ICAIF'24
5th ACM International Conference on AI in Finance
Brooklyn, NY, November 14-17, 2024

Autoregressive DRL with Learned Intrinsic Rewards for Portfolio Optimisation

MAGDALENE HUI QI LIM
NIXIE S LESMANA*
CHI SENG PUN

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE
NUS National University of Singapore

Overview

- BACKGROUND
  - Deep Reinforcement Learning: Policy Gradient (PG) via A2C
  - Portfolio Optimization as DRL problem.
  - Reward Design in (D)RL: motivation + approaches ⇛ Learned Intrinsic Reward (LIRPG).
  - Multi-Dimensional Action Spaces in (D)RL: motivation + approaches ⇛ Autoregressive RL.
- METHOD
  - Autoregressive A2C via MMDP
  - Extending LIRPG to Autoregressive A2C.
- EXPERIMENTS
  - Effects of Intrinsic Reward
  - Best Strategies (DRL trading x Reward)
- CONCLUSION

Background: Deep Reinforcement Learning (DRL)

- Env: **S**tates $\mathcal{S}$, **A**ctions $\mathcal{A}$, **T**ransition Probabilities $T: \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$, **R**ewards $r: \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathbb{R})$
- Policy (parameterized) $\pi_\theta: \mathcal{S} \to \mathcal{P}(\mathcal{A})$
- Objective (parameterized)

$$J(\theta) \coloneqq \mathbb{E}_\theta\left[\sum_{t=0}^{\infty} \gamma^t r_t\right] = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \middle| s_t \sim T(\cdot\,|s_{t-1}, a_{t-1}), a_t \sim \pi_\theta(\cdot\,|s_t)\right]$$

- Learning Policy
  - Policy Gradient (PG)[S99]

$G(s_t, a_t) \coloneqq \sum_{\tau=t}^{\infty} \gamma^{\tau-t} r_\tau$ return till termination

$$\nabla J(\theta) \propto \mathbb{E}_\theta[G(s_t, a_t) \nabla_\theta log \pi_\theta(a_t|s_t)]$$ (via Backpropagation)

  - (this work) A2C: replace return G(.) with Advantage Function $A(.)$ for variance reduction[M16],

$$\nabla J(\theta) \propto \mathbb{E}_\theta[A(s_t, a_t) \nabla_\theta log \pi_\theta(a_t|s_t)]$$

Risk adjusted return $A(s_t, a_t) \coloneqq G(s_t, a_t) - \hat{V}_{\theta_v}(s_t)$; $\hat{V}(.)$ is (parameterized) value estimate.

[S99] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. Advances in neural information processing systems 12 (1999).

[M16] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asyn-chronous Methods for Deep Reinforcement Learning. In Proceedings of The 33rd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 48). PMLR, 1928–1937.

Background: Portfolio Optimization as DRL Problem

- The portfolio optimization Objective: find $\pi_\theta: \mathcal{S} \to \mathcal{P}(\mathbb{Z}^d)$ to trade $d > 1$ stocks daily.
- SATR matching[Y21]
  - States $\boldsymbol{s} = [b, \boldsymbol{p}, \boldsymbol{h}, \boldsymbol{m}]$: portfolio balance $b \in \mathbb{R}_+$, stock prices $\boldsymbol{p} \in \mathbb{R}_+^d$, stock holdings $\boldsymbol{h} \in \mathbb{Z}_+^d$, M market indicators $\boldsymbol{m} \in \mathbb{R}^{Md}$.
  - Actions $\boldsymbol{a} = (a^1, a^2, \ldots, a^d)$: Sell/Buy/Hold for each stock dimension $k \in \{1,2, \ldots, d\}$.
  - Transitions: (i) $\boldsymbol{p}, \boldsymbol{m}$-observed from Market, (ii) $\boldsymbol{b}, \boldsymbol{h}$-computed as follows,
    - $a_t^k < 0$: Sell $-a_t^k \in (0, h_t^k]$ shares $\Rightarrow h_{t+1}^k = h_t^k - a_t^k \in \mathbb{Z}_+$.
    - $a_t^k \geq 0$: Buy a_t^k (or hold if $a_t^k = 0$) shares $\Rightarrow h_{t+1}^k = h_t^k + a_t^k$.
    - $b_{t+1} = \mathrm{b_t} - \mathrm{p}_t^k \times a_t^k - 0.1\% \times p_t^k \times |a_t^k|$.
  - Rewards: $\mathrm{r}_{t+1} = U(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1})$

E.g., Profit $U(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1}) = (b_{t+1} + \boldsymbol{p}_{t+1}^T \boldsymbol{h}_{t+1}) - (b_t + \boldsymbol{p}_t^T \boldsymbol{h}_t)$

logReturn $U(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1}) = logR_t$, Returns $R_t \coloneqq \frac{b_{t+1} + \boldsymbol{p}_{t+1}^T \boldsymbol{h}_{t+1}}{b_t + \boldsymbol{p}_t^T \boldsymbol{h}_t} - 1$.

[Y21] Hongyang Yang, Xiao-Yang Liu, Shan Zhong, and Anwar Walid. 2021. Deep reinforcement learning for automated stock trading: an ensemble strategy. In Proceedings of the First ACM International Conference on AI in Finance. ACM, Article 31.

Background: Reward Design (RD) x Learned Intrinsic Rewards (LIR) Motive

- Recall:
  - RL Objective $J(\theta; r) \coloneqq \mathbb{E}_\theta[\sum_{t=0}^{\infty} \gamma^t r_t]$
  - Rewards $\mathrm{r}_{t+1} = U(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1})$

- RD Problem: r_t is commonly *hand-designed* to match the *task* domain.
  - Here onward, refer to *task-based* rewards as *external reward* $r^{ex} \sim U^{ex}$
  - In finance: $U^{ex} \in$ {Profit, logReturn, CSR[1], DSR1[2], DSR2[3,[M98]]}

- LIR Motive: *Optimal Reward* hypotheses[SSL10, SLB09]
  - Setting $r = r^{ex}$ assumes ability to expect everything that happens during the Env-Agent feedback loop $\Rightarrow$ insufficient for "bounded, learning" agents
  - Formally: consider the space of *all* reward functions $\mathcal{U} \ni r^{ex}$ and let $\theta^*(r) \coloneqq \underset{\theta}{\operatorname{argmax}} J(\theta; r)$. Then,

$$\exists r \in \mathcal{U}, r \neq r^{ex}, J(\theta^*(r); r^{ex}) > J(\theta^*(r^{ex}); r^{ex})$$

[M98] John Moody, Matthew Saffell, Yuansong Liao, and Lizhong Wu. 1998. Reinforcement learning for trading systems and portfolios: Immediate vs future rewards. Springer US, 129–140.

[SSL10] Jonathan Sorg, Satinder Singh, and Richard Lewis. 2010. Internal rewards mitigate agent boundedness. In Proceedings of the 27th International Conference on International Conference on Machine Learning. 1007 – 1014.

[SLB09] Satinder Singh, Richard L Lewis, and Andrew G Barto. 2009. Where do rewards come from. In Proceedings of the annual conference of the cognitive science society.Cognitive Science Society, 2601–2606.

1 Cumulative Sharpe Ratio
2 Difference in Cumulative Sharpe Ratios
3 Differential Sharpe Ratios

Background: RD Approaches x LIRPG

- Recall $J(\theta; r) := \mathbb{E}_\theta[\sum_{t=0}^{\infty} \gamma^t r_t]$. Set $r \leftarrow r^{ex+in} := (1-\lambda) r^{ex} + \lambda r^{in}$. $\longrightarrow$ intrinsic rewards
  - Hand-design r^{in}. E.g. $r^{in} \in$ {Entropy[L18], Surprise[AS17], Novelty/visitation counts[B16]}
  - Search for r^{in}. E.g., evolutionary search[S10], online gradient ascent[SLS10]
  - Compared to hand-design, learned r^{in} is more general:

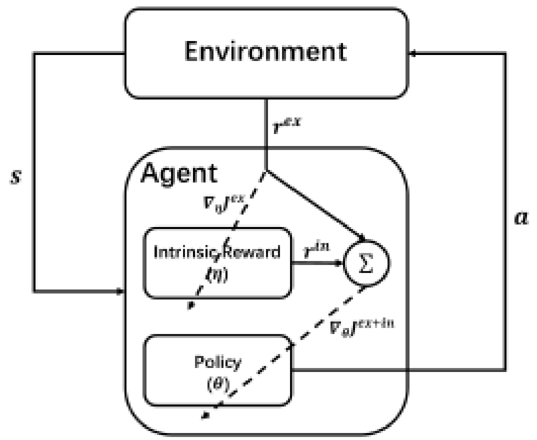
    the issue that r^{in} is supposed to resolve $\neq$ the specific issue that the hand-designed r^{in} are expected to resolve, e.g., max Entropy/novelty for balancing exploration and exploitation

