A Theory on How LLMs Make Decisions

Shangtong Zhang

Assistant Professor, Department of Computer Science, University of Virginia

LLMs can maximize rewards during inference

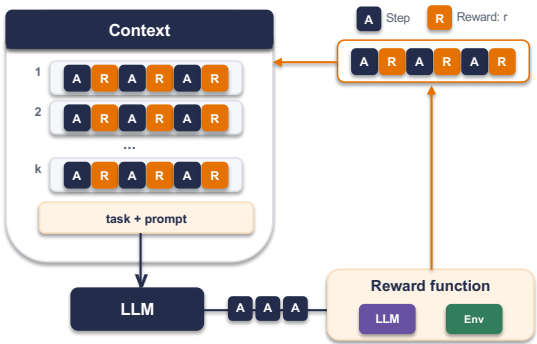
just like a reinforcement learning agent.

Game of 24

4	4	6	8

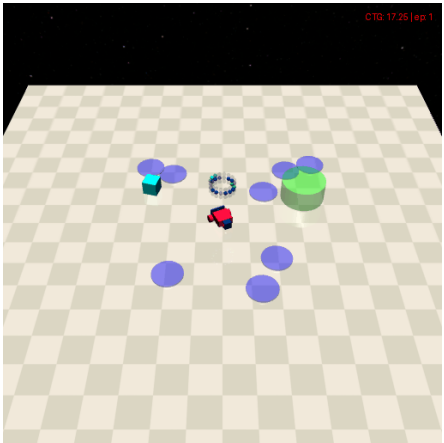
Prompt LLMs with Chain of Thought

Step 1	4 + 8 = 12	left: 4, 6, 12
Step 2	6 - 4 = 2	left: 2, 12
Step 3	2 * 12 = 24	left: 24
Answer	(6 - 4) * (4 + 8) = 24	



Transformers can maximize rewards and minimize costs simultaneously

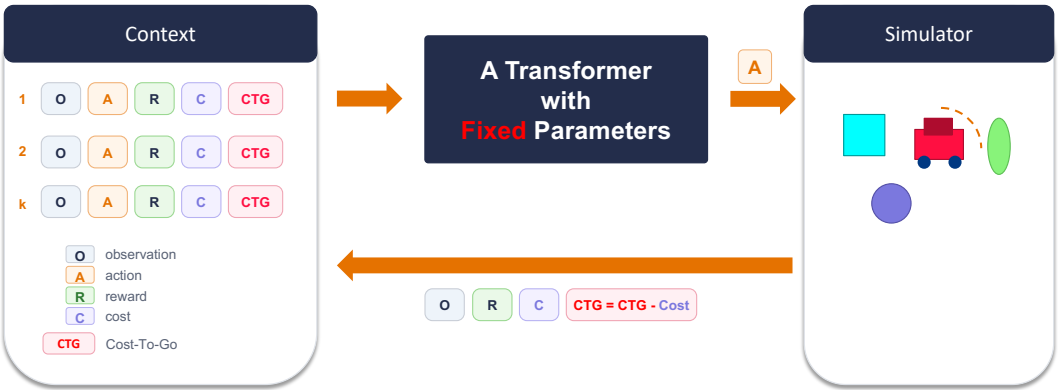
also during inference.



- The **car** must reach the **green cylinder goal**.
- The **light blue cube** and **dark circle** are obstacles.
- The **car** receives only local observations.

Transformers can maximize rewards and minimize costs simultaneously

also during inference.



“

Transformer’s forward pass implements some RL algorithm.

— An instance of the iterative inference hypothesis (Jastrzębski et al., 2017)

Layer-by-layer forward pass can be viewed as step-by-step update

of an iterative algorithm.

- Z_0: input tokens to a transformer
- $Z_{l+1} = Z_l + \text{Attention}_\theta(Z_l)$: the update rule for each layer
- $v_{l+1} = v_l + \text{TD-Error}(v_l)$: what an RL algorithm looks like

Transformers can implement temporal difference learning

in the forward pass.

What is TD?

- $A_t \sim \pi(\cdot|S_t), R_{t+1} = r(S_t, A_t), S_{t+1} \sim p(\cdot|S_t, A_t)$
- $v_\pi(s) = \mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i R_{t+i+1} \mid S_t = s\right]$
- $\phi : \mathcal{S} \to \mathbb{R}^d$: state feature map
- Goal of TD: find w such that $\phi(s)^\top w \approx v_\pi(s)$ for all s
- TD algorithm: $w_{t+1} = w_t + \alpha_t(R_{t+1} + \gamma w_t^\top \phi(S_{t+1}) - w_t^\top \phi(S_t))\phi(S_t)$

Transformers can implement temporal difference learning

in the forward pass.

