CASA-RL: EXACT SET POLICIES FOR SPARSE PORTFOLIO CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse portfolio reinforcement learning couples unordered asset selection with continuous allocation, while financial priors complicate attribution of learned gains. We introduce CASA-RL, a framework for *Cardinality-Aware Sparse Allocation* built around three requirements. First, an exactly normalized fixed-cardinality distribution assigns probability directly to the selected asset set. Second, a support-conditioned logistic-normal allocation yields feasible stock/cash weights and an explicit joint likelihood for PPO. Third, matched initialization, support-prior ablations, and checkpoint comparisons separate prior-based performance from learned changes. In a 45-run study on reconstructed historical S&P 500 constituents, the validation-selected configurations of CASA-RL and two ordered-policy implementations achieve approximately **315.92%** mean cumulative net return on a retrospective test interval. The selected trained policies match their initialized counterparts' stock supports on every test day. Removing the explicit support prior exposes substantial policy changes and checkpoint-dependent method rankings. The resulting framework aligns feasible portfolio actions, likelihood-based learning, and empirical attribution.

1 INTRODUCTION

At each rebalance, a portfolio agent faces three connected decisions: which stocks to hold, how much capital to commit to each, and how much to retain as cash. Together, these decisions determine the investment that enters the market. Consider a portfolio with 55% in one stock, 35% in another, and 10% in cash. Listing the two stocks in a different order leaves this portfolio unchanged; shifting capital between them changes its exposure. Sparse portfolio control makes this structure explicit: the agent chooses exactly K stocks with positive weights, assigns zero to all others, and completes the unit budget with cash, which is additional to K.

Reinforcement learning (RL) connects these decisions across time through trading costs and future holdings. Today's allocation affects both immediate net return and the positions from which subsequent trades begin. Classical portfolio optimization encodes risk, return, and cardinality as objectives and constraints(Markowitz (1952); Chang et al. (2000)); portfolio RL learns sequential trading and allocation rules(Moody & Saffell (2001); Jiang et al. (2017)). Bringing these perspectives together requires a probability model that follows the structure of the feasible action: an unordered stock support with continuous weights. The selected set determines which coordinates can carry capital, and the allocation determines the exposure on those coordinates.

Policy updates must score the portfolio that earned the reward, and evaluation must identify changes from the initialized strategy. We therefore organize sparse portfolio RL around three linked requirements: represent the executed portfolio, evaluate its joint likelihood, and identify the learned adjustments. Three limitations motivate the design in Figure 1.

L1: the represented choice differs from the portfolio support. Dense allocations require an additional rule to enforce a position count.Plackett–Luce (PL(Plackett (1975))), provides a valid likelihood for a retained sequence, but several sequences represent the same stock set. Neither representation directly makes the unordered, exactly-K support the object assigned probability.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

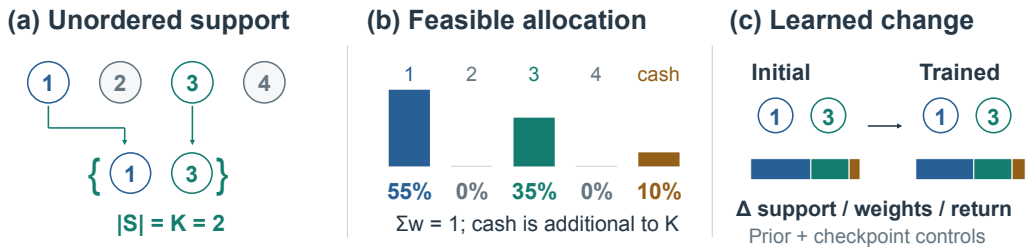

Figure 1: Three requirements for sparse portfolio reinforcement learning.

L2: selection alone does not specify the joint action likelihood. After choosing stocks, the agent must allocate capital on the corresponding simplex face. A support sampler or differentiable selection surrogate(Jang et al. (2017); Maddison et al. (2017)) does not by itself define a density for those weights. Likelihood-based learning needs both terms to score the same stored portfolio under the old and updated policies.

L3: total return does not identify the learned adjustment. Financial priors can define a strong starting strategy. Total return mixes initialization with learned support and allocation changes; checkpoint selection determines which policy is reported. Final scores alone leave the contribution of learning unresolved.

The key insight is to carry the same portfolio object through representation, learning, and evaluation. Figure 1 follows the two-stock example from its unordered support to a feasible stock/cash allocation and an initial–trained comparison. CASA-RL combines a weighted fixed-size subset law with a conditional logistic-normal allocation(Chen et al. (1994); Aitchison & Shen (1980)). An elementary symmetric polynomial (ESP) recurrence aggregates the weights of feasible supports and enables exact sampling without enumerating every set. Cash-reference coordinates then map continuous latent variables to feasible allocations with an explicit density. The support mass and conditional density form a single portfolio likelihood, allowing both decisions to learn from the same reward.

Our evaluation makes the distinction between portfolio performance and learned change concrete. Removing the explicit support prior reveals substantial changes in holdings; in this regime, validation-selected and terminal checkpoints produce different method rankings. Our results show why the starting policy and checkpoint rule are essential to interpreting what an RL agent learns. Section 2.2 translates limitations(L1–L3) into paired motivations and challenges.

