From Optimization to Generation: Learning Portfolio Distributions with Simplex-Space Flow Matching

Anonymous authors
Paper under double-blind review

Abstract

Portfolio construction typically seeks a single optimal portfolio from a market state, while the uncertainty and diversity of feasible investment decisions remain largely unexplored. In this work, we formulate portfolio construction as a conditional generative modeling problem, where the objective is to learn a distribution over feasible portfolio allocations conditioned on the observed market state, rather than predicting a single deterministic portfolio. We introduce Simplex-Space Flow Matching (SS-FM), which learns continuous probability flows directly over the portfolio simplex and generates diverse allocations that reflect different portfolio preferences. SS-FM generates diverse feasible portfolio candidates, while reinforcement learning learns how to make decisions within this generated candidate space using realized portfolio-level feedback, enabling the model to move beyond imitation of predefined portfolio construction rules. To provide informative conditioning signals, we incorporate overnight market information into alpha prediction through gated residual fusion, capturing information that is not contained in intraday price and volume dynamics. Experiments on the CSI 500 and S&P 500 demonstrate that modeling portfolio construction as conditional generation provides an effective alternative to deterministic and conventional optimization-based approaches.

1 Introduction

Portfolio construction is traditionally formulated as a deterministic optimization or prediction problem. Given market state s_t, an optimizer or learned model produces one allocation,

$$w_t^* = f(s_t), \qquad w_t \in \mathbb{R}^K. \tag{1}$$

Classical methods such as mean–variance optimization and risk-based portfolio construction optimize a single allocation under a predefined objective (Markowitz, 1952; Maillard et al., 2010). Modern neural approaches similarly map market observations or asset characteristics to portfolio weights (Heaton et al., 2017; Gu et al., 2020).

A single allocation can be restrictive when several return, risk, turnover, and diversification trade-offs are defensible under the same market state (Michaud, 1989; DeMiguel et al., 2009; Lai et al., 2011). We therefore formulate portfolio construction as conditional generation:

$$w_t \sim p_\theta(w \mid s_t), \qquad w_t \in \Delta^{K-1}, \tag{2}$$

where Δ^{K-1} denotes the long-only portfolio simplex. This formulation raises two challenges. First, portfolio weights are compositional variables constrained to a simplex rather than an unconstrained Euclidean space. Second, a generator alone does not determine which generated allocation should be executed under realized market feedback.

We address both challenges with a generation-to-decision framework. *Simplex-Space Flow Matching* (SS-FM) learns conditional probability flows in a centered log-ratio representation of the simplex (Lipman et al., 2023; Chen & Lipman, 2024; Aitchison, 1982; 1986). For each state, SS-FMgenerates a set of feasible portfolio candidates. A downstream PPO

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.

policy (Schulman et al., 2017) then uses the market state and candidate set to determine the executed allocation from realized portfolio-level rewards. Thus, SS-FMmodels which allocations are plausible under complementary portfolio-construction principles, while reinforcement learning determines which proposal is useful sequentially.

We evaluate the framework under strict walk-forward protocols on CSI 500 and S&P 500 data. CSI 500 additionally supports controlled analysis of overnight and opening-auction information; S&P 500 provides a cross-market test of the core generation-to-decision framework.

Our contributions are:

- We formulate portfolio construction as conditional generation over feasible simplex-valued allocations rather than deterministic allocation prediction.
- We propose SS-FM, a CLR-space conditional flow-matching model trained with multiple optimization-induced portfolio targets.
- We introduce a generation-to-decision framework in which PPO or GRPO refines SS-FM proposals using realized portfolio-level feedback.
- We evaluate deterministic, diffusion, Euclidean flow-matching, and simplex-aware generation methods under a common walk-forward protocol.

2 Related Work

2.1 Learning-Based Portfolio Optimization

Classical portfolio construction methods typically produce a single allocation by optimizing a predefined risk–return objective. Mean–Variance Optimization (MVO) balances expected return against portfolio variance (Markowitz, 1952), Maximum Sharpe Ratio optimization targets risk-adjusted return (Sharpe, 1966), and Risk Parity seeks balanced risk contributions across assets (Maillard et al., 2010). Black–Litterman further combines equilibrium information with investor views for portfolio construction (Black & Litterman, 1992).

Recent learning-based methods replace hand-crafted portfolio rules with flexible nonlinear representations. Deep learning has been used for return forecasting and asset allocation (Heaton et al., 2017; Simon et al., 2026), while machine-learning asset-pricing models learn nonlinear relations between asset characteristics and expected returns (Gu et al., 2020). More recent studies investigate machine learning for alpha generation, empirical asset pricing, and portfolio decisions (Kelly et al., 2024; Choi et al., 2025; Jensen et al., 2026). Despite these differences, classical and learned methods generally map the current market state to one allocation.

2.2 Generative Modeling for Decision Making

Generative policies provide a flexible alternative to deterministic decision rules by modeling a distribution over actions rather than producing a single point estimate. Conventional stochastic policies often use simple parametric families, such as Gaussian policies (Schulman et al., 2017; Haarnoja et al., 2018). More expressive approaches employ normalizing flows (Rezende & Mohamed, 2015; Grathwohl et al., 2019; Papamakarios et al., 2021), diffusion models (Ho et al., 2020; Song et al., 2021), and flow matching (Lipman et al., 2023; Tong et al., 2024) to represent complex conditional distributions.

Generative methods have subsequently been adapted to sequential decision making. Trajectory-based methods generate action sequences conditioned on states, goals, or returns (Janner et al., 2022); Decision Diffuser formulates offline decision making as conditional trajectory generation (Ajay et al., 2023); and Diffusion-QL uses diffusion models to represent expressive action distributions in offline reinforcement learning (Wang et al., 2023). Financial applications have applied generative models to forecasting, portfolio construction, and high-frequency trading (Cho et al., 2026; Li et al., 2025).

2.3 Reinforcement Learning for Portfolio Management

Reinforcement learning formulates portfolio management as sequential decision making, in which an agent observes market information, chooses an allocation, and receives realized portfolio-level feedback. Deep reinforcement learning has been applied to portfolio selection using policy-gradient and actor–critic methods (Jiang et al., 2017; Liang et al., 2018; Liu et al., 2021). More recent methods incorporate richer temporal representations, market interactions, and autoregressive decision structures (Wang et al., 2021; Winkel et al., 2024).

