DOI: 10.15514/ISPRAS-2026-38(4)-7

The EG-TD3 Machine Learning Architecture: Evolutionary-Guided Twin Delayed Deep Deterministic Policy Gradient

H.-D. Djambong Tenkeu, ORCID: 0009-0002-4689-1665 <Dzhambong.T.K@hse.ru>
National Research University "Higher School of Economics",
11, Pokrovsky blvd, Moscow, 109028, Russia.

Abstract. This paper presents the Evolutionary-Guided Twin Delayed Deep Deterministic Policy Gradient (EG-TD3), a hybrid learning architecture that couples TD3 policy improvement with Differential Evolution for adaptive packet-buffer control under non-stationary traffic. Through empirical evaluation on a standard continuous-control benchmark (Pendulum-v1) and a synthetic traffic-burst queue simulator, we show that EG-TD3 improves training stability (coefficient of variation 0.005 versus 0.258 for TD3) and action smoothness (mean action delta 0.052 versus 0.164 for TD3) while maintaining competitive robustness under observation and action noise. The architecture combines periodic elite transplantation and fitness-guided exploration. Beyond learning results, we outline a service-mesh deployment path based on a split control/data plane. Full EG-TD3 training and multi-layer inference are kept in userspace next to the mesh control plane, because in-kernel machine learning inside eBPF remains constrained by verifier and instruction-set limits. On the node data plane, eBPF programs attached at TC or socket hooks, operating alongside a sidecar proxy such as Envoy, collect buffering and traffic telemetry into BPF maps and enforce quantized control decisions (target buffer size, timeout, and urgency thresholds) at control-loop frequency (10 – 100 Hz), without training on the packet path. Shared BPF maps mediate the interaction: eBPF writes telemetry; the EG-TD3 agent reads state and writes control parameters; eBPF applies the updated policy. Offline or sidecar training on logged trajectories does not block the fast path. An open-source implementation and full hyperparameter specifications are provided for reproducibility.

Keywords: network control; adaptive buffering; deep reinforcement learning; EG-TD3; twin delayed deep deterministic policy gradient (TD3); differential evolution (DE).

For citation: Djambong Tenkeu H.-D. The EG-TD3 Machine Learning Architecture: Evolutionary-Guided Twin Delayed Deep Deterministic Policy Gradient. Trudy ISP RAN/Proc. ISP RAS, 2026, vol. 38, issue 4, part 1, pp. 135-152. DOI: 10.15514/ISPRAS-2026-38(4)-7.

Acknowledgements. This work continues prior research published in the Proceedings of the Institute of System Programming of the Russian Academy of Sciences and was supported by the HSE University Basic Research Program.

Архитектура машинного обучения EG-TD3: эволюционно-направленный двойной задержанный глубокий детерминированный градиент политики

Х.-Д. Джамбонг Тенке, ORCID: 0009-0002-4689-1665 <Dzhambong.T.K@hse.ru>
Национальный исследовательский университет «Высшая школа экономики»,
Россия, 109028, г. Москва, ул. Покровский бульвар, д. 11.

Аннотация. В работе представлена архитектура EG-TD3 (Evolutionary-Guided Twin Delayed Deep Deterministic Policy Gradient) – гибридный конвейер обучения, объединяющий алгоритм TD3 и дифференциальную эволюцию для адаптивного управления буферизацией пакетов в нестационарных условиях. На стандартном тесте непрерывного управления (Pendulum-v1) и синтетическом симуляторе очередей с всплесками трафика показано, что EG-TD3 повышает стабильность обучения (коэффициент вариации 0,005 против 0,258 у TD3) и плавность управляющих воздействий (средняя дельта действий 0,052 против 0,164), сохраняя конкурентоспособную устойчивость к шуму наблюдений и действий. Архитектура использует периодическую замену элитных параметров и исследование, направляемое функцией приспособленности. Помимо результатов обучения описан путь развёртывания в сервисной сети на основе разделённых плоскостей управления и данных. Обучение и вывод EG-TD3 выполняются в пользовательском пространстве рядом с плоскостью управления сервисной сети, поскольку полноценное машинное обучение внутри eBPF ограничено верификатором и набором инструкций. На узловой плоскости данных программы eBPF, подключаемые к точкам перехватка TC или сокетов и работающие совместно с прокси в дополнительном контейнере (например, Envoy), собирают телеметрию буферизации и трафика в BPF-карты и применяют квантованные управляющие решения (целевой размер буфера, таймаут и пороги срочности) с частотой контура управления 10–100 Гц, без обучения на пути пакета. Взаимодействие опосредуется общими BPF-картами: eBPF записывает телеметрию; агент EG-TD3 считывает состояние и записывает параметры управления; eBPF применяет обновлённую политику. Обучение на журналах траекторий (офлайн или в прокси в дополнительном контейнере) не блокирует быстрый путь. Приведены полные спецификации гиперпараметров и открытая реализация для воспроизводимости результатов.

Ключевые слова: управление сетями; адаптивная буферизация; глубокое обучение с подкреплением; EG-TD3; двойной отсроченный глубокий детерминированный градиент политики (TD3); дифференциальная эволюция (DE).

Для цитирования: Джамбонг Тенке Х.-Д. Архитектура машинного обучения EG-TD3: эволюционно-направленный двойной задержанный глубокий детерминированный градиент политики. Труды ИСП РАН, 2026, том 38, вып. 4, часть 1, стр. 135–152 (на английском языке). DOI: 10.15514/ISPRAS-2026-38(4)-7.

Благодарности. Работа продолжает исследования, опубликованные в Трудах ИСП РАН, и выполнена при поддержке Программы фундаментальных исследований НИУ ВШЭ.

1. Introduction

The increasing complexity of modern computer networks, driven by microservices, real-time applications, and heterogeneous traffic patterns, demands adaptive control mechanisms that operate under uncertainty. Fixed buffering and congestion-control rules struggle to balance latency, throughput, and stability when workloads change over time.

Among deep reinforcement learning (RL) methods for continuous control, Twin Delayed Deep Deterministic Policy Gradient (TD3) [1] reduces value overestimation via clipped twin critics, delayed policy updates, and target policy smoothing. Applied to network buffering, however, TD3 remains sensitive to initialization (high cross-seed variance), can converge to local optima, and may produce abrupt action changes that are undesirable in production control loops.

Differential Evolution (DE) [2] provides complementary global search in policy parameter space but is sample-inefficient and lacks TD3's replay-based credit assignment. Existing hybrid

evolutionary–gradient methods (e.g., Evolutionary Policy Gradients [3]) often alternate between phases rather than tightly coupling evolution with an actor–critic learner, and they are rarely evaluated for kernel-adjacent network control.

