DECISION-UTILITY TEST-TIME TRAINING UNDER DELAYED MARKET FEEDBACK

Anonymous authors
Paper under double-blind review

ABSTRACT

Deployed return forecaster must adapt online to non-stationary environments. Standard test-time training (TTT) strategies fail here for two reasons: **misalignment**, as mean-squared error ignores decision boundaries and portfolio mechanics, and **delayed, constrained feedback**, as labels arrive h days later under strict leverage and fee constraints. To address these challenges, we propose **D**ecision-**U**tility **T**est-**T**ime **T**raining (**DU-TTT**), which adapts a lightweight affine head on a frozen feature extractor directly using downstream portfolio utility. DU-TTT buffers historical decision contexts, evaluates matured h-day returns, and updates the adaptation head via a covariance-weighted utility gradient. To focus feedback where it matters most, DU-TTT incorporates a soft top-K selection mechanism, concentrating adaptation on the active trading margin while dynamically adjusting step size based on forecast dispersion. Evaluated across 109 equity universes over two distinct market regimes, DU-TTT consistently outperforms standard MSE-based adaptation in both portfolio growth and Sharpe ratio. Our analysis reveals that DU-TTT achieves its primary advantage by suppressing unrewarded turnover and trading friction. Furthermore, ablation studies confirm the necessity of both the margin-focused selection and dispersion-scaled step sizes. By directly aligning test-time adaptation with financial utility, DU-TTT offers a robust framework for real-world algorithmic trading under distribution shift.

1 INTRODUCTION

Return forecasters must adapt online to non-stationary environments, a challenge addressed by test-time training (TTT) in vision (Sun et al., 2020; Wang et al., 2021) and time-series forecasting (Zhang et al., 2023; Kim et al., 2025). However, adapting financial forecasters involves delayed feedback and execution constraints: labels mature h trading days post-decision, and portfolio updates incur position caps and trading fees. Prior work has separately addressed TTT for forecasting (Zhang et al., 2023; Kim et al., 2025), decision-focused learning (Elmachtoub & Grigas, 2022; Mandi et al., 2024), and delayed online learning (Joulani et al., 2013; Amoh et al., 2026).

Adapting a return forecaster directly for downstream execution faces two key limitations:❶ Mismatch between forecast error and decision utility: Mean-squared error (MSE) poorly aligns with portfolio mechanics. Uniform shifts change MSE without altering rank-based allocations, whereas scale shifts preserve ranks while driving unrewarded turnover (Figure 1, middle). ❷Delayed and constrained feedback: Model states drift before h-day labels arrive. Updates must map outcomes back to their original decision contexts under tight leverage constraints, where position caps concentrate choices onto a narrow margin (Figure 1, left and right). Existing decision-focused approaches remain offline and static (Linghu et al., 2025; Wang & Hasuike, 2026), while delayed online methods ignore portfolio utility. This necessitates an online adaptation framework that updates forecasters directly via the utility of constrained, real-world portfolio decisions under delayed feedback.

To address these issues, we introduce **D**ecision-**U**tility **T**est-**T**ime **T**raining (**DU-TTT**), which adapts a lightweight $2n$-parameter affine head on a frozen forecaster using the mean–variance utility of the induced portfolio rather than squared prediction error. DU-TTT operates through three steps: **delayed calibration** buffers decision contexts (forecasts and covariances) so that matured h-day labels evaluate outcomes at the exact state that generated them; **utility feedback** updates the head along the derivative of portfolio utility through an equality-relaxed allocation map, removing forecast

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.

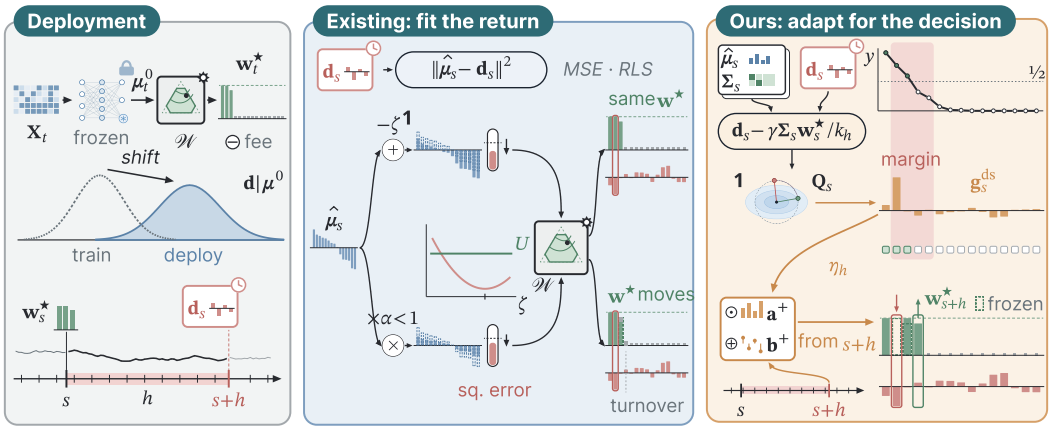


Figure 1: Compare Learning target choice in test-time adaptation. Frozen forecaster generates predictions under drift, producing capped, fee-constrained allocations with delayed labels. Standard MSE adaptation optimizes statistical error over utility, DU-TTT derives marginal utility from matured contexts, isolates the active trading margin and directly updates the allocation set.

level noise and weighting errors by their execution impact; and **margin-driven feedback** applies a soft top-K selection mechanism to focus updates on assets near the trading boundary while scaling the step size by forecast dispersion. Because adaptation is restricted to the output head, DU-TTT applies seamlessly across diverse backbones, including ridge regression, boosted trees, and frozen deep models. Across 109 equity universes under realistic trading friction (25 bps) over two evaluation periods, DU-TTT consistently suppresses unrewarded turnover while improving net portfolio growth and Sharpe ratio compared to standard squared-error calibration.

The main contributions of this paper are: ❶ **Decision-utility test-time training.** We formulate test-time adaptation for return forecasting as learning directly from delayed labels through the utility of constrained downstream portfolio decisions, offering a unified output-head adaptation framework across diverse backbones. ❷ **Margin-driven feedback.** We introduce a soft top-K selection mechanism that derives both a margin-focused feedback reweighting and a dispersion-scaled adaptation step size to concentrate learning where decisions are actively made. ❸ **Extensive empirical validation.** Across two distinct evaluation periods and four representative backbones, DU-TTT consistently outperforms standard squared-error calibration. Comprehensive ablation studies confirm that aligning adaptation with decision utility suppresses unrewarded turnover and provides robust, cost-effective gains under realistic trading friction and real trading company practice.

2 Related Work

Table 1: Taxonomy of related work vs. DU-TTT. Columns: online updates, delayed labels, decision loss, portfolio risk, trading costs; ○ for partial.

Method	Onl.	Delay	Dec.	Risk	Cost	Method	Onl.	Delay	Dec.	Risk	Cost
Tent (Wang et al., 2021)	✓	✗	✗	✗	✗	KKT-embedded DFL (Nosaka et al., 2026)	○	✗	✓	✓	✗
TAFAS (Kim et al., 2025)	✓	✓	✗	✗	✗	IPMO (Linghu et al., 2025)	○	✗	✓	✓	✓
PETSA (Medeiros et al., 2025)	✓	✓	✗	✗	✗	Online DFL (Capitaine et al., 2026)	✓	✗	✓	✗	✗
Proceed (Zhao & Shen, 2025)	✓	✓	✗	✗	✗	IGT-OMD (Amoh et al., 2026)	✓	✓	✓	✗	✗
Mean–variance IPO (Butler & Kwon, 2023)	○	✗	✓	✓	✗	ADOWIP (Wang, 2026)	✓	✓	○	○	✓
Sparse tangency layer (Jeon et al., 2026)	✗	✗	✓	✓	✗	**DU-TTT (ours)**	✓	✓	✓	✓	✓

Table 1 taxonomizes representative adaptation paradigms across five essential deployment criteria in quantitative trading. Extended discussions appear in Appendix C.

Test-Time Training (TTT) & Online Forecasting. TTT adapts deployed models at test time via self-supervision (Sun et al., 2020; Wang et al., 2021) or sequential targets under non-stationary drift (Zhang et al., 2023; Kim et al., 2025). Recent lightweight variants adapt only output components to preserve backbone representations (Chen et al., 2024; Lee et al., 2025b). Under delayed labels,

protocols evaluate strictly on matured ground truth (Zhao & Shen, 2025; Wang et al., 2026). However, these approaches optimize purely statistical forecast errors rather than downstream portfolio utility.

Decision-Focused Learning (DFL). DFL integrates optimization tasks directly into model training (Elmachtoub & Grigas, 2022; Mandi et al., 2024), including mean–variance portfolio objectives (Butler & Kwon, 2023; Lee et al., 2025a). Financial DFL methods typically treat turnover via explicit penalty heuristics (Linghu et al., 2025; Wang & Hasuike, 2026) and remain static or periodically refitted offline across market regimes. In contrast, DU-TTT operates continuously online via an equality-relaxed decision map without costly full solver backpropagation.

Online Learning with Delayed Feedback. Delayed online learning maps late-arriving losses back to historical decision contexts (Joulani et al., 2013; Amoh et al., 2026). Recent online DFL frameworks (Capitaine et al., 2026; Wang, 2026) and online portfolio updates (Li & Hoi, 2014) address non-stationarity, but none update return forecaster against utility under position cap and fees.

Learning to Rank & Parametric Policies. Ranking surrogates (Xia et al., 2008; Lin et al., 2026) and parametric policies (Brandt et al., 2009) optimize relative orderings or direct weight maps. Continuous top-K operators (Sander et al., 2023; Jeon et al., 2026) enable smooth selection but discard asset covariance and portfolio risk. DU-TTT uses a soft top-K operator solely to isolate the active selection margin and scale steps by dispersion, while guiding updates strictly along the true portfolio utility gradient aligned with real quant investment company practice.

3 PRELIMINARIES

Evaluation universe of $n > 1$ assets, indexed by trading days t. For each asset i, $x_{t,i} \in \mathbb{R}^p$ ($p = 8$) contains standardized price features stacked into $X_t \in \mathbb{R}^{n\times p}$. A frozen forecaster generates a base return forecast μ_t^0 (e.g., $\mu_t^0 = X_t\theta_0$ for Ridge, or non-linear outputs for foundation models). Because adaptation is restricted to μ_t^0, no backpropagation through the feature extractor is required. Let $P_t \in \mathbb{R}_+^n$ denote adjusted closing prices; the decision target at date s is the h-day realized return

$$d_s = P_{s+h} \oslash P_s - \mathbf{1}, \qquad h \in \{1, 5, 20\},$$

where $\oslash$ is componentwise division and $\mathbf{1}$ is the all-ones vector. The label d_s matures and becomes observable at close $s + h$, ensuring strict temporal causality without label leakage.