Most RL portfolio methods learn a direct policy from the market state to an allocation:

$$w_t \sim \pi_\phi(\cdot \mid s_t). \tag{3}$$

This requires the policy to simultaneously learn a feasible action representation, a state-conditioned allocation prior, and a sequential decision rule from realized rewards. In contrast, our framework separates these roles. SS-FM first learns a state-conditioned distribution of optimization-induced feasible portfolios and generates a candidate set $\mathcal{W}_t$. A downstream PPO or GRPO policy then conditions on $(s_t, \mathcal{W}_t)$ to construct the final allocation using realized portfolio-level rewards. This generation-to-decision decomposition lets the policy optimize sequential deployment while exploiting a structured candidate representation of the portfolio space.

3 Method

At decision time t, the model observes causal market information

$$s_t = \{x_{t,i}\}_{i=1}^{K} \tag{4}$$

for K selected assets and constructs a long-only portfolio

$$w_t \in \Delta^{K-1} = \left\{ w \in \mathbb{R}_+^K : \mathbf{1}^\top w = 1 \right\}. \tag{5}$$

Our framework consists of a causal market encoder, a simplex-space portfolio generator, and a candidate-conditioned decision policy:

$$s_t \to h_t \to \mathcal{W}_t \to \pi_\phi \to w_t. \tag{6}$$

3.1 Causal Market Representation

For each asset, the encoder processes causal price, volume, return, volatility, VWAP-related, technical, and liquidity features over a historical intraday window. A Transformer temporal encoder (Vaswani et al., 2017) captures within-asset temporal dependence, and cross-stock attention contextualizes representations across the tradable universe (Li et al., 2024):

$$h_t = f_{\text{cross}}\left(f_{\text{temp}}(X_{t-L:t-1,1}), \ldots, f_{\text{temp}}(X_{t-L:t-1,K})\right). \tag{7}$$

For CSI 500, we additionally encode overnight and opening-auction information a_t (Gerace et al., 2015; Chu et al., 2024) and apply gated residual fusion:

$$h_t \leftarrow h_t + \sigma\left(W_g[h_t; f_{\text{auc}}(a_t)] + b_g\right) \odot f_{\text{auc}}(a_t). \tag{8}$$

All inputs are available before execution. The representation is also used to predict next-day asset returns, which rank assets for the portfolio universe:

$$\hat{y}_{t,i} = f_\alpha(h_{t,i}). \tag{9}$$

Feature definitions, encoder hyperparameters, and auction-feature construction are provided in Appendix A.

3.2 Simplex-Space Flow Matching

Portfolio weights are compositional because non-negativity and the budget constraint couple all components. We therefore apply the centered log-ratio (CLR) transformation (Aitchison, 1982; 1986):

$$z_i = \mathrm{CLR}(w)_i = \log w_i - \frac{1}{K}\sum_{j=1}^{K} \log w_j, \qquad z \in \mathcal{T} := \left\{ z \in \mathbb{R}^K : \mathbf{1}^\top z = 0 \right\}. \tag{10}$$

Before transformation, we apply a small positive weight floor and renormalize teacher allocations. The inverse transformation is

$$\mathrm{CLR}^{-1}(z)_i = \frac{\exp(z_i)}{\sum_{j=1}^{K} \exp(z_j)}, \tag{11}$$

which yields a feasible long-only allocation.

For each state, we construct teacher portfolios from Mean–Variance Optimization, Maximum Sharpe, and Risk Parity:

$$w_t^{(m)} = \mathcal{T}_m(s_t), \qquad m \in \{\mathrm{MVO}, \mathrm{MS}, \mathrm{RP}\}. \tag{12}$$

These complementary targets expose the generator to distinct optimization-induced allocation preferences rather than one deterministic label.

Let

$$z_t^{(m)} = \mathrm{CLR}(w_t^{(m)}). \tag{13}$$

We sample a zero-sum base state $z_0 \sim p_0$ and, for $\tau \sim \mathcal{U}[0,1]$, define the conditional flow-matching path (Lipman et al., 2023; Tong et al., 2024):

$$z_{\tau,t}^{(m)} = (1-\tau)z_0 + \tau z_t^{(m)}, \qquad u_t^{(m)} = z_t^{(m)} - z_0. \tag{14}$$

A conditional vector field $v_\theta(z, \tau, h_t)$ is trained with

$$\mathcal{L}_{\text{SS-FM}} = \mathbb{E}_{t,m,\tau,z_0}\left[\left\| v_\theta\left(z_{\tau,t}^{(m)}, \tau, h_t\right) - u_t^{(m)} \right\|_2^2\right]. \tag{15}$$

At inference, independently sampled base states are transported according to

$$\frac{dz^{(g)}(\tau)}{d\tau} = v_\theta\left(z^{(g)}(\tau), \tau, h_t\right), \qquad w_t^{(g)} = \mathrm{CLR}^{-1}\left(z^{(g)}(1)\right), \tag{16}$$

producing G feasible candidates

$$\mathcal{W}_t = \left\{w_t^{(g)}\right\}_{g=1}^{G}. \tag{17}$$

3.3 Generation-to-Decision Policy

SS-FM provides a state-conditioned portfolio proposal set but does not directly determine which allocation should be deployed. The downstream policy therefore receives $(h_t, \mathcal{W}_t)$ and constructs the final allocation:

$$w_t \sim \pi_\phi\left(\cdot \mid h_t, \mathcal{W}_t\right), \qquad w_t \in \Delta^{K-1}. \tag{18}$$

We consider two implementations. In *Candidate-to-Weight*, an MLP outputs w_t directly. In *Candidate Selection*, the policy outputs a categorical distribution over generated candidates:

$$p_t^{(g)} = \pi_\phi\left(g \mid h_t, \mathcal{W}_t\right), \qquad w_t = w_t^{(g_t)}, \qquad g_t \sim p_t. \tag{19}$$

We optimize the policy using PPO (Schulman et al., 2017). For a selected candidate action, the clipped objective is

$$\mathcal{L}_{\text{PPO}} = \mathbb{E}_t\left[\min\left(\rho_t(\phi)\hat{A}_t, \mathrm{clip}\left(\rho_t(\phi), 1-\epsilon, 1+\epsilon\right)\hat{A}_t\right)\right], \tag{20}$$

where ρ_t is the ratio of current and behavior-policy probabilities and $\hat{A}_t$ is estimated by a separate critic.

