The Journal of Open Source Software

MFGLib: A Library for Mean-Field Games

Xin Guo[1], Anran Hu[3], Matteo Santamaria[1], Mahan Tajrobehkar[1], and Junzi Zhang[4]

1 University of California, Berkeley **3** Columbia University **4** Citadel Securities

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 04 January 2026
Published: unpublished

Summary

Mean-field games (MFGs) provide scalable models for large-population strategic interactions. They approximate an N-player game by analyzing the limiting regime as $N \to \infty$, replacing explicit multi-agent interactions with the interaction between a representative agent and the population distribution (Lasry & Lions, 2007). It has been shown that the Nash equilibrium policy of the mean-field game is an ϵ-Nash equilibrium of the N-player game with $\epsilon = O(1/\sqrt{N})$ (Huang et al., 2006) and in practice even games with small N on the order of tens can be well-approximated by MFGs (Cabannes et al., 2021; Guo et al., 2019; Kizilkale et al., 2019). Due to their tractability, MFG models have become widely used in applications such as digital advertising, high-frequency trading, dynamic pricing, transportation, and behavioral modeling.

Despite the rapid growth of the MFG literature, however, researchers and practitioners lack a unified, open-source software package for defining and solving their own MFG problems.

MFGLib is an open-source Python library that addresses this gap by providing:

- A modular and extensible API for defining arbitrary discrete-time finite-state MFGs
- Implementations of state-of-the-art algorithms for computing (approximate) Nash equilibria
- A collection of customizable benchmark environments drawn from the literature
- Tight integration with Optuna (Akiba et al., 2019) to provide automatic hyperparameter selection
- Clear documentation and examples, facilitating both research and industry use

The library is implemented in Python, maintained on GitHub, and can be installed via `pip install mfglib`. Full documentation, tutorials, and example notebooks are available at https://mfglib.readthedocs.io/en/latest/.

Statement of Need

Large-population games, where a massive number of agents interact, are ubiquitous in fields such as economic modeling, crowd dynamics, and smart-grid management. However, as the number of agents N increases, traditional game-theoretic methods face exponential computational growth. MFGs provide a tractable approximation by modeling the collective behavior of agents in the infinite-population limit ($N \to \infty$). Despite the mathematical maturity of MFG theory, the research ecosystem lacks a standardized software framework. This has led to a fragmented landscape where researchers must frequently re-implement population dynamics and equilibrium solvers from scratch for every publication, a process that is time-intensive and prone to implementation errors.

MFGLib is designed to provide a unified, modular foundation for MFG research. By offering a standardized API that decouples the environment definition from the equilibrium solver, the library lowers the barrier to entry for new researchers, enables rapid experimentation, and offers practitioners a way to prototype MFG-based models without requiring deep expertise in game theory or optimal control.

MFGLib is for both researchers and practitioners from a broad range of backgrounds working on large-population strategic interactions who need a standardized, extensible toolkit to model, solve, and experiment with mean-field games.

State of the Field

The current landscape of game-theoretic software is bifurcated between general-purpose N-player frameworks and specialized research scripts. N-player libraries like QuantEcon (Batista et al., 2024), Nashpy (Knight & Campbell, 2018), and ilqgames (Fridovich-Keil et al., 2020) are restricted to small N and lack the mean-field approximations necessary to handle the complexity of large-scale games. Conversely, MFG-specific repositories such as gmfg-learning (Cui & Koeppl, 2022) and entropic-mfg (Benamou et al., 2019) are typically "static" artifacts designed for single papers; they lack the unit testing, extensible interfaces, and documentation required for broader community adoption. Among the very few existing MFG libraries, OpenSpiel (Lanctot et al., 2019), a collection of environments and algorithms for research in reinforcement learning and planning in games, is the closest one to MFGLib. OpenSpiel has dedicated a module to MFGs implementing several environments and algorithms. However, it lacks customizability and a user-friendly API. In fact, according to its documentation, their code is still experimental and is only recommended for internal use.

Software Design

User-Friendly Environment Creation

Users can define custom MFG environments by providing reward functions, transition functions, and basic problem parameters (time horizon, state/action space sizes, initial distribution). The reward and transition functions are simple callables that map time and population distribution to tensors, allowing users to create environments with minimal code while maintaining mathematical clarity.