- **C1: An exact policy on unordered supports.** An ESP normalizer and backward sampler assign mass directly to feasible K-stock sets, addressing L1. We derive normalization, exact sampling, and inclusion-score identities that connect each asset's learning signal to its inclusion probability (Section 3.2).
- **C2: A joint likelihood for feasible portfolios.** A support-conditioned logistic-normal head allocates capital over the selected stocks and cash. Its density combines with the set mass to evaluate the same stored action in PPO(Schulman et al. (2017)), addressing L2 (Sections 3.3–3.4).
- **C3: An attribution-based evaluation of policy learning.** A 45-run comparison pairs shared architectures with initialization, support-prior, and checkpoint controls, addressing L3 (Section 5). It distinguishes prior-anchored performance from learned changes in selection and allocation, and identifies the dependence of method rankings on checkpoint choice.

2 Problem Formulation and Challenges

2.1 Sparse Portfolio Control

At decision time t, the observation x_t contains causal market features, the eligible asset set A_t, and current holdings. Let $N_t = |A_t| \geq K$. A feasible action comprises an unordered support $S_t \subseteq A_t$ of

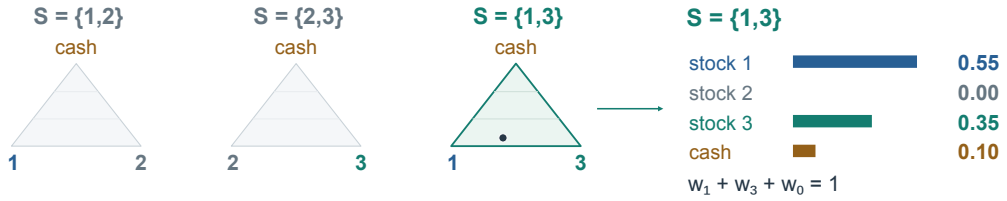

Figure 2: A support selects the simplex face on which capital is allocated.

size K and weights w_t on the corresponding simplex face:

$$w_{t,0} > 0, \quad w_{t,i} > 0 \ (i \in S_t), \quad w_{t,i} = 0 \ (i \notin S_t), \quad w_{t,0} + \sum_{i \in S_t} w_{t,i} = 1. \tag{1}$$

Cash is coordinate zero and is additional to the K stock positions. The action space is the union of their relative interiors over all eligible K-subsets. Trading costs and price movements update wealth and holdings, so an action also changes the context of the next decision. We optimize $\mathbb{E}_\pi[\sum_t \gamma^t R_t]$, where R_t sums daily net log-wealth rewards over one decision interval and K is prescribed.

Sequential investment connects portfolio control(Merton (1971)) with universal and online portfolio selection(Cover (1991); Li & Hoi (2014)). Sparsity penalties and norm constraints shape the allocation(Brodie et al. (2009); Gotoh & Takeda (2011)); here exactly K stocks belong to the policy's action domain. Parameterized-action RL likewise combines discrete and continuous decisions(Masson et al. (2016); Hausknecht & Stone (2016); Li et al. (2022)). A sparse portfolio specializes this structure: its support identifies the face on which its allocation density is defined.

Figure 2 shows how choosing a support determines the coordinates available for capital allocation. With three eligible stocks and $K = 2$, each support identifies a stock/stock/cash simplex; the illustrated interior point assigns $(0.55, 0.35, 0.10)$ to stocks 1, 3, and cash, with zero weight on stock 2.

2.2 Motivations and Design Challenges

M1: make the executed support the probabilistic object (L1). D1: normalize and sample an unordered, exactly-K choice. The policy must assign one mass to each feasible set while respecting eligibility. CASA-RL uses an additive set energy and an ESP normalizer. Its law differs from the PL set marginal; temperature-one hard Gumbel ranking and sequential PL both implement the ordered PL law(Kool et al. (2019a)). This is a change in the modeled action distribution, not a claim that ordered policies are invalid.

M2: learn selection and capital allocation as one action (L2). D2: preserve feasibility and an evaluable density across supports. Each selected set defines different active weight coordinates. The allocation must keep those stocks and cash positive, satisfy the budget, and expose the density needed to compare old and new policies. Cash-reference logistic-normal coordinates provide that density, which combines with the support mass in a single likelihood ratio.

M3: identify what training adds to the starting portfolio (L3). D3: separate initialization, prior strength, and checkpoint choice. Financial priors and model selection can change the interpretation of the same final-return metric. Matched initial parameters, a declared support-prior sweep, and both validation-selected and terminal checkpoints make these effects observable. The encoder, allocation head, and execution rules remain common across methods.

3 CASA-RL: A Joint Policy on Sets and Allocations

3.1 Joint Parameterization and Contextual Asset Scores

CASA-RL follows the structure of the action space: draw a stock support, then allocate capital within its simplex face. The joint policy factorizes as

$$\pi_\theta(S, w \mid x, K) = p_\theta(S \mid x, K)\, q_\theta(w \mid S, x). \tag{2}$$

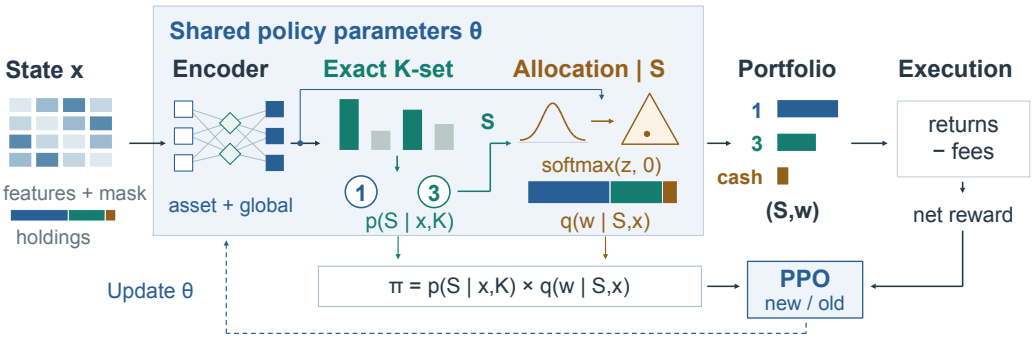

Figure 3: CASA-RL couples exact set selection with conditional allocation.

