

L2T-Hyena: Enhancing State-Space Models with an Adaptive Learn-to-Teach Framework

Fatemeh Sohbat
School of Electrical Engineering
Iran University of Science &
Technology
Tehran, Iran

Farzan Haddadi
School of Electrical Engineering
Iran University of Science &
Technology
Tehran, Iran

Hamid Salahinejad
School of Computer Engineering
Tarbiat Modares University
Tehran, Iran

Abstract— State-Space Models (SSMs) have emerged as efficient alternatives to computationally intensive architectures like Transformers, particularly for sequence modeling. However, a fundamental challenge in their training is the reliance on static loss functions, which may not be optimal across all learning stages. To address this issue, in this paper a hybrid model integrating the Hyena architecture with a Dynamic Loss Network (DLN) is proposed which is guided by a Learn-to-Teach (L2T) approach (L2T-DLN). In this framework, the Hyena model is a student, and its loss function is optimized adaptively. A teacher model, leveraging a memory of the student's past performance, guides the DLN in dynamically balancing the primary cross-entropy loss and a regularization term. Experiments on the Penn Treebank (PTB) dataset show that our approach significantly improves language modeling performance. Our proposed model achieved a validation Perplexity of 102.6, a notable improvement over the 110.4 achieved by a baseline Hyena model using a static loss function. This research indicates that combining SSMs with adaptive loss function markedly enhances the quality and efficiency of deep learning models for sequential data, showing potential for applications in Natural Language Processing (NLP), time-series analysis, and biological signal processing.

Keywords— *State-Space Models (SSMs), Learn to Teach (L2T), Dynamic Loss Network (DLN), Hyena, Language Modeling, Deep Learning.*

I. INTRODUCTION

Deep learning models, particularly Transformers, have achieved remarkable success in natural language processing [1, 2, 3, 4, 5]. However, their architectures face a significant bottleneck in processing long sequences due to the self-attention mechanism's quadratic computational complexity ($O(L^2)$) [6, 7, 8, 9]. The Hyena model, a State-Space Model (SSM), was introduced as a powerful solution, leveraging long convolutional operations to reduce this complexity to a more manageable $O(L \log L)$ [10, 11].

Despite the architectural advantages, a major challenge in the optimization process persists due to the reliance on static loss functions. These fixed functions cannot adapt to the dynamic states of learning, limiting the model's ability to achieve optimal and rapid convergence. To overcome this, the Learning to Teach with a Dynamic Loss Network (L2T-

DLN) framework provides an innovative solution by adjusting the loss function based on the model's performance during training, enabling more intelligent optimization [12,13].

This research introduces the L2T-Hyena model, which combines the efficient Hyena architecture with the adaptive L2T-DLN framework. The primary objective is to evaluate this hybrid approach on a language modeling task using the Penn Treebank (PTB) dataset and compare its efficiency against a baseline Hyena model trained with a conventional static loss function [10,11,12].

The main contributions of this paper are:

- Novel integration of the Hyena model with the L2T-DLN framework for adaptive loss optimization in a language modeling context.
- Demonstration of significant performance improvement on the PTB benchmark, achieving a lower Perplexity score compared to the baseline model.
- Detailed analysis of the dynamic interplay between the student (Hyena), the teacher model, and the DLN.

II. RELATED WORK

This section provides an overview of the core concepts that form the foundation of our proposed model: State-Space Models (SSMs), the Hyena architecture, and Learning to Teach with Dynamic Loss Networks (L2T-DLN) framework [9,10,12].

A. State-Space Models (SSMs)

State-Space Models (SSMs) are a class of mathematical models that describe a dynamic system using a set of first-order differential or difference equations. The core of an SSM is the state vector, a multi-dimensional vector that summarizes the past history of the system [14,15].

For a linear continuous-time system, the SSM is defined by two primary equations:

1. State Equation, which describes how the state vector $x(t)$ evolves over time based on current state and an input vector $u(t)$:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (1)$$

Where, A is the state evolution matrix that governs the system's internal dynamics, and B is the input transformation matrix that determines the influence of the input on the state [14,15].

