	FEDformer	59.51±9.03	59.84±8.10	58.57±8.97	57.73±9.99	89.75±5.37	60.88±9.63
	Informer	72.87±0.89	73.20±0.97	72.76±0.92	72.59±0.96	95.91±0.24	77.57±1.21
	iTransformer	75.15±0.48	75.09±0.53	75.14±0.47	74.91±0.51	96.64±0.14	80.81±0.60
	MTST	70.57±0.54	71.09±0.73	70.97±0.73	70.61±0.57	94.56±0.18	73.28±0.99
	Nonformer	74.85±1.76	75.19±1.37	74.51±1.85	74.39±1.80	96.43±0.27	79.29±1.90
	PatchTST	56.34±0.31	56.36±0.63	55.29±0.32	55.58±0.45	89.24±0.11	58.92±0.36
	Reformer	71.13±1.64	71.20±1.81	70.57±1.66	70.54±1.79	95.16±0.42	73.80±2.09
	Transformer	76.36±1.21	76.53±1.25	76.23±0.98	76.05±1.16	96.65±0.11	80.70±0.63
	Medformer	76.44±0.64	76.61±1.13	76.63±1.36	76.25±0.65	95.44±0.26	81.12±1.60
	MedGNN	68.54±1.32	68.91±1.21	68.23±1.39	67.89±1.11	94.77±0.18	71.82±0.45
	TeCh (CoTAR)	**80.60**±0.30	**80.29**±0.24	**80.36**±0.32	**80.23**±0.24	**97.67**±0.10	**86.18**±0.31
	GraphTSA (Ours)	79.37±0.57	79.12±0.64	78.92±0.51	78.87±0.57	97.61±0.08	85.33±0.55
UCI-HAR (6 Classes)	**Autoformer**	41.86±2.46	49.62±11.48	44.30±2.55	32.69±2.60	83.72±2.53	58.56±4.67
	FEDformer	76.89±9.59	75.66±9.46	77.56±9.79	75.03±9.77	95.16±4.66	83.28±8.14
	Informer	88.33±1.26	88.28±1.20	88.47±1.20	88.20±1.29	98.36±0.14	94.20±0.33
	iTransformer	92.41±0.63	92.24±0.63	92.33±0.67	92.39±0.64	99.07±0.07	96.01±0.39
	MTST	90.99±0.84	90.96±0.79	90.92±0.85	90.83±0.88	98.21±0.11	96.14±0.59
	Nonformer	91.04±0.58	90.98±0.60	91.14±0.56	91.01±0.60	99.02±0.09	96.07±0.37
	PatchTST	87.67±0.39	88.37±0.43	87.97±0.37	88.02±0.38	98.50±0.09	93.86±0.40
	Reformer	88.70±1.14	88.82±1.03	88.82±1.13	88.59±1.19	98.68±0.26	94.60±1.07
	Transformer	89.36±1.74	89.33±1.70	89.49±1.69	89.33±1.75	98.87±0.23	95.58±0.68
	Medformer	89.62±0.81	89.70±0.18	89.80±0.14	89.62±0.81	98.11±0.06	94.80±0.72
	MedGNN	89.63±0.77	89.80±0.59	89.66±0.82	89.65±0.77	98.94±0.07	95.07±0.74
	TeCh (CoTAR)	94.15±0.96	94.27±0.96	94.30±0.97	94.26±0.98	99.32±0.05	96.74±0.18
	GraphTSA (Ours)	**94.59**±0.59	**94.69**±0.64	**94.69**±0.55	**94.62**±0.60	**99.48**±0.08	**97.60**±0.43

4.3 Ablation and Rank Analysis

4.3.1 Residual Strength and Affinity Scaling

We examine two design choices motivated by prior studies of residual propagation and attention concentration. Residual-path design can affect Transformer optimization and signal propagation (Wang et al., 2024a), while attention-logit scaling controls softmax concentration (Vaswani et al., 2017; Zhai et al., 2023). We compare effective residual strengths $\alpha \in \{1, 2\}$ and affinity-logit scaling by $1/\sqrt{k}$ or $1/\sqrt{D}$. Here, $\alpha = 2$ retains GraphTSA's internal residual in addition to the encoder residual, while $1/\sqrt{D}$ yields smaller-magnitude logits than $1/\sqrt{k}$ when $D > k$, generally producing softer affinities. As shown in Table 4, the $(\alpha{=}2, 1/\sqrt{D})$ configuration provides the strongest overall performance across the five MedTS datasets. We use this configuration throughout.

Table 4: Ablation of effective residual strength α and affinity-logit scaling on the MedTS datasets.

	ADFTD		APAVA		TDBrain		PTB		PTB-XL	
	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1
$\alpha = 1, 1/\sqrt{k}$	55.24±0.42	49.57±0.96	86.97±1.10	86.04±1.27	93.15±0.85	93.14±0.86	86.12±0.87	79.77±1.12	73.44±0.28	62.71±0.42
$\alpha = 2, 1/\sqrt{k}$	55.34±0.33	49.83±1.23	86.16±1.46	85.05±1.73	92.98±0.79	92.97±0.80	86.64±2.32	83.04±3.61	73.65±0.23	62.47±0.59
$\alpha = 1, 1/\sqrt{D}$	54.20±0.64	49.83±0.85	86.14±1.00	84.96±1.22	92.94±1.06	92.93±1.07	86.53±2.05	82.95±3.04	73.74±0.17	62.56±0.44
$\alpha = \mathbf{2, 1/\sqrt{D}}$	**55.42**±0.21	**49.88**±1.17	**87.38**±0.57	**86.40**±0.65	**93.71**±0.37	**93.71**±0.37	**89.11**±1.16	**86.53**±1.64	**73.91**±0.20	**62.87**±0.15

4.3.2 Rank Sensitivity and Data Structure

We further examine how the affinity-logit rank k relates to data structure by contrasting PTB and FLAAP, where GraphTSA shows its largest improvement and only consistent degradation over TeCh, respectively. Fig. 4 compares their PCA spectra with GraphTSA's sensitivity to k.

PTB requires 9 of 15 components to explain 95% of variance, whereas FLAAP requires all 6, indicating greater linear compressibility in PTB under this spectral criterion (Fig. 4(a,c)). Correspondingly, PTB performance shows clear rank sensitivity, peaking at $k = 8$ (F1 range: 3.10 points), while FLAAP is comparatively insensitive to rank (F1 range: 0.50 points; Fig. 4(b,d)). Together, these results suggest that useful affinity-logit rank is dataset dependent and may relate to linear cross-channel redundancy.

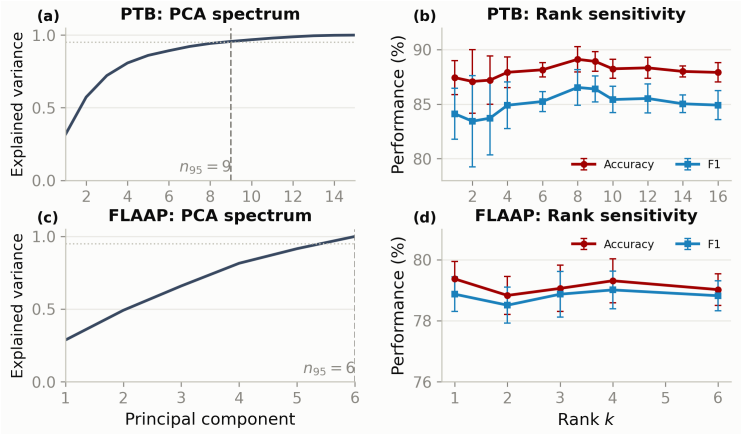

Figure 4: **Rank sensitivity and data structure on PTB and FLAAP.** (a,c) Cumulative PCA explained variance; n_{95} is the number of components required to explain 95% of the variance. (b,d) Accuracy and F1 across affinity-logit ranks k; error bars denote standard deviation over five seeds.

