TEMPiRL: Foundational Compounding Temporal Drift Theory for Temporal-Graph Adaptation in Large Language Models

Arnav Sharma
CortexPD Labs
arnavsharma.0914@gmail.com

Karthik Srikumar
CortexPD Labs
Karthiksrikumar83@gmail.com

Abstract

The basic architecture of a foundation model, including Large Language Models (LLMs), does not align with how time varies in sequential data as a continuous and conditioning variable, so they cannot learn from evolving information. This constraint is exemplified in interactive systems where LLMs are deployed. This study introduces a mathematical framework, called TEMPiRL for analysis of parameter-efficient temporal-graph adaptations. The framework posits an additive architecture that includes low-rank modulators and Time2Vec embeddings to make temporal and graph structured embeddings. TEMPiRL offers three main theoretical guarantees: a Lipschitz-based bound on drift of the model output that is proportional to the norm of the low-rank adapter; a Rademacher complexity bound on the generalization error that grows with the rank, r, of the low-rank adapter; and a formal condition for performance that captures the tradeoffs of the strength of the temporal signal, in terms of expectation with respect to a time average (decision rule), with the approximation error, and with the estimation error. This work provides a foundation for the future of additive foundation model architectures which allows for continually adaptable models.

1 Introduction

Foundation models (e.g., Large Language Models (LLMs)) are trained on a large, static corpora, with which there is a considerable misalignment in operating in a real feedback system such as a chatbot (1; 2). The misalignment results from a modeling architecture in which time is viewed as another token rather than as a conditioning variable, resulting in temporal inconsistencies and stochastic behavior from continuous data input (3; 4). This results in a fundamental theoretical gap: we do not have a theoretical basis on the rules of generalization over evolving data distributions, and we do not have a formal characterization on the trade-off between encoding new knowledge and retaining knowledge learned during pre-training (4). Addressing this gap requires a mathematical framework to model knowledge updates and guarantee model stability.

The challenge resides in extending learning theory from static function approximation to time-sensitive inquiry-based settings. Classical generalization theory considers the performance of a model on a fixed distribution of data (5). Yet, this theory provides us with no methods for measuring when a model changes with respect to a conditioning variable like time. Consequently, a framework that can advance stability into the temporal space and develop continuity characteristics is necessary (6).

1.1 Related Work

Current approaches for temporal adaptation depend on methods that are either computationally expensive or lack a formal theoretical basis. Retrieval-Augmented Generation (RAG) improves factual freshness by searching for external data, but, RAG functions as an unstructured, heuristic method because it cannot guarantee anything, including temporal reasoning (7; 8). Other methods with full-model fine-tuning, implicitly risk catastrophic forgetting, destabilizing the pre-trained model's knowledge base (9).

The more recent TG-LLM framework contributes to temporal reasoning in an empirical way with temporal-graph representations, dataset design, and augmentation procedures (10). However, TG-LLM contributions are purely methodological, without principled explanations on when such adaptations are stable, generalize, or provide a benefit. TEMPiRL stands as a complement by providing the theory for temporal-graph adaptation including formal guarantees on the stability under temporal changes (Lipschitz continuity), the generalization error (Rademacher complexity), and an exact performance condition indicating when adaptation will be beneficial.

1.2 The TEMPiRL Framework

TEMPiRL stands not as a new algorithm, but as a theoretical foundation that formalizes the principles of temporal-graph adaptation and provides guarantees. Its contributions include:

- **Stability Guarantees.** TEMPiRL demonstrates that small alterations in time only result in small, controlled changes in model output (Theorem 2.7); meaning that as the distributions change we can expect the predictions to remain tight.
- **Generalization Bounds.** It provides an upper bound for how accurately we can expect the adapted model to perform on new data (Theorem 2.8); it shows error scales with adapter size and embedding complexity, thus showing the tradeoff between flexibility and potential overfitting.
- **Performance Conditions.** TEMPiRL specifies when temporal-graph adaptation is useful (Theorem 2.9) by providing a condition comparing the potential strength of the temporal signal against the model's inherent limitations (approximation error) and the risk of overfitting to a small sample of limited data (generalization error).