This paper presents EG-TD3 as an *application-oriented* hybrid pipeline for adaptive buffering, not as a new universal class of optimizers. EG-TD3 keeps TD3 as the primary learner and uses DE through two mechanisms: (i) periodic transplantation of elite actor weights into the TD3 agent, and (ii) fitness-guided exploration that biases action selection toward high-fitness policy regions. We evaluate stability, policy smoothness, robustness, exploration, and generalization on Pendulum-v1 and a synthetic traffic-burst simulator, and we outline an eBPF-compatible deployment split between user-space inference and kernel-level measurement/enforcement [4].

The remainder of the paper is organized as follows. Section 2 reviews related work. Section 3 describes the EG-TD3 architecture and reproducibility specifications. Section 4 details the experimental methodology. Section 5 analyzes results. Sections 6–8 discuss limitations, future work, and conclusions.

2. Related works

2.1 Reinforcement Learning for Network Control

RL has been applied to congestion control [5], routing [6], and resource allocation [7]. Deep Q-networks [8] target discrete actions and are ill-suited to continuous buffer parameters. For continuous control, DDPG [9] and TD3 [1] are standard actor–critic baselines; TD3 is our gradient-based core.

2.2 Hybrid Evolutionary–Gradient Reinforcement Learning

Combining population-based search with policy gradients is an active area. Evolutionary Policy Gradients (EPG) [3] alternates gradient updates with evolutionary exploration. Evolution Strategies (ES) [10] optimize policy weights by fitness estimation but do not use off-policy replay. DEHB [11] applies evolutionary search to hyperparameters rather than online actor coupling. CEM-RL [12] mixes cross-entropy method elites with policy gradients but does not implement TD3-specific stabilization or networking deployment constraints.

EG-TD3 differs by operating DE inside the TD3 training loop on actor weights, using two explicit coupling mechanisms (transplantation and guided exploration), and targeting buffering control with an eBPF-oriented runtime model. **Ошибка! Источник ссылки не найден.** summarizes these distinctions.

Table 1. Comparison of EG-TD3 with representative hybrid and networking approaches.

Method	Ref.	Gradient learner	Evolutionary part	Coupling mode	Networking focus
EPG	[3]	Policy gradient	EA population	Alternating phases	No
ES for RL	[10]	ES (no replay)	Fitness on weights	Single optimizer	No
DEHB	[11]	Hyperband schedules	DE on HP config	Offline HP search	No
CEM-RL	[12]	DDPG/TD3 variants	CEM elites	Periodic mixing	No
PPO buffering	[13]	PPO	None	—	SDN buffering
DL buffer predictor	[14]	Supervised DL	None	—	No online RL
EG-TD3	This work	TD3	DE on actor weights	Transplant + guided exploration	Yes

2.3 Machine Learning in Networking and eBPF

Jay et al. [14] predict buffer sizes with supervised learning but without online adaptation. Schulze et al. [13] use PPO for buffer management without cross-seed stabilization analysis. Programmable data planes based on eBPF [4] enable kernel-level measurement and lightweight enforcement, but full MLP inference inside eBPF remains impractical. EG-TD3 therefore adopts a split design: training and inference in user space; eBPF for telemetry and actuation hooks.

2.4 Research Contribution and Positioning

Rather than claiming a principally new hybrid optimizer, we contribute:

- C1 – Buffering-oriented TD3+DE coupling. A reproducible training pipeline for continuous buffering parameters under online and offline traffic-burst scenarios.
- C2 – Dual integration mechanisms. Periodic elite transplantation and fitness-guided exploration, with measured effects on stability (CV = 0.005 vs. 0.258 for TD3) and smoothness (mean $\Delta a = 0.052$ vs. 0.164).
- C3 – Full implementation specification. Documented neural architectures, hyperparameters (Table 3), seeds, and open-source artifacts [15].
- C4 – eBPF deployment path. A user-space/kernel split compatible with programmable data-plane constraints [4].

3. The EG-TD3 Architecture

EG-TD3 implements the coupling strategy from Section 2.4: TD3 performs sample-efficient policy improvement while DE maintains a population of actor weights and supplies elites for transplantation and exploration guidance.

3.1 Problem formulation

Adaptive buffering is modeled as a contextual Markov decision process [16] with context $e_t \in E$ capturing non-stationary traffic. At step t the controller observes $s_t \in S$(RTT, loss rate ρ, jitter J, arrival rate λ, buffer occupancy B, latency sensitivity L, urgency U) and selects $a_t \in A$ (target buffer size B_{target}, timeout $T_{timeout}$, urgency threshold U_{th}). The environment returns reward $r_t = R(s_t, a_t, s_{t+1})$ and transitions via $P(s_{t+1} \mid s_t, a_t, e_t)$. The objective is to maximize expected discounted return $E[\sum_t \gamma^t r_t]$. EG-TD3 applies an existing TD3+DE solver to this formulation; simulator-specific state dimensions for Pendulum-v1 and the synthetic queue are given in Section 4.

3.2 Architecture overview

The architecture (Fig. 1) comprises two subsystems: (i) a TD3 actor–critic learner with replay buffer and twin critics [1], and (ii) a DE population evolving flattened actor weight vectors [2]. Six neural networks implement TD3: actor μ, critics Q_1, Q_2, and their targets. DE evaluates population members by episodic return (online) or dataset-based scores (offline).

3.2.1 Reinforcement learning subsystem

The RL subsystem is a standard TD3 implementation: experience tuples are stored in a replay buffer; twin critics minimize TD error with clipped double-Q targets; the actor is updated on a slower schedule with target policy smoothing. This provides efficient local improvement but remains initialization-sensitive in deceptive landscapes.

