Simplifying RL optimization for everyone, from cognitive scientists to machine learning experts

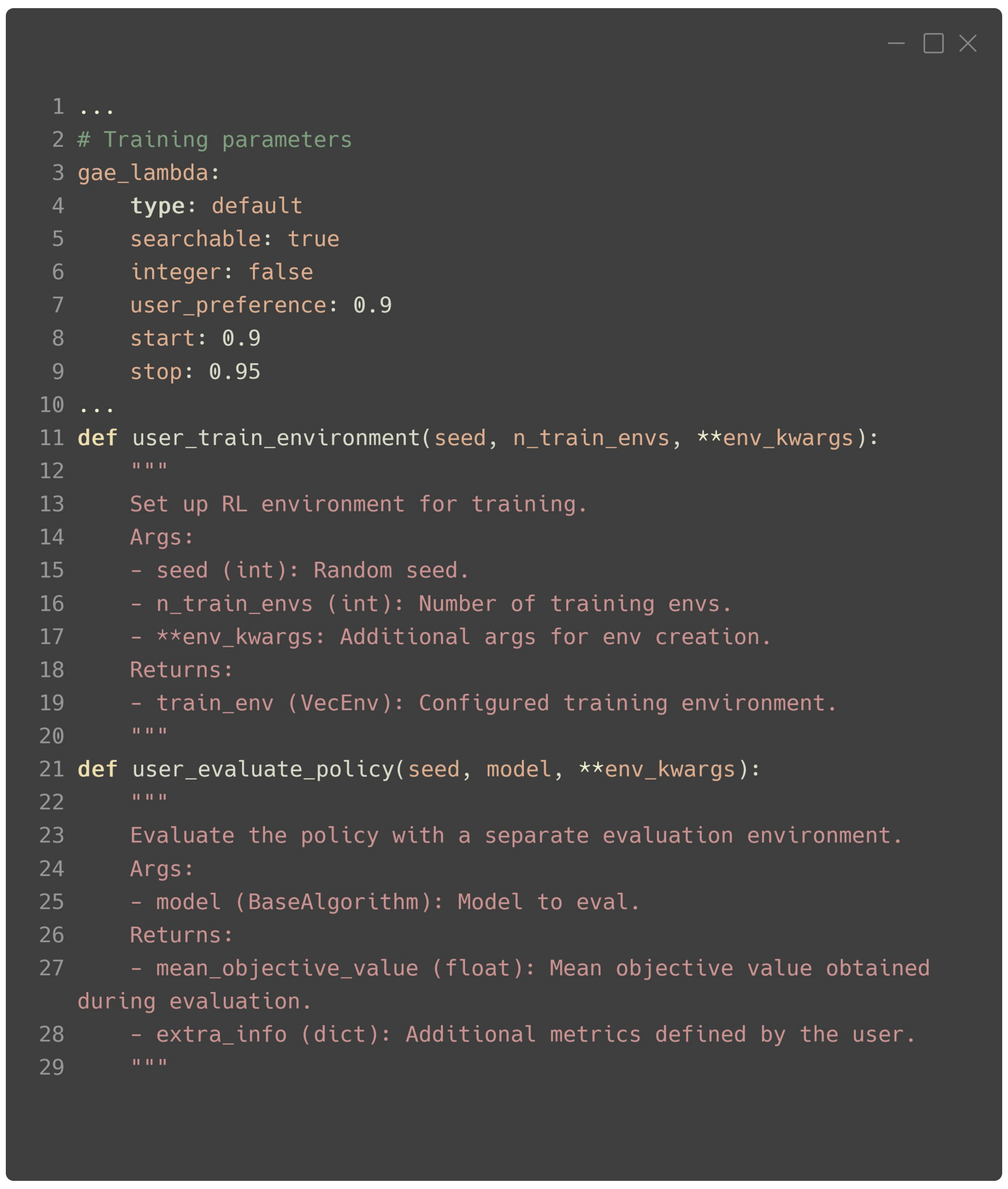
Aalto University
School of Engineering

AGENTFORGE: A Flexible Low-Code Platform for Reinforcement Learning Agent Design

Background:

- **Challenge:** RL agent design involves complex, interdependent parameters that are difficult for non-experts to optimize.
- **Solution:** A low-code platform simplifies parameter optimization and supports rapid iteration.
- **Target Users:** Cognitive scientists, behavioral researchers, and machine learning practitioners.