Pre-Implemented Algorithms

MFGLib implements several widely used algorithms, including **Online Mirror Descent** (Perolat et al., 2021), **Fictitious Play** (Perrin et al., 2020), **MFOMO** (Guo et al., 2024), **MFOMI** (Hu & Zhang, 2025), and **Prior Descent** (Cui & Koeppl, 2021). These algorithms encompass many other existing methods as special cases, such as fixed point iteration and **GMF-V** (Guo et al., 2019). The unified solver interface returns policy iterates, exploitability scores (which evaluate closeness to Nash equilibrium), and cumulative runtimes, with optional real-time logging to monitor convergence.

Automatic Hyperparameter Tuning

Every algorithm requires hyperparameters that can drastically influence convergence properties. MFGLib provides a built-in tuner based on Optuna (Akiba et al., 2019) to automatically select optimal hyperparameters. The tuner can optimize across single instances or environment suites with multiple policy initializations and customizable metrics (e.g., shifted geometric mean of exploitability). Users can also implement their own metrics with minimal effort.

High-Dimensional Representation

MFGLib uses PyTorch tensors to represent policies, mean-fields, and rewards while preserving the original structure of state and action spaces. Rather than flattening high-dimensional spaces into one-dimensional representations, the library maintains their natural structure, providing higher interpretability and more flexible user interactions.

Example

We demonstrate the usage of our library with a brief example: a mean-field variant of Rock-Paper-Scissors introduced by Cui & Koeppl (2021). Each agent chooses rock, paper or scissors, and receives a reward proportional to twice the number of beaten agents minus the number of agents that beat them.

Formally, the state and action spaces are $\mathcal{S} = \{0, R, P, S\}$ and $\mathcal{A} = \mathcal{S} \setminus \{0\}$, respectively. The initial state distribution is fixed at $\mu_0(0) = 1$, and the game occurs over timesteps $\mathcal{T} = \{0, 1, \dots, T\}$. Agent rewards are specified by

$$\begin{aligned} r(R, a, \mu_t) &= 2 \cdot \mu_t(S) - 1 \cdot \mu_t(P) \\ r(P, a, \mu_t) &= 4 \cdot \mu_t(R) - 2 \cdot \mu_t(S) \\ r(S, a, \mu_t) &= 6 \cdot \mu_t(P) - 3 \cdot \mu_t(R) \end{aligned}$$

The transition function allows agents to pick their next state directly and independently of the population; that is, for all $s, s' \in \mathcal{S}$ and $a \in \mathcal{A}$,

$$\Pr(s_{t+1} = s' \mid s_t = s, a_t = a) = \mathbf{1}_{\{s'\}}(a)$$

We tune two algorithms – Online Mirror Descent (Perolat et al., 2021) and Occupation Measure Inclusion (Hu & Zhang, 2025) – on this environment.

```python
from mfglib.env import Environment
from mfglib.alg import OnlineMirrorDescent, OccupationMeasureInclusion
from mfglib.tuning import GeometricMean

solve_kwargs = {"atol": None, "rtol": None, "max_iter": 300}

env = Environment.rock_paper_scissors(T=20)

omd_orig = OnlineMirrorDescent()
omi_orig = OccupationMeasureInclusion()

# Compute exploitability traces for untuned algorithms
_, omd_expls_orig, _ = omd_orig.solve(env, **solve_kwargs)
_, omi_expls_orig, _ = omi_orig.solve(env, **solve_kwargs)

metric = GeometricMean(shift=0.1)

# Optimize algorithms over hyperparameters
N_TRIALS = 50
omd_study = omd_orig.tune(
    metric=metric, envs=[env], n_trials=N_TRIALS, solve_kwargs=solve_kwargs
)
omi_study = omi_orig.tune(
    metric=metric, envs=[env], n_trials=N_TRIALS, solve_kwargs=solve_kwargs
)

# Initialize new algorithms objects from the tuning results
omd_tuned = omd_orig.from_study(omd_study)
omi_tuned = omi_orig.from_study(omi_study)
```

```python
# Compute exploitability traces for tuned algorithms
_, omd_expls_tuned, _ = omd_tuned.solve(env, **solve_kwargs)
_, omi_expls_tuned, _ = omi_tuned.solve(env, **solve_kwargs)
```

Plotting the exploitability scores of the two algorithms, before and after tuning, we observe that tuning significantly improves performance by achieving faster exploitability reduction.

