Optimal Control for Transformer Architectures: Enhancing Generalization, Robustness and Efficiency

Neural Information Processing Systems (NeurIPS) 2025

Paper: https://arxiv.org/pdf/2505.13499
Code: https://github.com/KelvinKan/OT-Transformer

Kelvin Kan[1]

Xingjian Li[2]

Benjamin Zhang[3]

Tuhin Sahai[4]

Stanley Osher[1]

Markos Katsoulakis[5]

1 UCLA

2 TEXAS The University of Texas at Austin

3 THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

4 SRI

5 UMASS

Overview

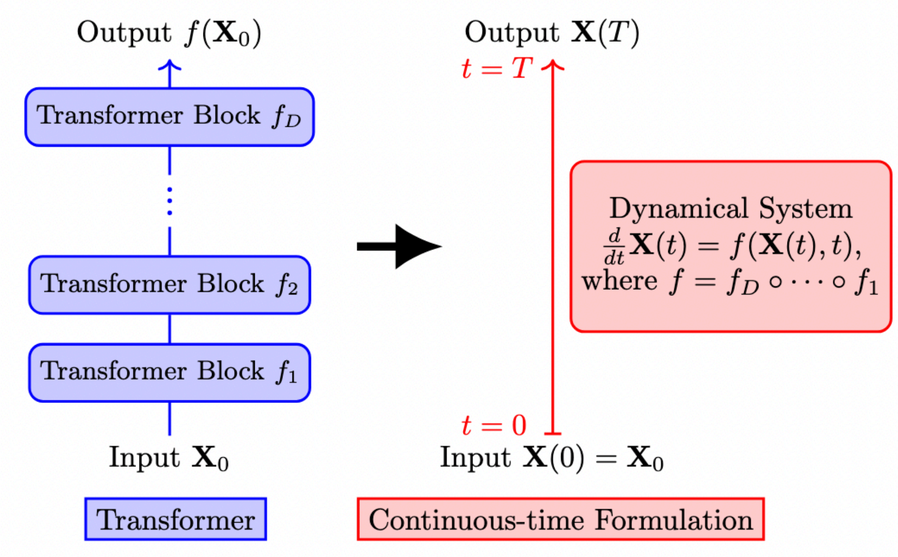
- **Transformers**
  - State-of-the-art in various applications
  - Effective architectures often discovered by empirical **trial and error**
- **Our Framework**
  - Analyze Transformers' training and architecture using **optimal control theory**
  - Propose our model: OT-Transformer
  - Achieve **theory-grounded improvements**

OT-Transformer Model

- Use an existing Transformer for f:

$$\frac{d\mathbf{X}(t)}{dt} = f(\mathbf{X}(t), t; \theta), \quad \text{for} \quad t \in [0, T], \quad \text{with} \quad \mathbf{X}(0) = \mathbf{X}_0 \quad \textbf{(hidden state dynamics)}$$

- **Leverage expressive power & retain (task-specific) benefits of Transformer variants**
- **Plug-and-play:** only requires slight code modification