A shared encoder supplies asset embeddings and a masked global summary to both heads. Assets route messages through learned factors and receive the aggregated context, allowing each score to depend on the surrounding market. The allocation head additionally pools embeddings over the chosen support, so its weights respond to the selected assets together. Shared transformations and invariant pooling make the joint stochastic policy equivariant when features, holdings, eligibility, and priors are permuted consistently (Appendix A.3).

The support logits take the form $\ell_i = f_\theta(x)_i + \alpha a_i(x)$, combining a learned score with a causal financial prior of strength α. Cross-asset context enters through f_θ; conditional on the observation, the support energy is additive in these logits. This design permits contextual asset scoring while preserving exact set normalization. Figure 3 traces the two decisions and their likelihoods through training. The sampled portfolio enters execution, whose net reward supplies the learning signal. The two probability terms form the joint likelihood used in PPO's new-to-old ratio, and dashed feedback updates the shared encoder and both policy heads. Feature values and portfolio weights in the diagram are schematic.

3.2 C1: Exact Unordered Support Selection

For eligible assets $1,\dots,N$, define the support probability by the sum of its logits, normalized over all size-K subsets:

$$p_\theta(S \mid x,K) = \frac{\exp(\sum_{i\in S}\ell_i)}{Z_K(\ell)}, \qquad Z_K(\ell) = \sum_{U:|U|=K} \exp\left(\sum_{i\in U}\ell_i\right). \tag{3}$$

This product-weighted family is equivalently a collection of independent Bernoulli variables conditioned on selecting exactly K assets(Chen et al. (1994)). Its normalizer is the degree-K ESP in $e^{\ell_1},\dots,e^{\ell_N}$. To compute it, let $L_{i,k}$ be the log total weight of all k-subsets of the first i assets. Partitioning those subsets according to whether they contain asset i gives

$$L_{i,k} = \operatorname{logaddexp}(L_{i-1,k}, \ell_i + L_{i-1,k-1}), \quad L_{i,0} = 0, \quad L_{0,k>0} = -\infty. \tag{4}$$

The final entry gives $\log Z_K = L_{N,K}$. An ineligible asset permits only exclusion. For sampling, traverse the table backward from (N,K): include asset i with probability $\exp(\ell_i + L_{i-1,k-1} - L_{i,k})$ and decrement the remaining count k upon inclusion. This produces exactly K selected assets with the law in Equation (3). The traversal order introduces no order into the returned action or its mass.

The learning signal has a set-level interpretation. The same normalizer also determines each asset's inclusion probability. Differentiating the log mass yields

$$\frac{\partial \log p_\theta(S \mid x,K)}{\partial \ell_i} = \mathbf{1}\{i \in S\} - \rho_i, \qquad \rho_i = \Pr_\theta(i \in S \mid x,K). \tag{5}$$

Each score compares realized inclusion with the policy's inclusion probability. The normalizer therefore accounts for competing supports in the derivative used for learning. Appendix A derives normalization, exact sampling in real arithmetic, and this score identity, and distinguishes the support law from the PL set marginal.

3.3 C2: Allocation on the Selected Support

Once the support is selected, cash supplies the reference coordinate for a K-dimensional logistic-normal allocation(Aitchison & Shen (1980)). Let $q_i > 0$ be prior stock weights normalized over S, $c_0 \in (0,1)$ the prior cash weight, and $s > 0$ a residual scale. The support-conditioned head produces means μ_i and positive scales σ_i. For independent standard Gaussian draws ε_i, set

$$b_i = \log \frac{(1-c_0)q_i}{c_0}, \qquad z_i = b_i + s(\mu_i + \sigma_i \varepsilon_i),$$
$$w_0 = \frac{1}{1+\sum_{j\in S} e^{z_j}}, \qquad w_i = \frac{e^{z_i}}{1+\sum_{j\in S} e^{z_j}} \quad (i \in S). \tag{6}$$

The transformation satisfies Equation (1) directly. It is invertible on the selected face, with $z_i = \log(w_i/w_0)$, so the policy can score the actual allocation. With respect to its K free stock-weight coordinates, the log density is

$$\log q_\theta(w \mid S, x) = \sum_{i\in S} \log \mathcal{N}\left(\log \frac{w_i}{w_0}; b_i + s\mu_i, s^2\sigma_i^2\right) - \log w_0 - \sum_{i\in S} \log w_i. \tag{7}$$

The final terms are the log Jacobian of the inverse transformation, following the change-of-variables principle also used in normalizing flows(Rezende & Mohamed (2015)). Support-conditioned parameters couple the allocation to the chosen assets, and the shared denominator couples their weights even with diagonal Gaussian coordinates. Every selected stock has positive weight, all other stock weights are zero, and cash completes the budget. This construction supplies a normalized density on each feasible face (Appendix A.4).