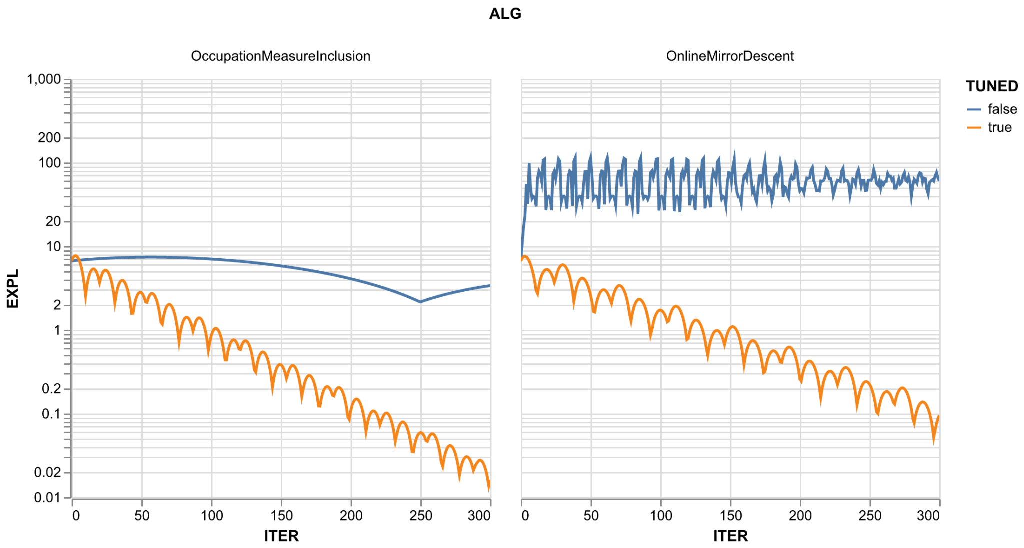

Figure 1: Exploitability curves before and after hyperparameter tuning

Research Impact Statement

The library has developed an active user community, including researchers and practitioners who have already used MFGLib in their work, contributed issues and pull requests on GitHub, and engaged with the tutorials and documentation. This activity demonstrates that the package is both useful to the community and actively maintained. Since its release, MFGLib has supported the development of several cutting-edge algorithms for MFGs, such as MF-OMO (Guo et al., 2024), MF-OMI (Hu & Zhang, 2025), and MESOB (Guo et al., 2023), as well as new models including MFG-MCDM (Becherer et al., 2025) and (α, β)-symmetric games (Yardim & He, 2024). It has also been used internally by Amazon Advertising for both research and production purposes. We believe it will continue to serve as an important building block for researchers in both academia and industry.

Acknowledgments

We thank the Amazon research teams for feedback and stress-testing the library in production settings. The authors would especially like to thank Sareh Nabi, Rabih Salhab, and Lihong Li of Amazon, Xiaoyang Liu of Columbia University, and Zhaoran Wang of Northwestern University for their valuable comments.

AI Usage Disclosure

The MFGLib library was developed without AI assistance; for the manuscript, Gemini 3 Flash was utilized for grammatical refinement and copy-editing. All AI-generated content was reviewed and edited for accuracy by the authors.

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2623–2631.

Batista, Q., Coleman, C., Furusawa, Y., Hu, S., Lunagariya, S., Lyon, S., McKay, M., Oyama, D., Sargent, T. J., Shi, Z., Stachurski, J., Winant, P., Watkins, N., Yang, Z., & Zhang, H. (2024). QuantEcon.py: A community based python library for quantitative economics. *Journal of Open Source Software*, *9*(93), 5585. https://doi.org/10.21105/joss.05585

Becherer, D., Reisinger, C., & Tam, J. (2025). *Mean-field games of speedy information access with observation costs*. https://arxiv.org/abs/2309.07877

Benamou, J.-D., Carlier, G., Di Marino, S., & Nenna, L. (2019). An entropy minimization approach to second-order variational mean-field games. *Mathematical Models and Methods in Applied Sciences*, *29*(08), 1553–1583. https://doi.org/10.1142/S0218202519500283

Cabannes, T., Lauriere, M., Perolat, J., Marinier, R., Girgin, S., Perrin, S., Pietquin, O., Bayen, A. M., Goubault, E., & Elie, R. (2021). Solving N-player dynamic routing games with congestion: A mean field approach. *arXiv Preprint arXiv:2110.11943*.