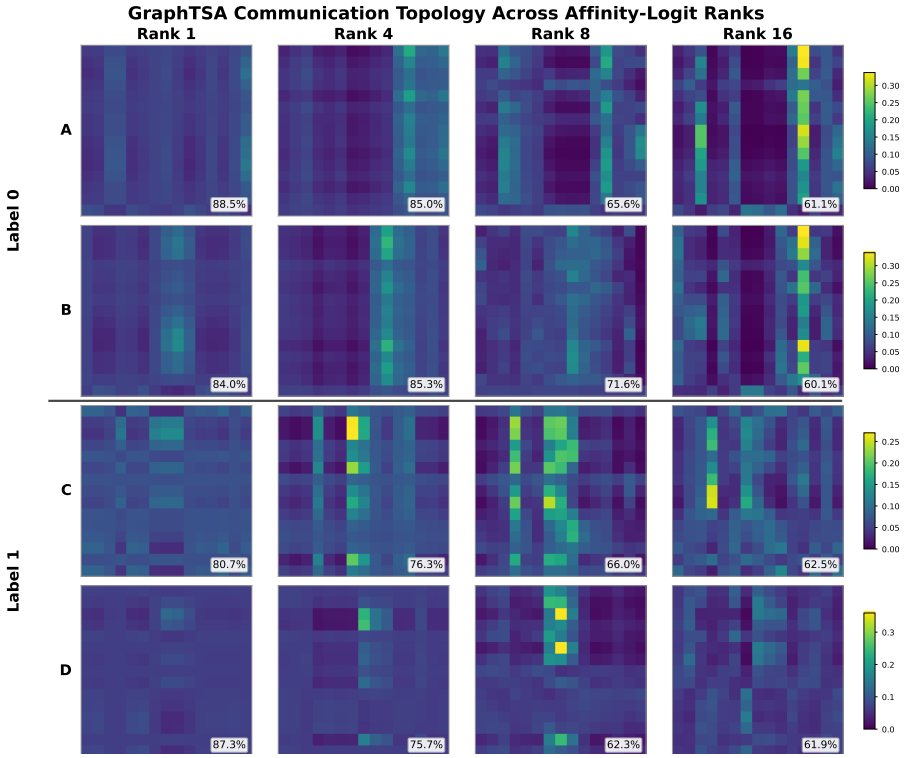

Figure 5: **Learned GraphTSA communication topology across affinity-logit ranks.** Samples A–B and C–D are representative examples from the two classes, respectively. At low rank, communication is concentrated in a dominant spectral mode, reflected by high leading singular-value mass. Increasing k yields richer heterogeneous structure while the dominant spectral mode persists. Within each sample, heatmaps share a color scale across ranks. Percentages indicate $\sigma_1 / \sum_i \sigma_i$.

4.4 Structural Analysis of Learned Token Communication

To characterize how affinity-logit rank k shapes the learned communication, Fig. 5 visualizes post-softmax communication matrices A across $k \in \{1, 4, 8, 16\}$ for representative samples from each class. Since A is directed and generally asymmetric, we analyze its singular-value spectrum. The percentage shown in each panel denotes the normalized leading singular-value mass, $\sigma_1 / \sum_i \sigma_i$.

Importantly, k constrains the pre-softmax affinity logits S, whereas the spectrum analyzed here characterizes the resulting post-softmax matrix A.

From concentrated to heterogeneous communication. As k increases, the leading singular-value mass generally decreases and the communication matrices develop richer secondary structure, while a dominant spectral mode persists through $k = 16$ (Fig. 5). PTB performance peaks at $k = 8$, suggesting that intermediate affinity-logit capacity can retain a dominant communication component while supporting richer heterogeneous interactions (Fig. 4).

Patient-specific channel communication. For PTB, the 15×15 communication matrix directly represents interactions among the 15 recorded ECG channels. At a fixed rank, the learned matrices differ across patients, showing that GraphTSA constructs patient-specific rather than fixed channel communication. This provides a structural form of interpretability: beyond identifying informative individual channels, GraphTSA exposes which channel-to-channel relationships are used in forming each patient's representation. The training-dynamics analysis in Appendix G further provides exploratory evidence that the spectral organization of these communication patterns can evolve differently across label groups. Additional examples are provided in Appendix F.

4.5 Model Efficiency Analysis

Unlike CoTAR's linear token mixing, GraphTSA materializes an $N \times N$ communication matrix. Forming $S = QK^\top/\sqrt{D}$ costs $\mathcal{O}(N^2k)$, while applying A to token messages costs $\mathcal{O}(N^2 d_c)$; graph-projection parameters grow only linearly with k. We evaluate the practical overhead.

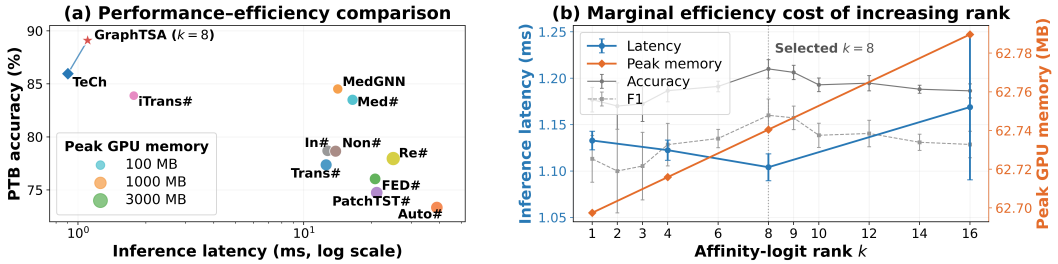

Figure 6: **Performance–efficiency analysis on PTB. (a)** Accuracy versus inference latency (log scale), with bubble area indicating peak GPU memory. **(b)** Latency and memory across GraphTSA ranks; rescaled accuracy and F1 curves show performance trends and do not correspond to the efficiency axes. Measurements use batch size 128 on an NVIDIA H100 NVL.

At $k = 8$, GraphTSA reaches 89.11% PTB accuracy with 1.561M parameters, 1.104 ms latency, and 62.74 MB peak memory (Fig. 6). Relative to TeCh with CoTAR, this corresponds to +3.15 accuracy points with +0.80% parameters, +0.08% memory, and 22.4% latency overhead. Even $k = 1$ produces input-dependent heterogeneous affinities rather than CoTAR's shared broadcast, improving accuracy from 85.96% to 87.43%. Thus, while CoTAR's linear token mixing remains computationally attractive, GraphTSA trades this structure for input-adaptive heterogeneous communication at modest measured overhead at the evaluated token lengths. Details are provided in Appendix H.

5 Conclusion

MedTS exhibit both local heterogeneity and global synchronization, motivating communication beyond either pairwise interaction or shared global aggregation alone. We introduced GraphTSA, which learns input-adaptive, directional token communication with controllable affinity-logit capacity. Across seven datasets, GraphTSA improves upon TeCh with CoTAR on six. Our analyses show dataset-dependent rank sensitivity, richer heterogeneous communication around a persistent dominant spectral mode, and patient-specific channel communication on PTB. Future work should test whether recurrent learned token relationships correspond to clinically meaningful physiological structure across larger cohorts. These results support adaptive structural communication as a useful inductive bias for MedTS while providing a representation-level view of learned token relationships.

AI USE STATEMENT

In this work, we used generative AI tools to propose and refine hypotheses, design or provide feedback on research methodology or experiments, create or modify scientific figures and images, summarize or analyse existing literature, suggest structures for sections, edit and polish the draft, and interpret results. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, provide critical ingredients for proving mathematical claims, implement methods, assist with translation, or support qualitative and thematic data analysis. Generating synthetic datasets, cleaning and reformating datasets, writing of proofs, and formulating mathematical claims are not applicable to this work. All model designs, experiments, data processing, quantitative results, and final scientific claims were reviewed and verified by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, Jorge Luis Reyes-Ortiz, et al. A public domain dataset for human activity recognition using smartphones. In *Esann*, volume 3, pp. 3–4, 2013.

Yara Badr, Usman Tariq, Fares Al-Shargie, Fabio Babiloni, Fadwa Al Mughairbi, and Hasan Al-Nashash. A review on evaluating mental stress by deep learning using eeg signals. *Neural Computing and Applications*, 36(21):12629–12654, 2024.

Selcan Kaplan Berkaya, Alper Kursat Uysal, Efnan Sora Gunal, Semih Ergin, Serkan Gunal, and M Bilginer Gulmezoglu. A survey on ecg analysis. *Biomedical Signal Processing and Control*, 43:216–235, 2018.

Yushu Chen, Shengzhuo Liu, Jinzhe Yang, Hao Jing, Wenlai Zhao, and Guangwen Yang. A joint time-frequency domain transformer for multivariate time series forecasting. *Neural Networks*, 176:106334, 2024.

FH Lopes da Silva and Ernst Niedermeyer. Electroencephalography, basic principles, clinical applications and related fields. 2005.

