On the O(1/T) Convergence of Alternating Gradient Descent-Ascent in Bilinear Games

Tianlong Nan†, Shuvomoy Das Gupta¶, Garud Iyengar†, Christian Kroer†

†Columbia University, IEOR ¶Rice University, CMOR

ICLR 2026

Motivations

Superhuman Poker AI

Adversarial Training

- ✦ Alternation: a numerical trick that greatly improved performance
- ✦ Question: whether alternation *provably* helps performance

Preliminaries

✧ Bilinear saddle point problems (SPPs):

$$\min_{x\in\mathcal{X}}\max_{y\in\mathcal{Y}} y^\top Ax$$

- $\mathcal{X}\subseteq\mathbb{R}^n$ and $\mathcal{Y}\subseteq\mathbb{R}^m$ are compact convex sets
- A is an $n\times m$ matrix

✧ Bilinear two-player zero-sum games (matrix games):

$$\min_{x\in\Delta_n}\max_{y\in\Delta_m} y^\top Ax$$

- $\Delta_n=\{x\in\mathbb{R}^n_+ \mid \sum_{i=1}^n x_i=1\}$
- $\Delta_m=\{y\in\mathbb{R}^m_+ \mid \sum_{j=1}^m y_j=1\}$

✧ Nash equilibrium (NE): $(x^*,y^*)\in\Delta_n\times\Delta_m$ satisfying

$$y^\top Ax^*\le (y^*)^\top Ax^*\le (y^*)^\top Ax \quad \forall\, x\in\Delta_n, y\in\Delta_m$$

✤ Simultaneous Gradient Descent-Ascent (SimGDA)

$$x^{t+1}=\Pi_{\mathcal{X}}(x^t-\eta A^\top y^t)$$
$$y^{t+1}=\Pi_{\mathcal{Y}}(y^t+\eta Ax^t)$$

✤ Alternating Gradient Descent-Ascent (AltGDA)

$$x^{t+1}=\Pi_{\mathcal{X}}(x^t-\eta A^\top y^t)$$
$$y^{t+1}=\Pi_{\mathcal{Y}}(y^t+\eta Ax^{t+1})$$

Performance Estimation Programming (PEP)

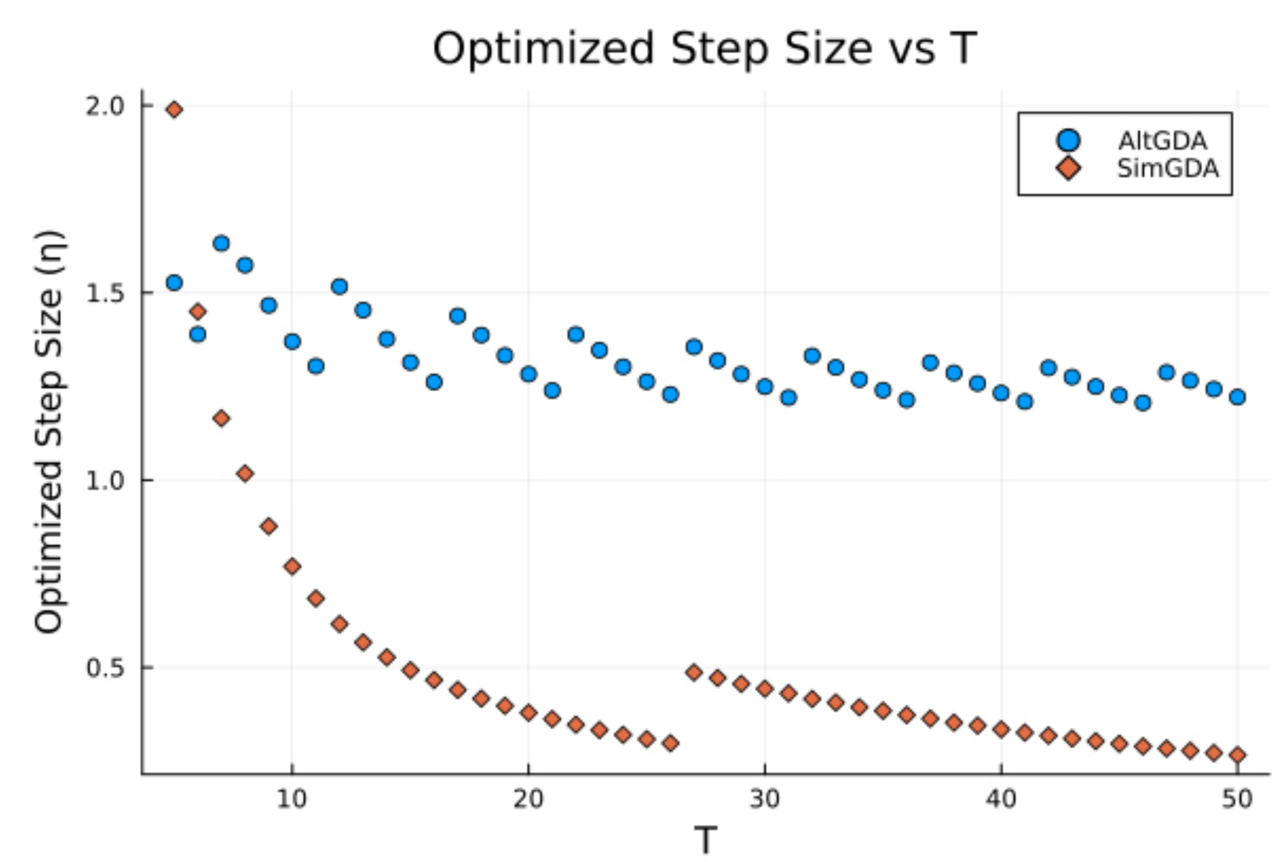
⊛ The worst-case performance measure over *all* bilinear saddle point problems can be computed by solving the following optimization problem, for any fixed iteration budget T and fixed step-size η