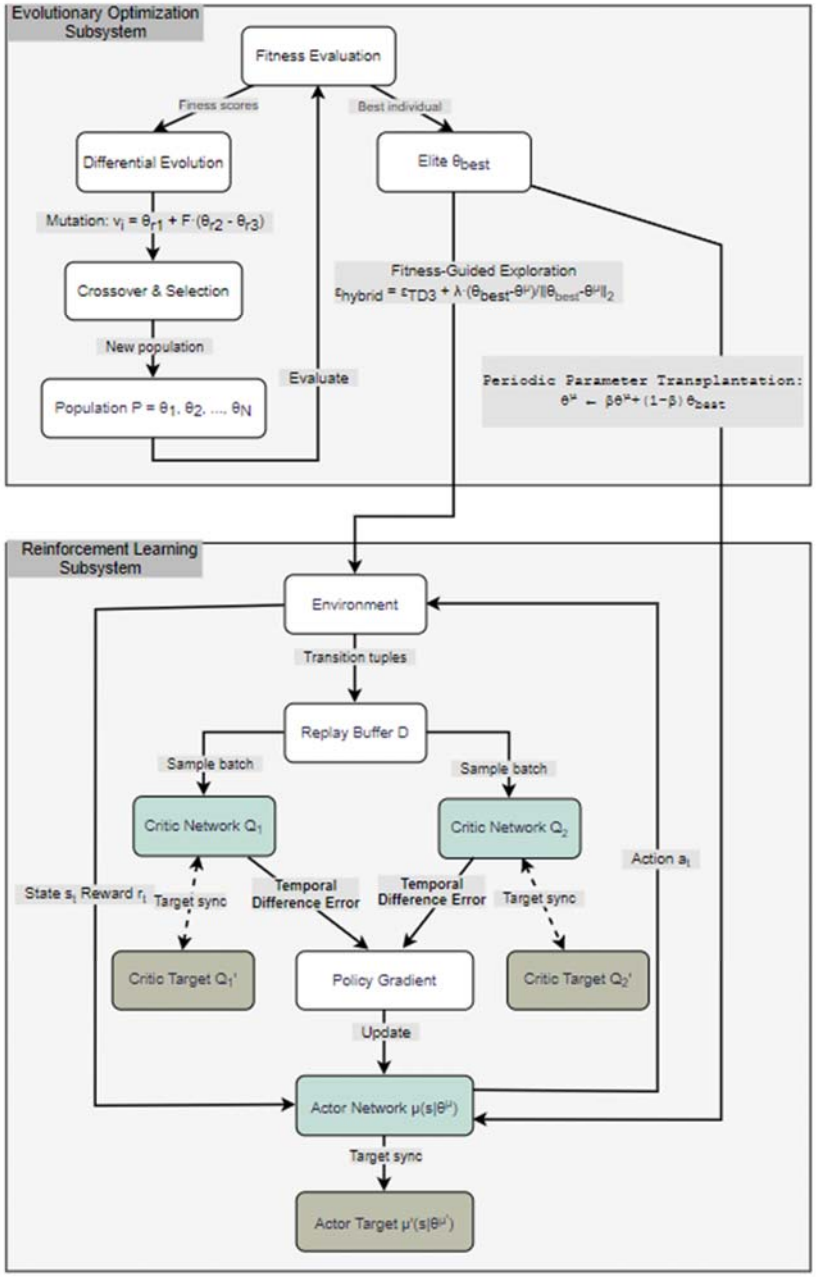

Fig. 1. EG-TD3 architecture: TD3 learning loop coupled with a DE population through transplantation and fitness-guided exploration.

3.2.2 Evolutionary optimization subsystem

DE maintains population $P = \{\theta_1,...,\theta_N\}$ in actor weight space and applies mutation, crossover, and greedy selection based on fitness F(θi) (mean episodic return). DE supplies global search; it does not replace TD3 updates.

3.2.3 EG-TD3 learning process

Primary networks (μ, Q_1, Q_2) interact with the environment; target networks (μ', Q_1', Q_2') stabilize bootstrapping. DE runs in parallel on actor parameters; coupling mechanisms in Section 3.3 connect the two subsystems.

3.3 Integration mechanisms

3.3.1 Periodic parameter transplantation

Every K environment steps, the best DE individual θ_{best} replaces the TD3 actor and actor target weights (full copy in our implementation). This provides coarse guidance and can help escape local optima without abandoning replay-based learning between transplants.

3.3.2 Fitness-guided exploration

With probability λ, action selection uses a blended actor $\theta_{blend} = \theta_\mu + \lambda(\theta_{best} - \theta_\mu)$; otherwise standard TD3 exploration applies. This biases local exploration toward regions DE has identified as high-fitness.

An overview of the integration mechanisms is presented in Table 2.

Table 2. Comparison of integration mechanisms in EG-TD3.

Aspect	Parameter transplantation	Fitness-guided exploration
Update mechanism	Full copy of elite DE actor weights into TD3 actor and actor-target	Blended actor θ_blend = θ_μ + λ(θ_{best} − θ_μ) used for action selection
Time scale	Coarse-grained (every K steps)	Fine-grained (per environment step, probability λ)
Primary role	Escape local optima via discontinuous policy jumps	Bias exploration toward high-fitness regions between transplants
Effect on policy	Can cause transient performance shifts after transplant	Smoother local evolution with directed exploration
Computational overhead	Moderate (N fitness evaluations every K steps)	Low (vector blend + occasional forward pass)

3.4 Neural network and training configuration

All networks are fully connected MLPs implemented in PyTorch, matching the open-source repository [15].

3.4.1 Actor μ(s | θμ)

Input: state dimension $|S|$ (3 for Pendulum-v1; 3 for the synthetic queue simulator). Two hidden layers of 256 units with ReLU; linear output scaled by tanh and a_{max}.

3.4.2 Twin critics Qi(s,a | θQi)

Input: concatenated [*s,a*]. Two hidden layers of 256 ReLU units; scalar linear output.
The neural network topology for actor and critic is shown in Table 3.

Table 3. Neural network topology (actor and each critic).

Layer	Actor	Critic
Input	|S|	|S|+|A|
Hidden 1	256 ReLU	256 ReLU
Hidden 2	256 ReLU	256 ReLU
Output	|A| (tanh × a_{max})	1 (linear)

3.4.3 Optimization and TD3 settings

Adam optimizer, learning rate 3×10^{-4} for actor and critics; $\gamma = 0.99$; Polyak $\tau = 0.005$; replay buffer 10^6; batch size 256; policy noise 0.2 (clip 0.5); policy delay 2; exploration noise $N(0,0.1)$ during training; 10,000 random warmup steps.

3.4.4 DE and EG-TD3 coupling

Population $N = 8$; mutation $F = 0.5$; crossover CR = 0.7; fitness = mean return over 3 episodes per candidate; transplant interval $K = 20{,}000$; guided exploration $\lambda = 0.2$. Complete values are presented in Table 4.

Table 4. Training hyperparameters (TD3, DE, and EG-TD3 coupling). Values match the open-source implementation [15].