We also compare Group Relative Policy Optimization (GRPO) (Shao et al., 2024). Initialized from the pre-trained SS-FM distribution, GRPO samples a group of candidates under the same state and uses the normalized within-group reward:

$$\hat{A}_t^{(g)} = \frac{r_t^{(g)} - \mu_t}{\sigma_t + \epsilon}. \tag{21}$$

PPO and GRPO share the market representation, candidate generator, action space, and portfolio-level reward; detailed optimization procedures are given in Appendix C.

4 Experimental Setup

We evaluate on CSI 500 and S&P 500 equity universes. CSI 500 is the primary setting for temporal-architecture and auction-fusion analysis; S&P 500 provides a cross-market test of the core generation-to-decision framework. Both datasets use intraday OHLCV-derived features and a ten-trading-day lookback. Only CSI 500 includes opening-auction features.

Table 1: Datasets and evaluation protocol.

Property	CSI 500	S&P 500
Intraday data	Yes	Yes
Auction features	Yes	No
Lookback	10 days	10 days
Test period	2025-05–2026-05	2025-05–2026-05
Portfolio universe	Top-30	Top-30

The alpha model predicts next-day Open-to-Close log return,

$$y_{t,i} = \log\left(\frac{C_{t+1,i}}{O_{t+1,i}}\right), \tag{22}$$

and ranks assets to form the daily Top-30 universe. We use monthly walk-forward evaluation: for test month T, the previous 12 complete months are used for training, month $T-1$ for validation, and T is strictly held out. No test-period observations are used for normalization, covariance estimation, teacher generation, model fitting, policy updates, or hyperparameter tuning. These safeguards address leakage risks in financial backtesting (Bailey et al., 2017).

We compare Equal Weight, MVO, Maximum Sharpe, and Risk Parity; Gaussian, MLP, Diffusion, Euclidean FM, and SS-FM generators; and SS-FM combined with PPO or GRPO. All methods use the same alpha model, selected universe, execution schedule, long-only constraint, and transaction-cost convention. Unless stated otherwise, the number of generated candidates is $G = 32$ and transaction cost is 3 basis points. We report net return, annualized Sharpe ratio, maximum drawdown, turnover, and, where relevant, alpha IC. Full feature definitions, hyperparameters, and reproducibility records are provided in the appendix.

5 Results

All results use the fixed out-of-sample walk-forward protocol described in Section 4. CSI 500 covers May 2025 to May 2026 (approximately 245 trading days), and S&P 500 covers June 2025 to May 2026 (approximately 235 trading days). Unless otherwise specified, the generator uses $G = 32$ candidates and the portfolio universe contains the Top-30 assets ranked by the frozen alpha model.

Because sampled-candidate selection can be sensitive to the aggregation rule, we distinguish two protocols throughout. *Pure* uses only generated portfolios, whereas *Meta* augments them with the three teacher portfolios. *Pred-Utility* selects the candidate with the largest ex-ante utility, while *Softmax* aggregates candidates using predicted utility weights. We use Pure or Softmax-Meta results for the main single-run comparisons and report multi-seed Pred-Utility results separately to quantify sampling sensitivity.

5.1 Main Comparison Across CSI 500 and S&P 500

SS-FM versus the generative-model family. Table 2 compares SS-FM with Gaussian, MLP, Diffusion, and Euclidean Flow Matching under the common Pure $G = 32$ protocol. On CSI 500, SS-FM achieves the highest cumulative net return and Sharpe ratio among the standalone generative models. On S&P 500, SS-FM is the only evaluated generator with positive cumulative net return and positive Sharpe ratio. These results indicate that SS-FM is competitive across both markets, although the absolute return level is substantially lower on S&P 500.

Table 2: SS-FM versus conditional portfolio generators. Both CSI 500 and S&P 500 use the Pure $G = 32$ protocol. Net Return is cumulative after transaction costs.

Market	Method	Net Return	Sharpe
CSI 500	Diffusion	31.9%	1.10
	Gaussian	40.5%	1.70
	MLP	36.6%	1.85
	FM	49.0%	2.15
	SS-FM	**60.8%**	**2.30**
S&P 500	Diffusion	-4.6%	-0.41
	Gaussian	-2.4%	-0.22
	MLP	-1.6%	-0.14
	FM	-0.5%	-0.04
	SS-FM	**7.5%**	**0.64**

Note. Multi-seed sensitivity and the effect of alternative candidate aggregation rules are reported in Section 5.2.

SS-FM versus optimization teachers. Table 3 compares SS-FM with the MVO, Maximum Sharpe, and Risk Parity teacher portfolios on CSI 500. SS-FM achieves the largest cumulative net return and the smallest maximum drawdown. Risk Parity remains the most conservative teacher, with the highest Sharpe ratio and the lowest turnover. Thus, SS-FM does not dominate every metric; rather, it provides a different return–drawdown–turnover trade-off from the individual optimization rules used for supervision.

Table 3: SS-FM versus optimization teachers on CSI 500 under the Pure $G = 32$ protocol. Net Return is cumulative after transaction costs. Turnover is average daily one-way turnover.

Method	Net Return	Sharpe	MDD	Turnover
MVO	32.6%	0.87	31.1%	0.935
Maximum Sharpe	36.9%	1.19	22.0%	0.917
Risk Parity	45.9%	**2.39**	12.9%	**0.055**
SS-FM	**60.8%**	2.30	**10.9%**	0.649

Figure 1 visualizes the daily cumulative net-return paths underlying Table 3. The figure is used to assess whether the performance difference is persistent throughout the evaluation period rather than driven by a small number of trading days. SS-FM remains above the individual teacher portfolios for most of the out-of-sample period and exhibits a shallower drawdown than MVO and Maximum Sharpe. Risk Parity follows a smoother but lower-return trajectory, consistent with its substantially lower turnover.

Generation-to-decision refinement. Table 4 evaluates whether a frozen SS-FM generator provides useful context for downstream policy optimization. PPO and GRPO use the same pre-trained SS-FM generator, so the comparison isolates the decision layer. On CSI 500, both GRPO and PPO improve on Pure SS-FM. PPO achieves the largest cumulative net-return improvement while reducing turnover from 0.649 to 0.260. Candidate-conditioned PPO has the lowest turnover, suggesting that candidate information can support cost-aware portfolio refinement.