Input sample

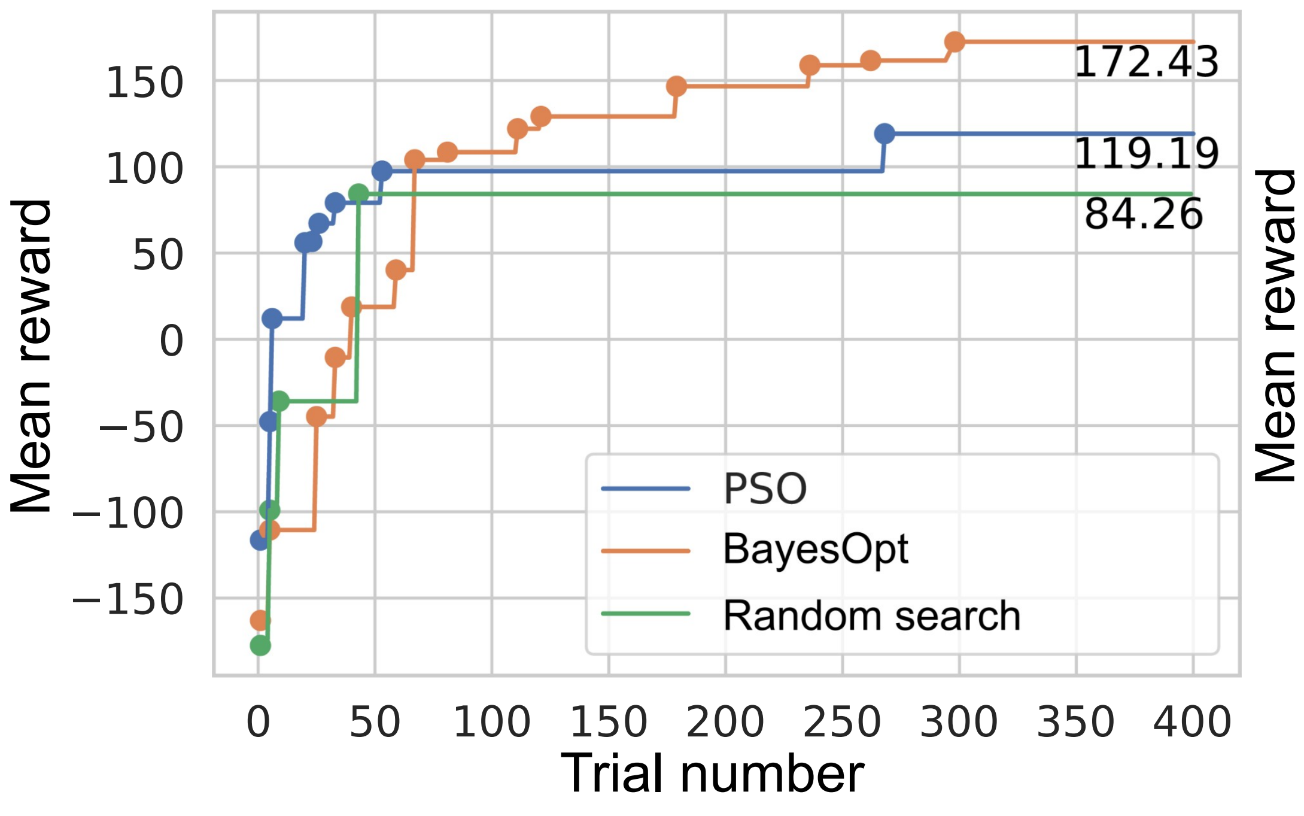


```
...
# Training parameters
gae_lambda:
    type: default
    searchable: true
    integer: false
    user_preference: 0.9
    start: 0.9
    stop: 0.95
...
def user_train_environment(seed, n_train_envs, **env_kwargs):
    """
    Set up RL environment for training.
    Args:
    - seed (int): Random seed.
    - n_train_envs (int): Number of training envs.
    - **env_kwargs: Additional args for env creation.
    Returns:
    - train_env (VecEnv): Configured training environment.
    """
def user_evaluate_policy(seed, model, **env_kwargs):
    """
    Evaluate the policy with a separate evaluation environment.
    Args:
    - model (BaseAlgorithm): Model to eval.
    Returns:
    - mean_objective_value (float): Mean objective value obtained
during evaluation.
    - extra_info (dict): Additional metrics defined by the user.
    """
```