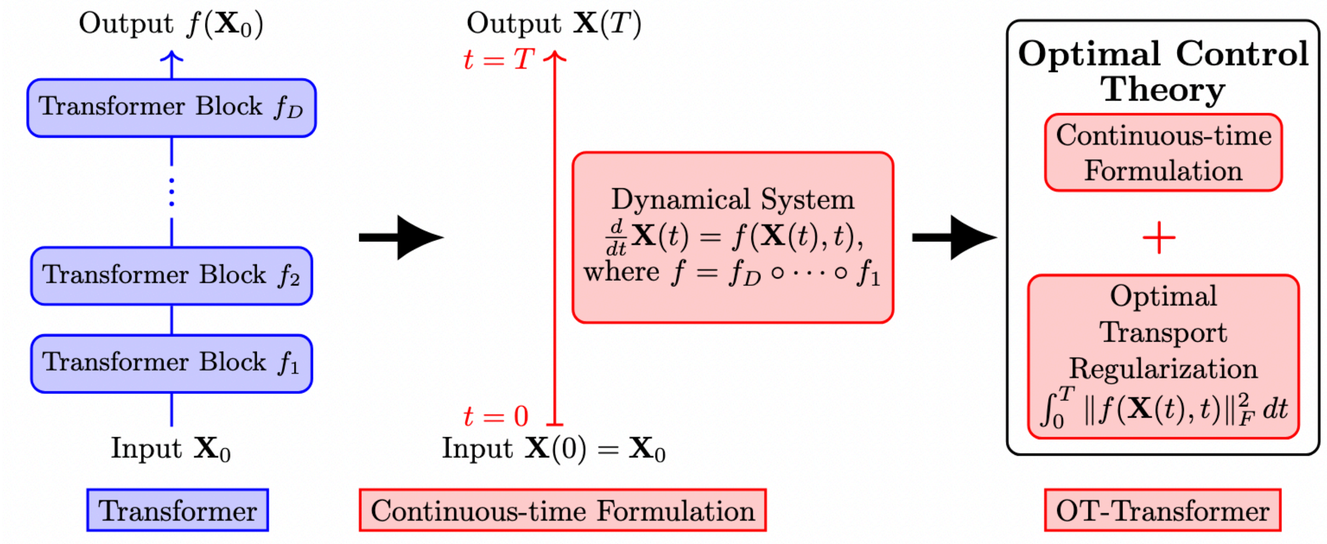
OT-Transformer Model

- Training formulation

$$\min_{\theta,\gamma} \; \mathbb{E}_{(\mathbf{X}_0,\mathbf{y})}\left\{ G(\mathbf{X}(T),\mathbf{y};\gamma) + \frac{\lambda}{2}\int_0^T \left\| f(\mathbf{X}(t),t;\theta) \right\|_F^2 dt \right\},$$

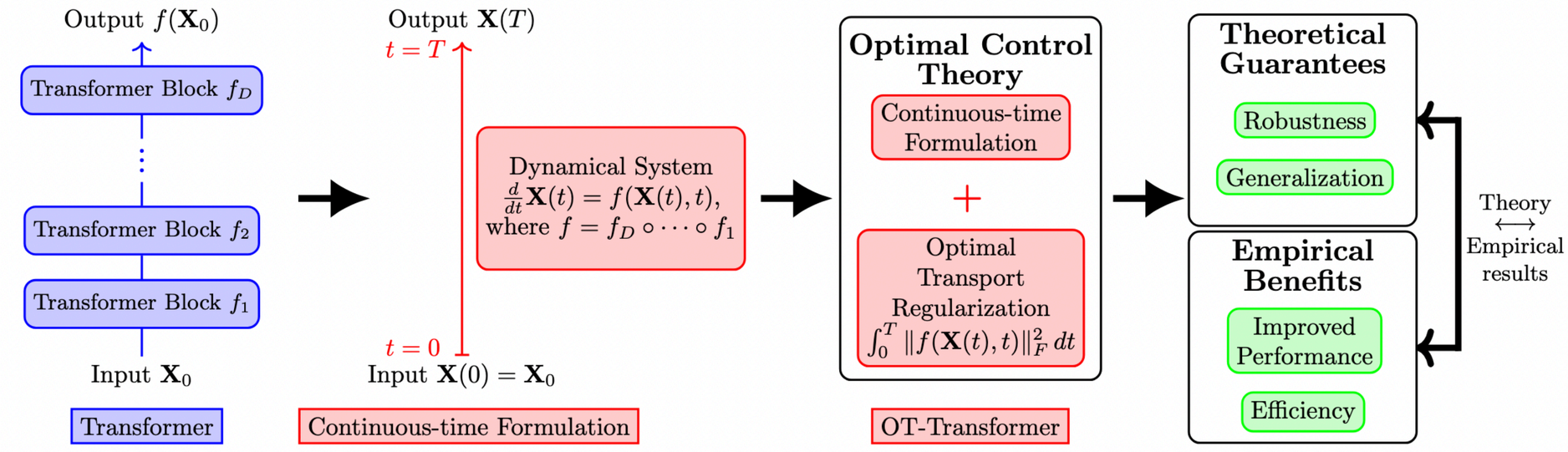
$$\text{subject to} \quad \frac{d\mathbf{X}(t)}{dt} = f(\mathbf{X}(t),t;\theta), \quad \text{for} \quad t \in [0,T], \quad \text{with} \quad \mathbf{X}(0) = \mathbf{X}_0 \quad \textbf{(hidden state dynamics)}$$

- First term G: data-fitting loss; Second term: OT regularization
- Other regularizations can be used



OT-Transformer Model

- OC theory ⇒ models with highly favorable properties
- Theory ⇔ empirical results



Main Theoretical Result

- **Stable forward propagation**

  Theorem 1. For input-label pairs $(\mathbf{X}_1(0), \mathbf{y}_1)$ and $(\mathbf{X}_2(0), \mathbf{y}_2)$, the corresponding model outputs $\tilde{\mathbf{y}}_1$ and $\tilde{\mathbf{y}}_2$ satisfy

$$\underbrace{\|\tilde{\mathbf{y}}_1 - \tilde{\mathbf{y}}_2\|_2}_{\text{model outputs}} \leq C_1 \underbrace{\|\mathbf{X}_1(0) - \mathbf{X}_2(0)\|_F}_{\text{Inputs}} + C_2 \underbrace{\|\mathbf{y}_1 - \mathbf{y}_2\|_2}_{\text{True labels}}$$

- If inputs & true labels are similar, then model outputs are similar
- ⇒ **Robustness**
  - **Robustness to Input Perturbations**
  - **Distributional Robustness**
- ⇒ **Generalization**
  - **In-distribution Generalization**
  - **Out-of-distribution Generalization**

Experimental Results

- Consistently **improves performance** while **enhancing parameter efficiency**
- Experimental results ⇔ theory on generalization and robustness
- **Implementation efficiency**: theory-driven improvements, instead of trial and error