J Escudero, Daniel Abásolo, Roberto Hornero, Pedro Espino, and Miguel López. Analysis of electroencephalograms in alzheimer's disease patients with multiscale entropy. *Physiological measurement*, 27(11):1091–1106, 2006.

Wei Fan, Jingru Fei, Dingyu Guo, Kun Yi, Xiaozhuang Song, Haolong Xiang, Hangting Ye, and Min Li. Medgnn: towards multi-resolution spatiotemporal graph learning for medical time series classification. *arXiv preprint arXiv:2502.04515*, 2025.

Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23):e215–e220, 2000.

Manuel Graña and Igone Morais-Quilez. A review of graph neural networks for electroencephalography data analysis. *Neurocomputing*, 562:126901, 2023.

Yu Guoqi, Juncheng Wang, Chen Yang, Jing Qin, Angelica Aviles-Rivero, and Shujun Wang. Decentralized attention fails centralized signals: Rethinking transformers for medical time series. In *International Conference on Learning Representations*, volume 2026, pp. 108935–108955, 2026.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.

Prabhat Kumar and S Suresh. Flaap: An open human activity recognition (har) dataset for learning and finding the associated activity patterns. *Procedia Computer Science*, 212:64–73, 2022.

Vernon J Lawhern, Amelia J Solon, Nicholas R Waytowich, Stephen M Gordon, Chou P Hung, and Brent J Lance. Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces. *Journal of neural engineering*, 15(5):056013, 2018.

Huaiyuan Liu, Donghua Yang, Xianzhang Liu, Xinglei Chen, Zhiyu Liang, Hongzhi Wang, Yong Cui, and Jun Gu. Todynet: temporal dynamic graph neural network for multivariate time series classification. *Information Sciences*, 677:120914, 2024a.

Xinwen Liu, Huan Wang, Zongjin Li, and Lang Qin. Deep learning in ecg diagnosis: A review. *Knowledge-Based Systems*, 227:107187, 2021.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in neural information processing systems*, 35: 9881–9893, 2022.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *International conference on learning representations*, volume 2024, pp. 11116–11140, 2024b.

PW Macfarlane, B Devine, and E Clark. The university of glasgow (uni-g) ecg analysis program. In *Computers in Cardiology, 2005*, pp. 451–454. IEEE, 2005.

Seyed Matin Malakouti, Mohammad Bagher Menhaj, and Amir Abolfazl Suratgar. Enhanced epilepsy detection using discrete wavelet transform and bandpass filtering on eeg data: integration of art-based and lvq models. *Clinical eHealth*, 2025.

Andreas Miltiadous, Katerina D Tzimourta, Theodora Afrantou, Panagiotis Ioannidis, Nikolaos Grigoriadis, Dimitrios G Tsalikakis, Pantelis Angelidis, Markos G Tsipouras, Euripidis Glavas, Nikolaos Giannakeas, et al. A dataset of scalp eeg recordings of alzheimer's disease, frontotemporal dementia and healthy subjects from routine eeg. *Data*, 8(6):95, 2023.

Alissa Müller, Manuel Scheibl, Tobias Uhe, Wolf-Rüdiger Schäbitz, and Britta Wrede. Application of graph neural networks on ecg data: a systematic literature review. *IEEE Journal of Biomedical and Health Informatics*, 2026.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

PhysioToolkit PhysioBank. Physionet: components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000.

Yannick Roy, Hubert Banville, Isabela Albuquerque, Alexandre Gramfort, Tiago H Falk, and Jocelyn Faubert. Deep learning-based electroencephalography analysis: a systematic review. *Journal of neural engineering*, 16(5):051001, 2019.

Charalampos Shimillas, Kleanthis Malialis, Konstantinos Fokianos, and Marios Polycarpou. Low rank transformer for multivariate time series anomaly detection and localization. In *International Conference on Learning Representations*, volume 2026, pp. 59972–60005, 2026.

Cornelis J Stam. Nonlinear dynamical analysis of eeg and meg: review of an emerging field. *Clinical neurophysiology*, 116(10):2266–2301, 2005.

Hanneke Van Dijk, Guido Van Wingen, Damiaan Denys, Sebastian Olbrich, Rosalinde Van Ruth, and Martijn Arns. The two decades brainclinics research archive for insights in neurophysiology (tdbrain) database. *Scientific data*, 9(1):333, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Patrick Wagner, Nils Strodthoff, Ralf-Dieter Bousseljot, Dieter Kreiseler, Fatima I Lunze, Wojciech Samek, and Tobias Schaeffter. Ptb-xl, a large publicly available electrocardiography dataset. *Scientific data*, 7(1):154, 2020.

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, and Furu Wei. Deepnet: Scaling transformers to 1,000 layers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(10):6761–6774, 2024a.

Yihe Wang, Nan Huang, Taida Li, Yujun Yan, and Xiang Zhang. Medformer: A multi-granularity patching transformer for medical time-series classification. *Advances in Neural Information Processing Systems*, 37:36314–36341, 2024b.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 753–763, 2020.

Kun Yi, Qi Zhang, Wei Fan, Hui He, Liang Hu, Pengyang Wang, Ning An, Longbing Cao, and Zhendong Niu. Fouriergnn: Rethinking multivariate time series forecasting from a pure graph perspective. *Advances in neural information processing systems*, 36:69638–69660, 2023.

Annan Yu, Danielle Maddix, Boran Han, Xiyuan Zhang, Abdul Fatir Ansari, Oleksandr Shchur, Christos Faloutsos, Andrew Gordon Wilson, Michael W Mahoney, and Bernie Wang. Understanding transformers for time series: Rank structure, flow-of-ranks, and compressibility. In *International Conference on Learning Representations*, volume 2026, pp. 42449–42489, 2026.

Daochen Zha, Kwei-Herng Lai, Kaixiong Zhou, and Xia Hu. Towards similarity-aware time-series classification. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*, pp. 199–207. SIAM, 2022.

Shuangfei Zhai, Tatiana Likhomanenko, Etai Littwin, Dan Busbridge, Jason Ramapuram, Yizhe Zhang, Jiatao Gu, and Joshua M Susskind. Stabilizing transformer training by preventing attention entropy collapse. In *International conference on machine learning*, pp. 40770–40803. PMLR, 2023.

Shilin Zhang, Yixian Fang, and Yuwei Ren. Ecg autoencoder based on low-rank attention. *Scientific Reports*, 14(1):12823, 2024a.

Yitian Zhang, Liheng Ma, Soumyasundar Pal, Yingxue Zhang, and Mark Coates. Multi-resolution time-series transformer for long-term forecasting. In *International conference on artificial intelligence and statistics*, pp. 4222–4230. PMLR, 2024b.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11106–11115, 2021.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pp. 27268–27286. PMLR, 2022.

A Subject-Independent Evaluation Protocol

MedTS datasets typically contain multiple correlated observations from each subject, obtained across sessions, trials, or segmented recording windows (Wang et al., 2024b). Consequently, randomly partitioning individual samples can place observations from the same subject in both training and evaluation sets. Such overlap may introduce subject-specific information, including individual physiology, sensor placement, baseline patterns, and recording conditions, allowing a model to exploit within-subject similarity rather than learning patterns that generalize across subjects. Sample-level splitting can therefore yield an overly optimistic estimate of performance when the intended application requires predictions for previously unseen subjects.

For datasets evaluated under the **subject-independent** protocol used by Medformer and subsequently TeCh with CoTAR (Wang et al., 2024b; Guoqi et al., 2026), we partition subjects into mutually disjoint training, validation, and test sets. Let

$$\mathcal{D} = \{(X^{(i)}, y^{(i)}, s^{(i)})\}_{i=1}^{M},$$

where $X^{(i)}$ denotes a MedTS sample, $y^{(i)}$ its label, and $s^{(i)} \in \mathcal{S}$ its subject identifier. We partition the set of subjects into mutually disjoint subsets,

$$\mathcal{S}_{\text{train}} \cap \mathcal{S}_{\text{val}} = \mathcal{S}_{\text{train}} \cap \mathcal{S}_{\text{test}} = \mathcal{S}_{\text{val}} \cap \mathcal{S}_{\text{test}} = \varnothing,$$

and assign every sample from a subject exclusively to that subject's partition. Thus, no subject observed during training or model selection appears in the test set.