2. The Output Equation, which defines how the output vector $y(t)$ is generated from the current state and input:

$$y(t) = Cx(t) + Du(t) \quad (2)$$

Where, C is the output matrix that maps the state to the output, and D is the feedthrough matrix that models the direct impact of the input on the output [14,15].

In deep learning, SSMs have become a promising alternative to attention-based architectures, as their structure is highly effective for processing time-series data. A key advantage is their linear computational complexity with respect to sequence length, a significant improvement over the quadratic complexity of attention mechanisms, making them exceptionally efficient for long sequences. Notable SSM architectures include S4 and Mamba [9,15,16,17].

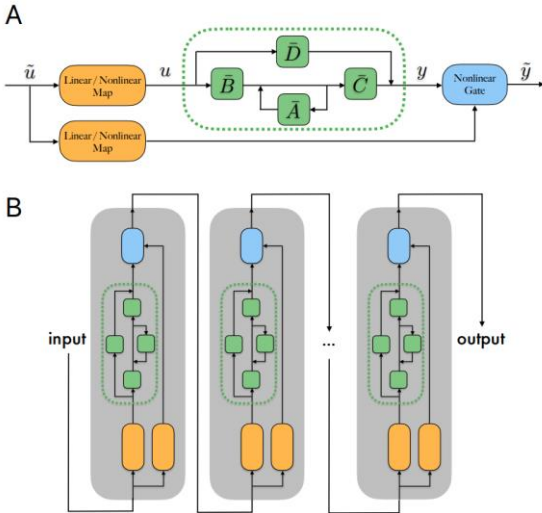


Fig. 1: A. General scaffolding of an SSM. The dynamical model is represented in green. The input to the SSM is pre-processed and forked off in a skip connection (lower signal). The nature of the pre-processing map (linear or nonlinear) depends on the specific scaffolding. The output of the recursion is then post-processed with a nonlinear gate. B. Overall architecture of an SSM. Each of the SSMs including its scaffolding is included in a layered fashion, where the output from one layer is the input to the next [14].

B. The Hyena Model

The Hyena model was proposed as a scalable, low-cost alternative to the attention mechanism in Transformers, specifically addressing its $O(L^2)$ computational complexity. Hyena models long-range dependencies with sub-quadratic complexity ($O(L \log L)$) by employing a combination of implicit long convolutions and data-dependent multiplicative gating. Instead of an attention layer, Hyena parameterizes its convolutional filters implicitly using a small feed-forward network and leverages Fast Fourier Transforms (FFTs) for highly efficient computation [10,11,18,19].

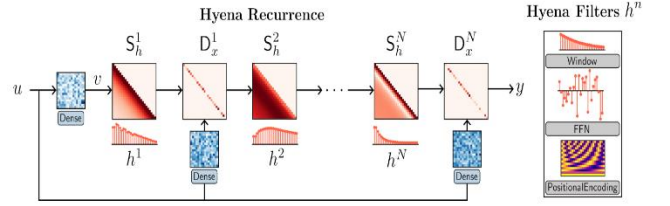


Fig 2: Hyena operator is defined as a recurrence of two efficient subquadratic primitives: an implicit long convolution h (i.e. Hyena filters parameterized by a feed-forward network) and multiplicative elementwise gating of the input. The depth of the recurrence specifies the size of the operator. Hyena can equivalently be expressed as a multiplication with data-controlled (conditioned by the input u) diagonal matrices D_x and Toeplitz matrices S_h . In addition, Hyena exhibits sublinear parameter scaling (in sequence length) and unrestricted context, similar to attention, while having lower time complexity [10].

C. Learning to Teach with Dynamic Loss Networks (L2T-DLN)

The Learning to Teach (L2T) paradigm is a meta-learning approach that aims to optimize a student model's training process through the guidance of a teacher model. L2T-DLN is an advanced adaptive optimization framework that uses a dynamic network to guide the student model toward better performance across different learning stages.