- This work builds on LIRPG[ZOS18] (online gradient ascent x DRL)
  - Workflow: $\Delta\eta \propto \nabla_\eta J^{ex}(\eta) := \nabla_\eta \underbrace{J(\eta; r^{ex})}_{\text{Task-based objective}} \mid \Delta\theta \propto \nabla_\theta J^{ex+in}(\theta) := \nabla_\theta \underbrace{J\left(\theta; r \sim r^{ex}, r_\eta^{in}\right)}_{\text{Augmented objective}}$.

    since $\theta^*(r) := \underset{\theta}{\operatorname{argmax}}\, J(\theta; r) \mid \exists r^{in},\ J\left(\theta^*\left(r = r^{ex+in}\right); r^{ex}\right) > J(\theta^*(r^{ex}); r^{ex})$
  - LIRPG framework is readily applicable to A2C, but remains to choose the r_η^{in} structure

    $\Rrightarrow$ Our contribution: LIRPG for Multi Dimensional Action Spaces (MDAS).

Fig 1: LIRPG Framework[ZOS18]



[L18] Sergey Levine. 2018. Reinforcement learning and control as probabilistic inference: Tutorial and review. arXiv preprint arXiv:1805.00909 (2018).
[AS17] Joshua Achiam and Shankar Sastry. 2017. Surprise-based intrinsic motivation for deep reinforcement learning. arXiv preprint arXiv:1703.01732 (2017).
[B16] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. 2016. Unifying count-based exploration and intrinsic motivation. Advances in neural information processing systems 29 (2016).
[S10] Satinder Singh, Richard L Lewis, Andrew G Barto, and Jonathan Sorg. 2010. Intrinsically motivated reinforcement learning: An evolutionary perspective. IEEE Transactions on Autonomous Mental Development 2, 2 (2010), 70–82.
[SLS10] Jonathan Sorg, Richard L Lewis, and Satinder Singh. 2010. Reward design via online gradient ascent. Advances in Neural Information Processing Systems 23 (2010).
[ZOS18] Zheng Zeyu, Oh Junhyuk, and Singh Satinder. 2018. On learning intrinsic rewards for policy gradient methods. In Proceedings of the 32nd International Conference on Neural Information Processing Systems. 4649–4659.

Background: Multi-Dimensional Action Spaces (MDAS)

- LIRPG allows flexible r_η^{in} and π_θ structures (e.g. MLP or CNN), up to the forms $r_\eta^{in}: \mathcal{S} \times \mathcal{A} \times \mathcal{S}' \to \mathbb{R}$ and $\pi_\theta: \mathcal{S} \to \mathcal{P}(\mathcal{A})$ ⇛ Insufficient, especially for portfolio management,
  - MDAS: $\mathcal{A} \subset \mathbb{R}^d, d > 1$.
  - Intricate dependencies across different action dimensions (i.e., *subactions*).
- Autoregressive RL for MDAS[Z18]: allow policies π_θ to explicitly model subaction dependencies.
  - via Modified MDP (MMDP)[M18]

$$\pi_\theta(\boldsymbol{a}_t|\boldsymbol{s}_t) \coloneqq \pi_\theta\left(a_t^1, \dots, a_t^d \middle| \boldsymbol{s}_t\right) = \pi_\theta(a_t^1|\boldsymbol{s}_t) \times \pi_\theta(a_t^2|\boldsymbol{s}_t, a_t^1) \times \cdots \times \pi_\theta\left(a_t^d \middle| \boldsymbol{s}_t, a_t^1, \dots, a_t^{d-1}\right)$$

- Just like policy, r_η^{in} will need to factor in such dependency, to better distinguish the contribution of one subaction from another

⇛ Our contribution: LIRPG for MDAS (via MMDP)

[Z18] Yiming Zhang, Quan Ho Vuong, Kenny Song, Xiao-Yue Gong, and Keith W Ross. 2018. Efficient entropy for policy gradient with multi-dimensional action space. arXiv preprint arXiv:1806.00589 (2018).

[M18] Luke Metz, Julian Ibarz, Navdeep Jaitly, and James Davidson. 2018. Discrete Sequential Prediction of Continuous Actions for Deep RL. https://openreview.net/forum?id=r1SuFjkRW

Method: Autoregressive A2C (via MMDP)

- MDP to MMDP
  - $(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1})$ to $(u_0^{s_t}, a_t^1, u_1^{s_t}, a_t^2, \dots, u_{d-1}^{s_t}, a_t^d, u_d^{s_t} = u_0^{s_{t+1}})$
  - Decompose actions. Remodel policies, states.
- Policies + implement
  - $\pi_\theta(\boldsymbol{a}_t|\boldsymbol{s}_t) = \prod_{k=1}^{d} \pi_\theta\left(a_t^k \middle| \overbrace{s_t, \boldsymbol{a}_t^{1:k-1}}^{u_{k-1}^{s_t}}, \overbrace{\boldsymbol{0}^{k:d-1}}^{\text{placeholder}}\right)$
  - $\forall k, \pi_\theta(\cdot\,|u_{k-1}^{s_t}) \in \mathcal{P}(\mathcal{A}^k)$ softmax subpolicies

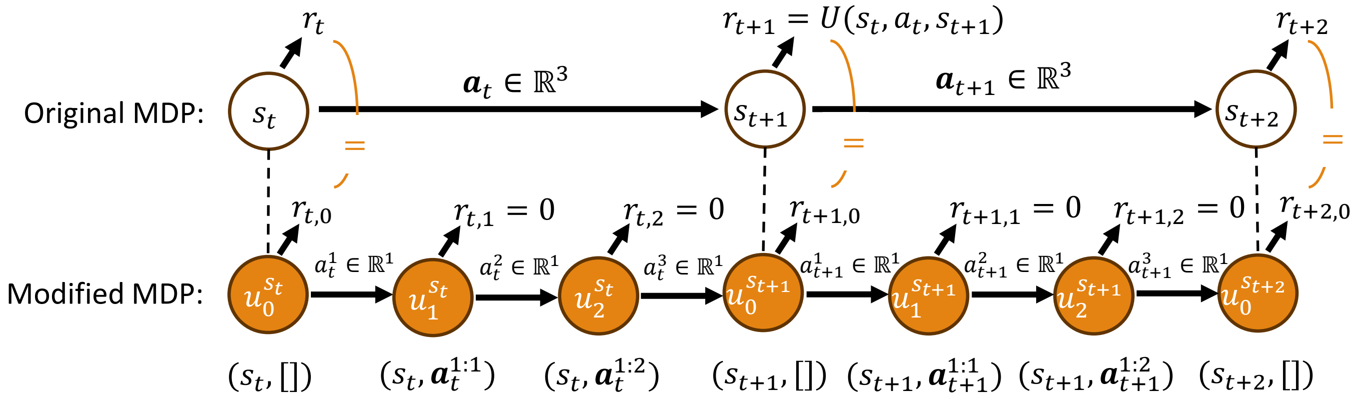


Fig 2: MMDP Example ($d = 3$)[M18]

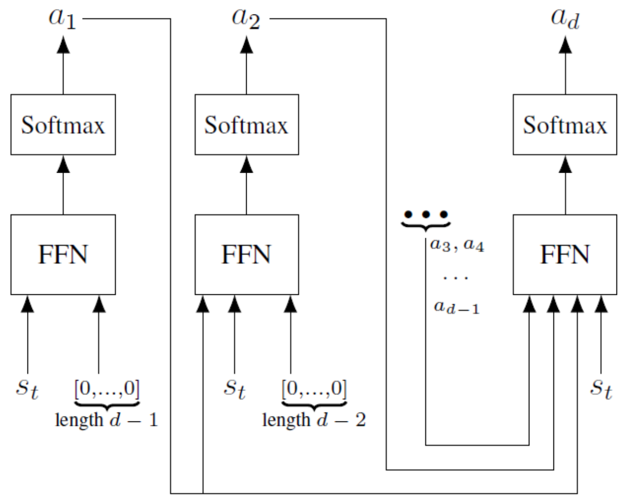


Fig 3: MMDP Architecture[Z18]

[Z18] Yiming Zhang, Quan Ho Vuong, Kenny Song, Xiao-Yue Gong, and Keith W Ross. 2018. Efficient entropy for policy gradient with multi-dimensional action space. arXiv preprint arXiv:1806.00589 (2018).