Decision layer. Let $\Sigma_s \succ 0$ be the annualized return covariance available at date s, $k_h = 252/h$, and $\gamma > 0$ the risk aversion parameter. For any forecast $v \in \mathbb{R}^n$, the mean–variance utility and the corresponding optimal allocation are defined as

$$U_{s,h}(w; v) = v^\top w - \frac{\gamma}{2k_h} w^\top \Sigma_s w, \qquad \mathcal{A}_{s,h}(v; \gamma) = \arg\max_{w\in\mathcal{W}} U_{s,h}(w; v), \tag{1}$$

over the feasible set $\mathcal{W} = \{w : 0 \le w \le w_{\max}\mathbf{1},\ \mathbf{1}^\top w = 1\}$ with per-asset cap $w_{\max} \ge 1/n$. All baseline methods share this exact decision layer and optimization setup.

Invariances. Due to the budget constraint $\mathbf{1}^\top w = 1$, a location shift ($v + c\mathbf{1}$) leaves the allocation $\mathcal{A}_{s,h}$ completely unchanged. Conversely, a positive scale ($\kappa v, \kappa > 0$) preserves asset rankings yet directly alters the trade magnitudes and portfolio execution costs. While mean-squared error is sensitive to location noise but insensitive to decision boundaries, downstream utility requires an adaptation objective aligned with these execution invariances (Appendix A.1).

4 METHODOLOGY

The core idea of DU-TTT is to adapt a frozen forecaster using the downstream utility of its induced decisions. As illustrated in Figure 2, DU-TTT consists of three core components: ❶ *Delayed affine calibration*, which maintains a lightweight $2n$-parameter head between the forecaster and allocator while buffering decision contexts for asynchronous updates (Section 4.1); ❷ *Utility feedback*, which updates the head along the derivative of mean–variance utility through an equality-relaxed allocation map, removing forecast level noise and weighting errors by asset covariance (Section 4.2); and ❸ *Margin-driven feedback*, which utilizes a soft top-K selection mechanism to concentrate feedback on assets near the trading boundary while scaling step sizes by forecast dispersion (Sections 4.3 and 4.3).

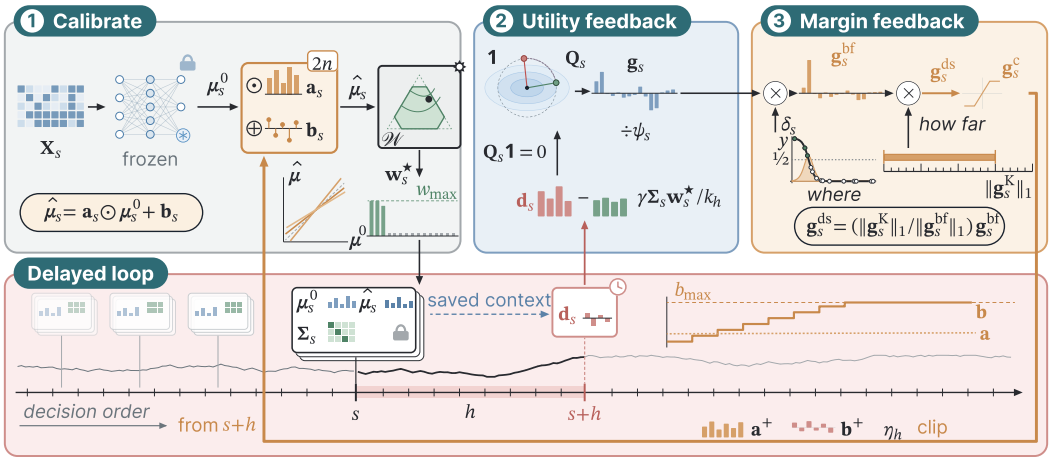


Figure 2: Overview of DU-TTT adaptation loop. *Calibrate:* Frozen model generates predictions, calibrated by a $2n$-parameter head to produce capped portfolio allocations while saving decision contexts at s. *Utility feedback:* Label maturation ($s + h$), marginal utility is evaluated at the saved context and level-projected. *Margin feedback:* Soft top-K mechanism reweights feedback onto the active trading margin and scales the update step by forecast dispersion. *Delayed loop:* Clipped updates adapt head parameters from close $s + h$ onward without modifying historical decisions.

4.1 Delayed Affine Calibration

To accommodate multi-day return latency without leaking future information, DU-TTT isolates adaptation to a lightweight per-asset affine head over the frozen base forecaster:

$$\hat{\mu}_{s,i} = a_{s,i}\mu^0_{s,i} + b_{s,i}, \tag{2}$$

initialized with gains $a_0 = \mathbf{1}$ and biases $b_0 = \mathbf{0}$. At decision date s, the learner buffers the context tuple $(\mu^0_s, \hat{\mu}_s, \Sigma_s)$. When the realized return d_s matures at close $s + h$, the loss gradient g^{ds}_s (Section 4.3) is evaluated strictly at the saved context $\hat{\mu}_s$ and applied to update the *current* head parameters (a^-, b^-).

Each gradient component is clipped to $[-g_{\max}, g_{\max}]$, yielding g^{c}_s. The update uses a horizon-adjusted learning rate $\eta_h = \eta(h/5)^\beta$, where η is the base step size and β is the horizon scaling exponent:

$$a^+ = \mathrm{clip}_{[0.2,3]}\big(a^- - \eta_h\, g^{\mathrm{c}}_s \odot \mu^0_s\big), \qquad b^+ = \mathrm{clip}_{[-b_{\max},b_{\max}]}\big(b^- - \eta_h\, g^{\mathrm{c}}_s\big), \tag{3}$$

where $\odot$ denotes elementwise multiplication and $b_{\max}$ bounds the bias range. Asynchronous matured records are processed sequentially in decision order. By decoupling context generation from online parameter updates, DU-TTT ensures strict temporal causality past decisions remain fixed while updated head parameters govern only future allocations across all backbones.

4.2 Utility Feedback

Standard squared error penalizes forecast level shifts that have no bearing on portfolio weights. DU-TTT replaces it with the derivative of mean–variance utility evaluated through an equality-relaxed allocation map. Define the projected inverse covariance and the corresponding allocation Jacobian as

$$Q_s = \Sigma_s^{-1} - \frac{\Sigma_s^{-1}\mathbf{1}\mathbf{1}^\top\Sigma_s^{-1}}{\mathbf{1}^\top\Sigma_s^{-1}\mathbf{1}}, \qquad J^{\mathrm{eq}}_{s,h} = \frac{k_h}{\gamma}Q_s. \tag{4}$$

The projection operator eliminates forecast level noise ($Q_s\mathbf{1} = \mathbf{0}$), while $k_h = 252/h$ converts the Jacobian to h-period forecast units. Evaluated at the actual held allocation $w^\star_s = \mathcal{A}_{s,h}(\hat{\mu}_s; \gamma)$, the normalized utility direction is

$$g_s = -\frac{J^{\mathrm{eq}}_{s,h}\big(d_s - \gamma\Sigma_s w^\star_s/k_h\big)}{\psi_s}, \qquad \psi_s = \max\Big\{\frac{1}{n}\sum_{i,j}\big|(Q_s/\gamma)_{ij}\big|,\, 10^{-12}\Big\}, \tag{5}$$

where the term in parentheses represents the marginal portfolio utility at realized return d_s, and ψ_s scales the gradient by the average magnitude of Q_s/γ prior to clipping.

Equality-relaxation mechanism. Following relaxed IPO (Butler & Kwon, 2023), while execution enforces position bounds, gradient computation treats coordinates as free. This avoids the vanishing gradients inherent to exact piecewise-constant maps under tight position caps (Appendix A.3).

Action-target interpretation. Let $w^{\mathrm{eq}}_{s,h}(d)$ denote the unconstrained equality-only portfolio under realized return d_s. The normalized direction simplifies directly to an action-minus-target residual:

$$g_s = \frac{w^\star_s - w^{\mathrm{eq}}_{s,h}(d_s)}{\psi_s}, \tag{6}$$

which corresponds to the exact gradient of a regularized convex potential (Blondel et al., 2020a). By projecting out common shifts, g_s isolates errors that actively alter asset weights, directly suppressing unrewarded portfolio turnover (Appendix B).

4.3 Margin-Driven Feedback

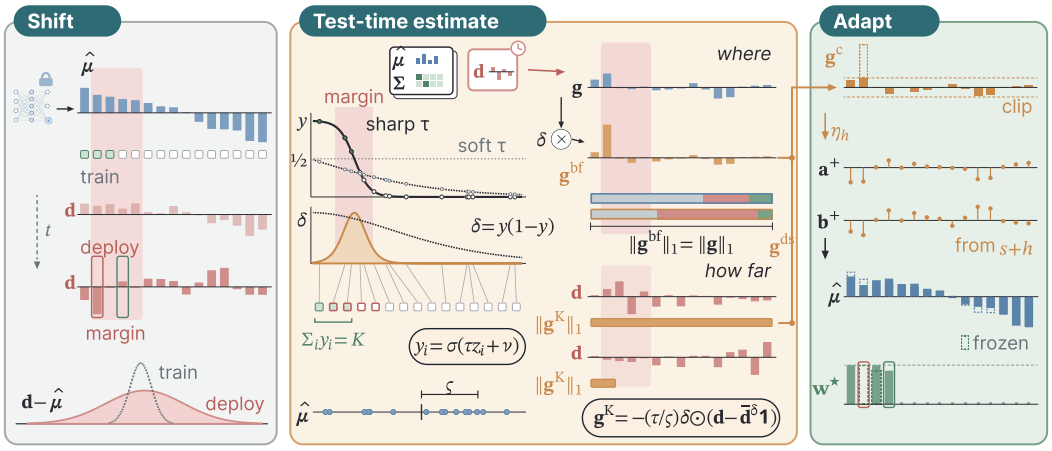


Figure 3: Test-time adaptation at the selection margin. When market drift induces errors near the selection threshold, our framework estimates marginal gradients to update the model head, dynamically replacing underperforming boundary assets.