This protocol evaluates *cross-subject generalization*: the model must recognize task-relevant patterns in subjects that were entirely unseen during training. It also prevents correlated samples from the same subject or recording from being distributed across splits. Results obtained under subject-dependent or sample-level splits should not be directly compared with this setting, since the evaluation protocols measure different generalization conditions.

We use this setting where specified by the dataset-specific protocols in Appendix C. Datasets evaluated under other protocols are described separately there and should not be interpreted as subject-independent evaluations.

B Data Augmentation

For a controlled comparison with Medformer and CoTAR (Wang et al., 2024b; Guoqi et al., 2026), we follow the same data augmentation protocol used by these methods. During training, an augmentation is sampled uniformly from a bank of six transformations and applied to the input before embedding. No augmentation is used during evaluation. The augmentation bank and default settings are summarized in Table 5.

Table 5: **Data augmentation bank.** We follow the augmentation protocol used by Medformer and CoTAR for a fair comparison.

Augmentation	Operation	Default
Temporal flipping	Reverse the temporal order	$p = 0.5$
Channel shuffling	Randomly permute channels	$p = 0.5$
Temporal masking	Mask a subset of timestamps	$r = 0.1$
Frequency masking	Mask frequency components	$r = 0.1$
Jittering	Add random perturbations	$s = 0.1$
Dropout	Randomly mask individual values	$r = 0.1$

Temporal flipping. The temporal axis is reversed, preserving the measurements while presenting their temporal ordering in the opposite direction. The transformation is activated with probability $p = 0.5$ when selected.

Channel shuffling. The channel order is randomly permuted while retaining the complete signal from each channel. The transformation is applied with probability $p = 0.5$.

Temporal masking. A randomly selected fraction of timestamps is masked simultaneously across channels. We use a masking ratio of $r = 0.1$.

Frequency masking. The signal is transformed to the frequency domain, where a randomly selected subset of frequency components is masked before transforming the signal back to the time domain. Following the augmentation adopted by Medformer and CoTAR, we use $r = 0.1$.

Jittering. Random perturbations are added to the input signal to reduce sensitivity to small variations in the measurements. The perturbation strength is controlled by a scale parameter $s = 0.1$.

Dropout. Individual input values are randomly masked, analogous to applying dropout directly to the observations rather than hidden representations. The masking ratio is $r = 0.1$.

C Data Preprocessing and Baselines

We use the same preprocessed releases of APAVA, TDBrain, ADFTD, PTB, PTB-XL, UCI-HAR, and FLAAP as Medformer and TeCh (Wang et al., 2024b; Guoqi et al., 2026); dataset-specific processing is summarized below for completeness.

C.1 Datasets

APAVA. Drawn from the Alzheimer's Patients' Relatives Association of Valladolid EEG collection (Escudero et al., 2006), APAVA covers 23 subjects (12 Alzheimer's patients, 11 healthy controls) recorded over 16 channels, with 30.0 $\pm$ 12.5 trials per subject on average, each a 5-second, 1280-timestamp recording. After per-trial standard scaling, we cut each trial into nine half-overlapping 1-second windows (256 timestamps), producing 5,967 subject-tagged samples. Under the subject-independent split, subjects $\{15, 16, 19, 20\}$ form validation and $\{1, 2, 17, 18\}$ form test, with the remainder in training.

TDBrain. The TDBrain archive (Van Dijk et al., 2022) spans 1,274 subjects across 33 channels under eye-open and eye-closed conditions and 60 possible co-occurring diagnostic labels. We use a 50-subject slice — 25 Parkinson's disease patients and 25 matched controls — restricted to the eye-closed condition, cut into non-overlapping 1-second (256-timestamp) windows with any sub-second remainder discarded, yielding 6,240 samples. Validation subjects are $\{18, 19, 20, 21, 46, 47, 48, 49\}$, test subjects are $\{22, 23, 24, 25, 50, 51, 52, 53\}$, and the rest form training.

ADFTD. ADFTD (Miltiadous et al., 2023) is a 3-class EEG task (36 Alzheimer's, 23 frontotemporal dementia, 29 healthy-control subjects) at 19 channels, originally sampled at 500 Hz, with per-group recording lengths averaging 13.5 min (AD, range 5.1–21.3), 12 min (FTD, range 7.9–16.9), and 13.8 min (HC, range 12.5–16.5). We bandpass-filter to 0.5–45 Hz, downsample to 256 Hz, and window into non-overlapping 1-second segments, discarding any partial final window, giving 69,752 samples. We split subjects 60/20/20% into train/val/test, assigning each subject's full sample set to their split.

PTB. Sourced from the PhysioNet PTB Diagnostic ECG Database (PhysioBank, 2000) (290 subjects, 15 leads, 1,000 Hz raw), we retain the 198-subject subset used by Medformer and TeCh for myocardial infarction versus healthy-control classification. Following their preprocessing, signals are resampled to 250 Hz and standardized. R-peaks are detected independently in each lead using NeuroKit, and each heartbeat is centered at the median R-peak position across leads. Trials with median R–R intervals exceeding 300 samples are treated as outliers, while retained heartbeats are symmetrically zero-padded to a fixed length of 300 samples. This yields 64,356 samples; following the Medformer/TeCh loader, subjects are split class-wise into approximately 55/15/30% training/validation/test partitions.

PTB-XL. PTB-XL (Wagner et al., 2020) is a 12-lead ECG resource covering 18,869 subjects and five diagnostic superclasses (four cardiac conditions plus healthy control); subjects whose repeated trials disagree on diagnosis are dropped, leaving 17,596 subjects. Each 10-second trial is available at

100 Hz and 500 Hz; we use the 500 Hz version, downsample to 250 Hz, standard-scale, and window into non-overlapping 1-second (250-timestamp) segments, discarding any incomplete final window, producing 191,400 samples, again split 60/20/20% by subject.

UCI-HAR. The UCI Human Activity Recognition dataset (Anguita et al., 2013) contains recordings from 30 subjects performing six daily activities with a waist-mounted smartphone. Three-axis accelerometer and gyroscope signals were sampled at 50 Hz and represented as 128-timestamp windows over nine input channels, yielding 10,299 samples. We follow the processed release and evaluation setup provided by Medformer and TeCh.

FLAAP. FLAAP (Kumar & Suresh, 2022) contains smartphone inertial recordings from eight subjects performing ten activities. Accelerometer and gyroscope signals were collected from a waist-mounted smartphone at 100 Hz. Following the processed release used by Medformer and TeCh, we use six sensor channels and 13,123 windows of 100 timestamps.

C.2 Baselines

Transformer-based baselines. We compare against ten Transformer-based models covering a range of attention and tokenization strategies. **Transformer** (Vaswani et al., 2017) provides the standard dense self-attention baseline. **Autoformer** (Wu et al., 2021) combines series decomposition with an auto-correlation mechanism for modeling temporal dependencies. **FEDformer** (Zhou et al., 2022) incorporates frequency-domain representations to efficiently capture global temporal structure, while **Informer** (Zhou et al., 2021) employs ProbSparse attention to reduce the computational cost of dense attention. **Reformer** (Kitaev et al., 2020) uses locality-sensitive hashing and reversible layers for memory-efficient sequence modeling, whereas the **Non-stationary Transformer** (Liu et al., 2022) introduces de-stationary attention to better model non-stationary time series.

Several baselines further modify the representation of multivariate time series. **PatchTST** (Nie et al., 2022) represents each channel through temporal patches. **iTransformer** (Liu et al., 2024b) instead treats individual variates as tokens and models dependencies across them. **MTST** (Zhang et al., 2024b) constructs patch representations at multiple temporal scales. **Medformer** (Wang et al., 2024b) is specifically designed for MedTS classification and combines cross-channel, multi-granularity patching with intra- and inter-granularity attention.

Graph-based baselines. We additionally compare against four graph-based time-series models: **TodyNet** (Liu et al., 2024a), **FourierGNN** (Yi et al., 2023), **SimTSC** (Zha et al., 2022), and **MedGNN** (Fan et al., 2025). These methods provide complementary comparisons with architectures that explicitly incorporate graph-structured dependencies. In particular, MedGNN is designed for MedTS classification and models adaptive multi-resolution spatiotemporal dependencies through learned graph representations.

For the five MedTS datasets, we use the graph-baseline results reported by MedGNN when available. Since these experiments do not report AUPRC, we leave the corresponding entries unreported rather than reconstructing them under potentially different training conditions. MedGNN does not report results on the two HAR datasets; we therefore reproduce these experiments using its official implementation under our evaluation protocol.