[M18] Luke Metz, Julian Ibarz, Navdeep Jaitly, and James Davidson. 2018. Discrete Sequential Prediction of Continuous Actions for Deep RL. https://openreview.net/forum?id=r1SuFjkRW

Method: Learned Intrinsic Reward

- Trajectory Generation: MDP to MMDP (with r^{in})

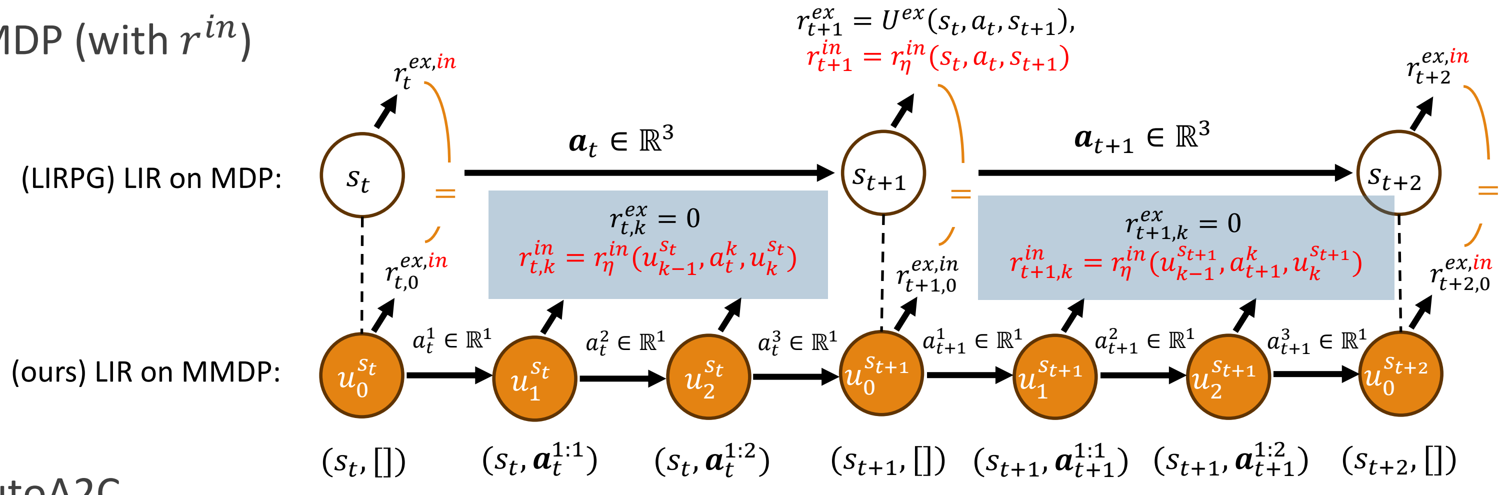


Fig 4: LIR-MMDP Example ($d = 3$)

- Updates: A2C→AutoA2C→AugmAutoA2C
  - Policy $\Delta\theta \propto \nabla J^{ex+in}; J^{ex+in}(\theta) \coloneqq \mathbb{E}_\theta[A^{ex+in}(u^{s_t}_{k-1}, a^k_t, u^{s_t}_k)]$
  - Intrinsic $\Delta\eta \propto \nabla J^{ex}; J^{ex}(\theta(\eta)) \coloneqq \mathbb{E}_{\theta(\eta)}[A^{ex}(u^{s_t}_{k-1}, a^k_t)]$
  - Advantages $A^{ex+in} = (1-\lambda)A^{ex} + \lambda A^{in}$

$$A^{ex}(u^{s_t}_{k-1}, a^k_t) \coloneqq G^{ex}(u^{s_t}_{k-1}, a^k_t) - \hat{V}^{ex}_{\eta_v}(u^{s_t}_{k-1}) \approx \overbrace{\left(\sum_{i=0}^{n-1} \gamma^i r^{ex}_{t+i}\right) + \gamma^n \hat{V}^{ex}_{\eta_v}(u^{s_{t+n}}_0)}^{V^{ex}_{target}(u^{s_t}_{k-1}) \text{ n-step bootstrap}} - \hat{V}^{ex}_{\eta_v}(u^{s_t}_{k-1}),$$

$$A^{in}(u^{s_t}_{k-1}, a^k_t, u^{s_t}_k) \coloneqq G^{in}(u^{s_t}_{k-1}, a^k_t, u^{s_t}_k) - \hat{V}^{in}_{\theta_v}(u^{s_t}_{k-1}) \approx \underbrace{\left(\sum_{j=k}^{d} \gamma^{j-k} r^{in}_{t,j-1} + \sum_{i=1}^{n-1}\sum_{j=1}^{d} \gamma^{id-k+j} r^{in}_{t+i,j-1}\right) + \gamma^{nd-k+1}\hat{V}^{in}_{\theta_v}(u^{s_{t+n}}_0)}_{V^{in}_{target}(u^{s_t}_{k-1}) \text{, MMDP t-scale}} - \hat{V}^{in}_{\theta_v}(u^{s_t}_{k-1})$$

  - Value estimates

$$\Delta\eta_v \propto \nabla MSE(\hat{V}^{ex}_{\eta_v}, V^{ex}_{target}), \Delta\theta_v \propto \nabla MSE(\hat{V}^{in}_{\theta_v}, V^{in}_{target})$$

Method: Learned Intrinsic Reward

- Trajectory Generation: MDP to MMDP (with r^{in})

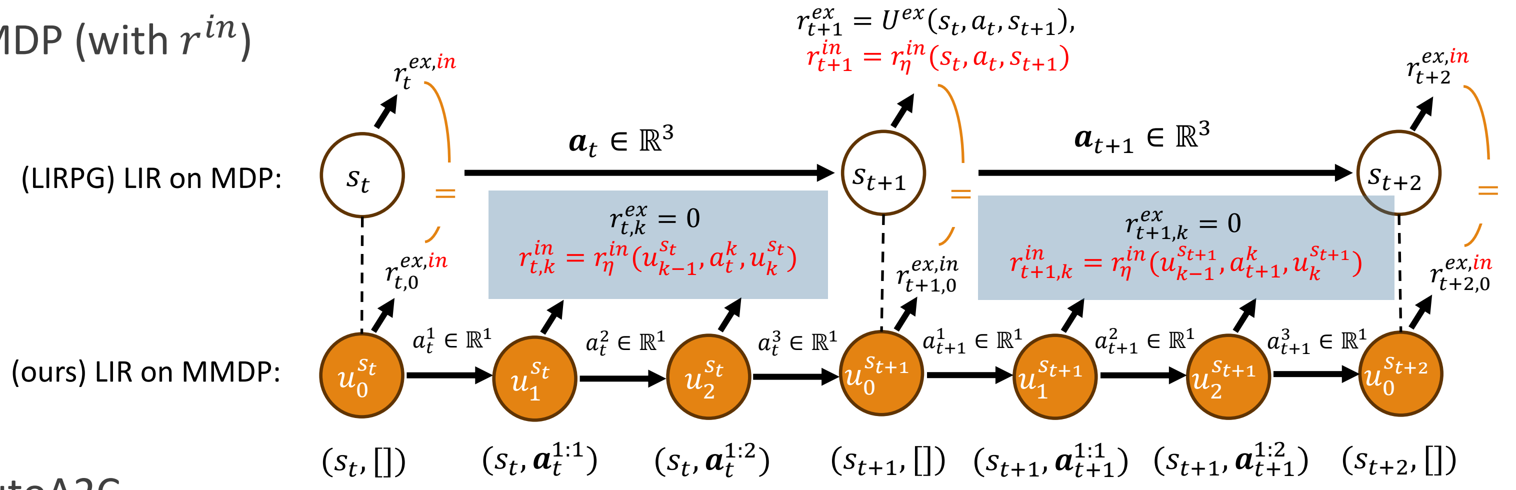


Fig 4: LIR-MMDP Example ($d = 3$)