Figure 2 plots the cumulative net-return paths for the strategies in Table 4. It provides the trajectory-level evidence for the generation-to-decision claim: the PPO curve separates from Pure SS-FM during the out-of-sample period and retains the improvement through the end of the evaluation window. The GRPO variants also remain above Pure SS-FM, but their improvement is smaller than that of PPO.

On S&P 500, candidate-conditioned PPO obtains a mean cumulative net return of 17.9% with a standard deviation of 3.5% across the reported seeds. This compares with 0.6% $\pm$ 6.8% for the SS-FM Meta baseline and 5.2% $\pm$ 2.8% for GRPO initialization. Although the current multi-seed comparison contains only three reported seeds, the candidate-conditioned

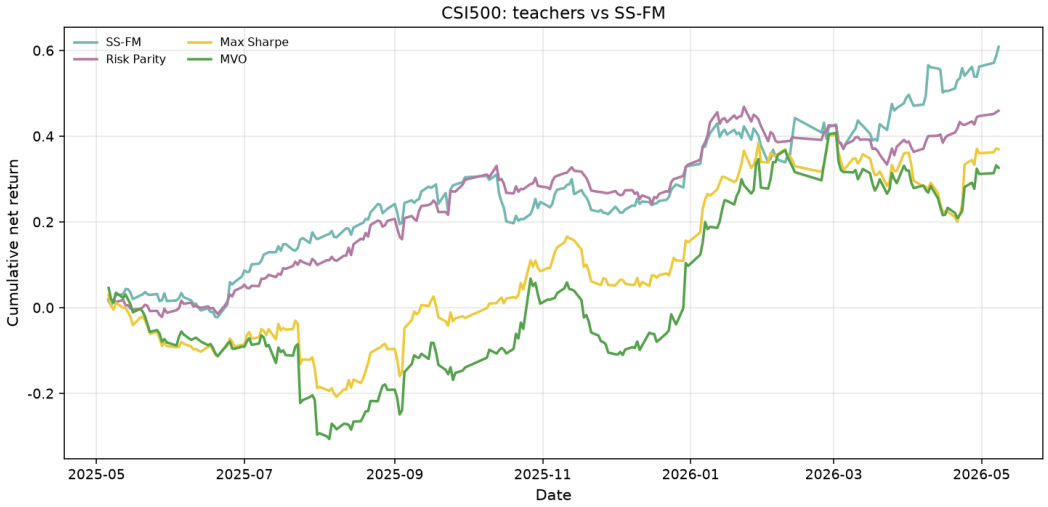

Figure 1: CSI 500 cumulative net-return trajectories of SS-FM and the optimization teachers under the Pure $G = 32$ protocol. SS-FM produces the highest terminal net return, whereas Risk Parity provides a more conservative, low-turnover trajectory.

Table 4: Generation-to-decision results on CSI 500. PPO and GRPO use the same frozen SS-FM generator under the Pure $G = 32$ protocol. Turnover denotes average daily one-way turnover.

Method	Net Return	Sharpe	Turnover
Pure SS-FM	60.8%	2.30	0.649
SS-FM + GRPO (Init.)	67.8%	2.36	0.274
SS-FM + GRPO	71.7%	2.45	0.263
SS-FM + Cand.-PPO	88.7%	2.14	**0.214**
SS-FM + PPO	**105.1%**	**2.81**	0.260

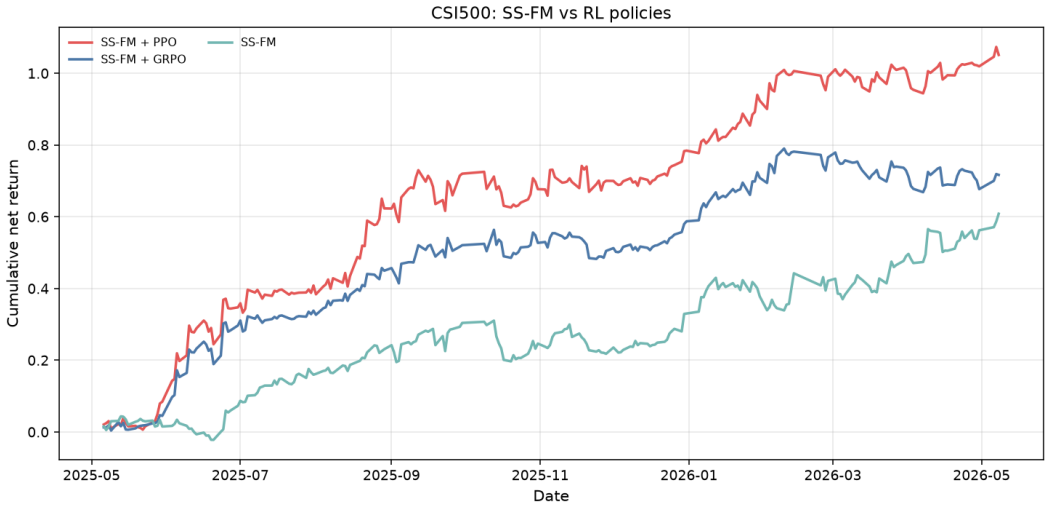

Figure 2: Generation-to-decision refinement on CSI 500. PPO and GRPO are trained on top of the same frozen SS-FM generator. PPO achieves the highest terminal cumulative net return, while candidate-conditioned PPO attains the lowest turnover in Table 4.

PPO variant is the best-performing method in each seed. These results suggest that the decision layer improves realized return on CSI 500 and improves stability on S&P 500.

5.2 Multi-Seed Stability and Performance of Generative Portfolios

To evaluate the intrinsic robustness and sampling stability of generative models, we freeze each generator and evaluate multi-seed performance across candidate budgets $G \in \{8, 32, 64, 128\}$ over 20 date-keyed random seeds in both CSI 500 and S&P 500 markets. We examine both winner-selection (Pred-Utility) and noise-resilient distributional aggregation (Softmax / Mean) protocols.

Table 5 summarizes the out-of-sample cumulative net returns across seeds. In CSI 500, SS-FM demonstrates remarkable stability and consistent growth under Softmax aggregation, achieving $51.1\% \pm 1.4\%$ at $G = 128$. Compared to Diffusion (2.4% std) and FM (0.7% std), SS-FM effectively suppresses sampling noise while maintaining highly competitive cumulative returns (50.0% $\rightarrow$ 51.1%). Under Pred-Utility selection, SS-FM steadily improves with candidate size G, scaling from $36.7\% \pm 17.5\%$ at $G = 8$ to $48.8\% \pm 14.6\%$ at $G = 128$, showing significantly lower seed variance than Diffusion (22.8% ... 29.0% std).