2 Theoretical Foundations

2.1 LLM Temporal-Graph Problem Formulation

Look at a pre-trained LLM $\mathcal{M}_\Theta : \mathcal{X} \to \mathcal{Y}$ containing L layers, hidden dimension d, and frozen parameters Θ. Unlike typical NLP tasks, temporal reasoning requires the processing of data tuples $(x, t, \mathcal{G}_t)$ where $x \in \mathcal{X}$ represents input text, $t \in [0, T]$ represents temporal context and $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t)$ represents temporal relational structures with fixed entities $\mathcal{V}$ and temporal edges $\mathcal{E}_t$.

The challenge remains in realizing these transformer attention mechanisms and layers, conditioned both on temporal continuity and graph structure, in an efficient manner along with theoretical guarantees.

Definition 2.1 (Low-Rank Adapter)**.** *At transformer layer ℓ processing hidden states $\mathbf{H}^\ell \in \mathbb{R}^d$, the TEMPiRL adapter computes:*

$$\Delta \mathbf{H}^\ell = \mathbf{B}^\ell \left(\mathbf{A}_t^\ell \varphi(t) + \mathbf{A}_g^\ell f_g(\mathcal{G}_t) + \mathbf{A}_h^\ell \mathbf{H}^\ell \right) \tag{1}$$

where $\mathbf{A}_t^\ell \in \mathbb{R}^{r \times d_t}$, $\mathbf{A}_g^\ell \in \mathbb{R}^{r \times d_g}$, $\mathbf{A}_h^\ell \in \mathbb{R}^{r \times d}$ are down-projection matrices, $\mathbf{B}^\ell \in \mathbb{R}^{d \times r}$ is the up-projection with bottleneck rank $r \ll d$, $\varphi : [0, T] \to \mathbb{R}^{d_t}$ implements Time2Vec temporal embeddings (11), and $f_g : \mathcal{G} \to \mathbb{R}^{d_g}$ computes graph neural network features (12).

The adapted layer output follows the residual connection: $\mathbf{H}^\ell_{\text{out}} = \mathbf{H}^\ell + \Delta \mathbf{H}^\ell$, involving $Lr(d_t + d_g + 2d)$ trainable parameters throughout, thus combining transformer characteristics with the principled temporal-graph conditioning (full exploration in Supplementary Material A.1)).

2.2 Theoretical Assumptions and Regularity Conditions

TEMPiRL contains 4 assumptions required for the sake of tractability and practicality.

Algorithm 1 Example: TEMPiRLs practicality as a Financial Assistant

Scenario: An LLM-based financial assistant must answer a time-sensitive user query (hypothetical information).

INPUTS:
Input Text (x): "What is the market sentiment on ACME Corp?"
Temporal Context (t): The current date, October 18, 2025.
Graph Context ($\mathcal{G}_t$): A knowledge graph with a new, time-stamped edge:
(ACME Corp, acquired, Innovate Inc., timestamp: Oct 15, 2025)

ADAPTATION PROCESS (within a transformer layer):
The TEMPiRL adapter combines these inputs to modulate the hidden state:
1. A Time2Vec embedding $\varphi(t)$ represents the current date as a dense vector.
2. A GNN embedding $f_g(\mathcal{G}_t)$ represents the new "acquired" relationship structure.
3. These are injected into the model's processing of the input text x.

OUTPUT:
An up-to-date, context-aware response is generated:
"Market sentiment for ACME Corp is currently influenced by its recent acquisition of Innovate Inc."