D Implementation Details

Common experimental protocol. Since GraphTSA directly replaces CoTAR within the TeCh architecture, we use TeCh (Guoqi et al., 2026) and Medformer (Wang et al., 2024b) as the primary reference implementations and follow their dataset-specific experimental protocols for controlled comparison. We keep the preprocessing, data splits, optimization procedure, model-selection criterion, and evaluation pipeline fixed when comparing GraphTSA with TeCh, such that their difference primarily reflects the token communication mechanism.

Following TeCh, models are trained for at most 100 epochs using Adam with a learning rate of 10^{-4} and early stopping after 10 epochs without improvement in validation F1. The checkpoint with the highest validation F1 is used for test evaluation. The batch size is 32 for APAVA and TDBrain and

128 for ADFTD, PTB, and PTB-XL. We evaluate accuracy, macro-averaged precision, recall, F1, AUROC, and AUPRC over five random seeds (42–46) and report the mean and standard deviation across runs.

For Transformer-based baselines, we retain the architecture-specific configurations used in the TeCh benchmark rather than modifying individual models to match GraphTSA. Results adopted from prior work therefore use their corresponding benchmark settings. For baselines reproduced in our study, we use the official implementations and preserve the same data splits and evaluation protocol whenever applicable.

Direct comparison with CoTAR. Our primary controlled baseline is **CoTAR** (Guoqi et al., 2026), implemented within the TeCh architecture. CoTAR replaces pairwise attention with a shared core aggregation–redistribution mechanism, whereas GraphTSA replaces this module with input-adaptive, heterogeneous token communication. We reproduce TeCh under the same preprocessing, data splits, training procedure, and evaluation pipeline used for GraphTSA. Consequently, the TeCh–GraphTSA comparison isolates the effect of replacing shared core communication with the proposed GraphTSA module while keeping the surrounding architecture and experimental protocol unchanged.

E GraphTSA Algorithm

Algorithm 1 GraphTSA Dual-Branch Forward Pass

Require: Recording $X \in \mathbb{R}^{T\times C}$; temporal depth L_t; channel depth L_c; rank k
Ensure: Class logits $\hat{y} \in \mathbb{R}^{N_{\text{cls}}}$

$H \leftarrow \text{ChannelEmbedding}(X)$ $\{H \in \mathbb{R}^{C\times D}\}$
for $\ell = 1$ **to** L_c **do**
  $\widetilde{H} \leftarrow G_{\text{GraphTSA},\ell}(H; k) + H$ {internal residual}
  $H \leftarrow \text{LayerNorm}(\widetilde{H} + H)$ {outer residual}
  $H \leftarrow \text{LayerNorm}(\text{FFN}_\ell(H) + H)$
end for
$z_c \leftarrow \text{Pool}(H)$
$E \leftarrow \text{TemporalEmbedding}(X)$ $\{E \in \mathbb{R}^{P\times D}\}$
for $\ell = 1$ **to** L_t **do**
  $\widetilde{E} \leftarrow G_{\text{GraphTSA},\ell}(E; k) + E$ {internal residual}
  $E \leftarrow \text{LayerNorm}(\widetilde{E} + E)$ {outer residual}
  $E \leftarrow \text{LayerNorm}(\text{FFN}_\ell(E) + E)$
end for
$z_t \leftarrow \text{Pool}(E)$
$z \leftarrow \text{Fuse}(z_t, z_c)$
$\hat{y} \leftarrow W_p z + b_p$
return $\hat{y}$

Unless otherwise stated, we scale the affinity logits by $1/\sqrt{D}$ and retain GraphTSA's internal residual. Because the encoder applies an additional outer residual, the first sublayer becomes $G(Z)+2Z$, corresponding to the effective residual weight $\alpha = 2$ described in Section 3.2. Pooling and fusion follow the default TeCh configuration. The model is trained using cross-entropy loss.

F Additional Visualization of Learned Communication Topologies

To complement the representative examples in Fig. 5, we visualize six additional PTB samples in Fig. 7, including three examples from each class. For each sample, we show the learned GraphTSA communication matrix across affinity-logit ranks $k \in \{1, 4, 8, 16\}$. The color scale is shared across ranks within each sample, allowing direct comparison of the rank-dependent structural changes for the same input, but is independently normalized across samples to preserve the visibility of their

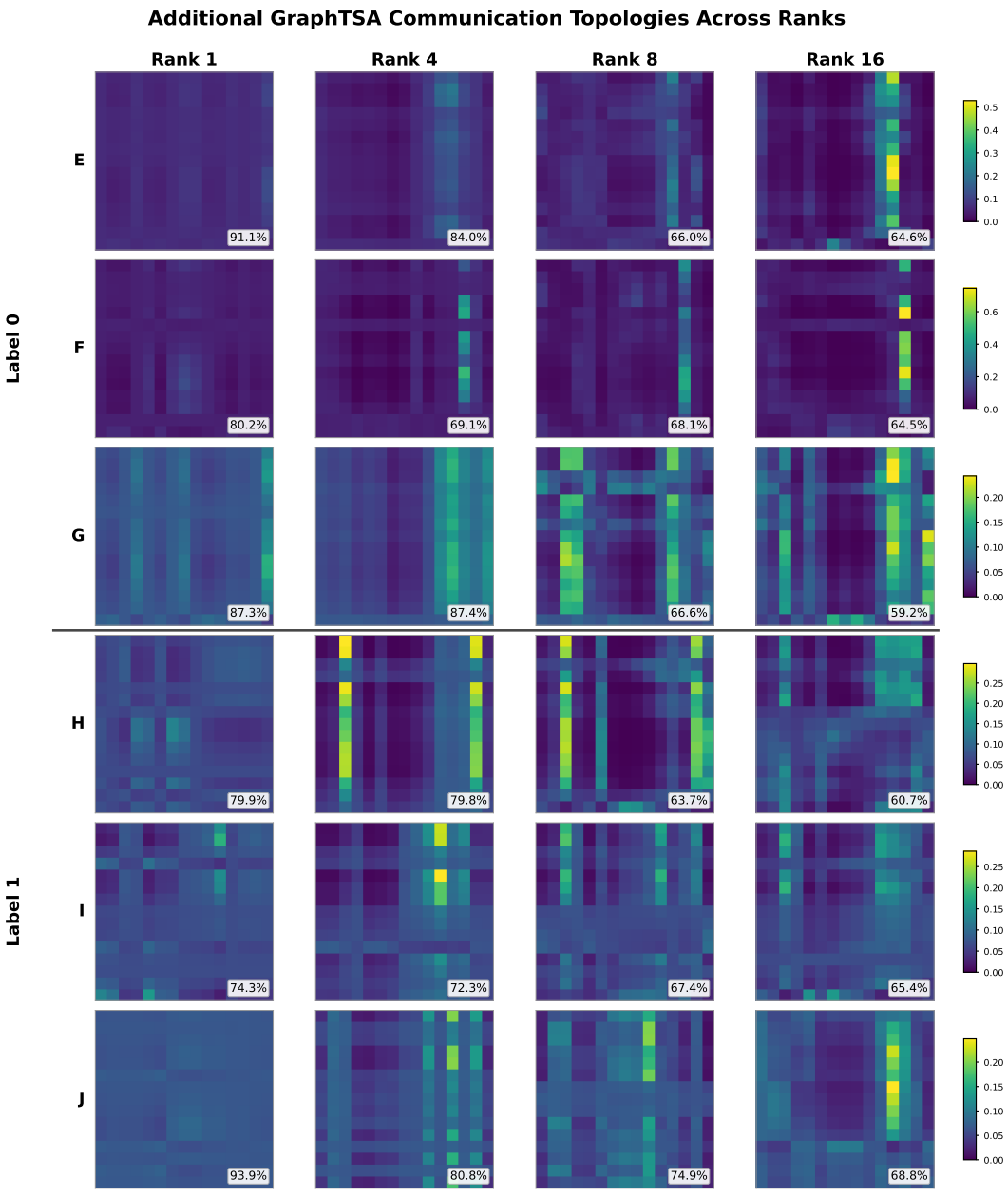

Figure 7: **Additional examples of learned GraphTSA communication topologies across affinity-logit ranks.** Samples E–G and H–J are additional examples from Label 0 and Label 1, respectively. Each row tracks the same sample across $k \in \{1, 4, 8, 16\}$. Color scales are shared across ranks within each sample but independently normalized across samples to facilitate visualization of rank-dependent structural changes. Percentages denote the normalized leading singular-value mass, $\sigma_1 / \sum_i \sigma_i$.

individual communication patterns. The percentage in each panel denotes the normalized leading singular-value mass, $\sigma_1 / \sum_i \sigma_i$.