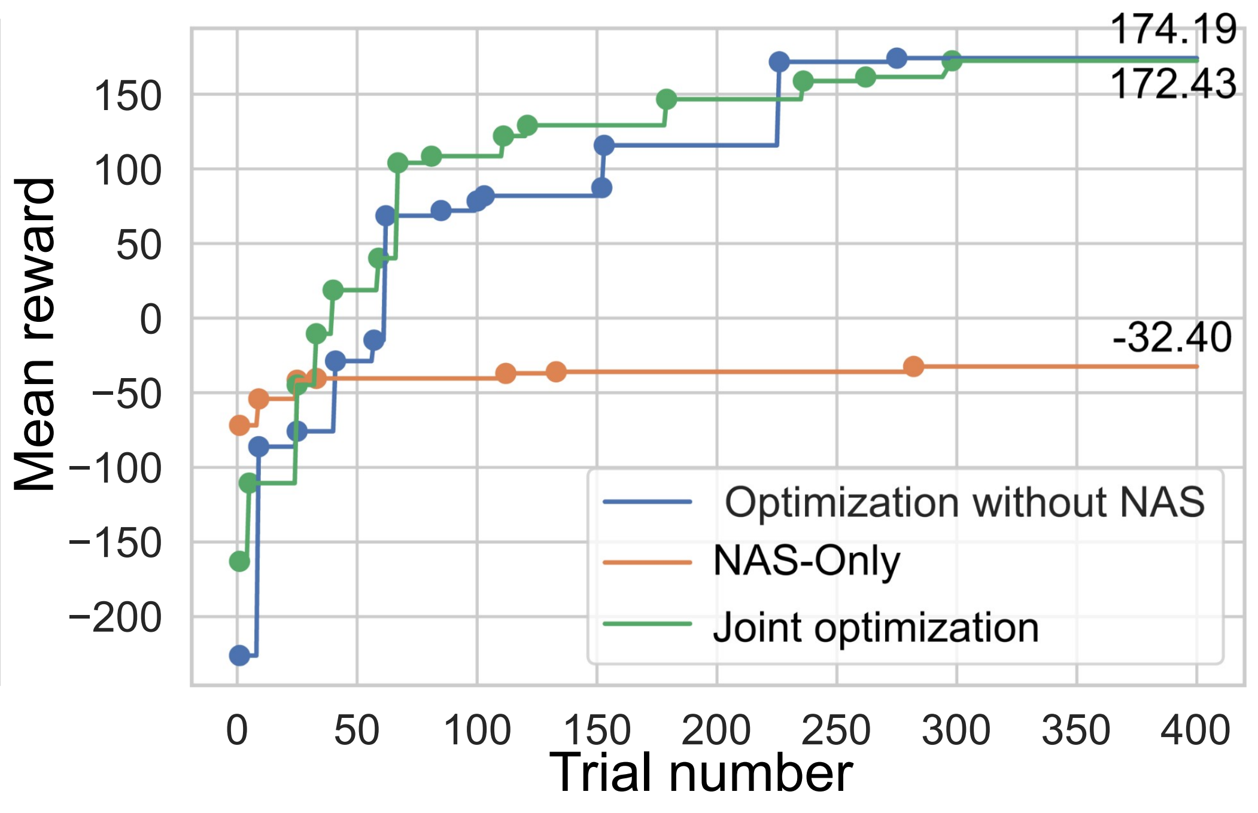
Results

- **Environment:** Pixel-based Lunar Lander POMDP.
- **Metrics:** Mean reward over 300 episodes.