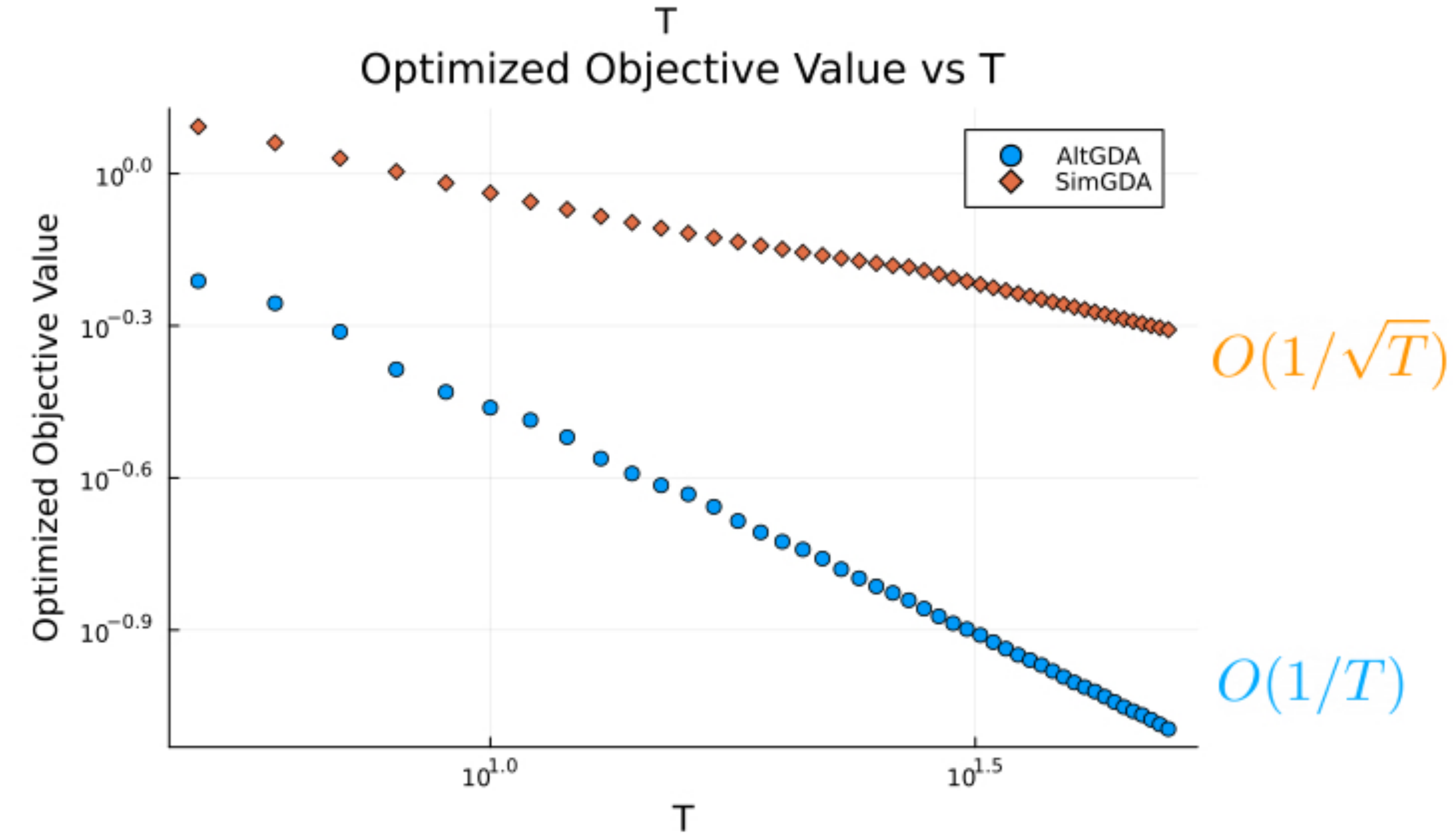
$$\mathcal{P}_T(\eta):=\left(\begin{array}{ll}\underset{x,y,\{(x^t,y^t)\}_{0\le t\le T},\mathcal{X},\mathcal{Y},A,m,n}{\text{maximize}} & \frac{1}{T}\sum_{t=1}^T\left(y^\top Ax^t-(y^t)^\top Ax\right)\\ \text{subject to} & \\ \quad \mathcal{X}\text{ is a convex compact set in }\mathbb{R}^n\text{ with radius }1, & \\ \quad \mathcal{Y}\text{ is a convex compact set in }\mathbb{R}^m\text{ with radius }1, & \\ \quad \sigma_{\max}(A)\le 1, & \\ \quad \{(x^t,y^t)\}_{1\le t\le T}=\texttt{AltGDA}(\eta,x^0,y^0), & \\ \quad (x^0,y^0),(x,y)\in\mathcal{X}\times\mathcal{Y}. & \end{array}\right)$$