Assumption 2.2 (Lipschitz Continuity)**.** *The transformer $\mathcal{M}_\Theta$ is L_{base}-Lipschitz continuous with respect to hidden state perturbations: $\|\mathcal{M}_\Theta(h_1) - \mathcal{M}_\Theta(h_2)\| \leq L_{base}\|h_1 - h_2\|$.*

The assumption stipulates that small alterations in internal representations (due to adapter modules) generate at least bounded variable output. Even though it is difficult to validate for large transformers, it can be approximately enforced via spectral normalization procedures, and is standard fare when theoretically analyzing deep networks (full exploration in Supplementary Material A.2) (13).

Assumption 2.3 (Bounded Temporal-Graph Embeddings)**.** *Embedding functions satisfy $\|\varphi(t)\|_2 \leq C_\varphi$ and $\|f_g(\mathcal{G}_t)\|_2 \leq R_g$ for all temporal and graph contexts.*

This constraint prevents temporal and structural information from unbounded numerical influence on transformer computations, and we can simply and easily enforce this through layer normalization (14) applied to the output of the embedding.

Assumption 2.4 (Adapter Matrix Norm Constraints)**.** *All adapter matrices have bounded Frobenius norms: $\|\mathbf{A}_t^\ell\|_F \leq \sigma_t$, $\|\mathbf{A}_g^\ell\|_F \leq \sigma_g$, $\|\mathbf{A}_h^\ell\|_F \leq \sigma_h$, $\|\mathbf{B}^\ell\|_F \leq \sigma_B$.*

The use of bounds such as those in this paper help derive stability and generalization guarantees, and these bounds can be retained by using weight decay regularization or explicit projection steps (full exploration in Supplementary Material A.3).

Assumption 2.5 (Transformer Hidden State Concentration)**.** *At each layer ℓ, hidden states concentrate with probability $1-\delta$: $\|\mathbf{H}^\ell\|_2 \leq C_h = \sqrt{2d\log(2/\delta)}$.*

This aligns with existing empirical findings that transformer activations have concentration properties in well-trained models as well as gives the probabilistic bounds necessary for the sample complexity analysis.

Remark 2.6 (Conditional Nature of Guarantees). The theoretical guarantees we present depend on convergence in training to a solution that satisfies Assumption 2.4. Analysis of joint optimization-generalization for training non-convex transformers is an open problem making the bounds contingent on algorithm success (full exploration in Supplementary Material A.4).

2.3 Core Theoretical Results

Compounding Temporal Stability: The first main result demonstrates that temporal perturbations on LLM inputs induce bounded changes to the outputs, taking into account how drift accumulates through transformer layers.

Theorem 2.7 (Compounding Temporal Drift)**.** *Under Assumptions 2.2-2.5, if each transformer layer ℓ is L_ℓ-Lipschitz w.r.t. hidden states and K_ℓ-Lipschitz w.r.t. temporal inputs, then temporal* *perturbations Δ produce bounded output drift:*

$$\|\mathcal{M}(x,t) - \mathcal{M}(x,t+\Delta)\|_2 \leq \sum_{\ell=1}^{L} \left(\prod_{j=\ell+1}^{L} L_j \right) K_\ell |\Delta|$$

This bound summarizes how temporal drift compounds as we move down transformer architectures, with multiplicatively larger effect in deeper layers. The layer-wise Lipschitz constraints K_ℓ depend on the adapter norms, adding clear incentives for regularization. The full proof, leveraging inductive reasoning of drift propagation, is in the Supplementary Material A.5.

Generalization Analysis: The second result provides insight into the fundamental tradeoff between a transformer's capacity for temporal adaptation and the risk of overfitting.

Theorem 2.8 (Adapter Sample Complexity)**.** *Let $\mathcal{H}_{TEMPiRL}$ denote the hypothesis class of TEMPiRL-adapted transformers. Under Assumptions 2.2-2.5, the Rademacher complexity over samples of size n satisfies:*

$$\mathfrak{R}_n(\mathcal{H}_{TEMPiRL}) \leq \frac{L_{base} L \sigma_B}{\sqrt{n}} \left(\sigma_t C_\varphi + \sigma_g R_g + \sigma_h C_h\right)$$

This bound explicitly connects generalization error to architectural choices (adapter rank r through $\sigma_\bullet$ terms), temporal embedding complexity (C_φ), and graph structure complexity (R_g). The $1/\sqrt{n}$ scaling provides standard PAC-learning rates while the linear dependence on layer count L reflects transformer depth effects. The complete proof using contraction principles and sub-additivity appears in Supplementary Material A.6.