L2T-DLN features a memory-augmented teacher model, which allows the system to learn from accumulated experiences during training. The framework involves a strong, dynamic interaction between three components [12,13]:

- The student model is trained using the loss weights provided by the DLN.
- The Dynamic Loss Network (DLN) is updated based on feedback from the teacher.

- The teacher model, equipped with memory units, learns to provide optimal guidance by retaining information about long-term training dependencies

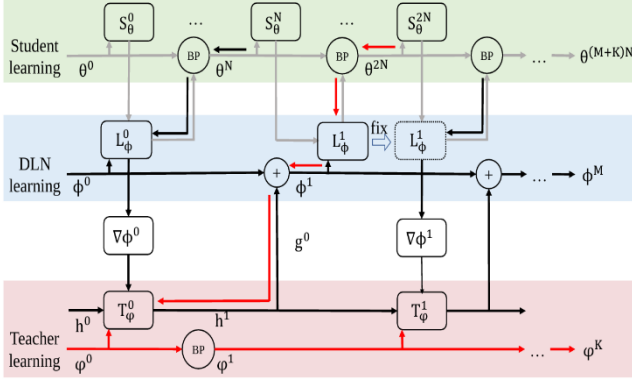


Fig 3: The pipeline of L2T-DLN [12]

III. METHODOLOGY

The proposed model, L2T-Hyena, integrates the Hyena architecture as a student model within a Learning to Teach (L2T) framework. This framework employs a Dynamic Loss Network (DLN) for adaptive loss function optimization, aiming to enhance language modeling by combining Hyena's powerful sequence modeling with adaptive learning mechanisms [10,12,13].

A. Student Model (Hyena)

The student model is a Hyena-based architecture designed for language modeling. It begins with an embedding layer that maps input tokens from a 10,000-token vocabulary to 256-dimensional vectors, which are then combined with positional embeddings. The core of the model is a stack of 6 Hyena Blocks, which model long-range dependencies using a combination of a short convolution (kernel size 3) and a long, FFT-based convolution whose filter is dynamically generated. Key architectural parameters include a model dimension of 256 and an order of 2. Finally, an output layer maps the processed sequence to the vocabulary size to produce logits for next-token prediction. The weights of this final layer are tied to the embedding layer to improve parameter efficiency [2,10,20].

B. Dynamic Loss Network

The DLN's primary role is to dynamically adjust the weights between the standard Cross-Entropy loss and an L2 logit regularization term. For each training instance, it takes the student's logits and target tokens as input to extract five statistical features, such as prediction confidence and error margin. This sequence of features is first normalized and then summarized into a single representative vector by a Gated Recurrent Unit (GRU). This summary vector is then passed to a 4-layer MLP which produces a single raw weight for the

regularization term. The final output is a scalar weight produced by a Sigmoid function [12,21,22,23].

C. Teacher Model and L2T Framework

The teacher model is a memory-augmented system whose core predictive component is a Multi-Layer Perceptron (MLP), guiding the DLN's learning process. Instead of processing text directly, it operates on a memory buffer that stores up to 500 past experiences. Each experience contains the statistical features generated by the DLN for a given sequence and the resulting student loss. During its training step, the teacher strategically samples from this memory, prioritizing high-loss examples to focus on more challenging cases. The teacher's task is to predict the student's final loss given a proposed regularization weight from the DLN. It is optimized by minimizing the Huber Loss between its prediction and the actual student loss, ensuring its training is robust to outliers [24].

IV. EXPERIMENTAL SETUP

This section details the dataset, implementation specifics, and the baseline model used for a comparative analysis.

A. Dataset

Penn Treebank (PTB) dataset was used for both training and evaluation. PTB is a standard benchmark for language modeling, consisting of approximately 930,000 training tokens, 74,000 validation tokens, and a vocabulary of 10,000 unique tokens.

B. Implementation Details

The system was implemented using a standard PyTorch training loop. Each of three primary components student, teacher, and DLN was optimized using a separate AdamW optimizer with distinct hyperparameters, as detailed in Table 1.