Category	Parameter	Value
TD3	Hidden layers (actor/critic)	256, 256 (ReLU)
	Learning rate (actor, critics)	3×10^{-4} (Adam)
	Discount γ / Polyak τ	0.99 / 0.005
	Batch size / replay buffer	256 / 10^6
	Policy noise / clip	0.2 / 0.5
	Policy delay / exploration σ	2 / 0.1
DE	Population N / mutation F / crossover CR	8 / 0.5 / 0.7
	Fitness episodes per candidate	3
	DE evaluation interval	every de_steps (default 1,000)
EG-TD3 coupling	Transplant interval K	20,000 steps
	Guided exploration strength λ	0.2
Training	Random warmup / total steps	10,000 / 200,000
Evaluation	Seeds / eval episodes per seed	5 (0–4) / 5

3.5 Deployment model for eBPF-integrated buffering

Full TD3 inference inside eBPF is impractical due to verifier and ISA limits [4]. EG-TD3 therefore uses a **split-plane design**:

- **User-space control plane**: periodic EG-TD3 inference (single MLP forward pass) maps telemetry features to ($B_{target}, T_{timeout}, U_{th}$).
- **Kernel data plane (eBPF)**: lightweight programs attached at TC or socket hooks collect queue occupancy, drop counts, and smoothed arrival metrics into BPF maps; quantized control decisions adjust thresholds or marking policies at control-loop frequency (10–100 Hz), not per-packet training.
- **Training plane**: TD3 and DE run offline or in a sidecar with logged trajectories; transplants and replay updates do not block the datapath.

Fig. 2 sketches this stack. The empirical results in Section 5 validate the learning behavior; kernel integration is an engineering path supported by the architecture rather than a claim of in-kernel backpropagation.

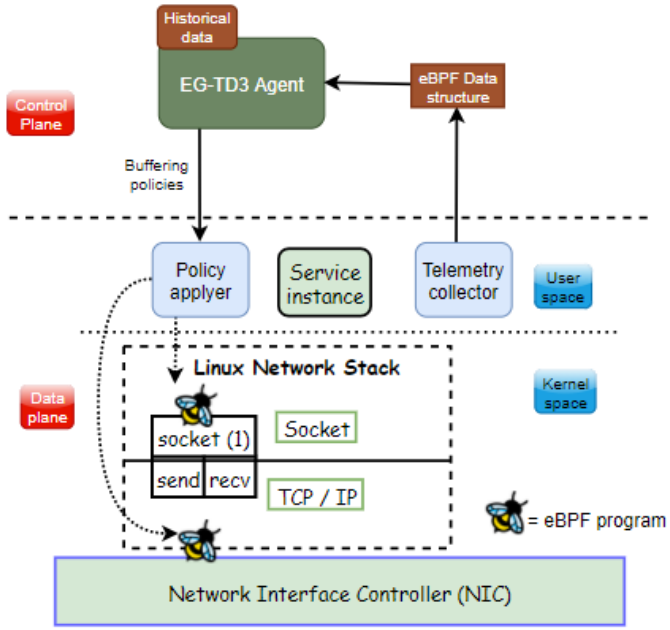

Fig. 2. Split deployment: user-space EG-TD3 inference and eBPF kernel hooks for telemetry and enforcement.

4. Experimental Methodology

Environments. The experiments were conducted over two evaluation settings:

- Standard (online RL) using Pendulum-v1, a continuous-control benchmark from OpenAI Gym [17] where the agent applies a bounded torque to swing a pendulum upright and keep it balanced. Observations include angle and angular velocity; actions are continuous torque. The reward penalizes angle error, velocity, and control effort, so higher (less negative) returns are better.

- Synthetic (offline RL) using a traffic-burst queueing simulator, where observations include normalized queue level, arrival rate ratio, and service rate ratio. Actions adjust service rate within a bounded range; rewards penalize queue buildup, packet drops, and large control changes.

Algorithms. We compare three methods:

- EG-TD3, the evolutionary-guided TD3 architecture.
- TD3, the pure actor–critic baseline [1].
- DE, a pure evolutionary baseline in policy parameter space using Differential Evolution [7].

Standard (online) training protocol. All methods are trained for 200,000 environment steps with 5 random seeds (0–4). Evaluation uses 5 episodes per seed without exploration noise. Final reported mean and standard deviation are computed across seeds.

Synthetic (offline) training protocol. A synthetic dataset of traffic bursts is generated by simulating queue dynamics with random actions. Offline TD3 and EG-TD3 are trained for 200,000 gradient updates; offline DE uses 50 generations with an initial critic pre-training phase. Evaluation is performed on the same synthetic simulator over 5 episodes per seed, and results are summarized across seeds.

EG-TD3 coupling parameters. For EG-TD3, we use guided exploration strength 0.2, fitness evaluation over 3 episodes per DE candidate, and a transplant interval of 20,000 steps. Other TD3/DE hyperparameters follow the default configuration in the implementation.

More about the experimental setup can be found in the Appendix section [15]. That includes the source code repository summary and how to run it.

4.1 Exploration Efficiency

We quantify exploration efficiency using two proxies computed from action sequences:

(i) Gaussian action entropy (higher implies broader action coverage);

(ii) Action diversity (mean per-dimension standard deviation; a higher value implies more varied actions).

Fig. 3 shows the diversity trajectory per-episode, and Fig. 4 summarizes the mean ± std between episodes. Table 5 shows exploration entropy and diversity.

Interpretation. The DE-only baseline exhibits the highest action entropy and diversity, which is expected for a population-based search that explores broadly in parameter space. However, high exploration does not necessarily translate into better returns, as DE still underperforms in mean return. TD3 shows the lowest entropy/diversity on average, reflecting a more exploitative policy that converges to a narrower action distribution. EG-TD3 lies between the two: it increases exploration relative to TD3 (higher entropy/diversity) without reaching the very broad exploration of DE.

The per-episode diversity time series further highlights this behavior. DE displays large episode-to-episode swings in diversity, indicating unstable exploration intensity. TD3 remains more consistent but also more limited in coverage. EG-TD3 shows moderate variability, suggesting a balance between exploration and exploitation. In practice, this balance can be desirable: it encourages discovery of new behaviors while avoiding excessive randomness that can harm performance, especially in control settings where large action fluctuations may be costly.

4.2 Generalization Under Parameter Shift

We evaluate generalization by modifying an environment parameter (gravity g in Pendulum-v1) and measuring the return relative to the base environment ($g \times 1.0$). The ratio *variant/base* close to 1 indicates robust generalization.