Table 5: Multi-seed out-of-sample cumulative net-return stability across 20 seeds (Mean $\pm$ Standard Deviation). Pred-Utility selects the candidate with the highest predicted utility, while Softmax aggregation ($\tau = 1$) provides noise-resilient distributional averaging over candidate sets.

Market	Aggregation	G	Diffusion	FM	SS-FM (Ours)
CSI 500	Pred-Utility	8	40.4% ± 23.3%	17.7% ± 7.2%	36.7% ± 17.5%
		32	73.2% ± 29.0%	25.7% ± 10.3%	41.2% ± 18.3%
		64	75.0% ± 28.0%	31.9% ± 11.9%	46.6% ± 19.9%
		128	**88.3% ± 22.8%**	37.4% ± 11.3%	**48.8% ± 14.6%**
	Softmax	8	50.9% ± 6.0%	49.9% ± 2.2%	50.0% ± 5.0%
		32	53.0% ± 3.5%	51.8% ± 1.1%	50.5% ± 2.7%
		64	54.1% ± 2.8%	52.6% ± 0.9%	51.1% ± 2.1%
		128	55.1% ± 2.4%	52.6% ± 0.7%	**51.1% ± 1.4%**
S&P 500	Pred-Utility	8	−2.4% ± 11.3%	8.0% ± 3.9%	**8.5% ± 6.3%**
		32	−8.8% ± 11.5%	8.4% ± 6.6%	7.5% ± 7.0%
		64	−10.7% ± 7.9%	7.8% ± 4.9%	2.2% ± 6.4%
		128	−17.2% ± 7.5%	6.5% ± 6.4%	1.5% ± 7.4%
	Softmax	8	−0.0% ± 2.8%	2.7% ± 1.4%	**6.9% ± 2.5%**
		32	−1.9% ± 2.6%	0.6% ± 0.7%	**6.9% ± 1.6%**
		64	−1.4% ± 1.7%	0.1% ± 0.4%	**7.0% ± 1.0%**
		128	−2.0% ± 1.2%	−0.2% ± 0.4%	**6.9% ± 0.6%**

In S&P 500, SS-FM exhibits marked superiority over baseline generators in thinner opportunity spaces. Under Softmax aggregation, SS-FM uniquely maintains a robust positive return of $6.9\% \pm 0.6\%$ across all candidate sizes, whereas FM degrades to -0.2% and Diffusion suffers negative returns (-2.0%). Under Pred-Utility selection, SS-FM achieves its peak performance of $8.5\% \pm 6.3\%$ at $G = 8$, outperforming Diffusion, which severely suffers from selection noise as candidate budget G expands (-17.2% at $G = 128$). These results validate SS-FM as a highly resilient baseline capable of producing superior conditional candidate distributions across diverse market conditions.

6 Limitations

Our study has several limitations. First, the training window covers only one year, which is relatively short for evaluating portfolio models under diverse market conditions. This limited history may constrain the stability and generalization of the learned models.

Second, we do not provide a fully matched comparison of all experimental settings across both the CSI 500 and S&P 500 markets. As a result, some cross-market comparisons are based on partially different experimental configurations.

Third, our reinforcement learning experiments focus primarily on SS-FM-generated candidates. We do not systematically evaluate how other portfolio generation models, including Gaussian, MLP, Diffusion, and Flow Matching (FM), perform when integrated with the same reinforcement learning framework. A more comprehensive comparison of different generators under identical RL settings would provide a stronger assessment of the generality of the proposed decision framework.

7 Conclusion

We introduced SS-FM, a generation-to-decision formulation of portfolio construction that replaces the conventional search for a single optimal allocation with the modeling of a structured distribution over feasible portfolio proposals. SS-FM learns this distribution in CLR-based simplex coordinates from optimization-induced supervision, while PPO or GRPO further refines the final allocation using realized portfolio-level rewards. Across strict walk-forward experiments, the proposed formulation demonstrates that simplex-aware portfolio generation can remain competitive with classical optimization and alternative generative models, while downstream policy optimization can further improve the quality of the generated portfolio prior. These results suggest that portfolio construction can be effectively viewed as two coupled problems: modeling a structured set of feasible allocations and learning which allocation to execute sequentially.

8 AI Use

We used AI assistance for revising and polishing the manuscript, as well as for deriving and checking the mathematical formulations.

References

John Aitchison. The statistical analysis of compositional data. *Journal of the Royal Statistical Society: Series B (Methodological)*, 44(2):139–160, 1982.

John Aitchison. *The Statistical Analysis of Compositional Data*. Chapman and Hall, London, 1986.

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua B. Tenenbaum, Tommi S. Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision making? In *The Eleventh International Conference on Learning Representations*, 2023.

David Bailey, Jonathan Borwein, Marcos Lopez de Prado, and Qiji Jim Zhu. The probability of backtest overfitting. *The Journal of Computational Finance*, 20(4):39–69, 2017.

Fischer Black and Robert Litterman. Global portfolio optimization. *Financial Analysts Journal*, 48(5):28–43, 1992.

Ricky T. Q. Chen and Yaron Lipman. Flow matching on general geometries. In *International Conference on Learning Representations*, volume 2024, pp. 47922–47945, 2024.

So-Yoon Cho, Jin-Young Kim, Kayoung Ban, Hyeng Keun Koo, and Hyun-Gyoon Kim. Diffolio: A diffusion model for multivariate probabilistic financial time-series forecasting and portfolio construction. *Information Fusion*, 133:104286, 2026.

Darwin Choi, Wenxi Jiang, and Chao Zhang. Alpha go everywhere: Machine learning and international stock returns. *The Review of Asset Pricing Studies*, 15(3-4):288–331, 2025.

Gang Chu, John W. Goodell, and Xiao Li. Are pre-opening periods important? Evidence from Chinese market lunch breaks. *Pacific-Basin Finance Journal*, 88:102577, 2024.

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.

Dionigi Gerace, Qigui Liu, Gary Gang Tian, and Willa Zheng. Call auction transparency and market liquidity: Evidence from China. *International Review of Finance*, 15(2):223–255, 2015.

Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. FFJORD: free-form continuous dynamics for scalable reversible generative models. In *The Seventh International Conference on Learning Representations*, 2019.