Cui, K., & Koeppl, H. (2021). Approximately solving mean field games via entropy-regularized deep reinforcement learning. *International Conference on Artificial Intelligence and Statistics*, 1909–1917.

Cui, K., & Koeppl, H. (2022). *Learning graphon mean field games and approximate nash equilibria*. https://arxiv.org/abs/2112.01280

Fridovich-Keil, D., Ratner, E., Peters, L., Dragan, A. D., & Tomlin, C. J. (2020). Efficient iterative linear-quadratic approximations for nonlinear multi-player general-sum differential games. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 1475–1481. https://doi.org/10.1109/ICRA40945.2020.9197129

Guo, X., Hu, A., Xu, R., & Zhang, J. (2019). Learning mean-field games. *Advances in Neural Information Processing Systems*, *32*.

Guo, X., Hu, A., & Zhang, J. (2024). MF-OMO: An optimization formulation of mean-field games. *SIAM Journal on Control and Optimization*, *62*(1), 243–270. https://doi.org/10.1137/22M1524084

Guo, X., Li, L., Nabi, S., Salhab, R., & Zhang, J. (2023). *MESOB: Balancing equilibria & social optimality*. https://arxiv.org/abs/2307.07911

Hu, A., & Zhang, J. (2025). MF-OML: Online mean-field reinforcement learning with occupation measures for large population games. *Appl. Math. Optim.*, *92*(3). https://doi.org/10.1007/s00245-025-10328-5

Huang, M., Malhamé, R. P., & Caines, P. E. (2006). Large population stochastic dynamic games: closed-loop McKean-Vlasov systems and the Nash certainty equivalence principle. *Communications in Information & Systems*, *6*(3), 221–252. https://doi.org/10.4310/CIS.2006.v6.n3.a5

Kizilkale, A. C., Salhab, R., & Malhamé, R. P. (2019). An integral control formulation of mean field game based large scale coordination of loads in smart grids. *Automatica*, *100*, 312–322. https://doi.org/10.1016/j.automatica.2018.11.029

Knight, V., & Campbell, J. (2018). Nashpy: A python library for the computation of nash equilibria. *Journal of Open Source Software*, *3*(30), 904. https://doi.org/10.21105/joss.00904

Lanctot, M., Lockhart, E., Lespiau, J.-B., Zambaldi, V., Upadhyay, S., Pérolat, J., Srinivasan, S., Finbarr Timbers, ', Tuyls, K., Omidshafiei, S., Hennes, D., Morrill, D., Muller, P., Ewalds, T., Faulkner, R., Kramár, J., Vylder, B. D., Saeta, B., Bradbury, J., … Ryan-Davis, J. (2019). OpenSpiel: A framework for reinforcement learning in games. *CoRR*, *abs/1908.09453*.http://arxiv.org/abs/1908.09453

Lasry, J.-M., & Lions, P.-L. (2007). Mean field games. *Japanese Journal of Mathematics*, *2*(1), 229–260.

Perolat, J., Perrin, S., Elie, R., Laurière, M., Piliouras, G., Geist, M., Tuyls, K., & Pietquin, O. (2021). Scaling up mean field games with online mirror descent. *arXiv Preprint arXiv:2103.00623*.

Perrin, S., Pérolat, J., Laurière, M., Geist, M., Elie, R., & Pietquin, O. (2020). Fictitious play for mean field games: Continuous time analysis and applications. *Advances in Neural Information Processing Systems*, *33*, 13199–13213.

Yardim, B., & He, N. (2024). *Exploiting approximate symmetry for efficient multi-agent reinforcement learning*.https://arxiv.org/abs/2408.15173