- Updates: A2C→AutoA2C→AugmAutoA2C
  - Policy $\Delta\theta \propto \nabla J^{ex+in}; J^{ex+in}(\theta) \coloneqq \mathbb{E}_\theta[A^{ex+in}(u^{s_t}_{k-1}, a^k_t, u^{s_t}_k)]$
  - Intrinsic $\Delta\eta \propto \nabla J^{ex}; J^{ex}(\theta(\eta)) \coloneqq \mathbb{E}_{\theta(\eta)}[A^{ex}(u^{s_t}_{k-1}, a^k_t)]$
  - Advantages $A^{ex+in} = (1-\lambda)A^{ex} + \lambda A^{in}$

$$A^{ex}(u^{s_t}_{k-1}, a^k_t) \coloneqq G^{ex}(u^{s_t}_{k-1}, a^k_t) - \hat{V}^{ex}_{\eta_v}(u^{s_t}_{k-1}) \approx \overbrace{\left(\textstyle\sum_{i=0}^{n-1} \gamma^i r^{ex}_{t+i}\right) + \gamma^n \hat{V}^{ex}_{\eta_v}(u^{s_{t+n}}_0)}^{V^{ex}_{target}(u^{s_t}_{k-1}) \text{ n-step bootstrap}} - \hat{V}^{ex}_{\eta_v}(u^{s_t}_{k-1}),$$

$$A^{in}(u^{s_t}_{k-1}, a^k_t, u^{s_t}_k) \coloneqq G^{in}(u^{s_t}_{k-1}, a^k_t, u^{s_t}_k) - \hat{V}^{in}_{\theta_v}(u^{s_t}_{k-1}) \approx \underbrace{\left(\textstyle\sum_{j=k}^{d} \gamma^{j-k} r^{in}_{t,j-1} + \sum_{i=1}^{n-1}\sum_{j=1}^{d} \gamma^{id-k+j} r^{in}_{t+i,j-1}\right) + \gamma^{nd-k+1}\hat{V}^{in}_{\theta_v}(u^{s_{t+n}}_0)}_{V^{in}_{target}(u^{s_t}_{k-1}) \text{, MMDP t-scale}} - \hat{V}^{in}_{\theta_v}(u^{s_t}_{k-1})$$

  - Value estimates

$$\Delta\eta_v \propto \nabla MSE(\hat{V}^{ex}_{\eta_v}, V^{ex}_{target}), \Delta\theta_v \propto \nabla MSE(\hat{V}^{in}_{\theta_v}, V^{in}_{target})$$

Method: Learned Intrinsic Reward

- Trajectory Generation: MDP to MMDP (with r^{in})

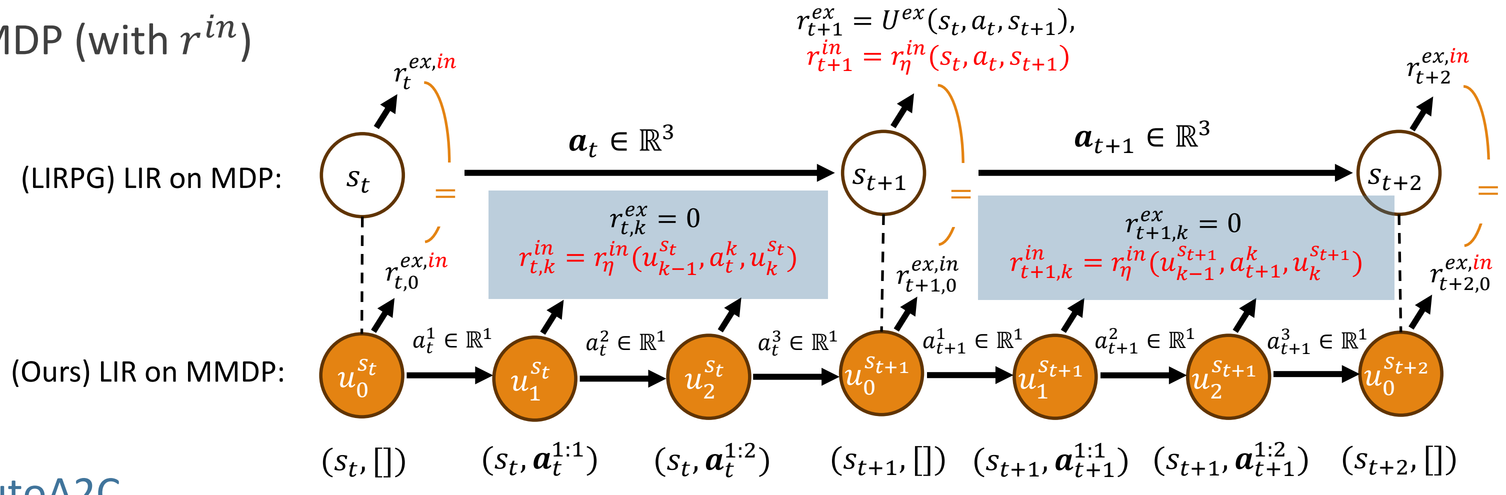


Fig 4: LIR-MMDP Example ($d = 3$)

- Updates: A2C→AutoA2C→AugmAutoA2C
  - Policy $\Delta\theta \propto \nabla J^{ex+in}; J^{ex+in}(\theta) \coloneqq \mathbb{E}_\theta[A^{ex+in}(u^{s_t}_{k-1}, a^k_t, u^{s_t}_k)]$
  - Intrinsic $\Delta\eta \propto \nabla J^{ex}; J^{ex}(\theta(\eta)) \coloneqq \mathbb{E}_{\theta(\eta)}[A^{ex}(u^{s_t}_{k-1}, a^k_t)]$
  - Advantages $A^{ex+in} = (1-\lambda)A^{ex} + \lambda A^{in}$

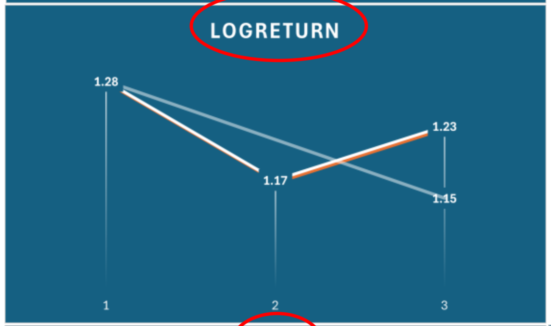
$$A^{ex}(u^{s_t}_{k-1}, a^k_t) \coloneqq G^{ex}(u^{s_t}_{k-1}, a^k_t) - \hat{V}^{ex}_{\eta_v}(u^{s_t}_{k-1}) \approx \overbrace{\left(\sum_{i=0}^{n-1} \gamma^i r^{ex}_{t+i}\right) + \gamma^n \hat{V}^{ex}_{\eta_v}(u^{s_{t+n}}_0)}^{V^{ex}_{target}(u^{s_t}_{k-1}) \text{ n-step bootstrap}} - \hat{V}^{ex}_{\eta_v}(u^{s_t}_{k-1}),$$

$$A^{in}(u^{s_t}_{k-1}, a^k_t, u^{s_t}_k) \coloneqq G^{in}(u^{s_t}_{k-1}, a^k_t, u^{s_t}_k) - \hat{V}^{in}_{\theta_v}(u^{s_t}_{k-1}) \approx \underbrace{\left(\sum_{j=k}^{d} \gamma^{j-k} r^{in}_{t,j-1} + \sum_{i=1}^{n-1}\sum_{j=1}^{d} \gamma^{id-k+j} r^{in}_{t+i,j-1}\right) + \gamma^{nd-k+1}\hat{V}^{in}_{\theta_v}(u^{s_{t+n}}_0)}_{V^{in}_{target}(u^{s_t}_{k-1}),\ \text{MMDP t-scale}} - \hat{V}^{in}_{\theta_v}(u^{s_t}_{k-1})$$

  - Value estimates

$$\Delta\eta_v \propto \nabla MSE(\hat{V}^{ex}_{\eta_v}, V^{ex}_{target}), \Delta\theta_v \propto \nabla MSE(\hat{V}^{in}_{\theta_v}, V^{in}_{target})$$

Experiments: Effect of Intrinsic Rewards given Fixed Extrinsic Rewards



- Recall:
  - RL Objective $J(\theta; \overbrace{r^{ex+in}}^{(1-\lambda)r^{ex}+\lambda r_\eta^{in}}) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t^{ex+in} \middle| s_t \sim T(\cdot|s_{t-1}, a_{t-1}), a_t \sim \pi_\theta(\cdot|s_t)\right]$
  - External rewards $\mathrm{r}_{t+1}^{ex} = U^{ex}(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1})$
  - In finance: $U^{ex} \in$ {Profit, logReturn, $\underbrace{\text{CSR[1], DSR1[2], DSR2[3,[M98]]}}_{\text{Sharpe-motivated}}$}