Performance Conditions: The final theoretical result establishes when TEMPiRL provides benefits over static LLM baselines.

Theorem 2.9 (TEMPiRL Performance Condition)**.** *Let $R(f)$ denote expected risk, $f^*_{optimal}$ the optimal predictor, and f^*_{static} the optimal static LLM. Define temporal-graph signal strength $S = R(f^*_{static}) - R(f^*_{optimal})$ and approximation error $\epsilon_{approx} = \inf_{f \in \mathcal{H}_{TEMPiRL}} R(f) - R(f^*_{optimal})$. TEMPiRL provides theoretical benefit when:*

$$S > \epsilon_{approx} + C \cdot \mathfrak{R}_n(\mathcal{H}_{TEMPiRL})$$

This condition captures the essential premise: a temporal-graph informed representation must be informative enough to accommodate both the limitations of the architecture (which is an approximation error), and the complexity of the statistics (which is a generalization error). While quantities like f^*_{optimal} are not able to be modeled directly, this result gives important theoretical understanding into the conditions under which adaptation is successful and provides direction for the empirical investigation process complete analysis in Supplementary Material A.7).

3 Limitations and Future Work

While the theoretical framework utilizes assumptions that are typical in deep learning theory, they are not easy to check directly. Specifically, it is difficult to directly compute the Lipschitz continuity assumption (Assumption 2.2) for larger transformers, meaning the bounds we presented are conditional on certain model properties that likely approximately hold in well trained networks. Second, the performance condition (Theorem 2.9) presents unmeasurable risks of optimal predictors, which suggests conceptual, rather than measurable, performance.

Future research on this work for future publication will examine empirical validation in comparing TEMPiRL-adapted transformers, against baselines on temporal reasoning tasks and temporal knowledge-graph completion tasks. Ablation studies will be included to see which mathematical and fundamental components are the most important or unimportant. Future work will also include developing practical methods for estimating effective Lipschitz constants during training. We first aim to engage with the research community to identify the most important and impactful directions for empirical validation and theoretical extensions.

4 Conclusion

This study establishes a theoretical foundation for temporal-graph adaptation in large language models (LLMs) by providing formal stability guarantees, generalization bounds, and performance conditions that can provide insight into areas of limited theoretical understanding for temporal adaptation. The compounding drift analysis describes how temporal inconsistencies propagate through multiple transformer layers, while the sample complexity bounds directly relate adapter design choices to overfitting risk.

The impact of the framework not only establishes the foundations for temporal reasoning theory, but also to the more general and more challenging problem of adaptive foundation models. As LLMs are increasingly deployed into practical implementation in feedback systems, which require continual adaptation and use of feedback, TEMPiRL provides the theoretical distribution for optimally deploying LLMs in real feedback systems.

However, it is important to note that the same mechanisms that allow for beneficial updates can also be exploited. For example, a model could be continually updated with biased, misleading, or malicious temporal data, creating a new and subtle vector for spreading misinformation.

References

[1] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, et al. On the Opportunities and Risks of Foundation Models. *arXiv preprint arXiv:2108.07258*, 2021.

[2] Angeliki Lazaridou, Adhi Kuncoro, Elena Gribovskaya, et al. Mind the gap: Assessing temporal generalization in neural language models. *Advances in Neural Information Processing Systems*, 34:29348–29363, 2021.

[3] Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, et al. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273, 2022.

[4] Zixuan Ke, Yijia Shao, Haowei Lin, et al. Continual pre-training of language models. *arXiv preprint arXiv:2302.03241*, 2023.