3.4 Learning from the Joint Portfolio Likelihood

The support mass and conditional density let PPO evaluate the same stored portfolio under its behavior policy and updated parameters. With $r_t(\theta) = \pi_\theta(S_t, w_t \mid x_t, K)/\pi_{\text{old}}(S_t, w_t \mid x_t, K)$, the clipped actor objective is

$$\mathcal{L}_{\text{clip}}(\theta) = \mathbb{E}_t\left[\min\left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_t\right)\right]. \tag{8}$$

The log ratio separates into a support term and a conditional-allocation term. Both refer to the executed (S_t, w_t), so the update needs neither a selection sequence nor a relaxed action. The allocation Jacobian cancels when scoring the same stored weights under both policies; the implementation evaluates the full densities. A value baseline and generalized advantage estimation(Schulman et al. (2016))supply $\hat{A}_t$, with discount and GAE coefficients defined per decision interval. Appendix A.5 derives the ratio, and Appendix B gives the complete training algorithm.

Likelihood-ratio gradients act on the mixed-action log density(Williams (1992); Sutton et al. (1999)). We use PPO across all support laws. Related approaches constrain policy changes(Schulman et al. (2015)), optimize deterministic actors(Silver et al. (2014)), or add an entropy objective(Haarnoja et al. (2018)). The joint policy is defined independently of the optimizer.

4 From Policy Samples to Portfolio Execution

Causal inputs and priors. Eligibility masks restrict selection to historical constituents, and features use information available at or before the observation date. Current holdings enter both the state and a retention buffer in the 252-day log-momentum support prior. The allocation prior uses equal stock weights and cash weight 10^{-4}. Expressing these choices as explicit offsets makes the initial policy reproducible and lets evaluation measure how learning changes its support and weights. Appendix B specifies the feature transformations and data construction.

Momentum studies motivate the declared prior(Jegadeesh & Titman (1993); Moskowitz et al. (2012)); matched initialization exposes the changes in holdings and wealth produced by training.

Numerical representation and evaluation. The ESP recurrence uses log-space addition and masks for unreachable states; batches retain 786 catalog slots. Allocation log scales are clamped to $[-5, 1.5]$, and logarithms use positive floors. At evaluation, top-K logits yield a maximum-probability support

under Equation (3), and the transformed latent mean gives a reproducible feasible allocation. This allocation rule differs from the logistic-normal mean or density mode; tie handling is specified in Appendix A.3.

Self-financing accounting. The execution model charges transaction costs to the capital being allocated. For pretrade stock weights v_i, target weights w_i measured after fees, and fee rate c, the cost fraction C and subsequent wealth satisfy

$$C = c\sum_i |(1-C)w_i - v_i|\,, \qquad \frac{V_{d+1}}{V_d} = (1-C)\left(1+\sum_i w_i r_{d+1,i}\right). \tag{9}$$

For long-only weights and $0 \le c < 1$, this cost map is a contraction with a unique feasible solution (Appendix A.7). Paying fees before allocating the remaining wealth enforces the self-financing budget. Prices then drift holdings forward between rebalances. The experiment uses observation-close execution, zero cash return, and no borrowing, with proportional fees but no modeled execution delay, slippage, market impact, or capacity constraints.

5 C3: Attributing Portfolio Changes to Learning

The policy derivations establish the representation and likelihood properties required by C1 and C2. The experiments address C3 through three questions: what portion of portfolio return is already present at initialization; how removing the explicit support prior changes the learned policy; and whether the resulting method comparison persists across checkpoint rules. A 45-run comparison holds the architecture and update counts fixed across support distributions.

5.1 Experimental setup

Data and protocol. Our universe is reconstructed from historical S&P 500 constituents. Training uses 2,515 realized-return days from January 2, 2014 to December 28, 2023; validation uses 501 days from January 2, 2024 to December 30, 2025. The test covers 174 days from January 2 to September 11, 2026 and is retrospective because this interval was inspected during earlier development. Every method selects $K = 3$ stocks, rebalances at a 40-trading-day decision interval, starts in cash, and pays 10-basis-point stock-turnover costs. Returns are cumulative and net of these costs. Appendix B details the data construction and split boundaries.

Baselines and budget. We compare CASA-RL with sequential Ordered PL and temperature-one hard Gumbel-TopK. Both comparators use the exact PL likelihood of the retained order; their samplers implement the same ordered distribution. All three methods share the encoder, allocation head, and initial parameter tensors within each configuration–seed pair. We evaluate $\alpha \in \{0, 0.5, 2\}$ with seeds $\{7, 19, 42, 73, 101\}$. Each run completes 200 PPO updates, with 128 collected decisions and eight optimizer steps per update. Entropy regularization and KL-based early stopping are disabled to match optimization counts.

Selection and metrics. For each run, maximum cumulative net return on the full validation interval selects a trained checkpoint among updates $25, 50, \ldots, 200$. Each method selects its configuration by the largest five-seed mean of these validation returns. All selected models are fixed before the present test evaluation. We report every seed, retaining initialization and terminal checkpoints to distinguish starting-policy performance from subsequent changes. Metrics are net return, Sharpe with 252-day scaling and zero risk-free rate, maximum drawdown, and return gain over initialization. Seed means summarize separately evaluated policies on the same historical path.

Interpreting the comparison. RL outcomes depend on seeds and implementation choices(Henderson et al. (2018); Agarwal et al. (2021); Engstrom et al. (2020); Andrychowicz et al. (2021)). We report five learning runs on one historical path and document shared training and selection settings in Appendix B, following reproducible RL practice(Raffin et al. (2021)).