Figure 3 illustrates our test-time adaptation workflow targeted at boundary assets. Under market regime shifts, frozen model predictions often lose accuracy near the portfolio selection threshold. By leveraging membership derivatives upon feedback maturation, our method focuses gradient estimates specifically on marginal assets to scale and directionally guide the update. Consequently, updating the prediction head enables the policy to dynamically swap degraded boundary assets for higher-yielding candidates in subsequent decisions. While the equality-relaxed Jacobian $J^{\mathrm{eq}}_{s,h}$ distributes feedback across all assets, position caps mean that only assets near the selection threshold actively drive allocation changes. To concentrate adaptation where decisions are actually made, we leverage the differentiable soft top-K membership from the limited multi-label layer (Amos et al., 2019):

$$y_{s,i} = \sigma(\tau z_{s,i} + \nu_s), \qquad \textstyle\sum_i y_{s,i} = K, \qquad z_s = \big(\hat{\mu}_s - \bar{\hat{\mu}}_s\big)/\varsigma_s, \tag{7}$$

where σ is the logistic function, ς_s is the cross-sectional standard deviation of $\hat{\mu}_s$, τ is temperature, ν_s enforces the budget, and $K = \lceil 1/w_{\max} \rceil$ is the portfolio cardinality. Its derivative $\delta_{s,i} = y_{s,i}(1-y_{s,i})$ peaks at $y_{s,i} = 1/2$ (the selection boundary) and vanishes for firmly held or excluded assets. We reweight the utility direction via

$$g^{\mathrm{bf}}_s = \varphi_s\big[(1-\alpha)\mathbf{1} + \alpha\, \delta_s/\bar{\delta}_s\big] \odot g_s, \qquad \varphi_s = \frac{\|g_s\|_1}{\big\|\big[(1-\alpha)\mathbf{1} + \alpha\, \delta_s/\bar{\delta}_s\big] \odot g_s\big\|_1}, \tag{8}$$

where $\bar{\delta}_s$ is the mean of δ_s and $\alpha \in [0,1]$. The normalization scalar φ_s guarantees $\|g^{\mathrm{bf}}_s\|_1 = \|g_s\|_1$, isolating the spatial allocation of feedback (*where*) from its step size. Because Eq. (7) operates on standardized predictions, g^{bf}_s inherits the scale and location invariances of the underlying portfolio.

Dispersion-scaled step size. While g_s^{bf} determines the directional focus, measuring gradient length across all n assets through ψ_s dilutes learning with unselected names. Differentiating the soft selection payoff $-y_s^\top d_s$ with respect to $\hat{\mu}_s$ yields a boundary-native magnitude:

$$g_s^K = -\frac{\tau}{\varsigma_s}\,\delta_s \odot \big(d_s - \bar{d}_s^{\delta}\mathbf{1}\big), \qquad \bar{d}_s^{\delta} = \frac{\sum_i \delta_{s,i} d_{s,i}}{\sum_i \delta_{s,i}}, \qquad g_s^{\mathrm{ds}} = \frac{\|g_s^K\|_1}{\|g_s^{\mathrm{bf}}\|_1}\, g_s^{\mathrm{bf}}. \tag{9}$$

The final update g_s^{ds} preserves the margin-focused direction g_s^{bf} while scaling step length by $\|g_s^K\|_1$. The step size expands with the cross-sectional dispersion of realized returns among marginal assets relative to forecast spread ς_s, ensuring larger adaptation steps when matured returns reshuffle boundary orderings (Figure 2) make the final decision aligned with the investment goal.

Temperature setting and effective support. The temperature τ controls the *effective support* $m_s(\tau)$ of the weighting:

$$m_s(\tau) = \Big(\textstyle\sum_i \delta_{s,i}\Big)^2 \Big/ \sum_i \delta_{s,i}^2, \tag{10}$$

defined as the participation ratio of δ_s. Matching $m_s(\tau)$ to the portfolio capacity K ensures that feedback covers exactly the active decision boundary. The adaptation signal combines three complementary components: covariance-aware risk via Σ_s, boundary isolation via the membership derivative δ_s, and step length scaled by marginal return dispersion.

5 Experiments

Data. We evaluate daily U.S. equity portfolios across 109 representative universes (6–25 assets) drawn from a 500-name adjusted-price panel, constructed via rules on 2022–2023 financial statistics. Operating each universe at decision horizons $h \in \{1, 5, 20\}$ yields 327 distinct universe–horizon *cells*. Table 7 details the chronological folds. Within each fold, validation selects hyperparameters, the base forecaster is refitted up to the fold cutoff, and test-time updates utilize only labels maturing inside the scored window. Primary text results report Folds 1–2, with an independent out-of-sample period evaluated in Appendix G. Confidence intervals use universe-cluster bootstrap across cells and circular block bootstrap for temporal return differences (Appendix E).

Baselines. All mean-forecast methods share the identical decision layer ($\gamma = 2$, $w_{\max} = 0.35$), chronology, and transaction cost model (Eq. (1)). We compare DU-TTT across four baseline families: (1) *No-adaptation nulls*: the frozen base forecast and best-fixed static validation calibration. (2) *Error-aligned calibration*: affine MSE (online gradient descent on squared error), recursive least squares with forgetting (RLS; Ljung (1999)), online Newton step (Hazan et al., 2007), absolute error tracking median outcomes (Zinkevich, 2003), adaptive conformal quantile tracking (Gibbs & Candès, 2021), and Hedge over calibration maps (Freund & Schapire, 1997). (3) *Decision-aligned objectives*: SPO+ decision regret and its deployed-scale variant (Elmachtoub & Grigas, 2022), alongside the risk-regularized Fenchel–Young loss (Blondel et al., 2020a). (4) *Ranking objectives*: ListMLE (Xia et al., 2008), LambdaRank-IC (Lin et al., 2026), and a soft Spearman loss surrogate (Blondel et al., 2020b). Reference non-predictive baselines include equal-weight ($1/N$) and minimum-variance allocations.

Implementation details. Portfolios rebalance weekly on Fridays. Realized returns are credited prior to updates, incurring proportional trading costs $c \cdot \|w_t - \widetilde{w}_t\|_1$ with $c \in \{0, 10, 25, 50\}$ bps. The aggregate strategy assigns equal capital to all 327 cells, resetting daily with cross-cell trade netting before fees. Hyperparameters are tuned per family via cross-cell validation net Sharpe at 25 bps over a 60-configuration grid. Backbones include Ridge regression, LightGBM, and two frozen 27B Qwen LLM checkpoints prompted with price features (Appendix F).

Evaluation metrics. We report annualized Sharpe ratio, CAGR, Sortino ratio, Calmar ratio, maximum drawdown (MDD), and annualized L_1 turnover across individual cells and for the aggregated portfolio same as portfolio manager performance review standard. Statistical significance of paired excess returns is established via 10,000 circular block bootstrap resamples (Appendix E).

5.1 Main results

Investment performance. As detailed in Table 2, DU-TTT achieves superior risk-adjusted returns and capital efficiency over all test-time adaptation baselines sharing the same frozen forecaster and

Table 2: Net performance of the aggregate strategy at 25 bps. All adaptive methods calibrate the same $2n$-parameter head on the same frozen ridge forecaster. **Bold** and underline mark the best and second-best value per column within each fold.

	Fold 1: 2024-01-02 to 2025-01-31 (272 days)							**Fold 2**: 2025-06-02 to 2026-04-15 (219 days)						
Strategy	CAGR ↑	SR ↑	Sort. ↑	Calm. ↑	MDD ↓	Vol.	Turn. ↓	CAGR ↑	SR ↑	Sort. ↑	Calm. ↑	MDD ↓	Vol.	Turn. ↓
No test-time adaptation														
Frozen ridge (null)	5.00	0.40	0.57	0.56	9.00	15.00	32.64	7.71	0.61	0.88	0.84	9.14	13.70	33.70
Static rescaling	5.54	0.43	0.61	0.62	9.00	15.51	32.07	9.46	0.70	1.02	1.00	9.46	14.36	33.77
Error-aligned calibration														
Affine MSE	9.42	0.71	1.02	0.95	9.90	14.00	32.45	5.27	0.48	0.68	0.53	9.99	12.34	38.21
Affine RLS	13.60	0.94	1.38	1.38	9.84	14.76	27.36	10.27	0.75	1.07	1.07	9.57	14.53	30.40
OGD (absolute error)	10.34	0.77	1.12	1.19	8.68	14.01	27.80	14.80	1.05	1.56	1.69	8.75	14.05	25.80
Adaptive quantile tracking	8.34	0.59	0.85	0.89	9.33	15.50	27.03	8.46	0.79	1.14	1.06	7.96	11.08	25.31
Hedge over calibration maps	5.05	0.40	0.57	0.56	9.00	15.04	32.61	7.91	0.62	0.90	0.86	9.17	13.74	33.75
Decision-aligned objectives														
SPO+	11.04	0.86	1.26	1.24	8.88	13.10	28.19	11.40	0.95	1.37	1.44	7.93	12.12	25.71
SPO+ (unit matched)	11.79	0.90	1.31	1.31	9.03	13.46	27.23	13.59	1.02	1.49	1.57	8.64	13.29	26.24
Fenchel–Young	6.46	0.54	0.76	0.75	8.65	13.26	39.07	6.44	0.56	0.81	0.80	8.10	12.45	37.06
Test-time-training														
TAFAS	6.83	0.58	0.82	0.75	9.05	12.92	40.81	7.42	0.64	0.93	0.81	9.20	12.35	38.01
KKT-embedded DFL	11.04	0.86	1.26	1.29	8.56	13.14	28.08	6.78	0.59	0.86	0.68	9.96	12.36	39.64
IPMO	4.98	0.46	0.65	0.58	8.52	12.00	37.85	5.92	0.53	0.77	0.62	9.61	12.21	41.44
DU-TTT (ours)	**16.24**	**1.21**	**1.79**	**2.06**	**7.88**	13.19	**13.64**	**17.24**	**1.31**	**1.99**	**1.98**	**7.69**	12.76	**10.39**

CAGR, maximum drawdown (MDD) and annualized volatility (Vol.) in %; SR, Sort. and Calm. are the Sharpe, Sortino and Calmar ratios; Turn. is annualized L_1 turnover in NAV multiples. The held-out period is reported in Appendix G.