Table 1: Optimizer Hyperparameters

Component	Optimizer	Learning Rate	Weight Decay
Student	AdamW	2×10^{-4}	0.15
Teacher	AdamW	2×10^{-6}	0.01
DLN	AdamW	5×10^{-7}	0.01

The loss functions were tailored to each component's role. The student model was trained on a dynamically weighted

combination of Cross-Entropy and an L2 logit regularization term. The models were trained for 10 epochs with a batch size of 128 and a sequence length of 64. A cosine annealing learning rate scheduler with a 2 epoch warmup period was used to ensure stable convergence. For additional stability, gradient clipping was applied to all models during training.

C. Baseline Model

For a fair and direct comparison, a baseline Hyena model was trained as a standalone model. It used the exact same hyperparameters (e.g., learning rate, weight decay, batch size) as the student model but was optimized using a standard, fixed Cross-Entropy loss function without L2T-DLN framework.

V. RESULTS AND DISCUSSION

This section presents the performance of the proposed L2T-Hyena model, comparing it against the baseline Hyena model on the PTB dataset. The analysis includes a quantitative comparison of key metrics and a visual analysis of the training dynamics.

A. Quantitative Performance Comparison

The final performance and training characteristics of both models are summarized in Table 2. The proposed L2T-Hyena model demonstrates a clear and significant improvement in predictive performance, albeit with an increase in computational cost.

Table 2: Detailed Performance Comparison on PTB Dataset

Metric	Baseline (Vanilla Hyena)	L2T-Hyena	Improvement
Validation Perplexity	110.4	102.6	7.1% Reduction
Validation Loss	4.7	4.6	Lower is Better
Epoch Achieved	3	5	Sustained Learning
Final Training Loss (Epoch 10)	3.1	1.91	36.1% Reduction

The L2T-Hyena model achieved a final validation perplexity that was 7.8 points lower than the baseline, which corresponds to a 7.1% relative reduction, confirming the substantial effectiveness of the adaptive training framework. Furthermore, the proposed model reached its optimal point at a later epoch (5 vs. 3), suggesting it was able to sustain productive learning for a longer period before overfitting. This improved performance comes with a computational

trade-off, as the L2T-Hyena framework required approximately 31% more training time due to the added complexity of the teacher and DLN components.

B. Visual Analysis of Training Dynamics

To provide deeper insight into the training process, key metrics were plotted over the 10 training epochs.

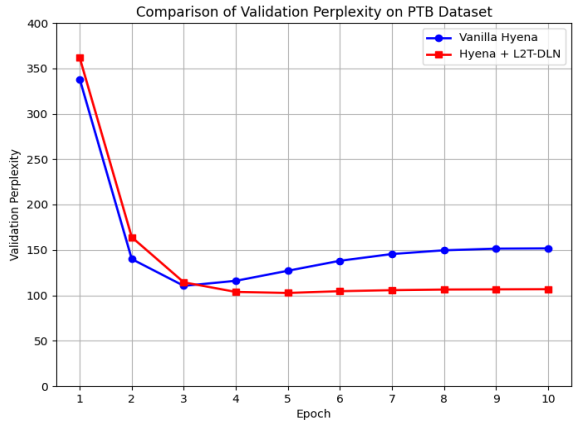


Fig 4: Comparison of Validation Perplexity.

Fig. 4 illustrates the validation perplexity for both models over the course of training. As shown, the baseline Vanilla Hyena model achieves its optimal performance at epoch 3 (Validation Perplexity of 110.4). After this point, its performance on the validation set consistently degrades, which is a clear sign of overfitting. In contrast, the L2T-Hyena model (red line) not only reaches a superior perplexity score of 102.6 at epoch 5, but also demonstrates remarkable stability in subsequent epochs. This highlights a key advantage of the adaptive L2T-DLN framework: its ability to mitigate overfitting and improve generalization capabilities. While the quantitative comparison in Table 2 correctly uses the baseline's best score from epoch 3, continuing the training for 10 epochs was essential to visually demonstrate these distinct training dynamics and the proposed model's superior stability.