[5] Vladimir N. Vapnik and Alexey Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280, 1971.

[6] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. MIT Press, 2018.

[7] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego De Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack Rae, Erich Elsen, Laurent Sifre. Improving Language Models by Retrieving from Trillions of Tokens. *Proceedings of the 39th International Conference on Machine Learning*, pages 2206–2240, 2022.

[8] Patrick Lewis, Ethan Perez, Aleksandra Piktus, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.

[9] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

[10] Siheng Xiong, Ali Payani, Ramana Kompella, and Faramarz Fekri. Large language models can learn temporal reasoning. *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10452–10470, 2024.

[11] Seyed Mehran Kazemi, Rishab Goel, Kshitij Jain, et al. Time2vec: Learning a vector representation of time. *arXiv preprint arXiv:1907.05321*, 2019.

[12] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.

[13] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *International Conference on Learning Representations*, 2018.

[14] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. Attention is all you need. *Advances in Neural Information Processing Systems*, 30:5998–6008, 2017.

[16] Edward J. Hu, Yelong Shen, Phillip Wallis, et al. LoRA: Low-rank adaptation of large language models. *International Conference on Learning Representations*, 2022.

[17] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, et al. Parameter-efficient transfer learning for nlp. *International Conference on Machine Learning*, pages 2790–2799, 2019.

[18] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, Qingsong Wen. Time-LLM: Time Series Forecasting by Reprogramming Large Language Models. *arXiv preprint arXiv:2310.01728*, 2023.

[19] Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha Talukdar. HyTE: Hyperplane-based temporally aware knowledge graph embedding. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2001–2011, 2018.

[20] Julien Leblay and Melisachew Wudage Chekol. Deriving validity time in knowledge graph. *Companion of the The Web Conference 2018 on The Web Conference*, pages 1771–1776, 2018.

[21] Zeyuan Allen-Zhu, Yuanzhi Li, Zhao Song. A Convergence Theory for Deep Learning via Over-Parameterization. *Proceedings of the 36th International Conference on Machine Learning*, pages 242–252, 2019.

A Supplementary Material

A.1 Parameter Complexity

Lemma A.1 (TEMPiRL Parameter Count)**.** *The total trainable parameters for TEMPiRL adapters across L transformer layers is exactly* $Lr(d_t + d_g + 2d)$.

Proof. For each layer ℓ, the parameter contributions are:

- $\mathbf{A}_t^\ell \in \mathbb{R}^{r \times d_t}$: rd_t parameters
- $\mathbf{A}_g^\ell \in \mathbb{R}^{r \times d_g}$: rd_g parameters
- $\mathbf{A}_h^\ell \in \mathbb{R}^{r \times d}$: rd parameters
- $\mathbf{B}^\ell \in \mathbb{R}^{d \times r}$: dr parameters

Per-layer total: $r(d_t + d_g + d + d) = r(d_t + d_g + 2d)$. Across L layers: $Lr(d_t + d_g + 2d)$. □

A.2 Assumption Practicality

Lipschitz Continuity (Assumption 2.2): Although it is impossible to compute exact Lipschitz constants for large transformers, it is possible to approximately impose this assumption via spectral normalization of weight matrices (13) and gradient-clipping at training time. This assumption is necessary for any stability analysis because it is the intuitive assumption that small perturbations to internal states should not induce unbounded perturbation to outputs.

Bounded Embeddings (Assumption 2.3): Simply monitored via layer normalization (14) applied to Time2Vec temporal embeddings and GNN outputs. It guarantees that temporal info and graph info doesn't overshadow transformer computations numerically and supports consistency in training dynamics.

Adapter Norm Constraints (Assumption 2.4): Essential for controlling adapter capacity and directly implementable via either ℓ_2 weight decay regularization or Frobenius norm projection after gradient updates. These bounds will give the constant factors in our main theorems and tunable parameters for practitioners.