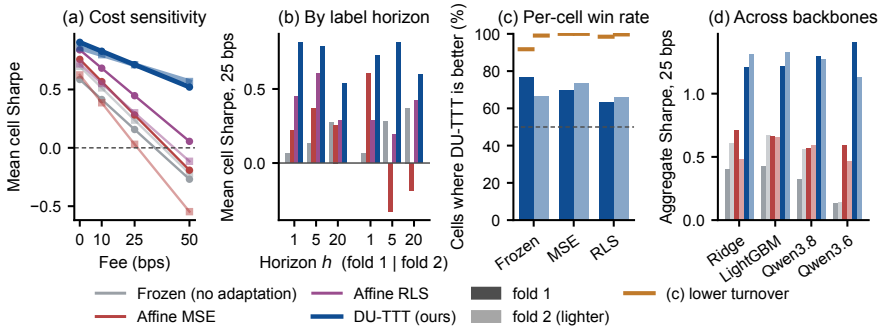


Figure 4: DU-TTT performance sensitivity and backbone transferability (Folds 1 and 2).

execution protocol. ❶ **Return and risk-adjusted growth:** Compared to standard Affine MSE, DU-TTT increases CAGR from 9.42% to 16.24% in Fold 1 and from 5.27% to 17.24% in Fold 2, elevating Sharpe ratios from $0.71 \rightarrow 1.21$ and $0.48 \rightarrow 1.31$, respectively. Calmar ratios similarly double across both test windows. ❷ **Turnover reduction:** By insulating the adaptation layer from forecast level shifts and focusing on the selection margin, DU-TTT cuts annualized L_1 turnover from $32.45\times$ and $38.21\times$ (Affine MSE) down to $13.64\times$ and $10.39\times$ NAV achieving lower turnover than even the unadapted frozen null. ❸ **Dominance across metrics:** DU-TTT consistently delivers the highest growth, Sharpe, Sortino, and Calmar ratios alongside the lowest turnover among all calibrated objectives in both folds.

Consistency across cells, horizons, and fees. The observed gains are consistent across individual sub-portfolios and robust to cost structures (Figure 4). DU-TTT outperforms Affine MSE at every transaction fee level including zero-cost regimes (0.904 vs. 0.758 and 0.856 vs. 0.625 mean cell Sharpe; panel a) and across all label maturation horizons $h \in \{1, 5, 20\}$ (panel b). At the baseline 25 bps cost, DU-TTT secures higher Sharpe ratios than Affine MSE in 70%–73% of cells and lower turnover in 92%–100% of cells (panel c).

Transfer across backbones. To evaluate model-agnostic generality, we examine whether DU-TTT transfers across diverse backbones by adapting only the $2n$-parameter head. As Table 3 and Figure 4(d) show, DU-TTT consistently outperforms across non-parametric trees (LightGBM), linear baselines (Ridge), and frozen LLMs (Qwen-27B). Across all backbones and evaluation folds, DU-TTT delivers top growth, superior Sharpe ratios, and the lowest turnover. The gains are most pronounced on frozen LLMs, where raw forecasts suffer from scale distortion and uncalibrated cross-sectional variance,

resulting in extreme turnover ($54.8\times$–$60.7\times$ NAV). Standard MSE calibration reduces noise but remains sensitive to level shifts. By insulating adaptation from level shifts and focusing on the active selection margin, DU-TTT boosts annual growth from 0.86%–6.05% (raw) and 5.07%–7.06% (MSE) to 13.80%–17.77%, while drastically reducing turnover to $13.0\times$–$14.1\times$ NAV.

Table 3: Transfer across forecasting backbones, net of 25 bps.

		Fold 1			Fold 2		
Backbone	Update	CAGR %↑	Sharpe↑	Turn./yr↓	CAGR %↑	Sharpe↑	Turn./yr↓
Ridge	Frozen	5.00	0.400	32.64	7.71	0.611	33.70
	Static rescaling	5.54	0.425	32.07	9.46	0.701	33.77
	MSE	9.42	0.713	32.45	5.27	0.478	38.21
	DU-TTT	**16.24**	**1.207**	**13.64**	**17.24**	**1.310**	**10.39**
LightGBM	Frozen	5.21	0.428	34.29	8.70	0.669	34.00
	Static rescaling	5.22	0.423	34.17	9.03	0.685	34.03
	MSE	8.26	0.662	34.58	7.68	0.652	38.29
	DU-TTT	**16.09**	**1.215**	**13.53**	**17.30**	**1.325**	**13.81**
Qwen3.8-27B	Frozen	3.21	0.32	57.7	6.05	0.56	54.8
	MSE	6.73	0.57	38.7	6.76	0.59	42.9
	DU-TTT	**16.11**	**1.29**	**13.9**	**16.54**	**1.27**	**14.1**
Qwen3.6-27B	Frozen	0.86	0.13	60.7	0.94	0.14	58.5
	MSE	7.06	0.59	38.7	5.07	0.46	43.7
	DU-TTT	**17.77**	**1.40**	**13.0**	**13.80**	**1.13**	**14.1**

5.2 Ablation study

Table 4 shows that each component ablation to performance. Removing the dispersion-scaled step lowers cell Sharpe from 0.712 to 0.563 in fold 1 and to 0.537 in fold 2. Re-tuning without the step recovers only part of this loss. Removing boundary focusing (α=0) lowers cell Sharpe to 0.627 and 0.648 and raises turnover. The relaxed update, which drops both components, gives the lowest cell Sharpe and the highest turnover among the structural variants. We keep the step fixed and replace the decision-utility direction with a squared-error gradient. This lowers cell Sharpe to 0.515 and 0.637 and aggregate Sharpe to 1.082 and 1.291, while turnover rises by 58% and 61%. The gain therefore comes from the objective, not from the step size. Affine MSE with its own step does much worse. Freezing the gains and updating only the biases reproduces DU-TTT to the third decimal. The adaptation therefore works through per-asset bias correction. When forecasts are permuted across assets, cell Sharpe falls to 0.56–0.64 and turnover rises to 12–20. The updates therefore depend on the asset-specific signal. Shrinking the bias by $1 - \omega$ each step (ω=0.1) lowers cell Sharpe to 0.479 and 0.558 and raises turnover, so retaining the accumulated bias matters.

5.3 Trading intensity and the matched execution control

Where the advantage comes from. Of the net advantage of DU-TTT over affine MSE, $+4.70$ and $+6.95$ are saved trading fees and $+1.24$ and $+3.87$ are gross active return, so trading intensity is the main channel and the positions themselves also improve. When every calibrated family receives the same sixteen predicted-utility trade gates (Appendix E) and is re-selected on validation, DU-TTT still exceeds affine MSE in both folds on CAGR, Sharpe and turnover.

5.4 Sensitivity Analysis

Every focus weight $\alpha > 0$ improves on $\alpha = 0$ in both folds, and the temperature has an interior optimum at $\tau = 8$–16, where the effective support (2.7–3.8 names) matches the three names the cap holds. The value of the margin components grows with the fee: DU-TTT's advantage over the relaxed update rises from $+0.068$/$+0.016$ at zero fee to $+0.310$/$+0.280$ at 50 bps while median cell turnover falls by 42%/47%, so the gain comes from trading less rather than from a better forecast (Figure 3). Table 5 varies one learning parameter at a time around the selected configuration. Scaling the step by 0.5–$2\times$ keeps mean cell Sharpe within 0.615–0.739. Looser $g_{\max}$ and $b_{\max}$ match or improve the selected setting, and tighter values lower it by up to 0.12. A uniform horizon weighting ($\beta = 0$) outperforms $\beta = 1$ in both folds prove DU-TTT is highly robust to hyperparameter choices.

Table 4: Ablation and mechanism control experiments for DU-TTT across Folds 1 and 2.

Variant (Fold 1 / Fold 2)	Cell SR ↑	CAGR % ↑	Sharpe ↑	Turn. ↓
DU-TTT	**0.712 / 0.712**	**16.24 / 17.24**	**1.207 / 1.310**	**13.64 / 10.39**
w/o dispersion-scaled step	0.563 / 0.537	13.87 / 13.88	1.034 / 1.116	17.45 / 16.11
w/o dispersion-scaled step, own selection	0.631 / 0.621	15.44 / 15.93	1.132 / 1.252	14.73 / 14.14
w/o boundary focusing (α=0)	0.627 / 0.648	14.37 / 15.92	1.071 / 1.243	15.96 / 12.47
relaxed update (*w/o* both, own selection)	0.523 / 0.563	13.22 / 14.36	0.991 / 1.172	19.77 / 17.00
Learning target				
squared-error direction, same step (own selection)	0.515 / 0.637	13.64 / 18.51	1.082 / 1.291	21.62 / 16.74
affine MSE (own step, own selection)	0.281 / 0.030	9.42 / 5.27	0.713 / 0.478	32.45 / 38.21
Mechanism controls at the selected settings				
forecast permuted across assets	0.574–0.637 / 0.562–0.618	13.8–15.5 / 13.5–15.0	1.13–1.25 / 1.09–1.18	19.4–19.8 / 12.2–12.4
bias multiplied by $1-\omega$, $\omega = 0.1$	0.479 / 0.558	12.36 / 13.46	0.954 / 1.128	20.85 / 15.17

Table 5: One-at-a-time sensitivity of DU-TTT: mean cell Sharpe, net of 25 bps.

	Base step η ($\times$ selected)					Horizon exp. β		Grad. threshold $g_{\max}$			Bias limit $b_{\max}$		
Fold	0.25	0.5	1	2	4	0	1	0.25	0.5	1	0.05	0.1	0.2
1	0.513	0.615	0.712	0.739	0.727	0.712	0.622	0.593	0.712	0.728	0.712	0.751	0.754
2	0.697	0.715	0.712	0.707	0.633	0.712	0.676	0.663	0.712	0.707	0.624	0.712	0.737

Each column changes one parameter at the fold's selected configuration (shaded; fold 1: $\eta = 0.02$, $b_{\max} = 0.05$; fold 2: $\eta = 0.05$, $b_{\max} = 0.10$; both $\beta = 0$, $g_{\max} = 0.5$, $\alpha = 0.5$, $\tau = 8$).

6 Limitation

Our evaluation is restricted to U.S. equity portfolios. We have not examined trading behaviour across asset classes, such as futures, fixed income or multi-asset allocations, where contract rolls, leverage and market-specific cost structures change the decision layer. While, DU-TTT only adapts a $2n$-parameter head above a frozen forecaster and leaves the allocator unchanged, extending it to multi-assets allocation settings requires no change to the method.