Table 1: Retrospective test performance at each method's validation-selected configuration. Values are five-seed means. Net return and maximum drawdown (MDD) are percentages; gain over initialization is in percentage points (pp).

Method	α	Net return	Sharpe	MDD	Gain (pp)
CASA-RL	2.0	315.9207	2.7574	43.1080	+0.002218
Ordered PL	0.5	315.9206	2.7574	43.1082	+0.002067
Gumbel-TopK	0.5	315.9209	2.7574	43.1082	+0.002437

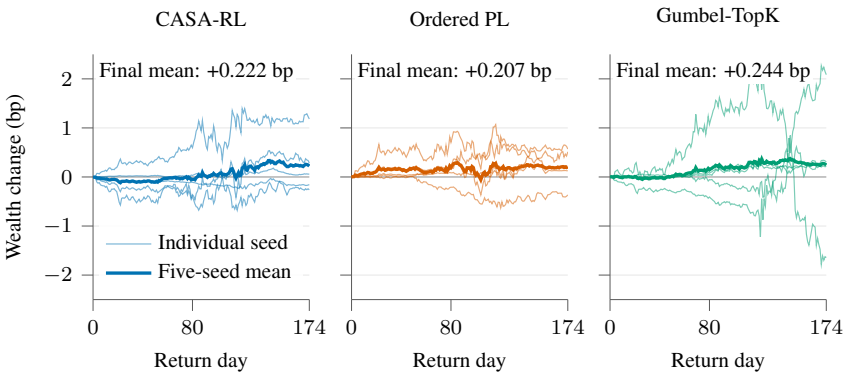

Figure 4: Learning gains relative to the matched initial policies.

5.2 Initialization Accounts for the Prior-Anchored Return

The validation-selected policies achieve nearly identical mean cumulative net returns of 315.92% (Table 1). Selection chooses $\alpha = 2$ for CASA-RL and $\alpha = 0.5$ for both ordered implementations. Thus the change to an exact unordered support likelihood is compatible with the portfolio performance attained by this shared architecture and prior.

Initialization explains the common return level. Figure 4 resolves the contribution of training by plotting each policy's wealth difference from its matched initialization. Thin paths show individual seeds and thick paths their five-seed means. All panels share a scale for $10^4(V_d^{\text{trained}} - V_d^{\text{initial}})$, the wealth difference in basis points of initial capital, with both wealth paths starting at one over January 2–September 11, 2026. On every one of the 174 test days, each selected trained policy holds the same stock support as its initialized counterpart. The mean stock/cash weight L_1 distances from initialization are 5.938×10^{-5} for CASA-RL, 5.855×10^{-5} for Ordered PL, and 8.230×10^{-5} for Gumbel-TopK. The measured gains therefore come from small allocation changes; the approximately 315.92% total return is largely present in the prior-based starting policies. This comparison makes portfolio performance and the contribution of policy learning separately observable.

5.3 Removing the Support Prior Exposes Learned Policy Changes

Setting $\alpha = 0$ produces a markedly different pattern of policy movement (Figure 5). Panel (a) shows each selected run's mean daily stock/cash weight L_1 distance from initialization as a marker on a logarithmic scale; horizontal marks give five-seed means. This ablation removes the explicit support-score prior while retaining the allocation prior, features, and execution rules. All 15 zero-support-prior policies hold different stock supports from their initialized counterparts on all 174 test days; all 30 policies at $\alpha = 0.5$ or 2 match their counterparts throughout. Across methods, mean daily stock/cash weight L_1 distance from initialization ranges from 1.909 to 1.969 at $\alpha = 0$, compared with 4.11×10^{-5}–8.23×10^{-5} at positive strengths. This distance compares separately evaluated policies, whereas turnover compares consecutive holdings within a policy.

Under validation checkpoint selection at $\alpha = 0$, CASA-RL averages 144.4102% net return, Ordered PL 12.8450%, and Gumbel-TopK 90.0658%. At both tested positive strengths, all three means remain

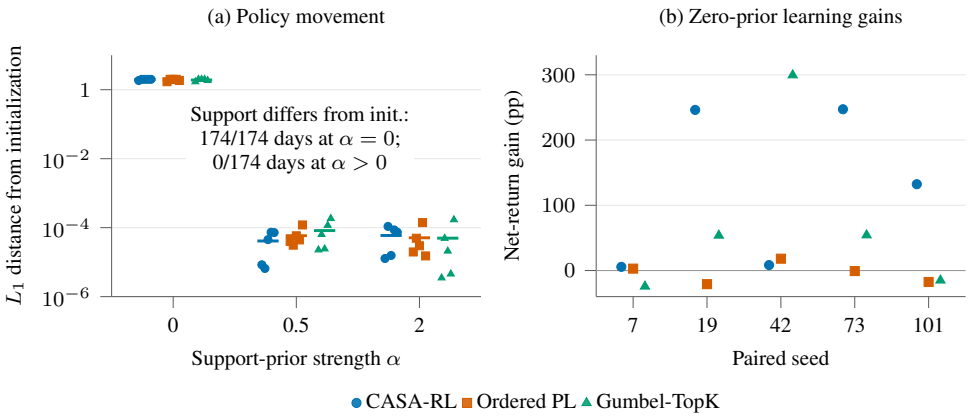

Figure 5: Support-prior strength controls departures from initialization.