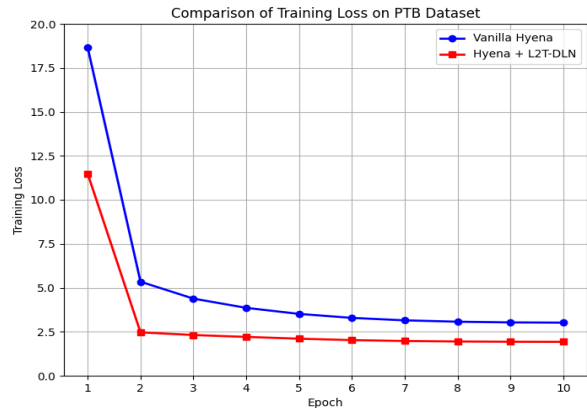


Fig 5: Comparison of Training Loss.

Fig. 5 visualizes the average training loss for both models. It demonstrates that the proposed L2T-Hyena model converges significantly faster and to a lower final loss value than the baseline. This indicates a more efficient and effective learning process on the training data itself. When viewed alongside Figure 4, this shows that the improved training efficiency does not come at the cost of generalization.

VI. CONCLUSION AND FUTURE WORK

This research demonstrated that integrating the Hyena architecture with the L2T-DLN framework significantly enhances language modeling performance. Evaluated on the PTB dataset, our proposed model achieved a validation Perplexity of 102.6, a substantial improvement over the 110.4 achieved by a standalone Hyena model. This result highlights the efficacy of using adaptive loss optimization and a guided learning strategy to unlock the full potential of advanced sequence models. The primary novelty of this work lies in the first-time combination of the Hyena architecture with the L2T-DLN framework for language modeling, establishing a new and effective training paradigm.

Future research could explore several directions:

- **Broader Evaluation:** The proposed L2T-Hyena model could be tested on larger and more complex language datasets (e.g., WikiText-103) or adapted for different NLP tasks, such as machine translation and text summarization, to assess its generalizability.
- **Deeper Framework Analysis:** A more in-depth analysis of the L2T-DLN components could yield further improvements. This includes investigating the impact of varying the teacher model's memory size and sampling strategies on training dynamics or exploring more advanced architectures for the teacher, such as an LSTM, to better model the student's learning history.
- **Comparative Studies:** To better position this work within the current landscape, the L2T-Hyena model should be benchmarked against other state-of-the-art language models, including other advanced SSMs (e.g., Mamba) and the latest Transformer variants.

REFERENCES

- [1] M. Shao, A. Basit, R. Karri and M. Shafique, "Survey of Different Large Language Model Architectures: Trends, Benchmarks, and Challenges," in *IEEE Access*, vol. 12, pp. 188664-188706, doi: 10.1109/ACCESS.2024.3482107, 2024.
- [2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proc. NAACL-HLT*, 2019.
- [3] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, and D. Amodei, "Language Models are Few-Shot Learners," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020.
- [4] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, and P. J. Liu, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," *J. Mach. Learn. Res.*, vol. 21, pp. 1–67, 2020.
- [5] S. Minaee, T. Mikolov, N. Nikzad, M. Asgari Chenaghlu, R. Socher, X. Amatriain, and J. Gao, "Large Language Models: A Survey," *arXiv preprint arXiv:2402.06196*, 2024.
- [6] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Barnes, and A. S. Mian, "A Comprehensive Overview of Large Language Models," *arXiv preprint arXiv:2307.06435*, 2023.
- [7] S. B. P. R. P, S. M. B and K. S, "The Evolution of Large Language Model: Models, Applications and Challenges," *International Conference on Current Trends in Advanced Computing (ICCTAC)*, Bengaluru, India, pp. 1-8, doi: 10.1109/ICCTAC61556.2024.10581180, 2024.
- [8] M. A. K. Raiaan, M. S. H. Mukta, K. Fatema, N. M. Fahad, S. Sakib, M. M. J. Mim, J. Ahmad, M. E. Ali, and S. Azam, "A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges," *IEEE Access*, vol. 12, pp. 26839–26874, 2024.
- [9] X. Wang, S. Wang, Y. Ding, Y. Li, W. Wu, Y. Rong, W. Kong, J. Huang, S. Li, H. Yang, Z. Wang, B. Jiang, C. Li, Y. Wang, Y. Tian, and J. Tang, "State Space Model for New-Generation Network Alternative to Transformers: A Survey," *arXiv preprint arXiv:2404.09516*, 2024.
- [10] M. Poli, S. Massaroli, E. Nguyen, D. Y. Fu, T. Dao, S. Baccus, Y. Bengio, S. Ermon, and C. Ré, "Hyena Hierarchy: Towards Larger Convolutional Language Models," in *Proceedings of the 40th International Conference on Machine Learning (ICML'23)*, vol. 202, pp. 28043–28078, 2023.
- [11] T. R. Ralambomihanta, S. Mohammadzadeh, M. S. N. Islam, W. Jabbour, and L. Liang, "Scavenging Hyena: Distilling Transformers into Long Convolution Models," *arXiv preprint arXiv:2401.17574*, 2024.