$U(s_t, a_t, s_{t+1}) = (b_{t+1} + p_{t+1}^T h_{t+1}) - (b_t + p_t^T h_t)$

$U(s_t, a_t, s_{t+1}) = logR_t,$

Returns $R_t := \frac{b_{t+1} + p_{t+1}^T h_{t+1}}{b_t + p_t^T h_t} - 1.$

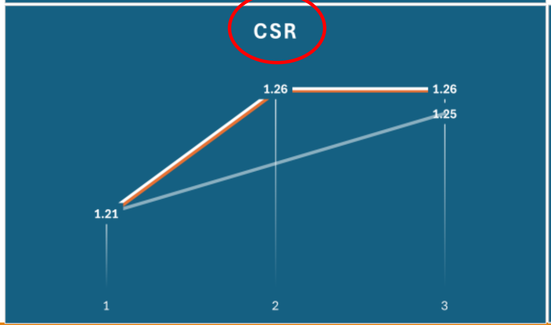
$U(s_{0:t}, a_{0:t}, s_{t+1}) = SR_{t+1} = \frac{E[\mathrm{R}_t]}{Std[\mathrm{R}_t]},$

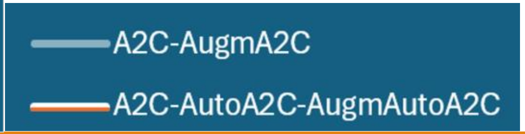
estimated from past returns $\Gamma_{t+1} := \mathrm{R}_{t-62:t}$

$U(s_{0:t}, a_{0:t}, s_{t+1}) = SR_{t+1} - SR_t$

$U(s_{0:t}, a_{0:t}, s_{t+1}) = \frac{B_t \Delta A_{t+1} - 1/2. A_t \Delta B_{t+1}}{(B_t - A_t^2)^{3/2}},$

A_t, B_t are exponential moving estimates of $E[\mathrm{R}_t], E[R_t^2]$ calculated with Γ_{t+1}.

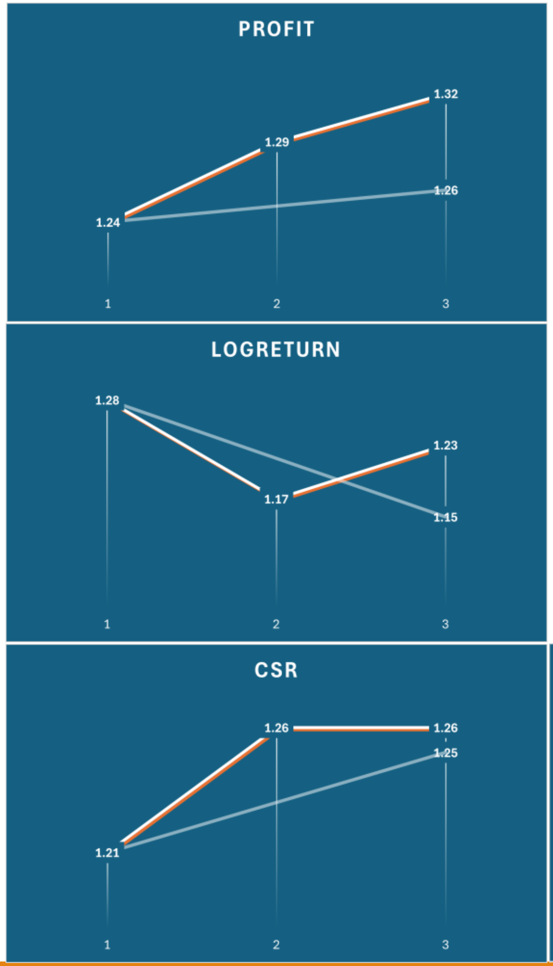




[M98] John Moody, Matthew Saffell, Yuansong Liao, and Lizhong Wu. 1998. Reinforcement learning for trading systems and portfolios: Immediate vs future rewards. Springer US, 129–140.

1 Cumulative Sharpe Ratio
2 Difference in Cumulative Sharpe Ratios
3 Differential Sharpe Ratios

Experiments: Effect of Intrinsic Rewards given Fixed Extrinsic Rewards



- Recall:
  - RL Objective $J(\theta; r^{ex+in}) := \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_t^{ex+in} | s_t \sim T(\cdot | s_{t-1}, a_{t-1}), a_t \sim \pi_\theta(\cdot | s_t)]$
  - External rewards $r_{t+1}^{ex} = U^{ex}(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1})$
  - In finance: $U^{ex} \in$ {Profit, logReturn, CSR[1], DSR1[2], DSR2[3,[M98]]}

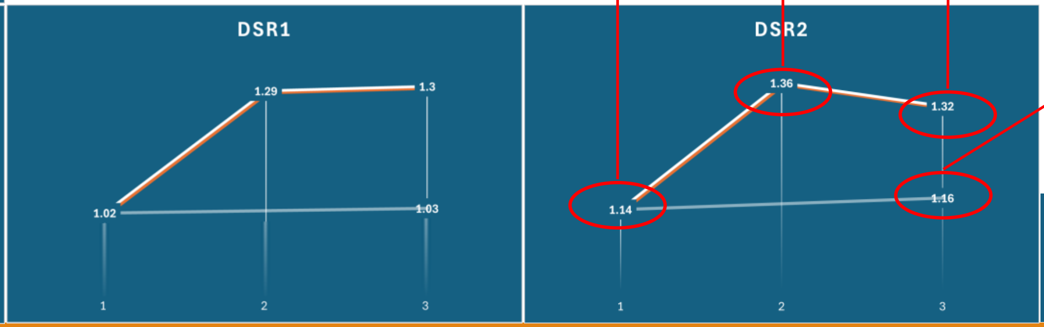
Sharpe Ratios of

A2C

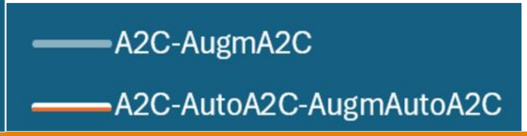
AutoA2C

LIR + AutoA2C (ours)

LIR + A2C (LIRPG)