7 Conclusion

This paper investigates how test-time adaptation should be designed for return forecasting under delayed label arrival, portfolio constraints, and transaction costs. We propose DU-TTT, a framework that logs each decision context, updates a compact affine head using a covariance-weighted utility direction as labels mature, and leverages soft top-K membership to concentrate gradients onto the selection boundary and scale the step size accordingly. By unifying delayed affine calibration with utility-driven and margin-focused feedback, DU-TTT consistently improves net annual growth and risk-adjusted Sharpe ratios over standard squared-error calibration while drastically curbing turnover across both evaluation folds. Decision utility serves as the superior learning target for test-time adaptation in constrained, cost-sensitive practical portfolio management.

Reproducibility statement

Section 5 and Appendix E specify the data construction, label maturity, trading chronology, selection rules and metric conventions; Appendices A and B give the decision geometry and proofs. The experiment archive retains configuration and source hashes, selected policies and dated daily paths from which every table is recomputed. The source code in `https://anonymous.4open.science/status/DUTTT-3369`

AI USE STATEMENT

In this work, we used generative AI tools to implement methods and to provide feedback on research methodology and experiments. We have not used generative AI tools to generate synthetic data sets, develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, or interpret results. Translation, dataset cleaning and reformatting, and qualitative or thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to create and edit software code, such as experiment and evaluation scripts, and to edit the text of the paper for readability. We have reviewed all AI-assisted work. All AI-assisted code was read, executed, and tested by the authors, and all reported results were checked against the saved experiment logs. Feedback from AI tools on methodology and experiments was treated as suggestions; the research questions, method design, and conclusions were determined by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of AI.

ETHICS STATEMENT

This work uses only public or reconstructible price data and no personal data. The results are exploratory historical evaluations and are not investment advice.

REFERENCES

Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J. Zico Kolter. Differentiable convex optimization layers. In *Advances in Neural Information Processing Systems*, volume 32, pp. 9558–9570, 2019.

Benjamin Amoh, Geoffrey G. Parker, and Wesley Marrero. IGT-OMD: Implicit gradient transport for decision-focused learning under delayed feedback. *arXiv preprint arXiv:2605.12693*, 2026. URL `https://arxiv.org/abs/2605.12693`.

Brandon Amos, Vladlen Koltun, and J. Zico Kolter. The limited multi-label projection layer. *arXiv preprint arXiv:1906.08707*, 2019.

Hassan T. Anis and Roy H. Kwon. End-to-end, decision-based, cardinality-constrained portfolio optimization. *European Journal of Operational Research*, 320:739–753, 2025. doi: 10.1016/j.ejor.2024.08.030.

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024.

Quentin Berthet, Mathieu Blondel, Olivier Teboul, Marco Cuturi, Jean-Philippe Vert, and Francis Bach. Learning with differentiable perturbed optimizers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Mathieu Blondel, André F. T. Martins, and Vlad Niculae. Learning with Fenchel-Young losses. *Journal of Machine Learning Research*, 21(35):1–69, 2020a.

Mathieu Blondel, Olivier Teboul, Quentin Berthet, and Josip Djolonga. Fast differentiable sorting and ranking. In *International Conference on Machine Learning*, pp. 950–959. PMLR, 2020b.

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.

Michael W. Brandt, Pedro Santa-Clara, and Rossen Valkanov. Parametric portfolio policies: Exploiting characteristics in the cross-section of equity returns. *The Review of Financial Studies*, 22(9): 3411–3447, 2009. doi: 10.1093/rfs/hhp003.

Andrew Butler and Roy H. Kwon. Integrating prediction in mean-variance portfolio optimization. *Quantitative Finance*, 23(3):429–452, 2023. arXiv:2102.09287.

Aymeric Capitaine, Maxime Haddouche, Eric Moulines, Michael I. Jordan, Etienne Boursier, and Alain Durmus. Online decision-focused learning. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=FJhtHBphCt.

Mouxiang Chen, Lefei Shen, Han Fu, Zhuo Li, Jianling Sun, and Chenghao Liu. Calibration of time-series forecasting: Detecting and adapting context-driven distribution shift. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 341–352, 2024. doi: 10.1145/3637528.3671926.

Giorgio Costa and Garud N. Iyengar. Distributionally robust end-to-end portfolio construction. *Quantitative Finance*, 23:1465–1482, 2023. doi: 10.1080/14697688.2023.2236148.

Marco Cuturi, Olivier Teboul, and Jean-Philippe Vert. Differentiable ranks and sorting using optimal transport. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Xilin Dai, Yiding Liu, Hongjie Xia, Yifan Hu, Zewei Dong, Jiang-Ming Yang, and Qiang Xu. Learning the context of errors: Black-box online adaptation of time series foundation models. *arXiv preprint arXiv:2606.14222*, 2026.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 2024.

Puja Das, Nicholas Johnson, and Arindam Banerjee. Online lazy updates for portfolio selection with transaction costs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 27, pp. 202–208, 2013. doi: 10.1609/aaai.v27i1.8693.

M. H. A. Davis and A. R. Norman. Portfolio selection with transaction costs. *Mathematics of Operations Research*, 15(4):676–713, 1990. doi: 10.1287/moor.15.4.676.

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.

Victor DeMiguel, Alberto Martín-Utrera, Francisco J. Nogales, and Raman Uppal. A transaction-cost perspective on the multitude of firm characteristics. *The Review of Financial Studies*, 33(5): 2180–2222, 2020. doi: 10.1093/rfs/hhz085.

Priya Donti, Brandon Amos, and J. Zico Kolter. Task-based end-to-end model learning in stochastic optimization. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5484–5494, 2017.

Adam N Elmachtoub and Paul Grigas. Smart "predict, then optimize". *Management Science*, 68(1): 9–26, 2022.

Fuli Feng, Xiangnan He, Xiang Wang, Cheng Luo, Yiqun Liu, and Tat-Seng Chua. Temporal relational ranking for stock prediction. *ACM Transactions on Information Systems*, 37(2):27:1–27:30, 2019.

Genevieve E. Flaspohler, Francesco Orabona, Judah Cohen, Soukayna Mouatadid, Miruna Oprescu, Paulo Orenstein, and Lester Mackey. Online learning with optimism and delay. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 3363–3373, 2021.

Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

Yossi Gandelsman, Yu Sun, Xinlei Chen, and Alexei A. Efros. Test-time training with masked autoencoders. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Nicolae Gârleanu and Lasse Heje Pedersen. Dynamic trading with predictable returns and transaction costs. *The Journal of Finance*, 68(6):2309–2340, 2013. doi: 10.1111/jofi.12080.

Isaac Gibbs and Emmanuel Candès. Adaptive conformal inference under distribution shift. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 2021.

Aditya Grover, Eric Wang, Aaron Zweig, and Stefano Ermon. Stochastic optimization of sorting networks via continuous relaxations. In *International Conference on Learning Representations*, 2019.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. Large language models are zero-shot time series forecasters. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2310.07820.

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2–3):169–192, 2007.

Elad Hazan, Kfir Y. Levy, and Shai Shalev-Shwartz. Beyond convexity: Stochastic quasi-convex optimization. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Michael Huang and Vishal Gupta. Decision-focused learning with directional gradients. In *Advances in Neural Information Processing Systems*, volume 37, 2024. arXiv:2402.03256.

Yoontae Hwang, Yaxuan Kong, Stefan Zohren, and Yongjae Lee. Decision-informed neural networks with large language model integration for portfolio optimization. *arXiv preprint arXiv:2502.00828*, 2025.

Haeun Jeon, Seunghoon Choi, Hyunglip Bae, Yongjae Lee, and Woo Chang Kim. Decision-focused sparse tangent portfolio optimization. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, 2026. arXiv:2607.00581.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-LLM: Time series forecasting by reprogramming large language models. In *International Conference on Learning Representations*, 2024.

Pooria Joulani, András György, and Csaba Szepesvári. Online learning under delayed feedback. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 1453–1461, 2013. URL https://proceedings.mlr.press/v28/joulani13.html.

Pooria Joulani, András György, and Csaba Szepesvári. Delay-tolerant online convex optimization: Unified analysis and adaptive-gradient algorithms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016. doi: 10.1609/aaai.v30i1.10320.

HyunGi Kim, Siwon Kim, Jisoo Mok, and Sungroh Yoon. Battling the non-stationarity in time series forecasting via test-time adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 17868–17876, 2025.

Ying-yee Ava Lau, Zhiwen Shao, and Dit-Yan Yeung. Fast and slow streams for online time series forecasting without information leakage. In *International Conference on Learning Representations*, 2025.

Junhyeong Lee, Haeun Jeon, Hyunglip Bae, and Yongjae Lee. Return prediction for mean-variance portfolio selection: How decision-focused learning shapes forecasting models. In *Proceedings of the 6th ACM International Conference on AI in Finance (ICAIF)*, pp. 114–122. ACM, 2025a. doi: 10.1145/3768292.3770423. arXiv:2409.09684.

Thomas L. Lee, William Toner, Rajkarn Singh, Artjom Joosen, and Martin Asenov. Lightweight online adaption for time series foundation model forecasts. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 33736–33764, 2025b.

Bin Li and Steven C. H. Hoi. Online portfolio selection: A survey. *ACM Computing Surveys*, 46(3): 35:1–35:36, 2014.

Bin Li, Jialei Wang, Dingjiang Huang, and Steven C. H. Hoi. Transaction cost optimization for online portfolio selection. *Quantitative Finance*, 18(8):1411–1424, 2018. doi: 10.1080/14697688.2017.1357831.

Yan Lin, Yihong Su, and Yi Yang. LambdaRankIC: Directly optimizing rank IC for financial prediction. *arXiv preprint arXiv:2605.00501*, 2026.

Yuxuan Linghu, Zhiyuan Liu, and Qi Deng. Integrated prediction and multi-period portfolio optimization. *arXiv preprint arXiv:2512.11273*, 2025. URL https://arxiv.org/abs/2512.11273.

Lennart Ljung. *System Identification: Theory for the User*. Prentice Hall, Upper Saddle River, NJ, 2nd edition, 1999.

Alejandro Lopez-Lira and Yuehua Tang. Can ChatGPT forecast stock price movements? return predictability and large language models. *arXiv preprint arXiv:2304.07619*, 2023.

Jayanta Mandi, Víctor Bucarey, Maxime Mulamba, and Tias Guns. Decision-focused learning: Through the lens of learning to rank. In *International Conference on Machine Learning*, 2022.