- [12] Z. Hai, L. Pan, X. Liu, Z. Liu, and M. Yunita, "L2T-DLN: Learning to teach with dynamic loss network," in *Proceedings of the 37th International Conference on Neural Information Processing Systems (NeurIPS '23)*, Red Hook, NY, USA, 2023.
- [13] L. Wu, F. Tian, Y. Xia, Y. Fan, T. Qin, J. Lai, and T.-Y. Liu, "Learning to Teach with Dynamic Loss Functions," *Neural Information Processing Systems*. arXiv:1810.12081, 2018.
- [14] C. A. Alonso, J. Sieber, and M. N. Zeilinger, "State Space Models as Foundation Models: A Control Theoretic Overview," *arXiv preprint arXiv:2403.16899*, 2024.
- [15] A. Gu, K. Goel, and C. Ré, "Efficiently Modeling Long Sequences with Structured State Spaces," *arXiv preprint arXiv:2111.00396*, 2021.
- [16] P. Baldi, "Autoencoders, Unsupervised Learning, and Deep Architectures," *JMLR: Workshop and Conference Proceedings*, vol. 27, pp. 37–50, 2012.
- [17] B. N. Patro and V. S. Agneeswaran, "Mamba-360: Survey of State Space Models as Transformer Alternative for Long Sequence Modelling: Methods, Applications, and Challenges," *arXiv preprint arXiv:2404.16112*, 2024.
- [18] T. Kudo, "Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates," in *Proc. Assoc. Comput. Linguistics (ACL)*, pp. 66–75, 2018.
- [19] V. Nair and G. E. Hinton, "Rectified Linear Units improve Restricted Boltzmann Machines," in *Proc. Int. Conf. Mach. Learn. (ICML)*, pp. 807–814, 2010.
- [20] R. Sennrich, B. Haddow, and A. Birch, "Neural Machine Translation of Rare Words with Subword Units," in *Proc. Assoc. Comput. Linguistics (ACL)*, pp. 1715–1725, 2016.
- [21] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2014.
- [22] D. Hendrycks and K. Gimpel, "Gaussian Error Linear Units (GELUs)," *arXiv preprint arXiv:1606.08415*, 2016.
- [23] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," *arXiv preprint arXiv:1607.06450*, 2016.

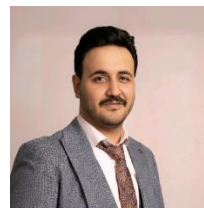
- [24] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.



FATEMEH SOHBATI is currently pursuing the M.Sc. degree in communication systems from the Iran University of Science and Technology, Tehran, Iran. Her research interests include language models and deep learning.



FARZAN HADDADI received the Ph.D. degree in communication systems from Sharif University of Technology, Tehran, Iran. He is currently a University Professor with the School of Electrical Engineering/Department of Communication Systems, Iran University of Science and Technology, Tehran, Iran. His research interests include deep learning, neural networks, and artificial intelligence.



HAMID SALAHINEJAD is currently pursuing the M.Sc. degree in secure computing from Tarbiat Modares University, Tehran, Iran. His research interests include language models and the Ethereum network.