⊛ This program can be reduced to a finite-dimensional SDP

⊛ For a fixed T, the best convergence rate of AltGDA can be found by solving

$$\mathcal{P}_T^*=\text{minimize}_\eta\mathcal{P}_T(\eta)$$





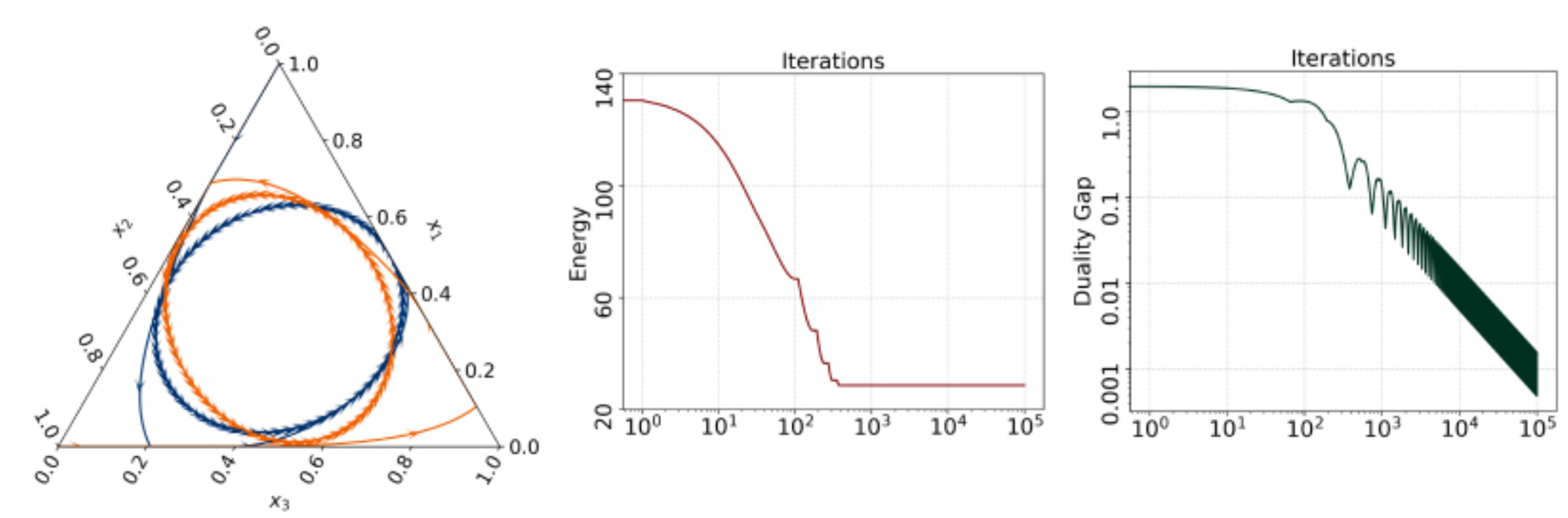
O(1/T) Convergence Rates

◈ We prove that AltGDA *can* achieve an O(1/T) convergence rate in bilinear games: if the game admits an interior Nash equilibrium, the duality gap of the averaged iterates of AltGDA with a constant step-size independent of T converges at an O(1/T) rate

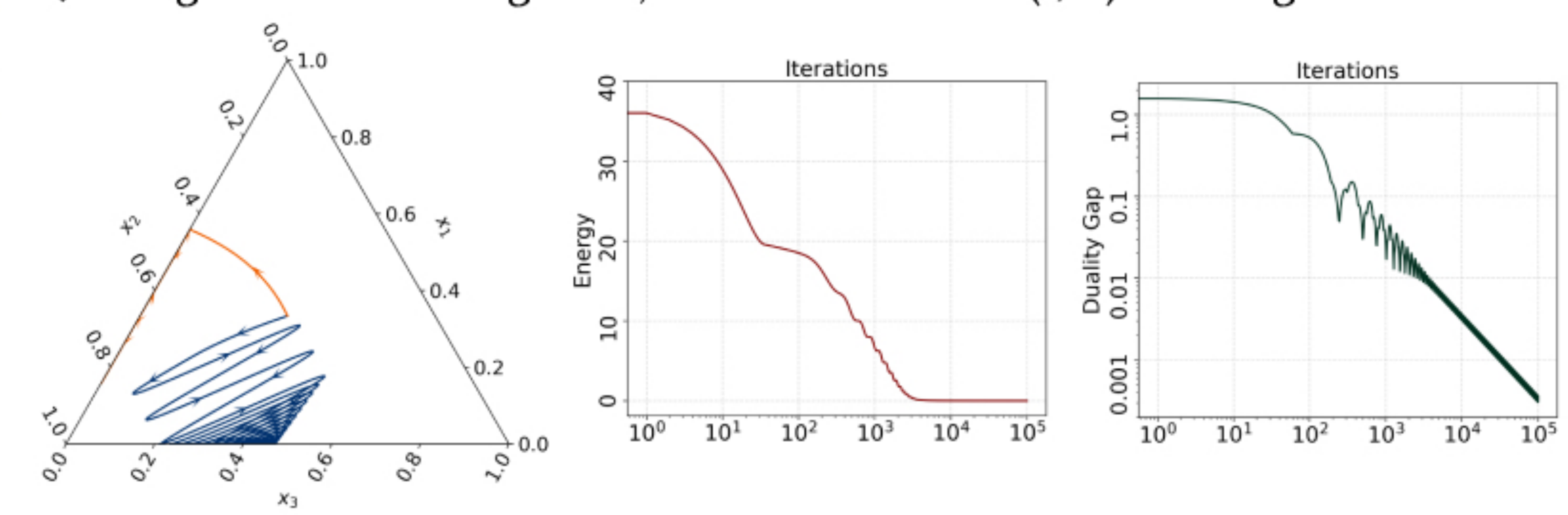
◈ The proof leverages the following observations and connections in AltGDA's two-phase behaviors

Summary for bilinear games with an interior NE

Phase 1	Phase 2
Trajectory collides with the boundary	Trajectory is in the interior
Energy function decays	Energy function remains constant
Duality gap decreases slowly	Duality gap decreases with O(1/T) rate



◈ For general bilinear games, we show a local O(1/T) convergence rate



Numerical Performances

❆ Numerical performance comparisons between SimGDA and AltGDA on larger instances: 60 * 120 matrix games with various value distributions