Jayanta Mandi, James Kotary, Senne Berden, Maxime Mulamba, Víctor Bucarey, Tias Guns, and Ferdinando Fioretto. Decision-focused learning: Foundations, state of the art, benchmark and future opportunities. *Journal of Artificial Intelligence Research*, 80:1623–1701, 2024. doi: 10.1613/jair.1.15320.

H. Brendan McMahan and Matthew Streeter. Delay-tolerant algorithms for asynchronous distributed online learning. In *Advances in Neural Information Processing Systems*, volume 27, 2014.

Heitor R. Medeiros, Hossein Sharifi-Noghabi, Gabriel L. Oliveira, and Saghar Irandoust. Accurate parameter-efficient test-time adaptation for time series forecasting. *arXiv preprint arXiv:2506.23424*, 2025. Second Workshop on Test-Time Adaptation at ICML 2025.

Maxime Mulamba, Jayanta Mandi, Michelangelo Diligenti, Michele Lombardi, Victor Bucarey, and Tias Guns. Contrastive losses and solution caching for predict-and-optimize. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 2021.

Miquel Noguer i Alonso and Rodolfo Pereira Franklin. Pretrained time-series foundation models for financial return forecasting. *arXiv preprint arXiv:2606.27100*, 2026.

Kensei Nosaka, Shunnosuke Ikeda, and Yuichi Takano. Decision-focused learning for mean–variance portfolio optimization via KKT-based reformulation. *arXiv preprint arXiv:2609.21427*, 2026. URL https://arxiv.org/abs/2609.21427. Accepted at PRICAI 2026.

Francesco Orabona and Dávid Pál. Scale-free online learning. *Theoretical Computer Science*, 716:50–69, 2018. doi: 10.1016/j.tcs.2017.11.021.

Quang Pham, Chenghao Liu, Doyen Sahoo, and Steven C. H. Hoi. Learning fast and slow for online time series forecasting. In *International Conference on Learning Representations*, 2023.

Daniel Poh, Bryan Lim, Stefan Zohren, and Stephen Roberts. Building cross-sectional systematic strategies by learning to rank. *The Journal of Financial Data Science*, 3:70–86, 2021. doi: 10.3905/jfds.2021.1.060.

Hao Qiu, Andrew Jacobsen, Emmanuel Esposito, and Mengxiao Zhang. Parameter-free dynamic regret: Time-varying movement costs, delayed feedback, and memory. *arXiv preprint arXiv:2602.06902*, 2026.

Kent Quanrud and Daniel Khashabi. Online learning with adversarial delays. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Utsav Sadana, Abhilash Chenreddy, Erick Delage, Alexandre Forel, Emma Frejinger, and Thibaut Vidal. A survey of contextual optimization methods for decision making under uncertainty. *European Journal of Operational Research*, 320(2):271–289, 2025.

Subham Sekhar Sahoo, Anselm Paulus, Marin Vlastelica, Vít Musil, Volodymyr Kuleshov, and Georg Martius. Backpropagation through combinatorial algorithms: Identity with projection works. In *International Conference on Learning Representations*, 2023.

Michael E. Sander, Joan Puigcerver, Josip Djolonga, Gabriel Peyré, and Mathieu Blondel. Fast, differentiable and sparse top-k: a convex analysis perspective. In *International Conference on Machine Learning*, 2023.

Noah Schutte, Krzysztof Postek, and Neil Yorke-Smith. Robust losses for decision-focused learning. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 4868–4875, 2024.

Sanket Shah, Kai Wang, Bryan Wilder, Andrew Perrault, and Milind Tambe. Decision-focused learning without differentiable optimization: Learning locally optimized decision losses. *arXiv preprint arXiv:2203.16067*, 2022.

Qiang Song, Anqi Liu, and Steve Y. Yang. Stock portfolio selection using learning-to-rank algorithms with news sentiment. *Neurocomputing*, 264:20–28, 2017. doi: 10.1016/j.neucom.2017.02.097.

Yongyi Su, Xun Xu, and Kui Jia. Revisiting realistic test-time training: Sequential inference and adaptation by anchored clustering. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9229–9248, 2020.

Bo Tang and Elias B. Khalil. PyEPO: A PyTorch-based end-to-end predict-then-optimize library for linear and integer programming. *Mathematical Programming Computation*, 16:297–335, 2024.

Ayse Sinem Uysal, Xiaoyue Li, and John M. Mulvey. End-to-end risk budgeting portfolio optimization with neural networks. *arXiv preprint arXiv:2107.04636*, 2021.

Marin Vlastelica, Anselm Paulus, Vít Musil, Georg Martius, and Michal Rolínek. Differentiation of blackbox combinatorial solvers. In *International Conference on Learning Representations*, 2020.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021.

Haochun Wang, Ruichen Xu, Georgios Kementzidis, Karen Cho, Sebastian Ramirez Villarreal, and Yuefan Deng. Towards principled test-time adaptation for time series forecasting. *arXiv preprint arXiv:2605.17250*, 2026.

Xibai Wang. Adapt only when it pays: Budgeted decision-loss priority for delayed online time-series adaptation. *arXiv preprint arXiv:2606.25068*, 2026.

Yi Wang and Takashi Hasuike. Decision-induced ranking explains prediction inflation and excessive turnover in SPO-based portfolio optimization. *arXiv preprint arXiv:2605.01176*, 2026.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53140–53164, 2024.

Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. Listwise approach to learning to rank: theory and algorithm. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 1192–1199, 2008.

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, 2020.

Wentao Zhang, Lingxuan Zhao, Haochong Xia, Shuo Sun, Jiaze Sun, Molei Qin, Xinyi Li, Yuqing Zhao, Yilei Zhao, Xinyu Cai, Longtao Zheng, Xinrun Wang, and Bo An. A multimodal foundation agent for financial trading: Tool-augmented, diversified, and generalist. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4314–4325, 2024. doi: 10.1145/3637528.3671801.

Xin Zhang, Lan Wu, and Zhixue Chen. Constructing long-short stock portfolio with a new listwise learn-to-rank algorithm. *Quantitative Finance*, 22:321–331, 2022. doi: 10.1080/14697688.2021.1939117.

Yi-Fan Zhang, Qingsong Wen, Xue Wang, Weiqi Chen, Liang Sun, Zhang Zhang, Liang Wang, Rong Jin, and Tieniu Tan. OneNet: Enhancing time series forecasting models under concept drift by online ensembling. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 69949–69980, 2023.

Zihao Zhang, Stefan Zohren, and Stephen Roberts. Deep learning for portfolio optimization. *arXiv preprint arXiv:2005.13665*, 2020.

Lifan Zhao and Yanyan Shen. Proactive model adaptation against concept drift for online time series forecasting. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2020–2031, 2025. doi: 10.1145/3690624.3709210.

Lifan Zhao, Shuming Kong, and Yanyan Shen. DoubleAdapt: A meta-learning approach to incremental learning for stock trend forecasting. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3492–3503, 2023. doi: 10.1145/3580305.3599315.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *International Conference on Machine Learning (ICML)*, pp. 928–936, 2003.

APPENDIX CONTENTS

A Decision geometry **17**

A.1 Forecast level and scale 17

A.2 SPO+ with a quadratic risk penalty 17

A.3 Exact and equality-relaxed Jacobians 17

A.4 Decision regret as projected prediction error 18

B Decision alignment and action stability **18**

B.1 The target of the relaxed direction 18

B.2 Population targets under noisy feedback 20

B.3 Constrained regret and allocation stability 20

C Extended Related Work **22**

C.1 Test-Time Adaptation and Label Timing Protocols 22

C.2 Decision-Focused Learning and Surrogates 22

C.3 Learning to Rank for Portfolio Selection 22

C.4 Online Learning with Delays and Financial Friction 23

C.5 Pretrained Time-Series and Financial Backbones 23

D Margin components: detailed analysis **23**

E Evaluation protocol details **25**

F Language-model protocol and deployment **26**

F.1 Checkpoint and input provenance 26

F.2 Hardware and software 27

G Held-out confirmation **28**

A DECISION GEOMETRY

Setting and units. Let s denote a decision date, h the target horizon in trading days, and n the number of assets. The return vector $d_s \in \mathbb{R}^n$ covers dates s through $s+h$ and becomes observable at the close of $s+h$. The forecast $v \in \mathbb{R}^n$ has the same units as d_s. Write $k_h = 252/h$ and let $\Sigma_s \succ 0$ be the annualized covariance available at date s. The results below hold for a fixed positive-definite covariance. For risk aversion $\gamma > 0$ and a return vector $d \in \mathbb{R}^n$, define the utility and risk penalty by

$$\begin{aligned} U_{s,h}(w; d) &= d^\top w - \Omega_{s,h}(w), \qquad \Omega_{s,h}(w) = \frac{\gamma}{2k_h} w^\top \Sigma_s w, \\ \mathcal{A}_{s,h}(v; \gamma) &= \arg\max_{w \in \mathcal{W}} U_{s,h}(w; v), \end{aligned} \tag{11}$$

where $\mathcal{W} = \{w \in \mathbb{R}^n : 0 \le w \le w_{\max}\mathbf{1}, \mathbf{1}^\top w = 1\}$, $\mathbf{1}$ is the all-ones vector, and $0 < w_{\max} \le 1$ with $n w_{\max} \ge 1$. The inequalities are componentwise. The feasible set is nonempty and compact, and strict concavity makes the allocation unique. Multiplying the objective in Eq. (11) by k_h gives the annual-unit allocator with inputs $k_h v$ and Σ_s. In implementation the allocator and the update use $\Sigma_s + 10^{-10} I_n$ (Appendix E). These results concern decision utility at a fixed date. Net returns also include the effects of holding drift, subsequent trades, and transaction costs.

A.1 FORECAST LEVEL AND SCALE

For any common shift $\zeta \in \mathbb{R}$ and positive scale $\kappa > 0$,

$$\mathcal{A}_{s,h}(v + \zeta\mathbf{1}; \gamma) = \mathcal{A}_{s,h}(v; \gamma), \qquad \mathcal{A}_{s,h}(\kappa v; \gamma) = \mathcal{A}_{s,h}(v; \gamma/\kappa).$$

The first identity holds because the added objective term is $\zeta \mathbf{1}^\top w = \zeta$ on every feasible action. The second follows by dividing the objective by κ. A common shift therefore has no effect, whereas a positive scale changes the effective risk aversion. The latter can change the active set while preserving every forecast rank. For a positive-semidefinite covariance, both identities hold for argmax sets; a single-valued map additionally requires a consistent tie rule.