Experiment	Method	Para. Count	Test Loss
nanoGPT on Shakespeare (Char.-level)	Baseline	10.65M	2.68 ± 0.006
	OT-Trans. (Ours)	6.16M	$\mathbf{1.44 \pm 0.005}$
GPT-2 on Shakespeare (Word-level)	Baseline	123.7M	5.18 ± 0.032
	OT-Trans. (Ours)	123.7M	$\mathbf{4.96 \pm 0.012}$
GPT-2 on OpenWebText (9B tokens)	Baseline	123.7M	3.21
	OT-Trans. (Ours)	123.7M	**3.03**

Experiment	Method	Para. Count	Test Accuracy
Point Cloud Classification	Baseline	0.86M	$87.4\% \pm 0.45\%$
	OT-Trans. (Ours)	0.65M	$\mathbf{89.9\% \pm 0.42\%}$
Image Classification (MNIST)	Baseline	93K	$93.0\% \pm 0.69\%$
	OT-Trans. (Ours)	18K	$\mathbf{97.1\% \pm 0.16\%}$
Image Classification (Cats & Dogs)	Baseline	1.77M	$77.6\% \pm 0.86\%$
	OT-Trans. (Ours)	1.48M	$\mathbf{79.0\% \pm 0.31\%}$
Sentiment Analysis	Baseline	4.74M	$83.9\% \pm 0.26\%$
	OT-Trans. (Ours)	2.37M	$\mathbf{84.6\% \pm 0.55\%}$

Theory: Empirical Validation

- Added typical noise per application
- Noise absent in training data
- Tests on **robustness** and **out-of-distribution generalization**
- Much better under high noise

Experiment 1: Point cloud classification with point dropout (test accuracy)

Experiment	Metric/drop rate	0.0	0.01	0.05	0.1	0.2	0.5
Point cloud (dropout)	Acc. (±std) Base.	86.6%±0.45%	86.6%±0.48%	85.8%±0.60%	84.3%±0.69%	76.9%±0.88%	34.5%±1.94%
	Acc. (±std) Ours	**89.3%±0.55%**	**89.3%±0.55%**	**88.8%±0.34%**	**87.6%±0.62%**	**83.9%±0.80%**	**55.4%±4.87%**
	Drop (↓) Base.	–	**0.0%**	0.8%	2.3%	9.7%	52.1%
	Drop (↓) Ours	–	**0.0%**	**0.5%**	**1.7%**	**5.4%**	**33.9%**

Experiment 2: NanoGPT with random text replacement (test loss)

Experiment	Metric/replace rate	0.0	0.005	0.01	0.05	0.1
NanoGPT (text replace)	Loss (±std) Base.	2.68±0.004	2.78±0.004	2.88±0.004	3.65±0.004	4.60±0.005
	Loss (±std) Ours	**1.44±0.005**	**1.49±0.004**	**1.55±0.003**	**1.95±0.010**	**2.42±0.022**
	Drop (↓) Base.	–	0.10	0.20	0.97	1.92
	Drop (↓) Ours	–	**0.05**	**0.11**	**0.51**	**0.98**

Experiments 3 & 4: MNIST with Gaussian/uniform noise (test accuracy)

Experiment	Metric/noise level	0.0	0.01	0.05	0.1	0.2	0.5
MNIST (Gauss. noise)	Acc. (±std) Base.	92.97%±0.67%	92.99%±0.66%	92.96%±0.69%	92.76%±0.72%	91.70%±0.68%	80.64%±1.58%
	Acc. (±std) Ours	**97.05%±0.15%**	**97.05%±0.16%**	**96.99%±0.18%**	**96.89%±0.15%**	**96.39%±0.11%**	**90.10%±1.10%**
	Drop (↓) Base.	–	**-0.02%**	**0.01%**	0.21%	1.27%	12.33%
	Drop (↓) Ours	–	0.00%	0.06%	**0.16%**	**0.66%**	**6.95%**
MNIST (Uni. noise)	Acc. (±std) Base.	92.97%±0.67%	92.99%±0.65%	92.98%±0.63%	92.90%±0.58%	92.64%±0.57%	90.02%±0.45%
	Acc. (±std) Ours	**97.05%±0.15%**	**97.03%±0.14%**	**97.00%±0.15%**	**96.97%±0.14%**	**96.79%±0.16%**	**95.57%±0.11%**
	Drop (↓) Base.	–	**0.02%**	**-0.02%**	**0.07%**	0.33%	2.95%
	Drop (↓) Ours	–	**0.02%**	0.05%	0.08%	**0.26%**	**1.48%**

Discussion & Summary

- OC framework for Transformer architecture and training
- OT-Transformer: Plug-and-play model grounded in theoretical guarantees
- Empirical results ⇔ theory
- Future directions/Ongoing work:
  - OC framework to analyze other components
  - e.g., layer normalization, attention mechanism, other regularizers