Across these additional examples, we observe a similar rank-dependent transition to that shown in Fig. 5. At low rank, the learned communication matrices exhibit strong spectral concentration, accompanied by high leading singular-value mass. As k increases, richer heterogeneous interaction patterns emerge and the relative contribution of the leading singular component generally decreases.

Nevertheless, the leading component remains substantial even at higher ranks, indicating that richer communication patterns emerge without eliminating the dominant spectral mode.

The specific topology also varies considerably across individual samples. For example, different samples emphasize different subsets and directional patterns of token interactions even under the same rank setting. Together with the examples in Fig. 5, these observations further illustrate the input-adaptive nature of GraphTSA: the model does not learn a single fixed communication topology, but generates heterogeneous structures conditioned on each input. We emphasize that these visualizations characterize learned representation-level communication and should not be interpreted as direct anatomical or causal connectivity.

G Evolution of Learned Communication During Training

We further investigate how the communication topology learned by GraphTSA emerges during optimization. Rather than examining only the final learned graph, we track a balanced set of 14 PTB samples, comprising seven examples from each class, at epochs $\{1, 3, 6, 10, 20, 30, 40, 50\}$. All communication matrices are extracted from the rank $k = 8$ GraphTSA configuration, which achieves the best predictive performance on PTB in our rank ablation. Thus, the analysis tracks how the communication topology of the empirically optimal rank configuration develops throughout training. For each learned communication matrix A, we compute its singular-value spectrum and quantify the concentration of its dominant mode as

$$m_1 = \frac{\sigma_1}{\sum_i \sigma_i}.$$

We additionally compute the effective spectral rank by normalizing the singular values as

$$q_i = \frac{\sigma_i}{\sum_j \sigma_j}, \qquad r_{\text{eff}} = \exp\left(-\sum_i q_i \log q_i\right).$$

The effective spectral rank increases as spectral mass is distributed across a larger number of singular modes. Together, m_1 and r_{eff} characterize complementary aspects of the learned communication structure: the concentration of its dominant mode and the diversity of its spectral components, respectively.

Rapid structural formation during early training. Fig. 8 reveals a clear temporal pattern. Averaged across all 14 tracked samples, the leading singular-value mass decreases from 84.6% at epoch 1 to 70.7% at epoch 3. The decrease then continues more gradually, reaching 59.9% at epoch 50. Notably, all 14 samples exhibit a decrease from epoch 1 to epoch 3, indicating that the pronounced early transition is shared across the tracked examples rather than being driven by a small number of samples. Consistent with this redistribution of spectral mass, the mean effective spectral rank increases from 1.85 at epoch 1 to 3.45 at epoch 50.

These observations suggest a characteristic learning process: GraphTSA undergoes substantial structural reorganization during the earliest stage of optimization, followed by progressively slower refinement. The learned communication is therefore not merely a static consequence of the architecture; its structure develops systematically during training.

From spectral concentration to richer communication. The spectral evolution provides a quantitative view of how the learned communication changes. At the beginning of training, a large fraction of the spectrum is concentrated in the first singular component, corresponding to a strongly dominant communication mode. As optimization proceeds, the leading component accounts for a progressively smaller fraction of the spectrum while the effective spectral rank increases. Communication is therefore redistributed across a richer collection of structural modes.

Importantly, this process does not eliminate the dominant component. Even at epoch 50, the leading singular mode remains substantial. The learned communication thus evolves from a highly concentrated structure toward one in which a persistent dominant spectral mode coexists with increasingly heterogeneous secondary interactions. This behavior is consistent with GraphTSA's motivating hypothesis that MedTS representations benefit from retaining a dominant global communication component while allowing differentiated token-to-token interactions to emerge.

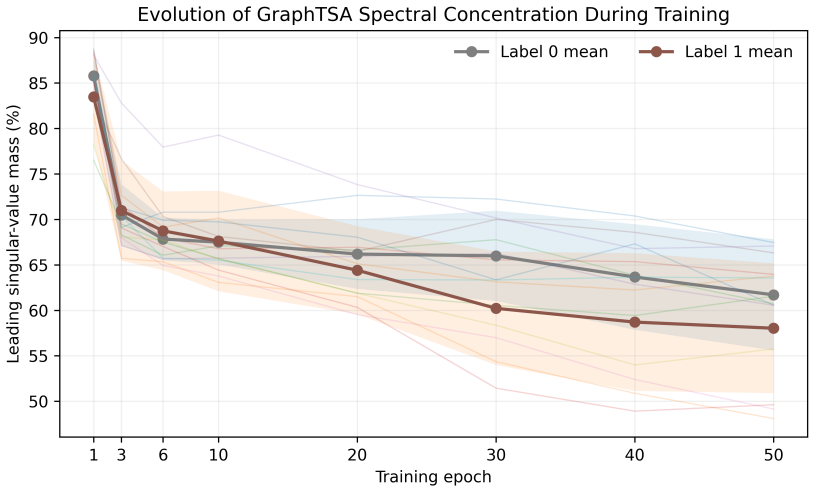

Figure 8: **Evolution of spectral concentration during training.** Leading singular-value mass, $\sigma_1 / \sum_i \sigma_i$, of the learned GraphTSA communication matrices for 14 balanced PTB samples, with seven samples from each class. Individual trajectories and class-level trends are shown across eight training checkpoints. Spectral concentration decreases sharply during early optimization and subsequently evolves more gradually.

Visualizing the emergence of token topology. Figs. 9 and 10 visualize this process directly for three representative samples from each class. To avoid selecting examples according to visual appearance, we choose the three samples whose eight-checkpoint leading-singular-value trajectories are closest to the corresponding class-average trajectory. Each sample is then followed through all eight training checkpoints.

The topology visualizations make the spectral transition tangible. Early communication matrices are comparatively spectrally concentrated. During the first few epochs, substantial reorganization becomes visible, consistent with the sharp decrease in spectral concentration in Fig. 8. Later checkpoints exhibit more differentiated token interactions while retaining a prominent dominant component. Although individual samples follow different detailed trajectories, the overall transition from concentrated communication toward richer heterogeneous structure is visible in both classes.

Emergent differences between the two groups. An additional observation in Fig. 8 is that the two class-average trajectories become increasingly separated during later training, despite the analysis not being designed to distinguish the two groups. The topology visualizations in Figs. 9 and 10 likewise show substantial sample-dependent variation in the structures that emerge.

We treat this observation as exploratory rather than evidence of class-specific physiological connectivity. The present analysis contains only seven tracked samples per class, and the learned matrices describe representation-level communication rather than anatomical or causal connectivity. Nevertheless, the observation is consistent with GraphTSA producing input-dependent communication structures whose spectral evolution can differ across groups. A systematic investigation of whether such training dynamics contain clinically meaningful information is an interesting direction for future work.

Relation to the rank analysis. The training dynamics provide a complementary perspective to the rank analysis in Section 4.3.2. While the rank analysis compares the structural capacity induced by different affinity-logit ranks, the present experiment fixes $k = 8$, the best-performing configuration on PTB, and examines how its communication structure develops during optimization. The two analyses therefore vary orthogonal dimensions: the former studies *structural capacity across ranks*, whereas the latter studies *structural evolution across training* at a fixed rank.