A.2 SPO+ WITH A QUADRATIC RISK PENALTY

To state the bound, let $S \subset \mathbb{R}^n$ be any nonempty compact convex set and let $\Omega : S \to \mathbb{R}$ be continuous and convex. For a true cost $\mathbf{c} \in \mathbb{R}^n$ and predicted cost $\hat{\mathbf{c}} \in \mathbb{R}^n$, choose $z(\mathbf{c}) \in \arg\min_{z \in S}\{\mathbf{c}^\top z + \Omega(z)\}$, with a fixed selector if necessary. Let $V(\mathbf{c}) = \mathbf{c}^\top z(\mathbf{c}) + \Omega(z(\mathbf{c}))$ be the optimal value. The decision regret is $\mathcal{R}(\hat{\mathbf{c}}, \mathbf{c}) = \mathbf{c}^\top z(\hat{\mathbf{c}}) + \Omega(z(\hat{\mathbf{c}})) - V(\mathbf{c})$. The regularized SPO+ surrogate

$$L_+(\hat{\mathbf{c}}, \mathbf{c}) = \max_{z \in S}\{(\mathbf{c} - 2\hat{\mathbf{c}})^\top z - \Omega(z)\} + 2\hat{\mathbf{c}}^\top z(\mathbf{c}) + 2\Omega(z(\mathbf{c})) - V(\mathbf{c})$$

is convex in $\hat{\mathbf{c}}$ and upper-bounds $\mathcal{R}$. Convexity follows from the maximum of affine functions. For the bound, evaluate that maximum at $z(\hat{\mathbf{c}})$ to obtain

$$L_+(\hat{\mathbf{c}}, \mathbf{c}) - \mathcal{R}(\hat{\mathbf{c}}, \mathbf{c}) \ge 2\big[\hat{\mathbf{c}}^\top (z(\mathbf{c}) - z(\hat{\mathbf{c}})) + \Omega(z(\mathbf{c})) - \Omega(z(\hat{\mathbf{c}}))\big] \ge 0,$$

where the last inequality follows from optimality at $\hat{\mathbf{c}}$. Danskin's theorem gives a subgradient $2[z(\mathbf{c}) - z(2\hat{\mathbf{c}} - \mathbf{c})]$ with respect to $\hat{\mathbf{c}}$. Taking $S = \mathcal{W}$, $\Omega = \Omega_{s,h}$, $\mathbf{c} = -d_s$, and $\hat{\mathbf{c}} = -v$ converts this to the h-period return-coordinate subgradient

$$2\big[\mathcal{A}_{s,h}(2v - d_s; \gamma) - \mathcal{A}_{s,h}(d_s; \gamma)\big].$$

The direction vanishes when the reflected forecast $2v - d_s$ and d_s induce the same action. Matching the actions at v and d_s alone does not suffice. All three actions in the bound use the same decision-date covariance.

A.3 EXACT AND EQUALITY-RELAXED JACOBIANS

Let $J_{s,h}(v) = \partial \mathcal{A}_{s,h}(v; \gamma)/\partial v$ wherever the derivative exists. On a neighborhood with a fixed active set, let $\mathcal{I} = \{i : 0 < [\mathcal{A}_{s,h}(v; \gamma)]_i < w_{\max}\}$ denote the free indices. For nonempty $\mathcal{I}$, let $B = \Sigma^{-1}_{s,\mathcal{I}\mathcal{I}}$ be the inverse covariance block. The Jacobian is zero outside its free block, and

$$[J_{s,h}(v)]_{\mathcal{I}\mathcal{I}} = \frac{k_h}{\gamma}\left(B - \frac{B\mathbf{1}\mathbf{1}^\top B}{\mathbf{1}^\top B \mathbf{1}}\right).$$

Each $\mathbf{1}$ in this formula has length $|\mathcal{I}|$; the Jacobian is zero if $\mathcal{I}$ is empty. The formula follows by differentiating the budget-constrained first-order conditions on the free coordinates. It is symmetric and positive semidefinite. At an active-set boundary, the ordinary derivative need not exist. For $w = \mathcal{A}_{s,h}(v;\gamma)$, the local negative-utility gradient is

$$\nabla_v[-U_{s,h}(w;d_s)] = -J_{s,h}(v)^\top \left(d_s - \frac{\gamma}{k_h}\Sigma_s w\right).$$

Treating all n coordinates as free gives $J^{\mathrm{eq}}_{s,h}$ in Eq. (4). DU-TTT applies this equality-only Jacobian to the capped action, then normalizes and clips the direction. The exact local derivative above and this relaxed update therefore describe different learning rules.

A.4 Decision regret as projected prediction error

Let $\mathcal{E} = \{w \in \mathbb{R}^n : \mathbf{1}^\top w = 1\}$ denote the equality-only feasible set, which permits short positions, and define

$$Q_s = \Sigma_s^{-1} - \frac{\Sigma_s^{-1}\mathbf{1}\mathbf{1}^\top\Sigma_s^{-1}}{\mathbf{1}^\top\Sigma_s^{-1}\mathbf{1}}, \qquad w^{\mathrm{eq}}_{s,h}(v) = \frac{\Sigma_s^{-1}\mathbf{1}}{\mathbf{1}^\top\Sigma_s^{-1}\mathbf{1}} + \frac{k_h}{\gamma}Q_s v.$$

The matrix Q_s is positive semidefinite and satisfies $Q_s\mathbf{1} = 0$ and $Q_s\Sigma_s Q_s = Q_s$. The displayed action solves Eq. (11) with $\mathcal{W}$ replaced by $\mathcal{E}$. Both actions obey the same budget, so their difference lies in a direction where the first-order term at the optimum vanishes. The remaining quadratic is

$$\begin{aligned} &U_{s,h}(w^{\mathrm{eq}}_{s,h}(d_s);d_s) - U_{s,h}(w^{\mathrm{eq}}_{s,h}(v);d_s) \\ &\quad= \frac{\gamma}{2k_h}[w^{\mathrm{eq}}_{s,h}(v) - w^{\mathrm{eq}}_{s,h}(d_s)]^\top\Sigma_s[w^{\mathrm{eq}}_{s,h}(v) - w^{\mathrm{eq}}_{s,h}(d_s)] = \frac{k_h}{2\gamma}(v-d_s)^\top Q_s(v-d_s). \end{aligned} \tag{12}$$

Covariance-projected error therefore equals equality-constrained decision regret. Raw squared error also penalizes the common-level direction, which cannot change the allocation.

The identity is exact when the same covariance defines both the projected error and the equality-only allocation. Regularizing and normalizing the learning metric, scaling forecasts, and deploying capped, cost-gated portfolios produce the implemented surrogate. The resulting investment performance is evaluated on the full trading path.

B Decision alignment and action stability

This section collects consequences of equality-constrained quadratic optimization (Butler & Kwon, 2023) and the convex-duality view of regularized prediction (Blondel et al., 2020a), expressed for the update in Section 4. The analysis identifies the action targets of different learning rules and relates forecast changes to constrained decisions. These are fixed-context statements, rather than a regret guarantee for the complete online trading algorithm.

Setting. Fix a decision date s, horizon h, risk aversion $\gamma > 0$ and positive-definite covariance Σ_s. Use k_h, $U_{s,h}$, $\mathcal{A}_{s,h}$ and $\mathcal{W}$ from Eq. (1), and Q_s from Eq. (4). Every expression within a statement uses the same covariance. Let $\psi_s > 0$ be the floored normalization denominator of Eq. (5). It is fixed in the context and does not depend on the forecast or matured label. The equality-only allocation $w^{\mathrm{eq}}_{s,h}$ is that of Appendix A.4. Define the optimal-value function $F_{s,h} : \mathbb{R}^n \to \mathbb{R}$ and the relaxed direction at arbitrary vectors $v, d \in \mathbb{R}^n$ by

$$F_{s,h}(v) = \max_{w\in\mathcal{W}} U_{s,h}(w;v),$$
$$g^{\mathrm{rel}}_{s,h}(v,d) = -\frac{(k_h/\gamma)Q_s[d - (\gamma/k_h)\Sigma_s\mathcal{A}_{s,h}(v;\gamma)]}{\psi_s}.$$

The implemented pre-clipping direction is $g_s = g^{\mathrm{rel}}_{s,h}(\hat{\mu}_s, d_s)$.

B.1 The target of the relaxed direction

Proposition 1 (Action-target representation)**.** *For all $v, d \in \mathbb{R}^n$,*

$$g^{\mathrm{rel}}_{s,h}(v,d) = \frac{\mathcal{A}_{s,h}(v;\gamma) - w^{\mathrm{eq}}_{s,h}(d)}{\psi_s} = \nabla_v\Psi_{s,h}(v,d), \tag{13}$$

where the differentiable convex potential is

$$\Psi_{s,h}(v,d) = \frac{F_{s,h}(v) - v^\top w^{\mathrm{eq}}_{s,h}(d)}{\psi_s}.$$

Proof. The definition of Q_s gives

$$Q_s\Sigma_s = I_n - \frac{\Sigma_s^{-1}\mathbf{1}\mathbf{1}^\top}{\mathbf{1}^\top\Sigma_s^{-1}\mathbf{1}}.$$

For every $w \in \mathcal{W}$, the budget constraint therefore implies $Q_s\Sigma_s w = w - w^{\mathrm{eq}}_{s,h}(0)$. Substitution in $g^{\mathrm{rel}}_{s,h}$ yields the first equality in Eq. (13). The function $F_{s,h}$ is convex because it is a maximum of affine functions of v. The strict concavity of $U_{s,h}(\cdot; v)$ and the nonempty compact feasible set make the maximizer unique. Danskin's theorem then gives $\nabla F_{s,h}(v) = \mathcal{A}_{s,h}(v;\gamma)$. Subtracting a linear function and dividing by $\psi_s > 0$ proves the remaining claims. □

Equation (13) concerns the capped forward allocation combined with an equality-only Jacobian. It does not identify that direction with the derivative of the capped decision regret derived in Appendix A.3. In the equality relaxation studied by Butler & Kwon (2023, Section 3.3), the predictor is fitted after dropping the inequalities and then evaluated with constrained allocations. Here the capped action remains in the update's forward computation, leading to the representation above.