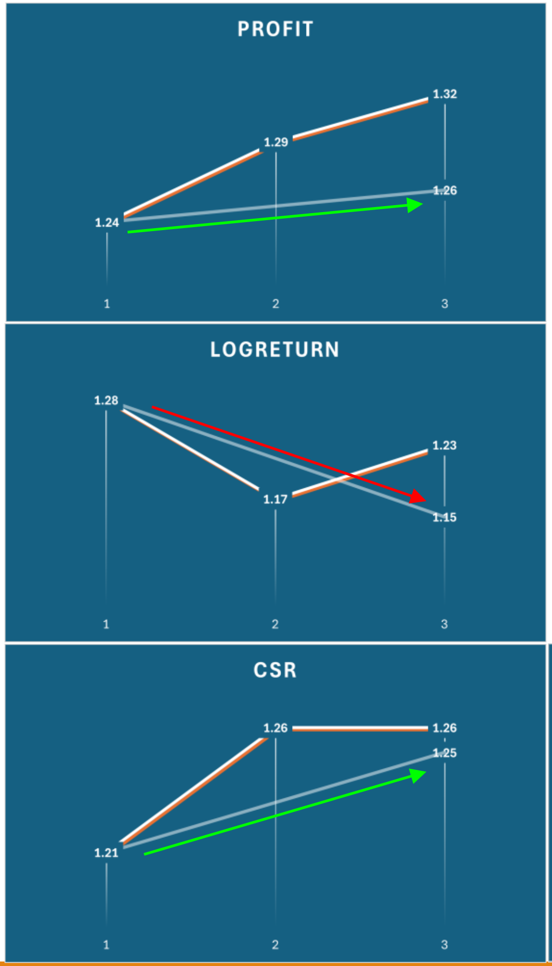


Mean-Var	0.77
DJI	0.55

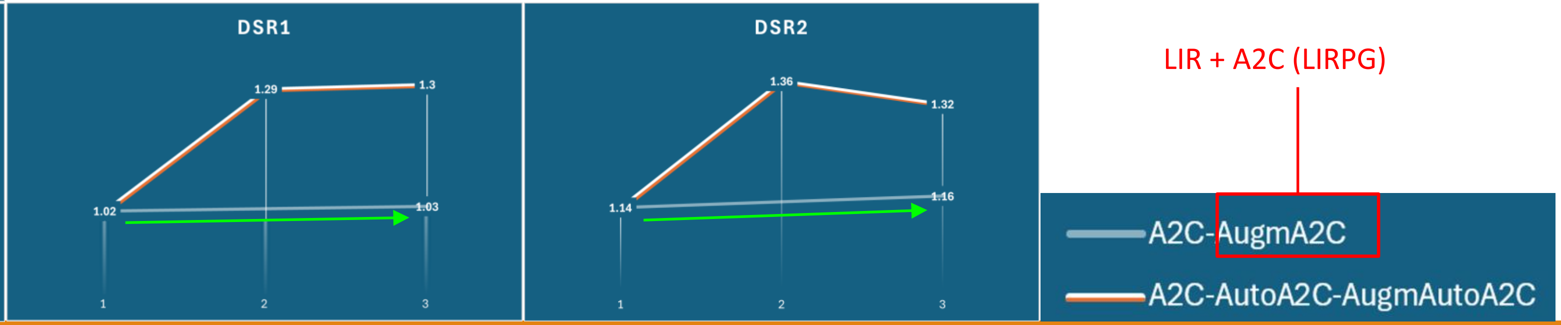


1 Cumulative Sharpe Ratio
2 Difference in Cumulative Sharpe Ratios
3 Differential Sharpe Ratios

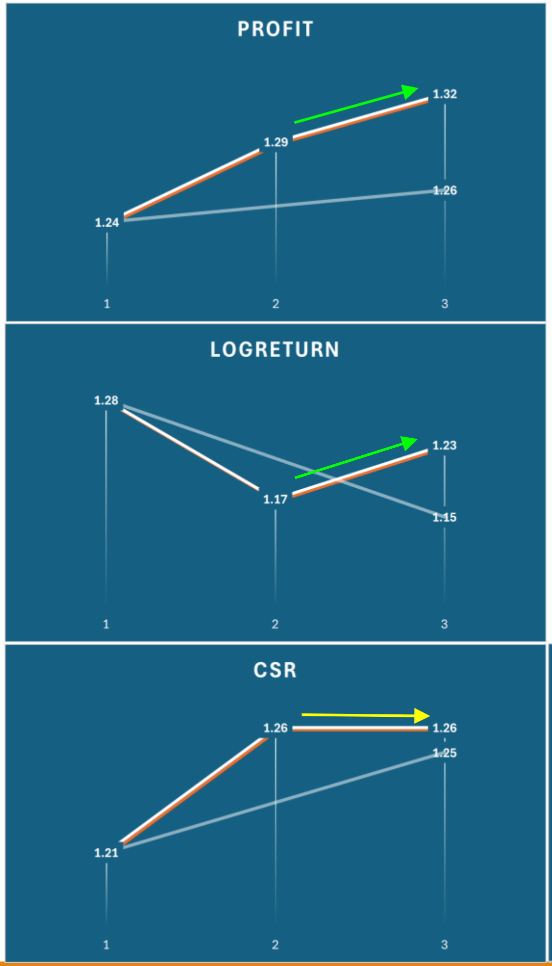
Experiments: Effect of Intrinsic Rewards given Fixed Extrinsic Rewards



- ❑ LIR on A2C ⇛ 4/5 Sharpe improved
- ❑ LIR on AutoA2C ⇛ 4/5 Sharpe improved (strictly, 3/5)
- ❑ Auto on LIR + A2C ⇛ 5/5 Sharpe improved



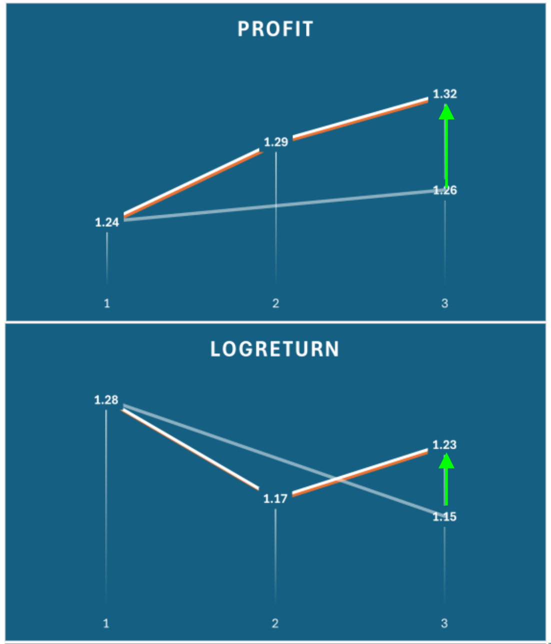
Experiments: Effect of Intrinsic Rewards given Fixed Extrinsic Rewards

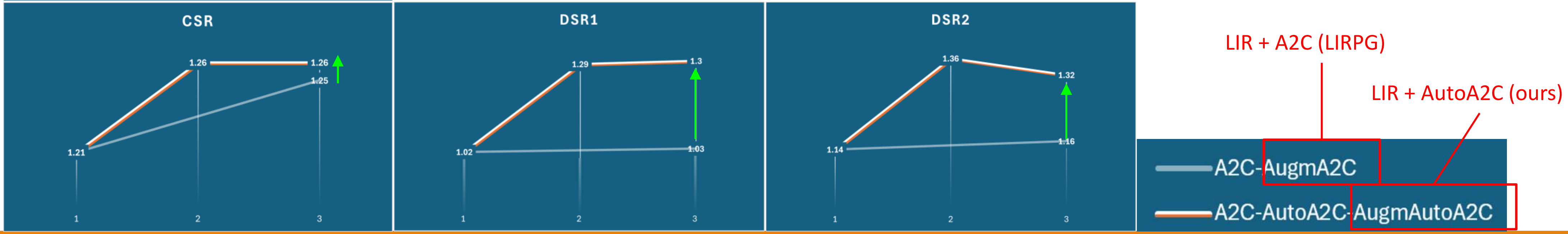


- ❑ LIR on A2C ⇛ 4/5 Sharpe improved
- ❑ LIR on AutoA2C ⇛ 4/5 Sharpe improved (strictly, 3/5)
- ❑ Auto on LIR + A2C ⇛ 5/5 Sharpe improved

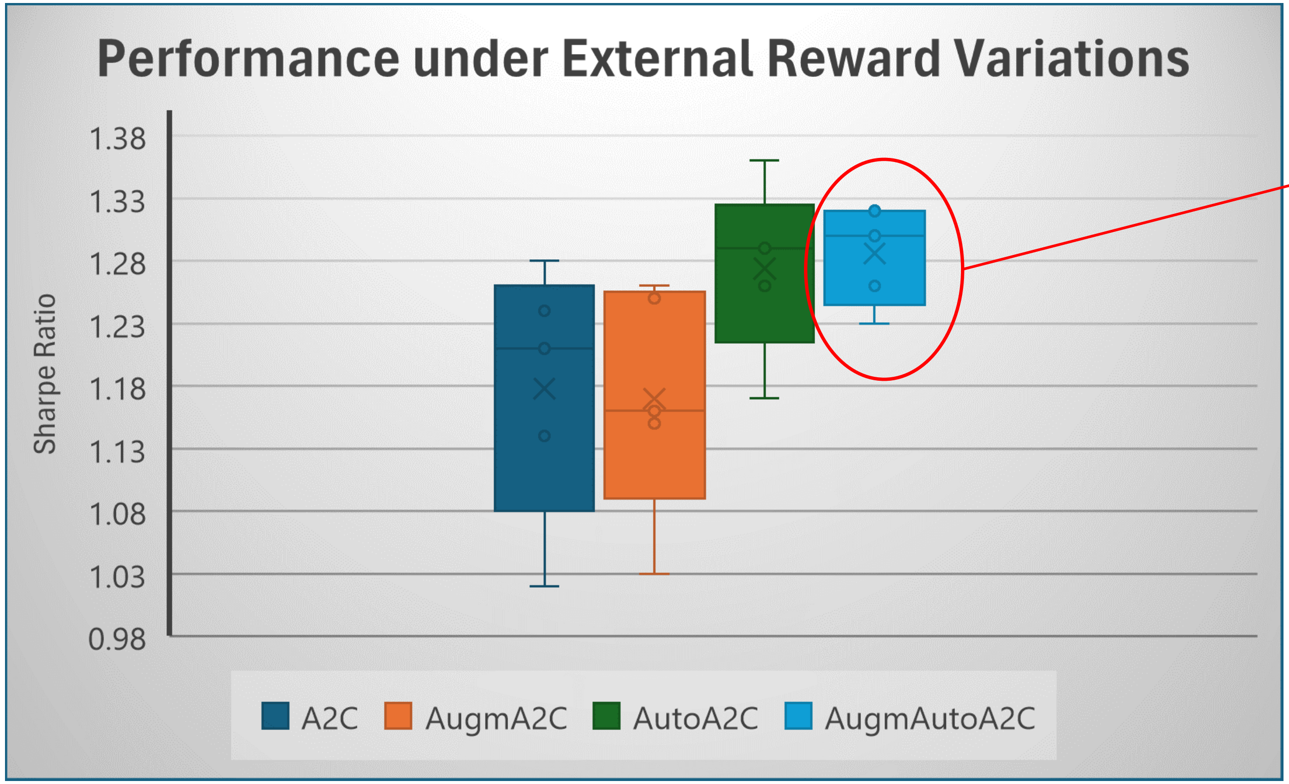
Experiments: Effect of Intrinsic Rewards given Fixed Extrinsic Rewards