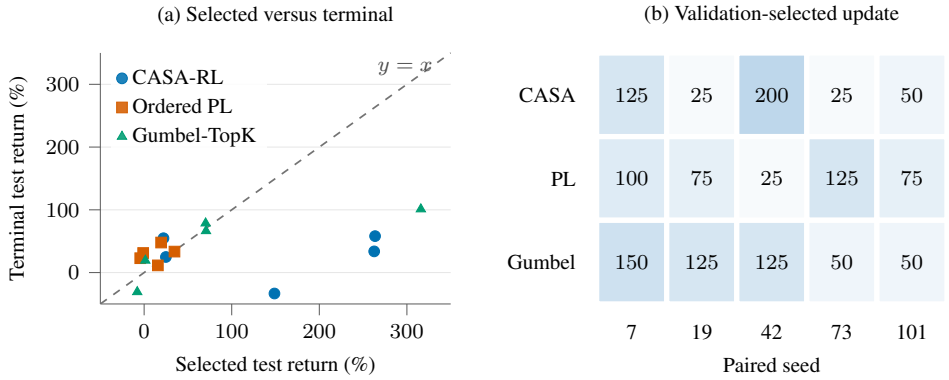

Figure 6: Checkpoint choice changes the zero-prior comparison.

close to 315.92%. The three configurations therefore distinguish a regime of changed asset selections from one that retains the initial support. Figure 5(b) shows each paired seed's net-return gain over initialization at zero support prior, in percentage points; the next comparison examines how these differences depend on the checkpoint rule.

5.4 Checkpoint Choice Determines the Zero-Prior Ranking

At $\alpha = 0$, validation-selected and fixed-budget policies give different method orderings (Figure 6). Panel (a) pairs each seed's validation-selected test return with its fixed 200-update terminal return; the dashed line marks equality. Panel (b) locates the validation-selected update for each method and seed. Starting from a common initialization return of 16.4673%, the terminal policies after 200 updates average 27.6184% for CASA-RL, 29.3716% for Ordered PL, and 46.8768% for Gumbel-TopK. Fourteen of the 15 validation-selected checkpoints occur before update 200. CASA-RL's larger mean at the validation-selected checkpoints thus describes a selection-dependent result within this budget. Reporting both rules alongside initialization identifies this dependence without using test performance to choose a reporting rule. Ordered PL and Gumbel-TopK can follow different trajectories because their sampled random draws differ, even though their ordered probability laws agree. Appendix C provides every fixed-configuration result and the validation histories.

Scope of the evidence. These results cover three prior strengths and five seeds on one retrospective market split. Shared architecture and execution choices come from earlier development whose tuning budgets were not equalized, and public reconstruction has incomplete quote and delisting coverage.

The controls characterize learning within 200 updates; normalization and likelihood properties follow from the construction.

6 Related Work

Portfolio optimization and sequential allocation. Mean–variance optimization formalizes portfolio risk and return(Markowitz (1952)), while cardinality-constrained formulations bound the number of positions(Chang et al. (2000)). Sparse and norm-constrained portfolios connect regularization to allocation structure(Brodie et al. (2009); Gotoh & Takeda (2011)); multi-period convex trading incorporates forecasts, risk, and transaction costs across successive decisions(Boyd et al. (2017)). Comparisons with equal weighting also emphasize the importance of a clearly specified allocation baseline(DeMiguel et al. (2009)). CASA-RL addresses the stochastic action representation within sequential allocation: the portfolio constraint defines the support of a trainable joint distribution, and self-financing execution turns each sampled action into a net reward.

Reinforcement learning for financial decisions. Direct reinforcement learning links trading rules to sequential performance(Moody & Saffell (2001)). Other financial RL settings include optimized trade execution(Nevmyvaka et al. (2006)), learned market representations for trading(Deng et al. (2017)), neural portfolio allocation(Jiang et al. (2017)), and trading policies evaluated under transaction costs(Zhang et al. (2020)). Our control variable is an unordered K-stock support with continuous stock/cash weights. The comparison varies its support law within a shared architecture and execution pipeline.

Normalized subset distributions. Weighted fixed-size sampling(Chen et al. (1994)) supplies CASA-RL's product-weighted subset law. Determinantal point processes provide another view of normalized subset models(Kulesza & Taskar (2012)): conditioned on size K, a diagonal positive L-ensemble kernel gives the same product-weighted law, whereas nondiagonal kernels model diversity through determinant interactions. CASA-RL uses contextual asset scores and an additive set energy, enabling a direct ESP recurrence over asset weights. Plackett–Luce assigns an exact mass to an ordered selection(Plackett (1975)), and hard Gumbel-TopK samples that ordering without replacement(Kool et al. (2019a)). Our likelihood is instead evaluated directly on the selected support under the product-weighted law; the ordered alternatives remain valid augmented-action policies.

Differentiable subsets, sorting, and ranking. Gumbel-Softmax and Concrete replace discrete variables with continuous relaxations(Jang et al. (2017); Maddison et al. (2017)). Reparameterizable subset sampling(Xie & Ermon (2019)) and optimal-transport TopK(Xie et al. (2020)) extend differentiable selection to multiple items. Related approaches relax sorting operations(Grover et al. (2019)) or construct efficient differentiable sorting and ranking operators(Blondel et al. (2020)); stochastic softmax tricks generalize relaxation-based gradient estimation to structured choices(Paulus et al. (2020)). These methods target differentiation through discrete operations. CASA-RL uses the discrete support mass in a likelihood-ratio update and differentiates the conditional allocation density.