Hidden State Concentration (Assumption 2.5): Observes empirical evidence regarding activation ordering patterns for transformers, as well as the high-probability bounds to apply for analyzing generalization of the PAC-style. The logarithmic dependence on confidence δ is standard in concentration inequalities.

A.3 Training Algorithm

Algorithm 2 TEMPiRL Training with Norm Constraints

Input: Training data $\{(x_i, y_i, t_i, \mathcal{G}_{t_i})\}$, rank r, learning rate η
Initialize adapter matrices with Xavier initialization for $\{\mathbf{A}^\ell\}$ and zeros for $\{\mathbf{B}^\ell\}$
for epoch = 1 to N **do**
  for each batch **do**
    Compute Time2Vec embeddings $\varphi(t_j)$ and GNN features $f_g(\mathcal{G}_{t_j})$
    for layer $\ell = 1$ to L **do**
      Compute adapter update: $\Delta\mathbf{H}^\ell = \mathbf{B}^\ell(\mathbf{A}_t^\ell \varphi(t_j) + \mathbf{A}_g^\ell f_g(\mathcal{G}_{t_j}) + \mathbf{A}_h^\ell \mathbf{H}^\ell)$
      Apply residual: $\mathbf{H}_{\text{out}}^\ell = \mathbf{H}^\ell + \Delta\mathbf{H}^\ell$
    end for
    Compute loss and update adapters via backpropagation
    Project adapters: $\mathbf{A}^\ell \leftarrow \text{proj}_{\|\cdot\|_F \le \sigma}(\mathbf{A}^\ell)$, $\mathbf{B}^\ell \leftarrow \text{proj}_{\|\cdot\|_F \le \sigma_B}(\mathbf{B}^\ell)$
  end for
end for

A.4 Optimization-Generalization Gap

In the conceptual analysis, we assumed convergence to solutions satisfying Assumption 2.4. However, the training of transformers involves massive non-convex optimization landscapes, and thus global convergence cannot be guaranteed as part of this assumptions about the underlying embedding structure. This constraint presents a well-known fundamental disconnection between a generalization bound (which assumes we will find good solutions exist) and any type of optimization guarantees (which we are not providing).

In the analysis, we make the assumption that if a good solution exists, the algorithm will converge to it. But in the setting of transformer training, the optimization landscape is non-convex and global convergence is not *possible* (in the sense of optimization guarantees). This is a gap between the generalization bounds, which assume the existence of good solutions, and the optimization guarantees, which we do not provide.

Recent research on optimization of neural networks seems to indicate that over-parameterized networks are able to attain good generalization even though they are non-convex. Extending these results to the adapter based architecture is still an open question. Practitioners should think of the bounds as conditional guarantees which hold if the training is successful and not as guarantees of performance.

A.5 Compounding Temporal Drift Proof

Proof of Theorem 2.7. We proceed by induction on transformer layers. Let $\Delta H_\ell = H_\ell(t+\Delta) - H_\ell(t)$ denote drift at layer ℓ.

Base Case ($\ell = 1$): The first layer's temporal sensitivity satisfies:

$$\|\Delta H_1\| = \|f_1(H_0, t+\Delta) - f_1(H_0, t)\| \leq K_1|\Delta|$$

by layer-wise Lipschitz continuity in temporal inputs.

Inductive Step: Assume $\|\Delta H_{\ell-1}\| \leq \sum_{j=1}^{\ell-1}\left(\prod_{k=j+1}^{\ell-1} L_k\right) K_j|\Delta|$.