Despite these different perspectives, both analyses reveal a consistent structural pattern: richer secondary communication modes emerge while a substantial dominant spectral component is preserved. Together, the results suggest that GraphTSA does not simply switch between concentrated and het-

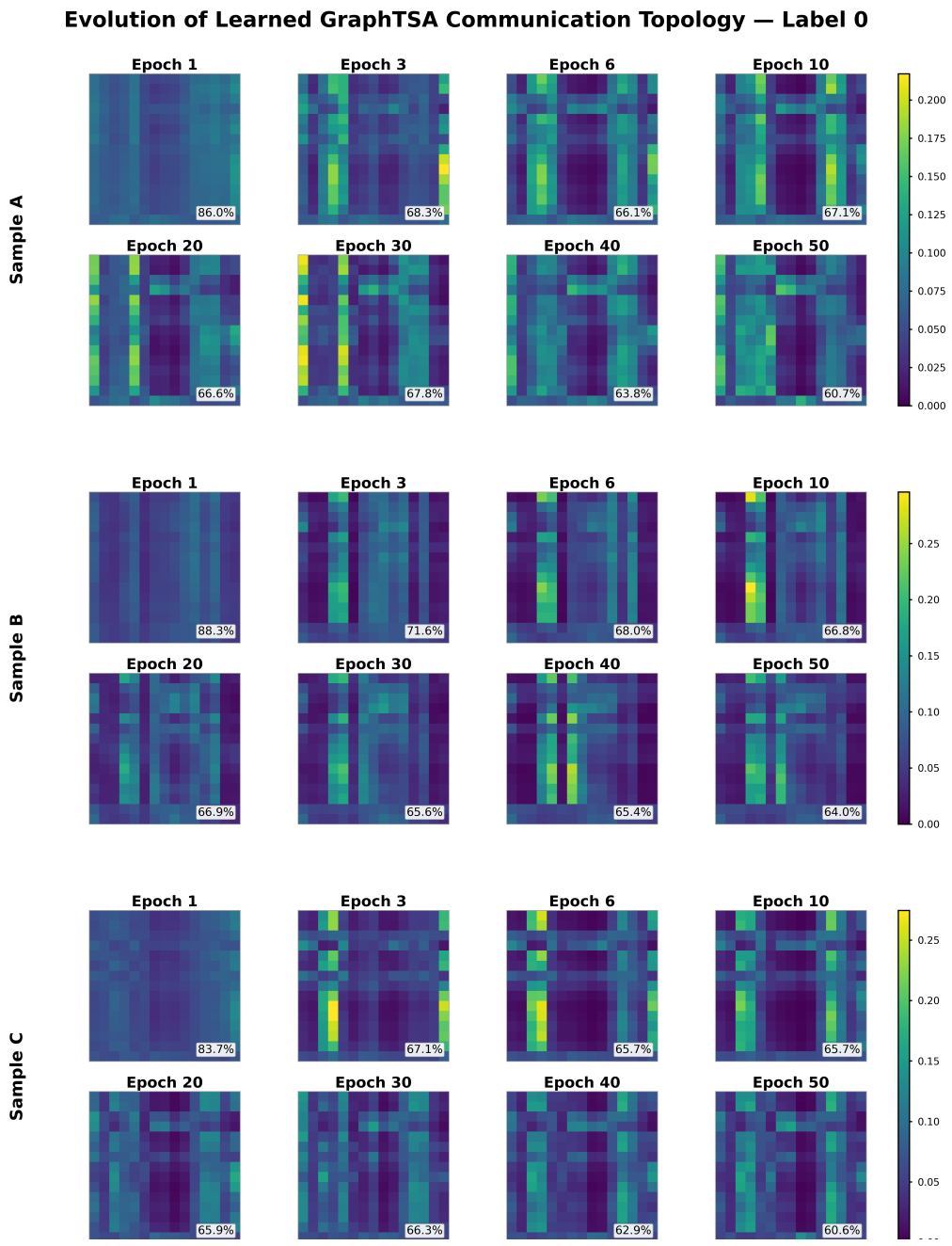

Figure 9: **Evolution of learned GraphTSA communication topology for representative Label-0 samples.** Three representative PTB samples are tracked across epochs $\{1, 3, 6, 10, 20, 30, 40, 50\}$. Each pair of rows corresponds to the same sample throughout training. Percentages denote the normalized leading singular-value mass, $\sigma_1 / \sum_i \sigma_i$. A single color scale is shared across all eight checkpoints of each sample, while scales are independently normalized between samples to preserve within-sample comparability.

erogeneous communication; instead, it learns differentiated token interactions around a persistent dominant mode. This representation-level behavior is consistent with the local-heterogeneity/global-synchronization motivation underlying GraphTSA.

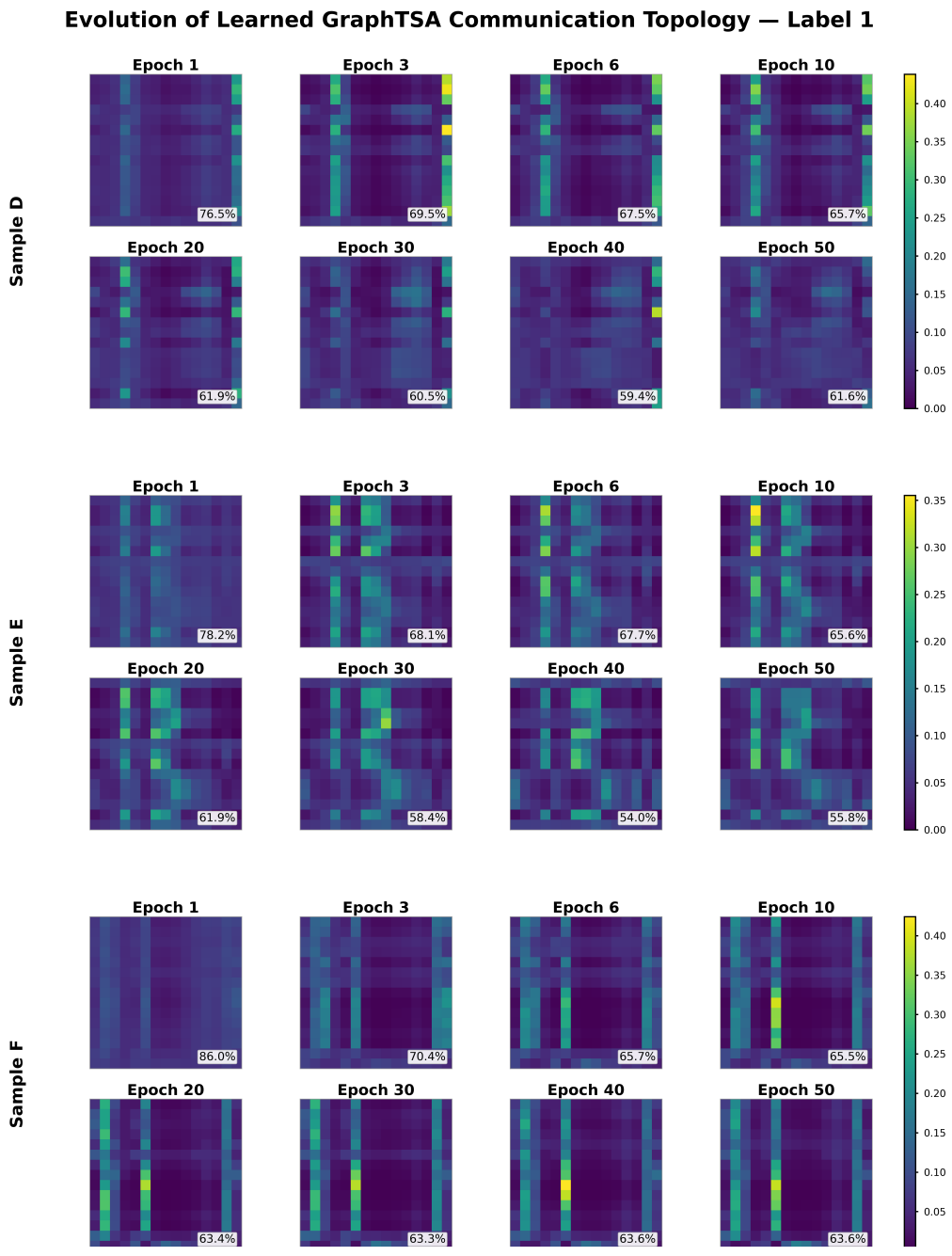

Figure 10: **Evolution of learned GraphTSA communication topology for representative Label-1 samples.** Three representative PTB samples are tracked across epochs $\{1, 3, 6, 10, 20, 30, 40, 50\}$. Each pair of rows corresponds to the same sample throughout training. Percentages denote the normalized leading singular-value mass, $\sigma_1 / \sum_i \sigma_i$. A single color scale is shared across all eight checkpoints of each sample, while scales are independently normalized between samples. Together with Fig. 9, these examples illustrate that the emergence of richer communication structure occurs across samples from both classes.

H Detailed Efficiency and Complexity Analysis

This section provides additional details for the efficiency analysis in Section 4.5. We first describe the profiling protocol and report the complete per-model measurements, and then analyze the com-

Table 6: **Efficiency comparison on PTB.** Inference latency is reported as mean ± standard deviation over 50 timed forward passes following 10 warm-up iterations. Peak memory denotes peak GPU memory during inference. GraphTSA uses $k = 8$.