With certain attention parameters θ_* and Z_0

$$Z_{l+1} = Z_l + \text{LinAttention}_{\theta_*}(Z_l)$$

With $w_0 = 0$

$$w_{l+1} = \text{BatchTD}(w_l)$$

$$Z_l[-1, -1] = w_l^\top \phi(s) \quad \forall l$$

Linear attention replaces softmax with identity

- $\text{Attention}_\theta(Z) = \text{softmax}\big((ZW_Q)(ZW_K)^\top\big)ZW_V$
- $\text{LinAttention}_\theta(Z) = ZW_Q(ZW_K)^\top ZW_V$

BatchTD applies TD in a fixed dataset

- A dataset $\mathcal{D} = \{(s_i, r_i, s_{i+1})\}_{i=1}^{N}$ of transitions sampled from (π, p)
- $w_{l+1} = w_l + \frac{\alpha_l}{N} \sum_{i=1}^{N} (r_i + \gamma w_l^\top \phi(s_{i+1}) - w_l^\top \phi(s_i)) \phi(s_i)$

Transformers can implement temporal difference learning

in the forward pass.

Task: given $\mathcal{D}$ and a state s, estimate $v_\pi(s)$

With certain attention parameters θ_*

$$Z_0 = \begin{bmatrix} \phi(s_1) & \dots & \phi(s_N) & \phi(s) \\ \gamma\phi(s_2) & \dots & \gamma\phi(s_{N+1}) & 0 \\ r_1 & \dots & r_N & 0 \end{bmatrix}$$

$$Z_{l+1} = Z_l + \text{LinAttention}_{\theta_*}(Z_l)$$

With $w_0 = 0$

$$w_{l+1} = \text{BatchTD}(w_l)$$

$$Z_l[-1, -1] = w_l^\top \phi(s) \quad \forall l$$

“

Multi-task pretraining discovers θ_*, i.e., discovers the TD algorithm, in a few minutes.

— It took humans more than 20 years to invent TD.

Pretraining is to use DQN to train a transformer in multiple Atari games

... an analogy to our pretraining process.

- Sample a task (π, p, r)
- Sample a trajectory $S_0, A_0, R_1, S_1, \ldots$
- DQN-style update

$$\theta_{t+1} = \theta_t + \alpha_t(R_{t+1} + \gamma \mathrm{TF}_{\theta_t}(S_{t+1}) - \mathrm{TF}_{\theta_t}(S_t))\nabla_\theta \mathrm{TF}_{\theta_t}(S_t)$$

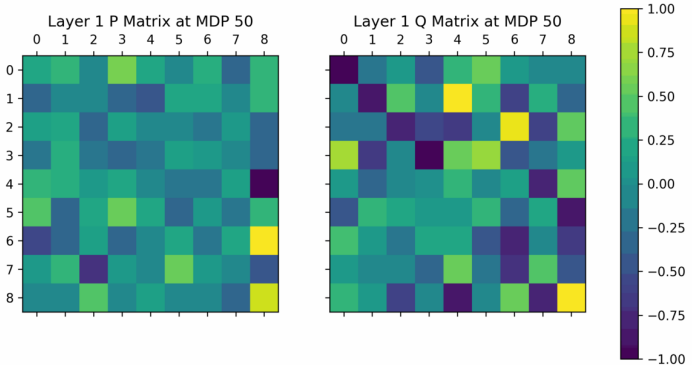
$$Z_0 \doteq \begin{bmatrix} \phi(S_0) & \ldots & \phi(S_{t-1}) & \phi(S_t) \\ \gamma\phi(S_1) & \ldots & \gamma\phi(S_t) & 0 \\ R_1 & \ldots & R_t & 0 \end{bmatrix} \xrightarrow{\theta} Z_1 \xrightarrow{\theta} \ldots \xrightarrow{\theta} \mathrm{TF}_\theta(S_t) \doteq Z_L[-1, -1]$$

Wang, J., Blaser, E., Daneshmand, H., and Zhang, S. (2025a). Transformers can learn temporal difference methods for in-context reinforcement learning. In *Proceedings of the International Conference on Learning Representations*

Transformer parameters converge to θ_* during pretraining

with random initialization

$$\text{LinAttention}_\theta(Z) = ZW_Q(ZW_K)^\top ZW_V = Z\left(W_Q W_K^\top\right)Z^\top ZW_V, \quad \theta = \left\{W_Q W_K^\top, W_V\right\}$$



Pretraining tasks can be very different from evaluation tasks

enabling maximal out-of-distribution generalization.