For layer ℓ:

$$\begin{aligned}
\|\Delta H_\ell\| &= \|f_\ell(H_{\ell-1}(t+\Delta), t+\Delta) - f_\ell(H_{\ell-1}(t), t)\| && (2)\\
&\leq L_\ell \|H_{\ell-1}(t+\Delta) - H_{\ell-1}(t)\| + K_\ell|\Delta| && (3)\\
&= L_\ell\|\Delta H_{\ell-1}\| + K_\ell|\Delta| && (4)
\end{aligned}$$

Substituting the inductive hypothesis:

$$\|\Delta H_\ell\| \leq L_\ell \sum_{j=1}^{\ell-1}\left(\prod_{k=j+1}^{\ell-1} L_k\right) K_j|\Delta| + K_\ell|\Delta| = \sum_{j=1}^{\ell}\left(\prod_{k=j+1}^{\ell} L_k\right) K_j|\Delta|$$

The final output bound follows by applying this result at layer L. □

A.6 Sample Complexity Proof

Proof of Theorem 2.8. We apply standard Rademacher complexity techniques adapted to the TEMPiRL architecture:

Step 1 - Base Model Contraction: Since $\mathcal{M}_\Theta$ is L_{base}-Lipschitz:

$$\mathfrak{R}_n(\mathcal{H}_{\text{TEMPiRL}}) \leq L_{\text{base}}\mathfrak{R}_n\left(\left\{\sum_{\ell=1}^{L} \Delta\mathbf{H}^\ell\right\}\right)$$

Step 2 - Layer Sub-additivity: By sub-additivity of Rademacher complexity:

$$\mathfrak{R}_n\left(\left\{\sum_{\ell=1}^{L} \Delta\mathbf{H}^\ell\right\}\right) \leq \sum_{\ell=1}^{L} \mathfrak{R}_n\left(\{\Delta\mathbf{H}^\ell\}\right)$$

Step 3 - Single Adapter Analysis: For layer ℓ, applying contraction to $\mathbf{B}^\ell$:

$$\mathfrak{R}_n\left(\{\Delta\mathbf{H}^\ell\}\right) \le \sigma_B \mathfrak{R}_n\left(\{\mathbf{A}_t^\ell\varphi(t) + \mathbf{A}_g^\ell f_g(\mathcal{G}_t) + \mathbf{A}_h^\ell\mathbf{H}^\ell\}\right)$$

Step 4 - Linear Function Class Bounds: Using triangle inequality and standard Rademacher bounds for linear functions:

$$\mathfrak{R}_n\left(\{\mathbf{A}_t^\ell\varphi(t) + \mathbf{A}_g^\ell f_g(\mathcal{G}_t) + \mathbf{A}_h^\ell\mathbf{H}^\ell\}\right) \tag{5}$$

$$\le \mathfrak{R}_n(\{\mathbf{A}_t^\ell\varphi(t)\}) + \mathfrak{R}_n(\{\mathbf{A}_g^\ell f_g(\mathcal{G}_t)\}) + \mathfrak{R}_n(\{\mathbf{A}_h^\ell\mathbf{H}^\ell\}) \tag{6}$$

$$\le \frac{1}{\sqrt{n}}(\sigma_t C_\varphi + \sigma_g R_g + \sigma_h C_h) \tag{7}$$

Step 5 - Final Assembly: Combining all steps and summing over L layers:

$$\mathfrak{R}_n(\mathcal{H}_{\text{TEMPiRL}}) \le \frac{L_{\text{base}} L \sigma_B}{\sqrt{n}}\left(\sigma_t C_\varphi + \sigma_g R_g + \sigma_h C_h\right)$$

□

A.7 Performance Condition Analysis

Proof of Theorem 2.9. Consider the learned predictor $\hat{f}$ from TEMPiRL training. Standard risk decomposition yields:

$$R(\hat{f}) - R(f^*_{\text{optimal}}) = \underbrace{\left(\inf_{f\in\mathcal{H}_{\text{TEMPiRL}}} R(f) - R(f^*_{\text{optimal}})\right)}_{\epsilon_{\text{approx}}} + \underbrace{\left(R(\hat{f}) - \inf_{f\in\mathcal{H}_{\text{TEMPiRL}}} R(f)\right)}_{\text{Generalization Error}}$$