Method	Params (M)	Acc. (%)	Latency (ms)	Memory (MB)
Transformer	6.593	77.37	12.572 ± 0.101	1213.49
Autoformer	6.584	73.35	38.886 ± 1.853	1550.71
FEDformer	17.131	76.05	20.724 ± 0.053	997.56
Informer	10.831	78.69	12.773 ± 0.067	920.51
Reformer	6.066	77.96	24.924 ± 0.826	3144.96
Nonformer	6.668	78.66	13.843 ± 0.243	1215.91
PatchTST	6.806	74.74	21.057 ± 0.967	1585.42
iTransformer	6.453	83.89	1.766 ± 0.012	231.07
Medformer	12.722	83.50	16.472 ± 0.455	774.82
MedGNN	11.478	84.53	14.158 ± 0.489	423.59
TeCh	1.549	85.96	0.902 ± 0.011	62.69
GraphTSA ($k = 8$)	**1.561**	**89.11**	$\mathbf{1.104 \pm 0.014}$	**62.74**

putational and parameter complexity introduced by GraphTSA's low-dimensional affinity parameterization.

H.1 Profiling Protocol

All models are profiled on PTB using the same hardware and inference protocol. Experiments are conducted on a single NVIDIA H100 NVL with batch size 128. We measure forward-pass inference without gradient computation. Each measurement consists of 10 warm-up iterations followed by 50 timed iterations, from which we report the mean latency and standard deviation. Peak GPU memory is measured under the same setting. We use randomly initialized weights for profiling, since the computational graph and input dimensions, rather than the learned parameter values, determine inference cost.

Table 6 reports the complete profiling results. We omit MTST from this comparison because a matched profiling result for the exact implementation used in our performance table is unavailable.

H.2 Computational Complexity

GraphTSA deliberately trades the linear token aggregation of CoTAR for input-adaptive pairwise communication. Importantly, however, the pairwise affinity is parameterized through a k-dimensional query–key space rather than directly in the full representation dimension. Given token representations

$$X \in \mathbb{R}^{N \times D},$$

GraphTSA first constructs two low-dimensional projections,

$$Q = XW_Q, \qquad K = XW_K, \qquad W_Q, W_K \in \mathbb{R}^{D \times k}, \tag{11}$$

such that

$$Q, K \in \mathbb{R}^{N \times k}.$$

The pre-softmax affinity logits are then

$$S = \frac{QK^\top}{\sqrt{D}} \in \mathbb{R}^{N \times N}, \qquad A = \mathrm{Softmax}_{\mathrm{row}}(S). \tag{12}$$

Before row-wise softmax, the affinity logits satisfy

$$\mathrm{rank}(S) = \mathrm{rank}(QK^\top) \le k. \tag{13}$$

We emphasize that Eq. 13 applies to the *pre-softmax affinity logits* S; the nonlinear row-wise softmax does not in general preserve this rank bound for A.

Constructing Q and K requires $\mathcal{O}(NDk)$ operations, while forming $QK^\top$ requires $\mathcal{O}(N^2k)$. GraphTSA subsequently applies the learned communication matrix to the core representation $O \in \mathbb{R}^{N \times d_c}$,

$$C = AO, \tag{14}$$

Table 7: **GraphTSA efficiency across affinity-logit ranks on PTB.** All configurations use the same profiling protocol described in Section H.1.

k	Params	Latency (ms)	Memory (MB)
1	1,550,280	1.133 ± 0.010	62.697
4	1,554,906	1.122 ± 0.011	62.716
8	1,561,074	1.104 ± 0.014	62.740
16	1,573,410	1.169 ± 0.078	62.790

which requires $\mathcal{O}(N^2 d_c)$ operations. Ignoring the shared MLP operations that are not specific to the graph construction, the additional communication computation can therefore be summarized as

$$\mathcal{C}_{\text{GraphTSA}} = \mathcal{O}(NDk) + \mathcal{O}(N^2 k) + \mathcal{O}(N^2 d_c). \tag{15}$$

The pairwise component is thus

$$\boxed{\mathcal{O}\big(N^2(k + d_c)\big)}. \tag{16}$$

GraphTSA therefore does *not* claim linear complexity in the number of tokens: explicitly constructing and applying A retains quadratic dependence on N. Instead, the role of k is to constrain the dimensional capacity used to generate the pairwise affinity logits. In particular, the rank-dependent affinity-construction term is $\mathcal{O}(N^2 k)$ rather than using the full representation dimension D for pairwise affinity construction. This provides a controllable structural bottleneck while still allowing explicit token-to-token communication.

H.3 Parameter Cost of Increasing Rank

The parameter overhead introduced by the affinity factorization is especially small. The two projections W_Q and W_K, including their biases, contain

$$P_{\text{rank}} = 2(Dk + k) = \boxed{2k(D + 1)} = \mathcal{O}(Dk) \tag{17}$$

parameters. Hence, unlike the quadratic dependence of explicit token communication on N, the number of additional learnable parameters grows only linearly with k.

Table 7 reports the measured cost at the four profiled ranks. Increasing the affinity-logit rank by $16\times$, from $k = 1$ to $k = 16$, increases the complete model size from 1,550,280 to 1,573,410 parameters, corresponding to only a 1.49% increase. Peak inference memory similarly changes from 62.70 MB to 62.79 MB.

Interestingly, measured wall-clock latency is nearly flat across these configurations rather than increasing monotonically with k. The measured means are 1.133, 1.122, 1.104, and 1.169 ms for $k = \{1, 4, 8, 16\}$, respectively. This does not contradict the $\mathcal{O}(N^2 k)$ rank-dependent term: asymptotic operation count and end-to-end GPU latency measure different quantities. At the token counts and small ranks considered here, the incremental rank-dependent computation is small relative to the complete forward pass and does not translate proportionally into measured GPU latency. We therefore interpret the measurements as evidence that increasing k has little *practical marginal cost in our evaluated setting*, rather than as evidence that inference complexity is independent of k.

This distinction is useful when interpreting the rank ablation. Increasing k expands the structural capacity available to the affinity logits, but does not require a proportionally larger model. On PTB, performance reaches its maximum at $k = 8$, whereas further increasing the rank provides no predictive benefit despite the additional capacity. Thus, k acts primarily as a controllable structural-capacity parameter rather than simply a model-size parameter.

H.4 Why Rank-One GraphTSA Does Not Reduce to Shared Broadcast

The lowest-rank setting also clarifies an important distinction between GraphTSA and CoTAR. When $k = 1$, the two projected token representations reduce to vectors

$$q, k \in \mathbb{R}^N,$$

and the affinity logits become

$$S = qk^\top, \qquad S_{ij} = q_i k_j, \qquad \operatorname{rank}(S) \le 1. \tag{18}$$

Although S is rank one, its entries need not be identical. Both q_i and k_j are functions of the input tokens, allowing the resulting affinity pattern to vary across token pairs and across samples.

For comparison, a uniform broadcast pattern can be represented by

$$S_{\text{uniform}} = c\,\mathbf{1}\mathbf{1}^\top, \tag{19}$$

which is also rank one but constitutes only a particular uniform rank-one structure. Therefore, "rank one" should not be conflated with "uniform": GraphTSA can produce heterogeneous, input-dependent affinity logits even when $k = 1$.

We do not claim that CoTAR is literally a special case of GraphTSA, since CoTAR additionally employs feature-dependent weighting during its core aggregation. Rather, the comparison highlights the difference in communication structure: CoTAR constructs a shared global representation that is broadcast back to the tokens, whereas rank-one GraphTSA already retains token-dependent pairwise affinity structure.

This distinction is reflected empirically. On PTB, rank-one GraphTSA achieves 87.43% accuracy and 84.11% F1, compared with 85.96% accuracy and 81.97% F1 for TeCh, while increasing the model size from 1,548,738 to only 1,550,280 parameters. Increasing the rank subsequently expands the available affinity structure, reaching 89.11% accuracy and 86.53% F1 at $k = 8$.

Together with the structural analysis in Section 4.4, these results suggest two complementary roles for GraphTSA. At low rank, GraphTSA already enables input-adaptive heterogeneous communication beyond CoTAR's shared global communication. Increasing k then controls the richness of the affinity structure that can be expressed, with only linear parameter growth and little additional measured inference cost over the ranks evaluated here.