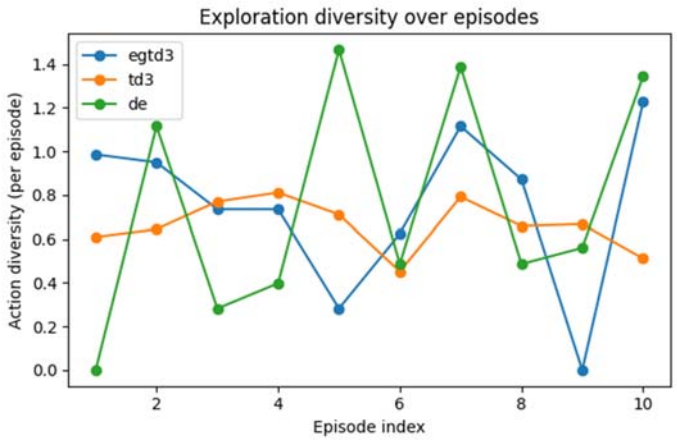

Fig. 3. Per-episode action diversity over time.

Table 5. Exploration efficiency statistics (mean ± std).

Method	Entropy	Diversity
EG-TD3	1.015 ± 0.323	0.705 ± 0.246
TD3	0.838 ± 0.500	0.620 ± 0.229
DE	1.543 ± 0.151	1.145 ± 0.164

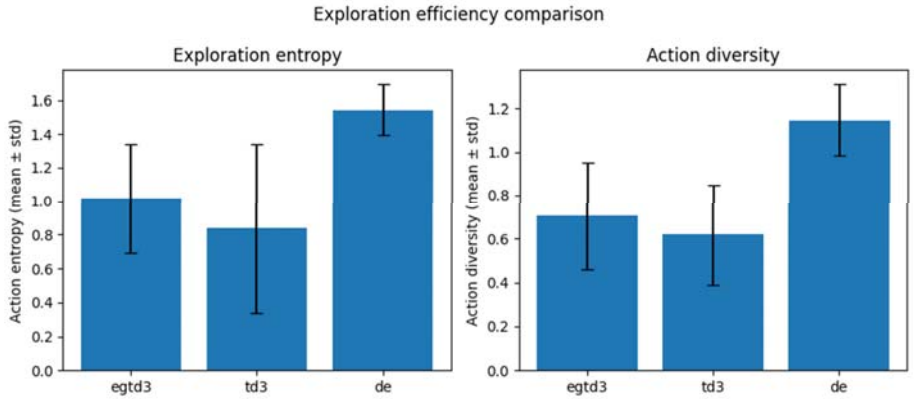

Fig. 4. Exploration efficiency summary.
Left: action entropy; right: action diversity (mean ± std).

Fig. 5 shows return ratios across gravity multipliers, while Table 6 show generalization under gravity shifts. EG-TD3 remains close to the base performance across shifts, while TD3 shows larger fluctuations (improvement at ×0.8 and ×1.2 in this run). DE remains near 1 but exhibits a drop at ×1.2, suggesting reduced robustness to stronger shifts.

Interpretation. Ratios above 1 do not necessarily imply better generalization; they may indicate that the modified dynamics are easier for the policy (e.g., reduced effective difficulty when g changes). Therefore, robust generalization is best interpreted as maintaining ratios close to 1 across shifts, with minimal variance across the tested multipliers. Under this view, EG-TD3 and DE appear more stable across gravity changes, while TD3 shows larger sensitivity to the parameter shift.

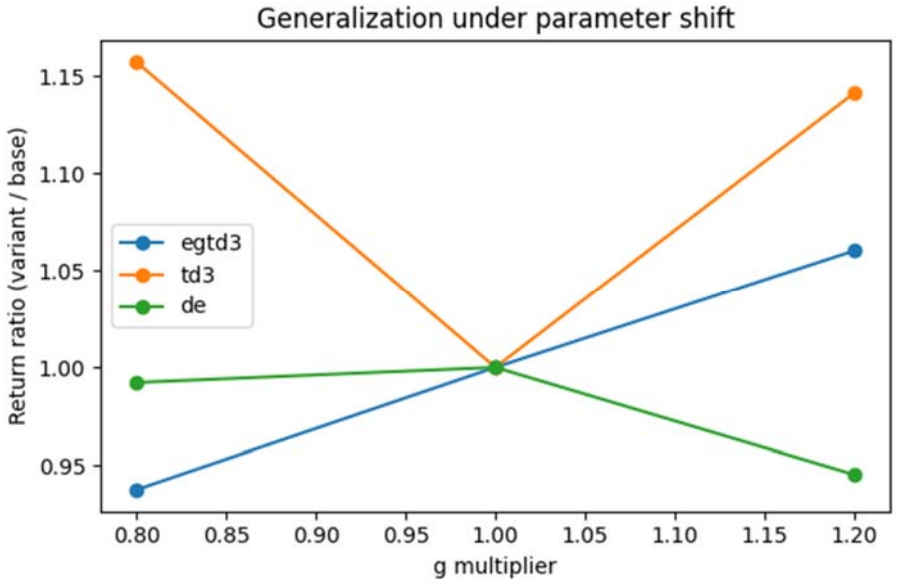

Fig. 5. Generalization under gravity shifts in Pendulum-v1. Ratios are variant/base return.

Table 6. Generalization under gravity shifts.

Method	g × 0.8	g × 1.0	g × 1.2
EG-TD3	0.938	1.000	1.060
TD3	1.157	1.000	1.141
DE	0.992	1.000	0.945

5. Results Analysis

5.1 Stability Analysis

Stability is quantified via standard deviation (std), coefficient of variation (CV), interquartile range (IQR), and min/max across seeds (lower std/CV indicates higher stability). For the standard setup, an overview is presented in Fig. 6 with extended statistic in Table 7. For the offline setup, it is presented on Fig. 7, with its statistic in Table 8.

Standard (online) results. EG-TD3 exhibits the smallest variance (std ≈ 7.17, CV ≈ 0.005), indicating the most consistent behavior across seeds. DE shows moderate variability (std ≈ 22.55, CV ≈ 0.013), while TD3, despite achieving the highest mean return, has the largest dispersion (std ≈ 36.31, CV ≈ 0.258). This reflects a trade-off between *absolute performance* and *stability*: TD3 is

more sensitive to initialization, whereas EG-TD3's evolutionary guidance reduces variability, even if mean return is lower.