Parameter	Type	Category	Range	Random search	BayesOpt	PSO
Field of view's size (pixels)	Integer	Environment	[40, 92]	71	92	92
Discount factor (γ)	Float	Agent	[0.4, 0.8]	0.7934	0.7984	0.8
Generalized advantage estimation (λ)	Float	Agent	[0.9, 0.95]	0.9433	0.9299	0.95
Learning rate	Float	Agent	$[3.5 \cdot 10^{-4}, 3.5 \cdot 10^{-3}]$	0.0006	0.0034	0.0035
Number of epochs	Integer	Agent	[3, 10]	6	3	5
Entropy coefficient	Float	Policy	[0.01, 0.1]	0.0284	0.0279	0.1
Clipping range	Float	Policy	[0.01, 0.3]	0.1407	0.0178	0.3
Activation function	Float	Policy	[0.0, 1.0]	0.5478	0.9259	1.0
No. of layers in policy	Integer	Policy	[1, 4]	1	3	4
No. of neurons per layer in policy	Integer	Policy	[64, 128]	83	89	127
No. of layers in value network	Integer	Policy	[2, 4]	4	2	4
No. of neurons per layer in value	Integer	Policy	[64, 128]	100	89	128
Mean reward				84.26	**172.43**	119.19

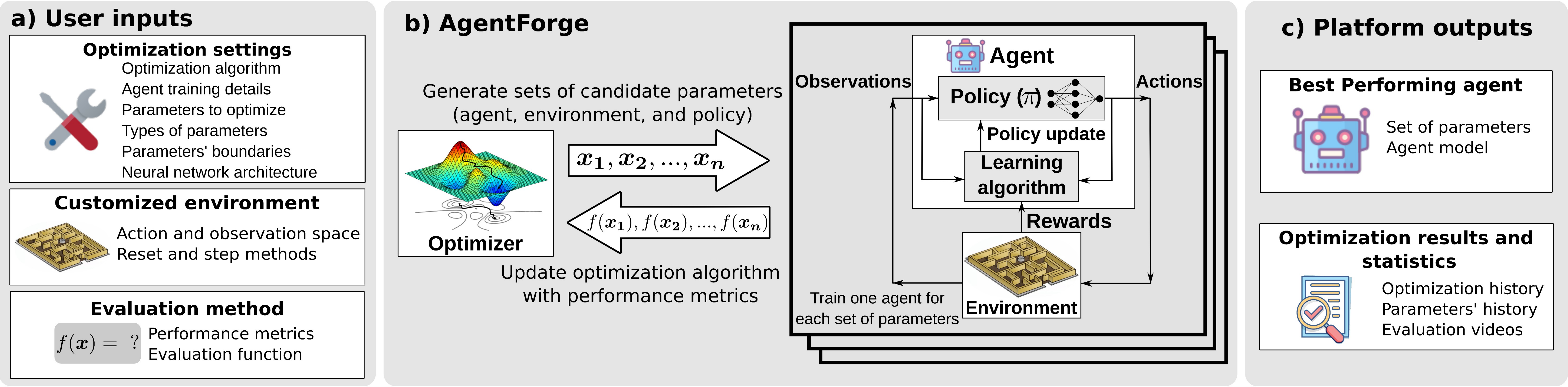


Joint optimization of parameters with NAS



BayesOpt, with and without NAS, and NAS-only optimization

Proposed Framework



a) User inputs

Optimization settings

Optimization algorithm
Agent training details
Parameters to optimize
Types of parameters
Parameters' boundaries
Neural network architecture

Customized environment

Action and observation space
Reset and step methods

Evaluation method

$f(x) = ?$ Performance metrics
Evaluation function

b) AgentForge

Generate sets of candidate parameters (agent, environment, and policy)

$x_1, x_2, \ldots, x_n$

$f(x_1), f(x_2), \ldots, f(x_n)$

Optimizer

Update optimization algorithm with performance metrics

Observations

Agent

Policy (π)

Actions

Policy update

Learning algorithm

Rewards

Train one agent for each set of parameters

Environment

c) Platform outputs

Best Performing agent

Set of parameters
Agent model

Optimization results and statistics

Optimization history
Parameters' history
Evaluation videos

Limitations and Future Work: We support only Gymnasium environments and we don't have a GUI yet. Future work includes broader compatibility, more algorithms, a GUI, and user studies to assess effectiveness.

arXiv

GitHub

Francisco Erivaldo Fernandes Junior
Antti Oulasvirta

erc
European Research Council
Established by the European Commission

FCAI Finnish Center for Artificial Intelligence