- ❑ LIR on A2C ⇛ 4/5 Sharpe improved
- ❑ LIR on AutoA2C ⇛ 4/5 Sharpe improved (strictly, 3/5)
- ❑ Auto on LIR + A2C ⇛ 5/5 Sharpe improved





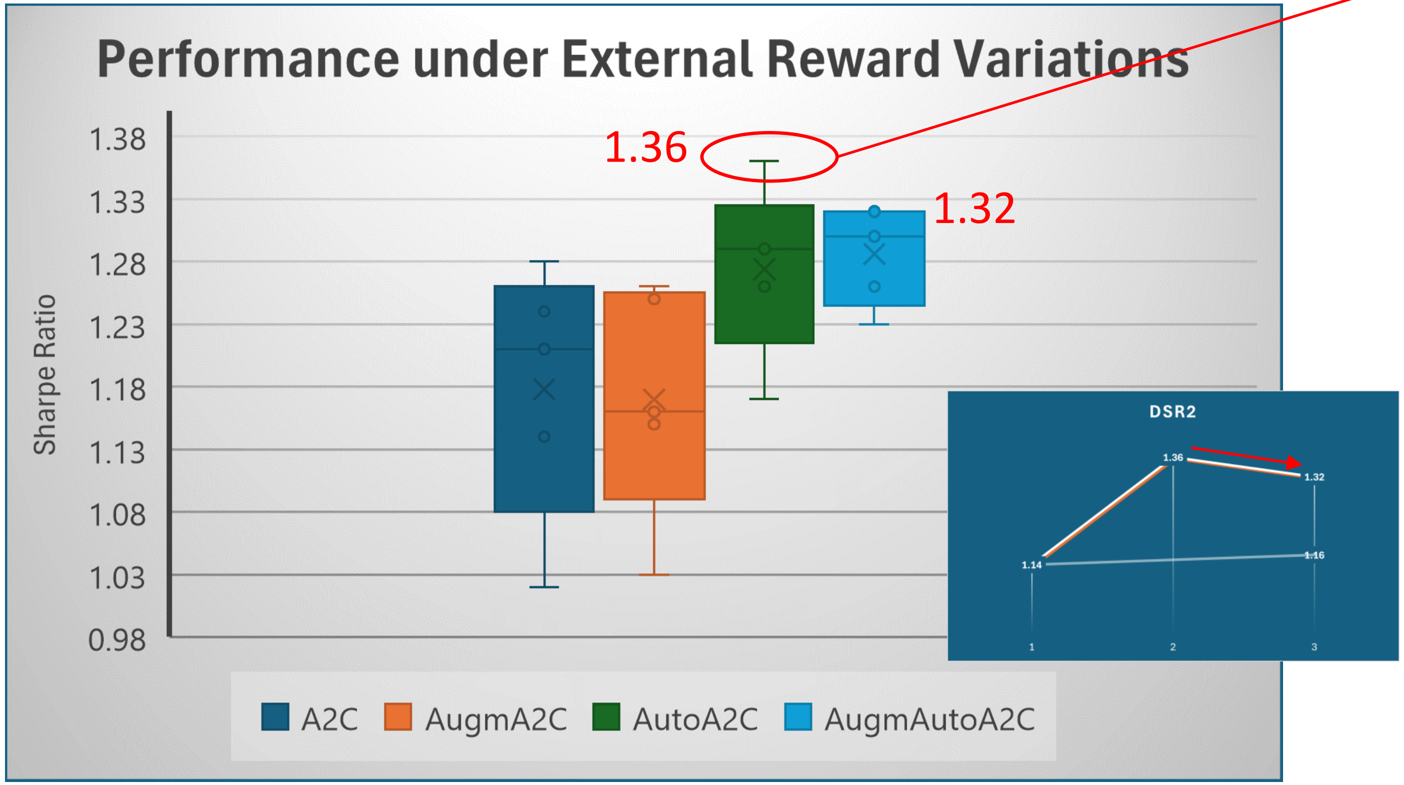
Experiments: Best Strategies (Algorithm x Extrinsic Reward)



AugmAutoA2C

- Best Sharpe statistics; robustness to r^{ex} choices.
- Ease r^{ex} design.

Experiments: Best Strategies (Algorithm x Extrinsic Reward)



AutoA2C x DSR2

- If willing to design r^{ex}, AutoA2C can gain highest Sharpe.
- Note: DSR2 is the most complicated r^{ex} out of 5 considered; has additional tuning parameter.
- Compare with AugmAutoA2C's max Sharpe: r^{ex} is simply Profit.

DSR2: LIR degrades AutoA2C

- DSR2 is close to optimal r for Sharpe under the agent bounds / bounds are minimal given Auto
- r^{in} has no meaningful improve direction
- r^{in} incurs training cost (hyperparameter, accuracy)

Conclusion

SUMMARY CONTRIBUTION

- adapted the idea of learned intrinsic rewards, paired with autoregressive RL, to portfolio optimization
- empirically studied the effect of learned intrinsic rewards under different
  - training objectives: Sharpe-motivated U^{ex}

    $\Rightarrow r^{in}$ improves Sharpe + robustness across U^{ex}
  - r^{in} structure: standard vs autoregressive

    $\Rightarrow$ autoregressive $>$ standard

FUTURE WORKS

- formalize cost-benefit analysis of r^{in} learning frameworks
  - Improvement direction given r^{ex} + DRL structure
  - r^{in} cost of learning given accuracy, hyperparameters
- explore alternative
  - r^{in} learning architectures (e.g., hybrid[vS17])
  - π structures (e.g., taking into account the order of subaction executions)
- generalize and scale up
  - beyond Portfolio Optimization task
  - higher dimensions (e.g., > 30 action-dims)

[vS17] Harm van Seijen, Mehdi Fatemi, Joshua Romoff, Romain Laroche, Tavian Barnes, and Jeffrey Tsang. 2017. Hybrid Reward Architecture for Reinforcement Learning. CoRR abs/1706.04208 (2017).

Thank You

ACM Paper Link: https://doi.org/10.1145/3677052.3698670

Full codes: https://github.com/cspun/ADRLwIntReward/

Correspondence: **Nixie S Lesmana**
Research Fellow, National University of Singapore, Singapore
nixiesap@nus.edu.sg | nixiesap001@e.ntu.edu.sg