Set representations and continuous allocation. Deep Sets and Set Transformer encode unordered inputs(Zaheer et al. (2017); Lee et al. (2019)); GraphSAGE aggregates neighborhood information(Hamilton et al. (2017)), and attention-based routing learns structured sequential decisions(Kool et al. (2019b)). These encoders supply context to the output distribution. Our conditional logistic-normal model(Aitchison & Shen (1980)) maps that context and the chosen support to a density on its allocation face.

7 Conclusion

CASA-RL connects requirements of sparse portfolio RL: representing an unordered K-stock support, learning its allocation through one explicit likelihood, and identifying the behavior added by training. ESP-normalized selection and conditional logistic-normal allocation supply the joint policy, with feasibility, normalization, and sampling established by construction. The evaluation distinguishes prior-anchored portfolios from the substantial policy changes exposed by removing the support prior, and locates the dependence of method rankings on checkpoint choice.

REFERENCES

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, volume 34, pp. 29304–29320. Curran Associates, Inc., 2021. URL https://papers.nips.cc/paper_files/paper/2021/hash/f514cec81cb148559cf475e7426eed5e-Abstract.html.

John Aitchison and S. M. Shen. Logistic-normal distributions: Some properties and uses. *Biometrika*, 67(2):261–272, 1980. doi: 10.1093/biomet/67.2.261.

Marcin Andrychowicz, Anton Raichuk, Piotr Stańczyk, Manu Orsini, Sertan Girgin, Raphael Marinier, Léonard Hussenot, Matthieu Geist, Olivier Pietquin, Marcin Michalski, Sylvain Gelly, and Olivier Bachem. What matters for on-policy deep actor-critic methods? a large-scale study. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=nIAxjsniDzg.

David H. Bailey, Jonathan M. Borwein, Marcos López de Prado, and Qiji Jim Zhu. The probability of backtest overfitting. *The Journal of Computational Finance*, 20(4):39–69, 2017. doi: 10.21314/JCF.2016.322. URL https://www.risk.net/journal-of-computational-finance/2471206/the-probability-of-backtest-overfitting.

Mathieu Blondel, Olivier Teboul, Quentin Berthet, and Josip Djolonga. Fast differentiable sorting and ranking. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 950–959. PMLR, 2020. URL https://proceedings.mlr.press/v119/blondel20a.html.

Stephen Boyd, Enzo Busseti, Steven Diamond, Ronald N. Kahn, Kwangmoo Koh, Peter Nystrup, and Jan Speth. Multi-period trading via convex optimization. *Foundations and Trends in Optimization*, 3(1):1–76, 2017. doi: 10.1561/2400000023. URL https://web.stanford.edu/~boyd/papers/cvx_portfolio.html.

Joshua Brodie, Ingrid Daubechies, Christine De Mol, Domenico Giannone, and Ignace Loris. Sparse and stable Markowitz portfolios. *Proceedings of the National Academy of Sciences*, 106(30):12267–12272, 2009. doi: 10.1073/pnas.0904287106. URL https://doi.org/10.1073/pnas.0904287106.

Gavin C. Cawley and Nicola L. C. Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11:2079–2107, 2010. URL https://jmlr.org/papers/v11/cawley10a.html.

T.-J. Chang, N. Meade, J. E. Beasley, and Y. M. Sharaiha. Heuristics for cardinality constrained portfolio optimisation. *Computers & Operations Research*, 27(13):1271–1302, 2000. doi: 10.1016/S0305-0548(99)00074-X.

Xiang-Hui Chen, Arthur P. Dempster, and Jun S. Liu. Weighted finite population sampling to maximize entropy. *Biometrika*, 81(3):457–469, 1994. doi: 10.1093/biomet/81.3.457.

Thomas M. Cover. Universal portfolios. *Mathematical Finance*, 1(1):1–29, 1991. doi: 10.1111/j.1467-9965.1991.tb00002.x. URL https://doi.org/10.1111/j.1467-9965.1991.tb00002.x.

Victor DeMiguel, Lorenzo Garlappi, and Raman Uppal. Optimal versus naive diversification: How inefficient is the 1/N portfolio strategy? *The Review of Financial Studies*, 22(5):1915–1953, 2009. doi: 10.1093/rfs/hhm075. URL https://doi.org/10.1093/rfs/hhm075.

Yue Deng, Feng Bao, Youyong Kong, Zhiquan Ren, and Qionghai Dai. Deep direct reinforcement learning for financial signal representation and trading. *IEEE Transactions on Neural Networks and Learning Systems*, 28(3):653–664, 2017. doi: 10.1109/TNNLS.2016.2522401. URL https://doi.org/10.1109/TNNLS.2016.2522401.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep policy gradients: A case study on PPO and TRPO. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=r1etN1rtPB.

Jun-ya Gotoh and Akiko Takeda. On the role of norm constraints in portfolio selection. *Computational Management Science*, 8(4):323–353, 2011. doi: 10.1007/s10287-011-0130-2. URL https://link.springer.com/article/10.1007/s10287-011-0130-2.

Aditya Grover, Eric Wang, Aaron Zweig, and Stefano Ermon. Stochastic optimization of sorting networks via continuous relaxations. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1eSS3CcKX.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1861–1870. PMLR, 2018. URL https://proceedings.mlr.press/v80/haarnoja18b.html.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1706.02216.