Shihao Gu, Bryan Kelly, and Dacheng Xiu. Empirical asset pricing via machine learning. *The Review of Financial Studies*, 33(5):2223–2273, 2020.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pp. 1861–1870, 2018.

James B. Heaton, Nick G. Polson, and Jan Hendrik Witte. Deep learning for finance: deep portfolios. *Applied Stochastic Models in Business and Industry*, 33(1):3–12, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. volume 33, pp. 6840–6851, 2020.

Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, volume 162, pp. 9902–9915, 2022.

Theis Ingerslev Jensen, Bryan Kelly, Semyon Malamud, and Lasse Heje Pedersen. Machine learning and the implementable efficient frontier. *The Review of Financial Studies*, 39(10): 3035–3078, 2026.

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

Bryan Kelly, Semyon Malamud, and Kangying Zhou. The virtue of complexity in return prediction. *The Journal of Finance*, 79(1):459–503, 2024.

Tze Leung Lai, Haipeng Xing, and Zehao Chen. Mean–variance portfolio optimization when means and covariances are unknown. *The Annals of Applied Statistics*, 5(2A):798–823, 2011.

Tong Li, Zhaoyang Liu, Yanyan Shen, Xue Wang, Haokun Chen, and Sen Huang. MASTER: Market-guided stock transformer for stock price forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 162–170, 2024.

Yang Li, Zhi Chen, and Steve Yang. FlowHFT: Imitation learning via flow matching policy for optimal high-frequency trading under diverse market conditions. *arXiv preprint arXiv:2505.05784*, 2025.

Zhipeng Liang, Hao Chen, Junhao Zhu, Kangkang Jiang, and Yanran Li. Adversarial deep reinforcement learning in portfolio management. *arXiv preprint arXiv:1808.09940*, 2018.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

Xiaoyang Liu, Hongyang Yang, Jiechao Gao, and Christina Dan Wang. FinRL: Deep reinforcement learning framework to automate trading in quantitative finance. In *Proceedings of the second ACM international conference on AI in finance*, pp. 1–9, 2021.

Sebastien Maillard, Thierry Roncalli, and Jerome Teiletche. The properties of equally weighted risk contribution portfolios. *The Journal of Portfolio Management*, 36(4):60–70, 2010.

Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952.

Richard O. Michaud. The Markowitz optimization enigma: Is 'optimized'optimal? *Financial Analysts Journal*, 45(1):31–42, 1989.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pp. 1530–1538, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. In *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y.K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

William F. Sharpe. Mutual fund performance. *The Journal of Business*, 39(1):119–138, 1966.

Frederik Simon, Sebastian Weibels, and Tom Zimmermann. Deep parametric portfolio policies. *Management Science*, 2026.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *The Ninth International Conference on Learning Representations*, 2021.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. volume 30, 2017.

Zhendong Wang, Jonathan J. Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023.

Zhicheng Wang, Biwei Huang, Shikui Tu, Kun Zhang, and Lei Xu. DeepTrader: A deep reinforcement learning approach for risk-return balanced portfolio management with market conditions embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 643–650, 2021.

David Winkel, Niklas Strauß, Maximilian Bernhard, Zongyue Li, Thomas Seidl, and Matthias Schubert. Autoregressive policy optimization for constrained allocation tasks. In *Advances in Neural Information Processing Systems*, volume 37, pp. 119150–119173, 2024.

A Feature Construction and Market Encoders

A.1 Causal Intraday Features

For each asset, the market encoder receives causal minute-level observations from the preceding ten complete trading days. Features include OHLCV, multi-horizon returns, realized volatility, volume statistics, VWAP-related statistics, moving-average features, liquidity measures, cross-sectional ranks, calendar features, overnight returns, and overnight volatility. All features are aligned to ensure that only observations available before the portfolio execution time are used.

The temporal encoder is Transformer-based (Vaswani et al., 2017):

$$h_{t,i}^{\text{minute}} = \text{Transformer}\left(X_{t-10:t-1,i}^{\text{minute}}\right). \tag{23}$$

Cross-stock attention further contextualizes each asset representation using the contemporaneously tradable universe (Li et al., 2024):

$$h_{t,i}^{\text{cross}} = f_{\text{cross}}\left(h_{t,1}^{\text{minute}}, \ldots, h_{t,N_t}^{\text{minute}}\right)_i. \tag{24}$$

A.2 Opening-Auction Fusion

For CSI 500, the model additionally receives overnight and opening-auction features. Let

$$h_t^{\text{auction}} = f_{\text{auction}}\left(X_{t-1:t}^{\text{auction}}\right) \tag{25}$$

denote the auction representation. Such information is motivated by the distinct price-formation process around market opening (Gerace et al., 2015; Chu et al., 2024).

We use gated residual fusion:

$$g_t = \sigma\left(W_g\left[h_t^{\text{minute}}; h_t^{\text{auction}}\right] + b_g\right), \tag{26}$$

$$h_t = h_t^{\text{minute}} + g_t \odot h_t^{\text{auction}}. \tag{27}$$

The S&P 500 experiments do not include auction features and should therefore be interpreted as tests of the core generation-to-decision framework rather than as auction-fusion comparisons.

A.3 Alpha Target and Universe Construction

The alpha model predicts next-day Open-to-Close log return:

$$y_{t,i} = \log\left(\frac{C_{t+1,i}}{O_{t+1,i}}\right). \tag{28}$$

Assets are ranked by $\hat{y}_{t,i}$ and the Top-30 ranked assets form the portfolio universe. The same daily selected universe is used across compared portfolio-construction methods.

B Simplex Modeling Details

B.1 Positive Weight Floor

CLR coordinates require strictly positive portfolio weights. For a teacher portfolio w, we use

$$\widetilde{w}_i = \frac{\max(w_i, \varepsilon_{\text{clr}})}{\sum_{j=1}^{K} \max(w_j, \varepsilon_{\text{clr}})}. \tag{29}$$

Here $\varepsilon_{\text{clr}} > 0$ is a small numerical floor. The CLR transformation is applied to $\widetilde{w}$ rather than directly to w.

B.2 Zero-Sum Base Distribution

To ensure that the base distribution lies in the CLR subspace, we sample

$$\epsilon \sim \mathcal{N}(0, I_K), \qquad z_0 = P\epsilon, \qquad P = I_K - \frac{1}{K}\mathbf{1}\mathbf{1}^\top. \tag{30}$$

The projector P satisfies $\mathbf{1}^\top P = 0$. In implementation, the vector-field output may also be projected onto the zero-sum subspace:

$$v_\theta^{\mathcal{T}} = P\widetilde{v}_\theta. \tag{31}$$

This keeps the flow in the CLR tangent subspace throughout numerical integration.