Synthetic (offline) results. On the traffic-burst dataset, TD3 is the most stable among the three (std ≈ 0.46), DE is comparable (std ≈ 0.37), and EG-TD3 exhibits the highest variability (std ≈ 1.70). This inversion is expected in offline training: EG-TD3's DE coupling and periodic transplants depend on fitness signals derived from the dataset, which can amplify dataset bias and produce larger variability when the offline data distribution is narrow. In contrast, TD3's purely gradient-based updates are more stable under fixed data.

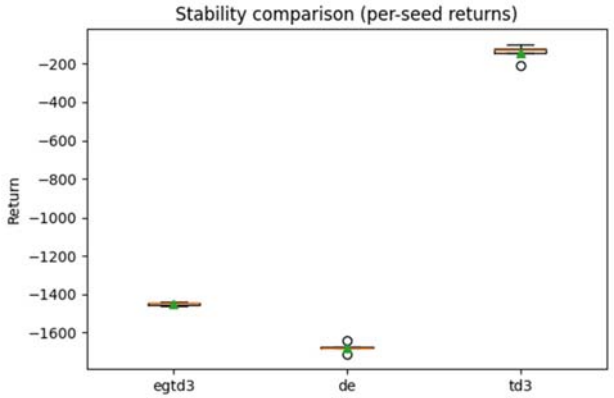

Fig. 6. Standard setup (Pendulum-v1): stability across 5 seeds.

Table 7. Stability statistics (standard setup).

Method	Mean	Std. Dev.	CV	IQR
EG-TD3	−1447.38	7.17	0.0050	11.19
DE	−1676.89	22.55	0.0134	3.93
TD3	−140.91	36.31	0.2577	21.46

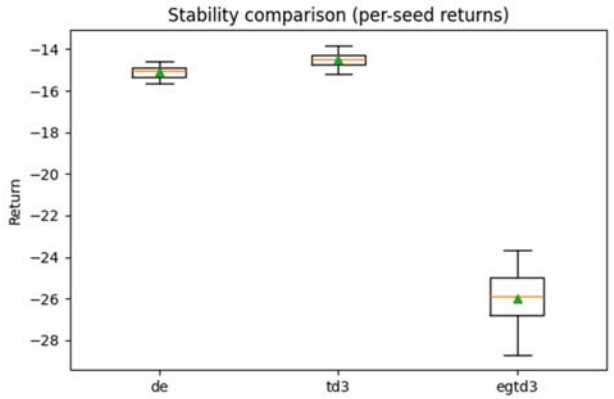

Fig. 7. Synthetic setup (traffic bursts, offline): stability across seeds.

Table 8. Stability statistics (synthetic setup).

Method	Mean	Std. Dev.	CV	IQR
DE	−15.10	0.37	0.0246	0.41
TD3	−14.51	0.46	0.0315	0.48
EG-TD3	−26.01	1.70	0.0653	1.83

5.2 Policy Smoothness

Smoothness is measured by the mean and standard deviation of the action delta ($\Delta a_t = \|a_t - a_{t-1}\|$), and by action variance. Lower Δa indicates smoother step-to-step control. Policy smoothness comparisons are presented on Fig. 8 (for standard setup) and on Fig. 9 (for synthetic setup). Their statistics are presented in Table 9 and Table 10.

Standard (online) results. TD3 exhibits the largest mean action-delta (0.164), indicating the most abrupt control changes. EG-TD3 and DE produce smaller mean deltas (0.052 and 0.056), suggesting smoother control. EG-TD3 also maintains lower action variance than DE, indicating smoother control without overly large action swings. This aligns with EG-TD3's guided exploration, which moderates rapid policy changes.

Synthetic (offline) results. In the offline setting, TD3 achieves the lowest action variance (0.104) and a low mean delta (0.045), suggesting stable control under fixed data. EG-TD3 remains smooth (mean delta 0.048) but shows higher action variance (0.958), indicating wider overall action spread despite locally smooth changes. DE reports zero variance and zero delta because the offline DE evaluation produces a nearly constant policy under the dataset-based fitness signal; this is consistent with a conservative policy that does not adapt dynamically during evaluation.

5.3 Robustness to Observation/Action Noise

We evaluate robustness by injecting Gaussian noise into both observations and actions and reporting the ratio of noisy to clean return (values closer to 1 indicate higher robustness), following common practices for evaluating policy robustness under input perturbations [18]. Results are shown for both standard (online) Pendulum-v1 and the synthetic (offline) traffic-burst dataset. In standard setup, robustness under observation (or action) noise is presented on Fig. 10. Corresponding ratios are presented on Table 11. The corresponding results for synthetic setup are presented on Fig. 11 and Table 12.

Standard (online) results. EG-TD3 and DE remain close to a ratio of 1 across noise levels, indicating comparatively stable behavior under perturbations. TD3 exhibits larger fluctuations, especially at low noise (σ=0.02), where the ratio drops substantially before recovering at higher noise. This suggests higher sensitivity to input perturbations, consistent with its more aggressive exploitation in the online setting.

Synthetic (offline) results. In the offline setting, all methods show larger deviations from 1, reflecting sensitivity to noise under a fixed dataset. EG-TD3 remains the closest to 1 overall, while TD3 and DE show larger drops, particularly at intermediate noise levels. This pattern is expected because offline policies are trained on a fixed distribution; noise perturbs observations/actions away from the dataset support, leading to reduced robustness for methods that overfit to the data.

5.4 Summary

To provide a clear and concise overview of the distinct characteristics of the three algorithms evaluated in this study, Table 13 summarizes their key differences across core operational dimensions.

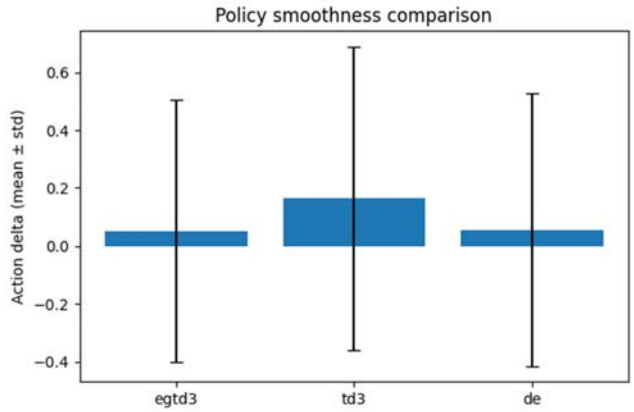

Fig. 8. Standard setup (Pendulum-v1): mean action-delta with ±1 std error bars.