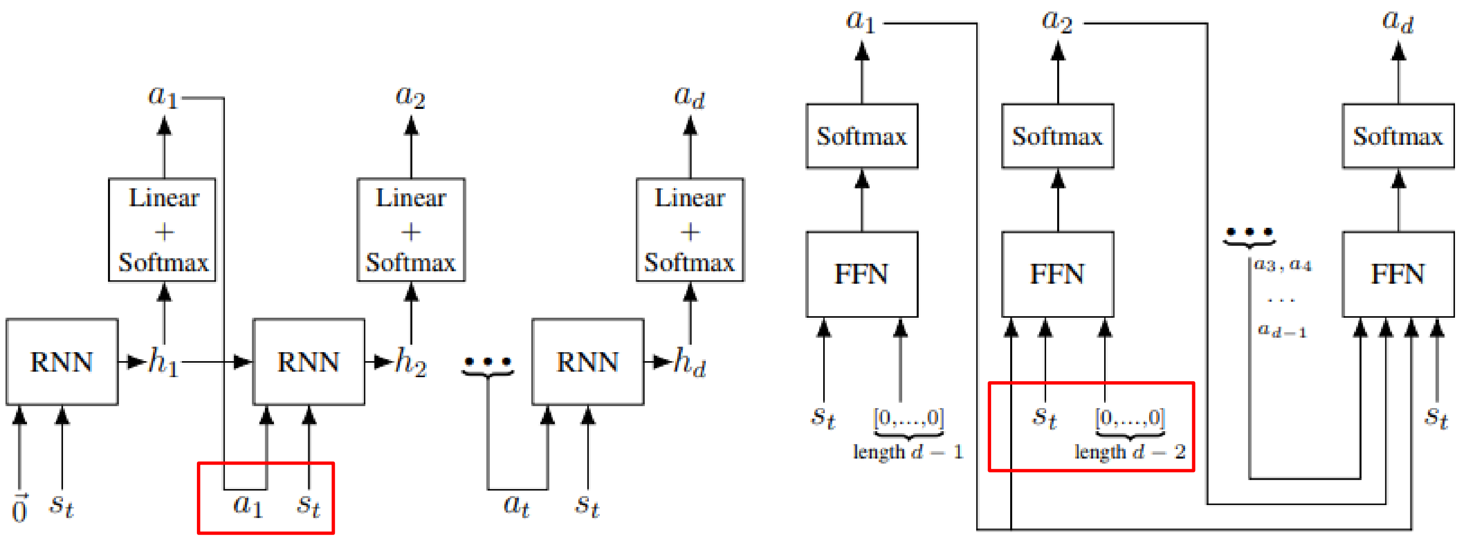
Acknowledgement:

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

NUS National University of Singapore

Chi Seng Pun gratefully acknowledges Ministry of Education (MOE) Singapore, AcRF Tier 2 grant (Reference No.: MOE-T2EP20220-0013)for the funding of this research. **Nixie S Lesmana** acknowledges the financial support from MoE AcRF grant R-144-000-457-733(A-0004550-00-00

Appendix: AutoA2C via MMDP vs RNN



(a) The RNN architecture. To generate a_i, we input s_t and a_{i-1} into the RNN and then pass the resulting hidden state h_i through a linear layer and a softmax to generate a distribution, from which we sample a_i.

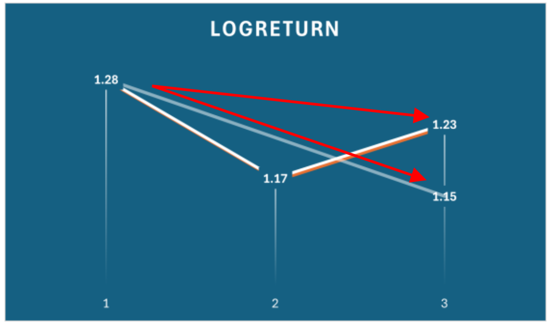
(b) The MMDP architecture. To generate a_i, we input s_t and $a_1, a_2, \ldots, a_{i-1}$ into a FFN. The output is passed through a softmax layer, providing a distribution from which we sample a_i. Since the input size of the FFN is fixed, when generating a_i, constants 0 serve as placeholders for $a_{i+1}, \ldots, a_{d-1}$ in the input to the FFN.

- RNN suffers from exploding gradients ⇛ inconsistent performance.
- When gradients do not explode, attains similar performance as MMDP.
- RNN takes longer to train.

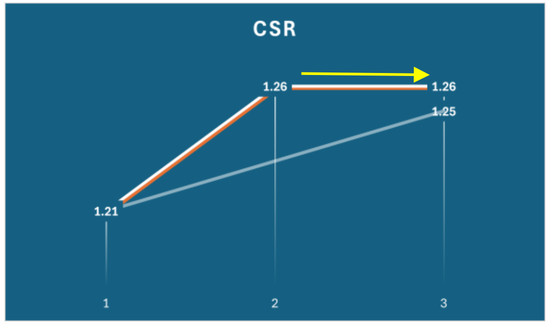
Fig 5: Implementation of Autoregressive Policies. (a) RNN (b) MMDP. [Z18]

[Z18] Yiming Zhang, Quan Ho Vuong, Kenny Song, Xiao-Yue Gong, and Keith W Ross. 2018. Efficient entropy for policy gradient with multi-dimensional action space. arXiv preprint arXiv:1806.00589 (2018).

Appendix: When LIR and AutoReg Does Not Improve



- No improve direction in both Auto and LIR. However, $\exists r^{ex}$ (e.g., Profit) where improve direction appears.
- r^{ex} :=logReturn is not yet optimal; instead, it seems to be some stationary point in r-space.
- No improvement maybe related to r^{ex} (logged) scale too.



- A2C Agent's bound is policy structure; LIR repairs this.
- After Auto, there is less (or none) to repair: harder to find improve direction for r^{in} (esp. as r^{in} needs to be learned).

Appendix: Other Financial Metrics

Reward function	Algorithm	Mean				Median			
		Sharpe	Vol	CAGR	Max DD	Sharpe	Vol	CAGR	Max DD
Profit	Augmented AutoA2C	1.32	6.51	8.69	−6.92	1.32	6.73	8.72	−6.76
	AutoA2C	1.29	7.03	9.33	−7.70	1.20	6.96	9.24	−7.73
	Augmented A2C	1.26	7.28	9.21	-6.65	1.28	7.15	8.87	-6.27
	A2C	1.24	7.16	9.03	−7.36	1.28	7.21	9.42	−7.43
Log(return)	Augmented AutoA2C	1.23	7.23	8.76	−8.90	1.30	6.99	9.72	−7.25
	AutoA2C	1.17	7.46	8.55	−9.15	1.26	7.04	9.15	−7.92
	Augmented A2C	1.15	7.39	8.55	−6.95	1.12	7.26	8.73	−6.41
	A2C	1.28	7.22	9.35	-6.40	1.27	7.34	9.40	-6.12
CSR	Augmented AutoA2C	1.26	6.61	8.47	−7.36	1.29	6.62	8.16	−7.63
	AutoA2C	1.26	7.84	10.08	−7.93	1.28	7.81	10.09	−7.70
	Augmented A2C	1.25	7.36	9.32	−7.13	1.34	7.36	10.11	−7.36
	A2C	1.21	7.34	8.92	-6.75	1.24	7.12	9.27	-6.62
DSR1	Augmented AutoA2C	1.30	6.91	9.17	−6.99	1.35	6.85	9.46	−7.30
	AutoA2C	1.29	7.70	10.11	−7.84	1.30	7.80	9.91	−7.48
	Augmented A2C	1.03	7.32	7.48	−7.19	1.08	7.22	7.79	−6.98
	A2C	1.02	7.42	7.55	-6.83	1.01	7.42	7.59	-6.64
DSR2	Augmented AutoA2C	1.32	7.20	9.76	−7.47	1.29	7.01	9.57	−7.77
	AutoA2C	1.36	7.11	10.02	−7.45	1.35	7.27	10.02	−7.61
	Augmented A2C	1.16	7.44	8.69	−7.33	1.14	7.30	8.55	−7.37
	A2C	1.14	7.22	8.28	-7.18	1.15	7.17	8.41	-6.86
	Mean-Var	0.77	23.30	16.30	−35.20	0.77	23.30	16.30	−35.20
	DJI	0.55	20.50	9.54	−37.10	0.55	20.50	9.54	−37.10

(1) *Annualized volatility* ("AVOL") is a measure of the average annualized risk of a strategy: $AVOL_T = \sigma_T \times \sqrt{252}$, where σ_T is the standard deviation of inter-day portfolio returns over T time periods.

(2) *Maximum drawdown* ("max DD") is the maximum loss between the portfolio value peak and the lowest point until next peak. It is an alternative way of measuring the risk of a strategy.

(3) *Compound annual growth rate* ("CAGR") is the annualized growth rate of return of a portfolio over a period of more than one year, given by CAGR $= (W_T/W_0)^{1/T} - 1$. CAGR smooths out the actual volatile growth rate each year by assuming this value is constant for each year.

(4) *Sharpe ratio* ("Sharpe") is the annualized average inter-day return in excess of the risk-free return per unit of volatility:

$$\text{Sharpe}_t = \frac{\mu_t - r_f}{\sigma_t}, \quad (15)$$

where μ_t and σ_T are the mean and standard deviation of inter-day portfolio returns and r_f is a risk-free rate. The Sharpe ratio evaluates the risk-adjusted returns of a strategy, requiring a good strategy to balance both profit and risk. This metric favours a far-sighted steady investment strategy, over a short-sighted strategy with short-term high profits.