B.3 Teacher Portfolios

At each training decision time, teacher portfolios are computed using only information available before execution:

$$\mathcal{Y}_t = \left\{ w_t^{(\text{MVO})}, w_t^{(\text{MS})}, w_t^{(\text{RP})} \right\}. \tag{32}$$

MVO follows the classical risk–return trade-off (Markowitz, 1952); Maximum Sharpe targets risk-adjusted return (Sharpe, 1966); and Risk Parity balances risk contributions (Maillard et al., 2010). Covariance estimation, expected-return estimation, solver configurations, regularization, and portfolio bounds must be estimated using historical training information only.

C Optimization and Policy Details

C.1 PPO

The candidate-selection policy produces

$$p_t^{(g)} = \pi_\phi\left(g \mid h_t, \mathcal{W}_t\right), \qquad g \in \{1, \ldots, G\}. \tag{33}$$

For the sampled candidate action g_t, the likelihood ratio is

$$\rho_t(\phi) = \frac{\pi_\phi\left(g_t \mid h_t, \mathcal{W}_t\right)}{\pi_{\phi_{\text{old}}}\left(g_t \mid h_t, \mathcal{W}_t\right)}. \tag{34}$$

PPO optimizes the clipped surrogate objective (Schulman et al., 2017):

$$L_{\mathrm{PPO}} = \mathbb{E}_t \left[\min \left(\rho_t(\phi) \hat{A}_t, \mathrm{clip} \left(\rho_t(\phi), 1-\epsilon, 1+\epsilon \right) \hat{A}_t \right) \right]. \tag{35}$$

A separate critic estimates the state–candidate value and provides the advantage estimate $\hat{A}_t$.

C.2 GRPO

For GRPO (Shao et al., 2024), the policy is initialized from the pre-trained SS-FM distribution. At each decision time, it samples a group of G portfolio candidates and evaluates each candidate under its realized portfolio-level reward. The normalized group-relative advantage is

$$\hat{A}_t^{(g)} = \frac{r_t^{(g)} - \mu_t}{\sigma_t + \epsilon}, \tag{36}$$

where μ_t and σ_t are the within-group reward mean and standard deviation. GRPO does not use a separate value function, instead relying on the within-group relative learning signal.

C.3 Reward

The reward is

$$r_t = 0.6 z \left(R_t^{\mathrm{net}} \right) + 0.4 z \left(R_t^{\mathrm{sharpe}} \right) - 0.05 z \left(\mathrm{TO}_t \right) - 0.05 z \left(\mathrm{Cost}_t \right). \tag{37}$$

Normalization statistics are computed using training data only. The same reward definition is used for PPO and GRPO to isolate the effect of policy optimization.

D Walk-Forward Protocol and Reproducibility

D.1 Temporal Leakage Control

Financial experiments are particularly sensitive to temporal leakage (Bailey et al., 2017). For every test month T, we use

$$\begin{aligned} \mathrm{Train} &= [T-13, T-2], \\ \mathrm{Validation} &= T-1, \\ \mathrm{Test} &= T. \end{aligned} \tag{38}$$

The held-out test month is not used for hyperparameter tuning, feature normalization, covariance estimation, teacher portfolio generation, flow matching updates, RL updates, reward normalization, or model selection.

D.2 Transaction Costs and Metrics

Turnover is defined by

$$\mathrm{TO}_t = \frac{1}{2} \sum_{i=1}^{K} \left| w_{t,i} - w_{t-1,i} \right|. \tag{39}$$

Net return is

$$r_t^{\mathrm{net}} = r_t^{\mathrm{gross}} - c\, \mathrm{TO}_t. \tag{40}$$

The transaction-cost coefficient is fixed at 3 basis points before out-of-sample evaluation. The annualized Sharpe ratio is

$$\mathrm{Sharpe} = \frac{\mathrm{mean} \left(r_t^{\mathrm{net}} \right)}{\mathrm{std} \left(r_t^{\mathrm{net}} \right)} \sqrt{A}, \tag{41}$$

where A is the annualization factor. Maximum drawdown is calculated from the realized cumulative net-return curve.

D.3 Recorded Artifacts

For every walk-forward run, we record random seeds, model configurations, data versions, feature schemas, train/validation/test splits, normalization statistics, checkpoints, optimizer states, generated candidates, final portfolio weights, and backtest outputs. All methods use identical evaluation scripts and transaction-cost definitions.

E Candidate Geometry and Generation Stability

E.1 Teacher Interaction and Candidate Geometry

We test whether SS-FM simply reproduces its three portfolio teachers and whether explicit teacher-aware interventions improve the generated candidate set. We conduct the teacher experiments on S&P 500 with 50 generation seeds. For teacher portfolio $w^{(m)}$ and generated candidate $w^{(g)}$, we define coverage and diversity as

$$\text{Coverage} = \frac{1}{3G} \sum_{m=1}^{3} \sum_{g=1}^{G} \mathbf{1} \left\{ \left\| w^{(g)} - w^{(m)} \right\|_1 \leq 0.35 \right\}, \tag{42}$$

$$\text{Diversity} = \frac{2}{G(G-1)} \sum_{i<j} \left\| w^{(i)} - w^{(j)} \right\|_1 . \tag{43}$$

E.1.1 Exp1: Teacher Coverage and Diversity

Table 6 shows that increasing G improves coverage, but coverage remains limited. At the default $G = 32$, only 5.0% of generated candidate–teacher pairs fall within the predefined L_1 neighborhood. Meanwhile, diversity remains near 1.41 across candidate budgets. The generated distribution is closest to Risk Parity and remains farther from MVO and Maximum Sharpe. Thus, SS-FM produces a dispersed candidate cloud rather than three deterministic clusters centered on its teacher portfolios.

E.1.2 Exp2: Teacher-Aware Sampling

We keep the generator fixed and modify only the candidate-sampling procedure. Teacher-aware sampling oversamples candidates, retains candidates closest to each teacher, and fills the remaining budget using a max–min diversity rule. It increases coverage from 0.019 to 0.039 at $G = 8$, from 0.050 to 0.081 at $G = 32$, and from 0.091 to 0.112 at $G = 128$.