Matthew Hausknecht and Peter Stone. Deep reinforcement learning in parameterized action space. In *International Conference on Learning Representations*, 2016. URL https://www.cs.utexas.edu/~pstone/Papers/bib2html-links/ICLR16-hausknecht.pdf.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. doi: 10.1609/aaai.v32i1.11694. URL https://ojs.aaai.org/index.php/AAAI/article/view/11694.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1611.01144.

Narasimhan Jegadeesh and Sheridan Titman. Returns to buying winners and selling losers: Implications for stock market efficiency. *The Journal of Finance*, 48(1):65–91, 1993. doi: 10.1111/j.1540-6261.1993.tb04702.x. URL https://doi.org/10.1111/j.1540-6261.1993.tb04702.x.

Zhengyao Jiang, Dixing Xu, and Jinjun Liang. A deep reinforcement learning framework for the financial portfolio management problem. *arXiv preprint arXiv:1706.10059*, 2017. URL https://arxiv.org/abs/1706.10059.

Wouter Kool, Herke van Hoof, and Max Welling. Stochastic beams and where to find them: The Gumbel-Top-k trick for sampling sequences without replacement. In *International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3499–3508, 2019a. URL https://proceedings.mlr.press/v97/kool19a.html.

Wouter Kool, Herke van Hoof, and Max Welling. Attention, learn to solve routing problems! In *International Conference on Learning Representations*, 2019b. URL https://arxiv.org/abs/1803.08475.

Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 5(2–3):123–286, 2012. doi: 10.1561/2200000044. URL https://arxiv.org/abs/1207.6083.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3744–3753, 2019. URL https://proceedings.mlr.press/v97/lee19d.html.

Bin Li and Steven C. H. Hoi. Online portfolio selection: A survey. *ACM Computing Surveys*, 46(3):1–36, 2014. doi: 10.1145/2512962. URL https://arxiv.org/abs/1212.2129.

Boyan Li, Hongyao Tang, Yan Zheng, Jianye Hao, Pengyi Li, Zhen Wang, Zhaopeng Meng, and Li Wang. HyAR: Addressing discrete-continuous action reinforcement learning via hybrid action representation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=64trBbOhdGU.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1611.00712.

Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952. doi: 10.1111/j.1540-6261.1952.tb01525.x.

Warwick Masson, Pravesh Ranchod, and George Konidaris. Reinforcement learning with parameterized actions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016. doi: 10.1609/aaai.v30i1.10226. URL https://doi.org/10.1609/aaai.v30i1.10226.

Robert C. Merton. Optimum consumption and portfolio rules in a continuous-time model. *Journal of Economic Theory*, 3(4):373–413, 1971. doi: 10.1016/0022-0531(71)90038-X. URL https://doi.org/10.1016/0022-0531(71)90038-X.

J. Moody and M. Saffell. Learning to trade via direct reinforcement. *IEEE Transactions on Neural Networks*, 12(4):875–889, 2001. doi: 10.1109/72.935097.

Tobias J. Moskowitz, Yao Hua Ooi, and Lasse Heje Pedersen. Time series momentum. *Journal of Financial Economics*, 104(2):228–250, 2012. doi: 10.1016/j.jfineco.2011.11.003. URL https://doi.org/10.1016/j.jfineco.2011.11.003.

Yuriy Nevmyvaka, Yi Feng, and Michael Kearns. Reinforcement learning for optimized trade execution. In *Proceedings of the 23rd International Conference on Machine Learning*, pp. 673–680, 2006. doi: 10.1145/1143844.1143929. URL https://doi.org/10.1145/1143844.1143929.

Max Paulus, Dami Choi, Daniel Tarlow, Andreas Krause, and Chris Maddison. Gradient estimation with stochastic softmax tricks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5691–5704, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/3df80af53dce8435cf9ad6c3e7a403fd-Abstract.html.

R. L. Plackett. The analysis of permutations. *Applied Statistics*, 24(2):193–202, 1975. doi: 10.2307/2346567.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-Baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL https://jmlr.org/papers/v22/20-1364.html.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1530–1538. PMLR, 2015. URL https://proceedings.mlr.press/v37/rezende15.html.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1889–1897. PMLR, 2015. URL https://proceedings.mlr.press/v37/schulman15.html.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1506.02438.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 387–395. PMLR, 2014. URL https://proceedings.mlr.press/v32/silver14.html.

Richard S. Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999. URL https://papers.nips.cc/paper/1999/hash/464d828b85b0bed98e80ade0a5c43b0f-Abstract.html.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, 1992. doi: 10.1007/BF00992696. URL https://link.springer.com/article/10.1007/BF00992696.

Sang Michael Xie and Stefano Ermon. Reparameterizable subset sampling via continuous relaxations. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pp. 3919–3925, 2019. doi: 10.24963/ijcai.2019/544. URL https://www.ijcai.org/proceedings/2019/544.

Yujia Xie, Hanjun Dai, Minshuo Chen, Bo Dai, Tuo Zhao, Hongyuan Zha, Wei Wei, and Tomas Pfister. Differentiable top-k with optimal transport. In *Advances in Neural Information Processing Systems*, volume 33, pp. 20520–20531, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/ec24a54d62ce57ba93a531b460fa8d18-Abstract.html.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander Smola. Deep sets. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1703.06114.

Zihao Zhang, Stefan Zohren, and Stephen Roberts. Deep reinforcement learning for trading. *The Journal of Financial Data Science*, 2(2):25–40, 2020. doi: 10.3905/jfds.2020.1.030. URL https://arxiv.org/abs/1911.10107.