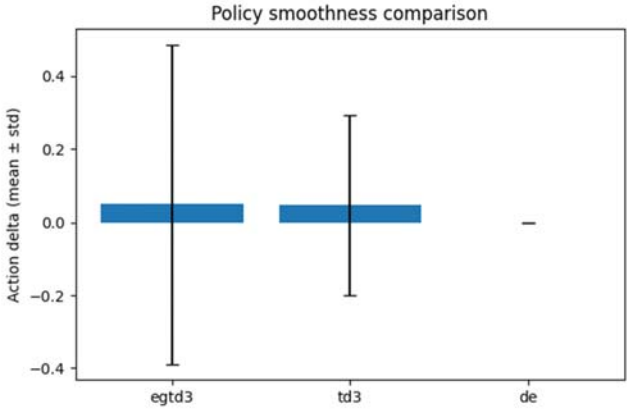

Fig. 9. Synthetic setup (traffic bursts, offline): mean action-delta with ±1 std error bars.

Table 9. Policy smoothness metrics (standard setup).

Method	Action Var.	Mean Δa	Std. Δa
EG-TD3	0.526	0.0523	0.4542
TD3	0.437	0.1636	0.5246
DE	0.860	0.0563	0.4711

Table 10. Policy smoothness metrics (synthetic setup).

Method	Action Var.	Mean Δa	Std. Δa
EG-TD3	0.958	0.0482	0.4366
TD3	0.104	0.0455	0.2453
DE	0.00	0.00	0.00

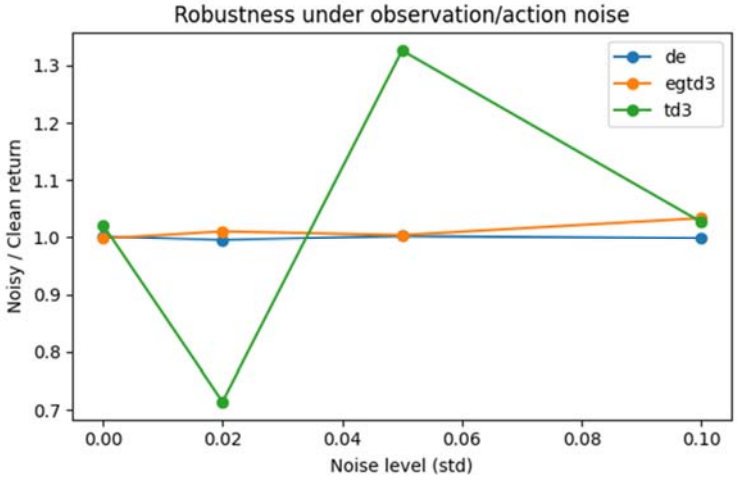

Fig. 10. Standard setup: robustness under observation/action noise (noisy/clean return ratio).

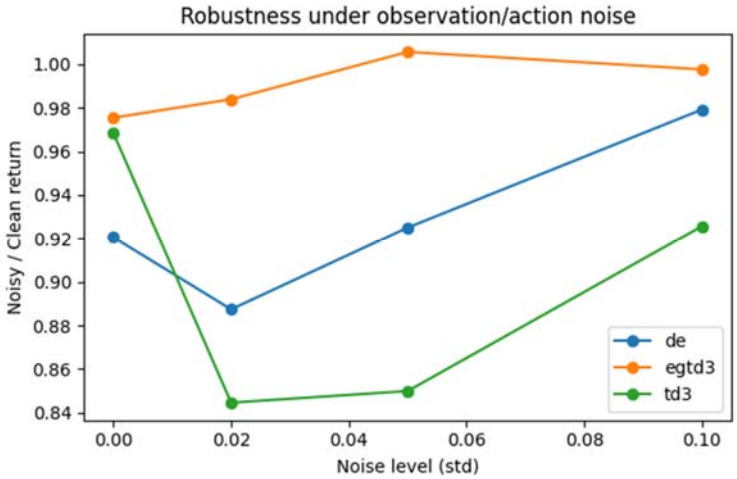

Fig. 11. Synthetic setup: robustness under observation/action noise (noisy/clean return ratio).

Table 11. Robustness ratios (standard setup; closer to 1 is more robust).

Method	σ = 0.02	σ = 0.05	σ = 0.10
EG-TD3	1.011	1.004	1.034
TD3	0.714	1.326	1.027
DE	0.995	1.002	0.999

Table 12. Robustness ratios (synthetic setup; closer to 1 is more robust).

Method	σ = 0.02	σ = 0.05	σ = 0.10
EG-TD3	0.984	1.006	0.998
TD3	0.845	0.850	0.926
DE	0.887	0.925	0.979

Table 13. Comparative Summary of TD3, DE, and EG-TD3.

Aspect	TD3	DE	EG-TD3
Core Strategy	Gradient-based local optimization	Population-based global search	Hybrid: local + global search
Exploration	Action noise	Population diversity	Biased exploration + diversity
Sample Efficiency	High	Low	Moderate
Stability (Std Dev)	Low (36.31)	Moderate (22.55)	High (7.17)
Policy Smoothness	Abrupt (0.164)	Smooth (0.056)	Smoothest (0.052)
Noise Robustness	Sensitive	Stable	Most consistent
Exploration Diversity	Lowest	Highest	Balanced
Offline Performance	Stable (0.46)	Stable (0.37)	Variable (1.70)
Local Optima Escape	Poor	Excellent	Good
Computational Cost	Low	High	Moderate
Hyperparameter Count	Moderate	Moderate	High
Deployment Complexity	Low	High	Moderate
Best Use Case	Smooth, data-rich problems	Deceptive landscapes	Non-stationary environments

The comparison highlights the fundamental trade-offs between the gradient-based local optimization of TD3, the population-driven global search of Differential Evolution (DE), and the hybrid approach of our proposed EG-TD3. As the table illustrates, EG-TD3 effectively balances the sample efficiency of TD3 with the exploratory diversity of DE, achieving superior stability and policy smoothness, though this comes at the cost of increased computational overhead and hyperparameter complexity compared to its standalone counterparts.

6. Risks and Limitations

EG-TD3 shows favorable stability and smoothness in online experiments, but several limitations must be acknowledged for real-world network control.

6.1 Computational Overhead

Maintaining and evaluating a DE population multiplies environment interactions during fitness evaluation (population size N, fitness over 3 episodes per candidate, every K steps). On resource-

constrained nodes or under strict control-loop latency, this overhead may be prohibitive. Mitigations include less frequent DE evaluation, surrogate fitness models, or parallel population evaluation.