Under Pred-Utility, teacher-aware sampling improves the small-budget configuration: at $G = 8$, cumulative net return increases from $7.4\% \pm 7.0\%$ to $10.1\% \pm 8.6\%$. The gain is small at $G = 32$, where random and teacher-aware sampling yield $5.8\% \pm 7.6\%$ and $6.2\% \pm 8.2\%$, respectively. Under Mean aggregation, forcing teacher coverage reduces performance at $G = 32$, from approximately 6.8% to 2.7%. Hence, higher geometric coverage does not mechanically imply better out-of-sample returns.

E.1.3 Exp3: Teacher-Aware Retraining

Finally, we augment the generator objective with coverage and diversity terms and evaluate retraining jointly with teacher-aware sampling. Table 7 summarizes Pred-Utility performance. The original generator with teacher-aware sampling gives the highest reported value at $G = 8$ in this experiment. Teacher-aware retraining does not uniformly improve performance, and Mean aggregation after retraining is often close to zero or negative. These experiments indicate that the original multi-teacher SS-FM objective already encodes a useful proposal distribution, whereas forcing closer geometric alignment to the teacher set can reduce useful variation.

Table 6: SS-FM teacher coverage and candidate diversity on S&P 500. Coverage is the fraction of candidate–teacher pairs within the predefined L_1 neighborhood. Diversity is average pairwise candidate L_1 distance.

G	Coverage	Diversity	Min. L_1 to MVO	Min. L_1 to MS	Min. L_1 to RP
8	0.0185	1.405	1.666	1.572	0.750
32	0.0500	1.412	1.413	1.333	0.675
64	0.0702	1.413	1.285	1.226	0.645
128	0.0914	1.411	1.160	1.127	0.612

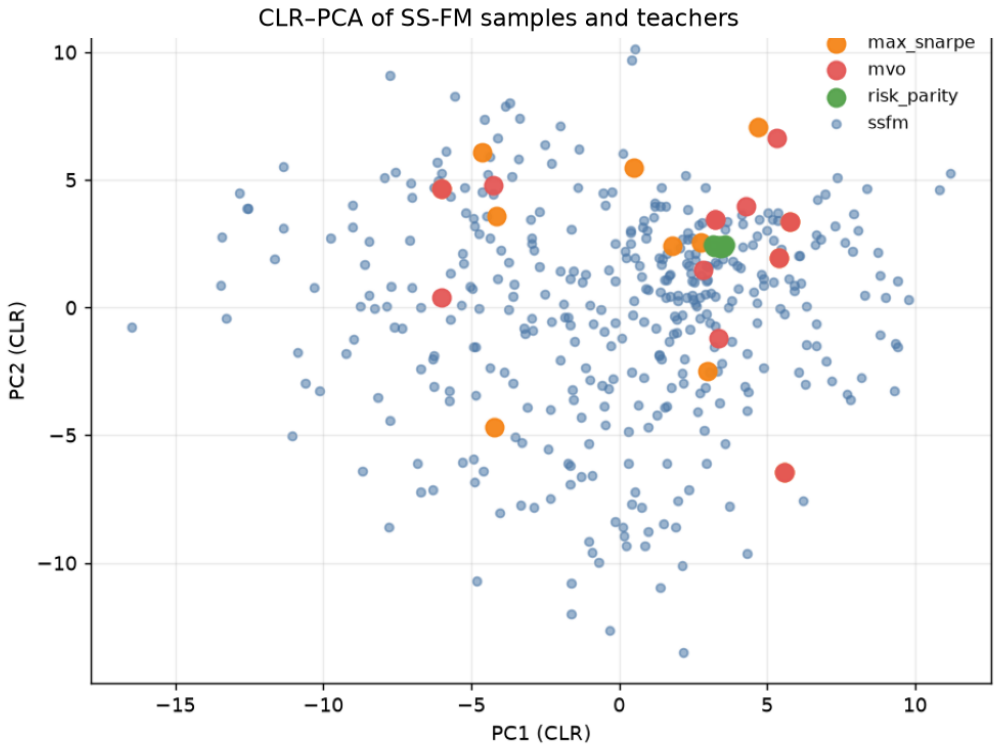

Figure 3: SS-FM candidate geometry on S&P 500. Increasing the candidate budget improves teacher coverage, while pairwise candidate diversity remains approximately stable. The visualization uses CLR-space PCA.

Table 7: Teacher-intervention experiments on S&P 500. Values are cumulative net returns under Pred-Utility aggregation. "TA" denotes teacher-aware.

Generator	Sampling	$G = 8$	$G = 32$	$G = 64$
Pure	Random	8.5%	7.5%	2.2%
Pure	**TA**	**11.2%**	4.8%	4.8%
TA-retrained	Random	9.3%	5.8%	3.1%
TA-retrained	TA	7.7%	6.5%	7.0%

E.2 Generation Stability Across Seeds

We further evaluate the sensitivity of generated candidate sets to the initial sampling noise. The experiment uses 20 independent generation seeds for each candidate budget. Pred-Utility selects the candidate with the highest predicted utility, while Mean aggregation averages the generated candidate portfolios.

Table 8: Multi-seed cumulative net return on CSI 500 under Pred-Utility aggregation. Values are mean ± standard deviation across 20 seeds.

G	Diffusion	FM	SS-FM
8	40.4% ± 23.3%	17.7% ± 7.2%	36.7% ± 17.5%
32	73.2% ± 29.0%	25.7% ± 10.3%	41.2% ± 18.3%
64	75.0% ± 28.0%	31.9% ± 11.9%	46.6% ± 19.9%
128	**88.3% ± 22.8%**	37.4% ± 11.3%	48.8% ± 14.6%

Pred-Utility selection can improve best-case portfolio performance but remains seed-sensitive. Mean and Softmax aggregation reduce this sensitivity and are therefore used as more stable primary distributional aggregation protocols.

F Additional RL Stability Results

For the S&P 500 multi-seed experiment, candidate-conditioned PPO achieves mean cumulative net return of 17.9% with standard deviation 3.5% across reported seeds. The corresponding SS-FM Meta baseline achieves $0.6\% \pm 6.8\%$, while GRPO initialization achieves $5.2\% \pm 2.8\%$. Candidate-conditioned PPO is the best-performing method in each reported seed. Future experiments should increase the number of independent training and sampling seeds and report confidence intervals for all portfolio metrics.