The θ_* minimizes the norm of the expected updates (NEU)

under some (arguably restrictive) assumptions

$$\theta_{t+1} = \theta_t + \alpha_t(R_{t+1} + \gamma \mathrm{TF}_{\theta_t}(S_{t+1}) - \mathrm{TF}_{\theta_t}(S_t))\nabla_\theta \mathrm{TF}_{\theta_t}(S_t)$$

Assume

- Enumeration of states in the context
- Expected next state features

$$\begin{bmatrix} \phi(S_0) & \dots & \phi(S_{t-1}) & \phi(S_t) \\ \gamma\phi(S_1) & \dots & \gamma\phi(S_t) & 0 \\ R_1 & \dots & R_t & 0 \end{bmatrix} \rightarrow \begin{bmatrix} \phi(s_1) & \dots & \phi(s_{|\mathcal{S}|}) & \phi(S_t) \\ \gamma \sum_{s'} p_\pi(s'|s_1)\phi(s_1) & \dots & \gamma \sum_{s'} p(s'|s_{|\mathcal{S}|})\phi(s_{|\mathcal{S}|}) & 0 \\ r_\pi(s_1) & \dots & r_\pi(s_{|\mathcal{S}|}) & 0 \end{bmatrix}$$

Then

$$\lim_{l \to \infty} \|\mathbb{E}[(R_{t+1} + \gamma \mathrm{TF}_{\theta_*}(S_{t+1}) - \mathrm{TF}_{\theta_*}(S_t))\nabla_\theta \mathrm{TF}_{\theta_t}(S_t)]\| = 0$$

A single-layer Transformer can also implement TD

if we use Chain-of-Thought prompting.

- CoT is to allow intermediate data
- Without CoT:

$$Z_L \leftarrow \mathrm{TF}_\theta(Z_0)$$

- With CoT:

$$\begin{aligned}
&Z_L \leftarrow \mathrm{TF}_\theta(Z_0) \xleftarrow{L=1} w_1 \\
&Z_0 \leftarrow [Z_0, Z_L] \\
&Z_L \leftarrow \mathrm{TF}_\theta(Z_0) \xleftarrow{L=1} w_2 \\
&Z_0 \leftarrow [Z_0, Z_L] \\
&\quad \ldots \\
&Z_L \leftarrow \mathrm{TF}_\theta(Z_0) \xleftarrow{L=1} w_k
\end{aligned}$$

Transformers with softmax attention implement a kernel TD

Given $(S_t, A_t, R_{t+1}, S_{t+1})$

- TD:

$$v(S_t) \leftarrow v(S_t) + \alpha(R_{t+1} + \gamma v(S_{t+1}) - v(S_t))$$

- Kernel TD:

$$\forall s, v(s) \leftarrow v(s) + \alpha(R_{t+1} + \gamma v(S_{t+1}) - v(S_t))\kappa(s, S_t)$$

“

LLMs (Transformers) make good decisions because they are computational machines

— not just statistical models that fit data.

Thanks & Questions

Collaborators

Alper Bozkurt
Amir Moeini
Claire Chen
Ethan Blaser
Hadi Daneshmand
Jiuqi Wang
Kefan Song
Lu Feng
Minjae Kwon
Rohan Chandra
Shuze Liu
Xinyu Liu
Yanjun Qi
Yuichi Motai
Zixuan Xie

Funding Agencies

National Science Foundation
Virginia's Commonwealth Cyber Initiative
Cisco Faculty Research Award
Nvidia Academic Grant
Google Research Award

Jastrzębski, S., Arpit, D., Ballas, N., Verma, V., Che, T., and Bengio, Y. (2017). Residual connections encourage iterative inference. *arXiv Preprint*.

Moeini, A., Kwon, M., Bozkurt, A. K., Motai, Y., Chandra, R., Feng, L., and Zhang, S. (2026). Safe in-context reinforcement learning. In *Proceedings of the International Conference on Machine Learning*.

Song, K., Moeini, A., Wang, P., Gong, L., Chandra, R., Zhang, S., and Qi, Y. (2026). Reward is enough: LLMs are in-context reinforcement learners. In *Proceedings of the International Conference on Learning Representations*.

Wang, J., Blaser, E., Daneshmand, H., and Zhang, S. (2025a). Transformers can learn temporal difference methods for in-context reinforcement learning. In *Proceedings of the International Conference on Learning Representations*.

Wang, J., Chandra, R., and Zhang, S. (2025b). Towards provable emergence of in-context reinforcement learning. In *Advances in Neural Information Processing Systems*.

Xie, Z., Liu, X., Chandra, R., and Zhang, S. (2026a). Convergence and emergence of in-context reinforcement learning with chain of thought. *arXiv Preprint*.

Xie, Z., Liu, X., Chen, C., Liu, S., Chandra, R., and Zhang, S. (2026b). Beyond linear attention: Softmax transformers implement in-context reinforcement learning. *arXiv Preprint*.