6.2 Offline Performance Degradation

In the offline traffic-burst setting, EG-TD3 exhibits higher variability (std. 1.70 vs. 0.46 for TD3). Fitness-guided exploration depends on critic quality; with fixed datasets, unreliable value estimates can amplify bias. EG-TD3 is therefore better suited to online settings with fresh interaction data than to purely offline pipelines without additional off-policy corrections.

6.3 Hyperparameter Sensitivity

EG-TD3 adds coupling parameters (K, λ, N, F, CR) atop TD3 hyperparameters. Suboptimal coupling can negate evolutionary benefits. Our reported configuration (Table 3) was chosen from a focused grid search; new domains may require re-tuning.

7. Future Work

7.1 Algorithmic Enhancements

Adaptive coupling (K, λ, N) based on measured policy improvement could reduce tuning burden. Alternative evolutionary operators (CMA-ES, CEM) and multi-objective fitness (latency vs. throughput) are natural extensions. Off-policy fitness estimators may improve offline behavior.

7.2 Deployment Engineering

Concrete eBPF prototypes should validate telemetry map design, control-loop frequency, and safe policy hot-swapping between user-space inference and kernel enforcement [4]. Tooling for debugging hybrid learners in production meshes is another priority.

7.3 Application Domains

Beyond buffering, EG-TD3-style coupling may apply to adaptive load balancing in service meshes, SDN routing, and 5G/6G resource slicing, provided state/action dimensions and safety constraints are addressed.

8. Conclusion

This paper presented EG-TD3, an application-oriented hybrid pipeline that couples TD3 with Differential Evolution for adaptive buffering control. Rather than proposing a new universal optimizer class, we documented a reproducible TD3+DE integration with two explicit mechanisms–periodic elite transplantation and fitness-guided exploration–together with full neural-network and hyperparameter specifications aligned with the open-source implementation.

Empirical evaluation on Pendulum-v1 and a synthetic traffic-burst simulator shows that EG-TD3 substantially reduces cross-seed variance (CV 0.005 vs. 0.258 for TD3) and action jerk (mean Δa 0.052 vs. 0.164) while maintaining competitive robustness under observation and action noise. Offline experiments highlight a trade-off: evolutionary coupling can increase variability when critics are trained on fixed data. We also outlined an eBPF-compatible split deployment in which kernel programs collect telemetry and enforce decisions while inference remains in user space.

Future work should validate the architecture in high-fidelity network simulators, refine kernel integration prototypes, and study adaptive coupling under non-stationary production traffic. The implementation and configuration tables provided here are intended to support reproducibility and revision of the ISPRAN submission.

This paper is the continuation of a research, with a previous paper published in the journal “Proceedings of the Institute of System Programing” of the Russian Academic of Sciences [15]

References

[1]. Fujimoto S., van Hoof H., Meger D. Addressing function approximation error in actor-critic methods. In Proceedings of the 35th International Conference on Machine Learning (ICML). PMLR, 2018, pp. 1587-1596.

[2]. Storn R., Price K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. In Journal of Global Optimization 11.4 (1997), pp. 341-359.

[3]. Khadka S., Tumer K. Evolution-guided policy gradient in reinforcement learning. In arXiv preprint, 2018. DOI: 10.48550/arXiv.1805.07917.

[4]. Zhang W., Liu F., Bo Zhao B. Towards in-kernel machine learning with eBPF: opportunities and challenges. In Proceedings of the ACM SIGCOMM Workshop on eBPF and Kernel Extensions, 2024, pp. 45-52.

[5]. Winstein K. Balakrishnan H. TCP ex machina: Computer-generated congestion control. In Proceedings of the ACM SIGCOMM Conference, 2013.

[6]. Valadarsky A., Schapira M., Shahaf D. Learning to route. In Proceedings of the ACM Workshop on Hot Topics in Networks (HotNets). 2017.

[7]. Xu M., Zhu H., Li Y., Liu J. Deep reinforcement learning for dynamic spectrum access in wireless networks. In IEEE Transactions on Cognitive Communications and Networking, 2020.

[8]. Mnih V., Kavukcuoglu K., Silver D. et al. Human-level control through deep reinforcement learning. In Nature 518.7540, 2015, pp. 529-533.

[9]. Lillicrap T.P., Hunt J.J., Pritzel A. et al. Continuous control with deep reinforcement learning. In:International Conference on Learning Representations (ICLR), 2016.

[10]. Salimans T., Ho J., Chen X. et al. Evolution strategies as a scalable alternative to reinforcement learning. In arXiv preprint, 2017. DOI: 10.48550/arXiv.1703.03864.

[11]. Awad N., Mallik N., Hutter F. DEHB: Evolutionary hyperband for scalable, robust and efficient hyperparameter optimization. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI). 2021, pp. 2147-2153.

[12]. Pourchot A., Sigaud O. CEM-RL: Combining evolutionary and gradientbased methods for policy search. In International Conference on Learning Representations (ICLR), 2019.

[13]. Schulze M., Krämer M., Wehrle K. Adaptive buffer management with deep reinforcement learning. In IEEE Conference on Computer Communications (INFOCOM), 2021, pp. 1-10.

[14]. Jay N., Rotman A., Godfrey P.B. Deep learning for adaptive buffer sizing. In Proceedings of the ACM Workshop on Network Meets AI & ML (NetAI), 2019, pp. 1-7.

[15]. Tenkeu H.-D. D. EG-TD3: Evolutionary-Guided Twin Delayed Deep Deterministic Policy Gradient. Version 0.1.0. 2026. Available at: https://github.com/14karra/eg-td3, accessed 27.07.2026.

[16]. Puterman M.L. Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons, 2014.

[17]. Brockman G., Cheung V., Pettersson L. et al. OpenAI Gym. In arXiv preprint, 2016. DOI: 10.48550/arXiv.1606.01540.

[18]. Gleave A., Dennis M., Wild C. et al. Adversarial policies: attacking deep reinforcement learning. In International Conference on Learning Representations (ICLR), 2020.

Информация об авторах / Information about authors

Ханк-Дебэн ДЖАМБОНГ ТЕНКЕ – магистр программной инженерии, аспирант НИУ “Высшая Школа Экономики”, приглашенный преподаватель на факультет компьютерных наук НИУ “Высшая Школа Экономики”. Сфера научных интересов: инженерия программного обеспечения и компьютерных систем.

Hank-Debain DJAMBONG TENKEU – Master of Science in Software Engineering, postgraduate student at the National Research University “Higher School of Economics”, invited lecturer at the Faculty of Computer Science of NRU “Higher School of Economics”. Research interests: Software and Computer Systems Engineering.