Standard PAC-learning theory bounds the generalization error through Rademacher complexity. With high probability:

$$R(\hat{f}) - \inf_{f\in\mathcal{H}_{\text{TEMPiRL}}} R(f) \le C \cdot \mathfrak{R}_n(\mathcal{H}_{\text{TEMPiRL}})$$

Therefore: $R(\hat{f}) \le R(f^*_{\text{optimal}}) + \epsilon_{\text{approx}} + C \cdot \mathfrak{R}_n(\mathcal{H}_{\text{TEMPiRL}})$

For TEMPiRL to provide benefit over the best static baseline: $R(\hat{f}) < R(f^*_{\text{static}})$

Substituting the bound and rearranging using the definition $S = R(f^*_{\text{static}}) - R(f^*_{\text{optimal}})$:

$$R(f^*_{\text{optimal}}) + \epsilon_{\text{approx}} + C \cdot \mathfrak{R}_n(\mathcal{H}_{\text{TEMPiRL}}) < R(f^*_{\text{static}})$$
$$\epsilon_{\text{approx}} + C \cdot \mathfrak{R}_n(\mathcal{H}_{\text{TEMPiRL}}) < S$$

Rearranging yields the stated condition. □

A.8 Graph Neural Network Dimension Analysis

Proposition A.2 (GNN Dimension Selection Heuristic)**.** *For temporal graphs with $|\mathcal{V}|$ vertices and spectral gap $\gamma > 0$, we propose:*

$$d_g = O\left(\min\left\{\frac{\log|\mathcal{V}|}{\gamma}, \frac{r\cdot d}{4}, 128\right\}\right)$$

The first requirement pertains to information-theoretic issues about how we represent the structure of the graph, the second requirement relates to avoiding imbalance in adapter component, and the third requirement is about computational tractability. The formal proof remains as future work.

A.9 Controlled Sensitivity Analysis

Theorem A.3 (TEMPiRL Jacobian Analysis)**.** *The influence of temporal and graph embeddings on hidden state updates follows:*

$$\frac{\partial(\Delta\mathbf{H}^\ell)}{\partial\varphi(t)} = \mathbf{B}^\ell\mathbf{A}_t^\ell, \quad \frac{\partial(\Delta\mathbf{H}^\ell)}{\partial f_g(\mathcal{G}_t)} = \mathbf{B}^\ell\mathbf{A}_g^\ell$$

This enables direct analysis of temporal versus structural influence through learned adapter matrices, providing interpretability for adaptation mechanisms.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the paper's contribution, which is TEMPiRL, and its specific theoretical contributions listed as three theoretical guarantees: stability, generalization bounds, and performance conditions.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a specific "Limitations and Future Work" section where theoretical and practicality cases are considered for TEMPiRL and the limitations/uncertainties are listed. It explicitly points out the difficulty of verifying some assumptions (e.g., Lipschitz continuity for large transformers) as the main limitation.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The 4 central assumptions to the framework proposed in the paper are stated in section 2.2 and the complete proofs are presented in the supplementary material in 5.5, 5.6, and 5.7.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [NA]

   Justification: The paper proposes foundational theory/math, without formal experiments, leading this question to be not applicable.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [NA]

   Justification: The paper does not involve any data or code, leading to this question being not applicable.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [NA]

   Justification: The paper does not involve any empirical training or experiments leading to this question being not applicable.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [NA]

   Justification: The paper does not involve any empirical experiments leading to this question being not applicable.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [NA]

   Justification: The paper does not involve any empirical training or experiments leading to this question being not applicable.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This paper is simply a theoretical analysis with no involvement of data collection, human subjects, or any other potentially unehtical routes.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: This is discussed in the conclusion

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not contain any data, models, or code.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper only uses existing concepts and math, which when used were respectively cited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not contain any assets such as data, models, or code.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper did not involve crowdsourcing experiments.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve any form of study participants.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: This paper is about a theoretical framework. While the paper discusses LLMs, an LLM was not a component in the process.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.