Relation to Fenchel–Young losses. For a feasible action $y \in \mathcal{W}$, define

$$L^{\mathrm{FY}}_{s,h}(v; y) = F_{s,h}(v) + \frac{\gamma}{2k_h} y^\top \Sigma_s y - v^\top y.$$

This is the Fenchel–Young loss generated by the quadratic risk penalty restricted to $\mathcal{W}$, and its gradient is $\mathcal{A}_{s,h}(v;\gamma) - y$ (Blondel et al., 2020a, Proposition 2). If $w^{\mathrm{eq}}_{s,h}(d)$ is feasible, the potential in Proposition 1 differs from $L^{\mathrm{FY}}_{s,h}(v; w^{\mathrm{eq}}_{s,h}(d))/\psi_s$ by a constant in v. If that action is infeasible, it is outside the domain of the constraint-restricted regularizer, so the finite potential $\Psi_{s,h}$ is not a finite Fenchel–Young loss with that target.

Proposition 2 (Target feasibility)**.** *If $w^{\mathrm{eq}}_{s,h}(d) \in \mathcal{W}$, the minimizers of $\Psi_{s,h}(\cdot, d)$ over $\mathbb{R}^n$ are exactly the forecasts whose capped action equals $w^{\mathrm{eq}}_{s,h}(d)$. Such minimizers exist. If $w^{\mathrm{eq}}_{s,h}(d) \notin \mathcal{W}$, then* $\inf_{v\in\mathbb{R}^n} \Psi_{s,h}(v,d) = -\infty$.

Proof. When the target is feasible, the equality-only optimizer is also the capped optimizer at $v = d$. Proposition 1 then gives a zero gradient at d. For a differentiable convex function, zero gradient is necessary and sufficient for global optimality, proving the first claim. For an infeasible target, compact convex separation supplies a vector $q \in \mathbb{R}^n$ and a scalar $\epsilon > 0$ such that

$$q^\top w^{\mathrm{eq}}_{s,h}(d) \geq \max_{w\in\mathcal{W}} q^\top w + \epsilon.$$

For $t > 0$, let $B_s = \max_{w\in\mathcal{W}} |\gamma w^\top \Sigma_s w/(2k_h)| < \infty$. Then

$$\psi_s\Psi_{s,h}(tq, d) \leq t \max_{w\in\mathcal{W}} q^\top w + B_s - tq^\top w^{\mathrm{eq}}_{s,h}(d) \leq B_s - t\epsilon,$$

which tends to $-\infty$. □

The parameter bounds in the implemented head restrict its attainable forecasts, so Proposition 2 does not assert divergence of that learner. It identifies why a finite parameter domain can affect the solution for infeasible targets. Nor does frequent sample-target infeasibility establish that the mean target is infeasible, as the next result makes explicit.

B.2 Population targets under noisy feedback

Let $\mathcal{H}_s$ contain the information available when the forecast is issued, including the saved covariance. Assume $\mathbb{E}[\|d_s\|_2 \mid \mathcal{H}_s] < \infty$, and define $\bar{d}_s = \mathbb{E}[d_s \mid \mathcal{H}_s]$. The following statements condition on this fixed context and consider optimization over the forecast vector $v \in \mathbb{R}^n$.

Proposition 3 (Conditional action targets)**.** *The relaxed direction has conditional expectation*

$$\mathbb{E}[g^{\mathrm{rel}}_{s,h}(v, d_s) \mid \mathcal{H}_s] = \frac{\mathcal{A}_{s,h}(v;\gamma) - w^{\mathrm{eq}}_{s,h}(\bar{d}_s)}{\psi_s}. \tag{14}$$

If $w^{\mathrm{eq}}_{s,h}(\bar{d}_s) \in \mathcal{W}$, minimizing the conditional expected potential produces the unique action maximizing $\mathbb{E}[U_{s,h}(w; d_s) \mid \mathcal{H}_s]$ over $\mathcal{W}$. By comparison, minimizing the conditional expected Fenchel–Young loss with target $\mathcal{A}_{s,h}(d_s;\gamma)$ produces the action

$$y^{\mathrm{FY}}_s = \mathbb{E}[\mathcal{A}_{s,h}(d_s;\gamma) \mid \mathcal{H}_s] \in \mathcal{W}, \tag{15}$$

which need not equal $\mathcal{A}_{s,h}(\bar{d}_s;\gamma)$.

Proof. The equality-only map is affine, so $\mathbb{E}[w^{\mathrm{eq}}_{s,h}(d_s) \mid \mathcal{H}_s] = w^{\mathrm{eq}}_{s,h}(\bar{d}_s)$. Proposition 1 yields Eq. (14), and the expected potential equals $\Psi_{s,h}(v, \bar{d}_s)$. Its feasible target is the capped optimizer at $\bar{d}_s$ by Proposition 2. Expected utility equals $U_{s,h}(w; \bar{d}_s)$ because the risk term is fixed, proving the first optimal-action assertion.

For the feasible-target loss, the action-valued target and its quadratic penalty are bounded on $\mathcal{W}$. Its conditional expected loss is therefore $F_{s,h}(v) - v^\top y^{\mathrm{FY}}_s$ plus a constant. The closed convex set $\mathcal{W}$ contains y^{FY}_s. This target is attainable: for $v = (\gamma/k_h)\Sigma_s y^{\mathrm{FY}}_s$, maximizing $U_{s,h}(w; v)$ is equivalent to minimizing $(w - y^{\mathrm{FY}}_s)^\top \Sigma_s (w - y^{\mathrm{FY}}_s)$ over $\mathcal{W}$. The minimizer is y^{FY}_s, which gives zero gradient of the conditional expected loss. Convexity then proves Eq. (15). The capped map is nonlinear, so in general its application and conditional expectation do not commute. □

This is a specialization of the expected-target identity for Fenchel–Young losses (Blondel et al., 2020a, Section 3.3) to portfolio action targets. The distinction is present even when every target of the feasible-action loss satisfies the portfolio constraints. For an explicit example, take $n = 2$, $\Sigma_s = I_2$, $\gamma/k_h = 1$ and $w_{\max} = 1$. Let the second return be zero and the first be 2 or -0.6, each with probability $1/2$. Then $\bar{d}_s = (0.7, 0)$ and the expected-utility action is $(0.85, 0.15)$. The two sample actions are $(1, 0)$ and $(0.2, 0.8)$, whose mean is $(0.6, 0.4)$. Here the equality-only action at the mean remains feasible, although one sample's equality-only action is infeasible.

The population statements precede componentwise clipping. In general, expectation does not commute with clipping, and a bounded affine predictor need not represent the minimizing forecast in every context. The statements therefore identify learning targets without guaranteeing their recovery by the delayed online update.

B.3 Constrained regret and allocation stability

For any vector u, write $\|u\|_{\Sigma_s} = \sqrt{u^\top \Sigma_s u}$. For forecasts $v, d \in \mathbb{R}^n$, define

$$w_v = \mathcal{A}_{s,h}(v;\gamma), \qquad w_d = \mathcal{A}_{s,h}(d;\gamma),$$
$$\mathcal{R}_{s,h}(v, d) = U_{s,h}(w_d; d) - U_{s,h}(w_v; d).$$

Proposition 4 (Projected control of constrained regret)**.** *For the capped feasible set, and more generally for any nonempty closed convex feasible set contained in $\{w : \mathbf{1}^\top w = 1\}$,*

$$\frac{\gamma}{2k_h}\|w_d - w_v\|^2_{\Sigma_s} \le \mathcal{R}_{s,h}(v, d) \le \frac{k_h}{2\gamma}(v - d)^\top Q_s (v - d). \tag{16}$$

The upper bound is attained whenever both equality-only optimizers $w^{\mathrm{eq}}_{s,h}(v)$ and $w^{\mathrm{eq}}_{s,h}(d)$ belong to the feasible set.

Proof. Let $\Delta w = w_d - w_v$. First-order optimality at w_v gives

$$[v - (\gamma/k_h)\Sigma_s w_v]^\top \Delta w \le 0.$$

Expanding the regret around w_v therefore gives

$$\mathcal{R}_{s,h}(v,d) \le (d-v)^\top \Delta w - \frac{\gamma}{2k_h}\Delta w^\top \Sigma_s \Delta w.$$

Since $\mathbf{1}^\top \Delta w = 0$, its right-hand side is at most

$$\max_{\mathbf{1}^\top u = 0} \left\{ (d-v)^\top u - \frac{\gamma}{2k_h} u^\top \Sigma_s u \right\} = \frac{k_h}{2\gamma}(v-d)^\top Q_s (v-d).$$

The maximizing vector is $(k_h/\gamma)Q_s(d-v)$. For the lower bound, expansion around w_d gives

$$\mathcal{R}_{s,h}(v,d) = [d - (\gamma/k_h)\Sigma_s w_d]^\top \Delta w + \frac{\gamma}{2k_h}\Delta w^\top \Sigma_s \Delta w,$$

whose first term is nonnegative by optimality at w_d. If both equality-only optimizers are feasible, they are also the constrained optimizers, and the identity in Eq. (12) gives equality in the upper bound. □

Equation (16) is a consequence of quadratic strong concavity and budget feasibility. It retains the same projected error as the equality-only identity while allowing the deployed position bounds. Its coefficient cannot be reduced uniformly: with $n = 3$, $\Sigma_s = I_3$, $\gamma/k_h = 1$, $w_{\max} = 0.35$, $v = 0$ and $d = (0.01, -0.01, 0)$, both equality-only actions are strictly inside the capped set and both sides of the upper bound equal 10^{-4}.

Proposition 5 (Allocation stability)**.** *For any two forecasts $v, v' \in \mathbb{R}^n$, let $\Delta w = \mathcal{A}_{s,h}(v;\gamma) - \mathcal{A}_{s,h}(v';\gamma)$. Then*

$$\|\Delta w\|_{\Sigma_s} \le \frac{k_h}{\gamma}\sqrt{(v-v')^\top Q_s (v-v')}. \tag{17}$$

If $\lambda_{\min}(\Sigma_s) > 0$ denotes the smallest eigenvalue, then

$$\|\Delta w\|_1 \le \frac{k_h}{\gamma}\sqrt{\frac{n}{\lambda_{\min}(\Sigma_s)}}\sqrt{(v-v')^\top Q_s (v-v')}. \tag{18}$$

Proof. Adding the two first-order optimality inequalities yields

$$\frac{\gamma}{k_h}\Delta w^\top \Sigma_s \Delta w \le (v-v')^\top \Delta w.$$

For any u with $\mathbf{1}^\top u = 0$ and any vector e, the budget projection satisfies

$$|e^\top u| = |(Q_s e)^\top \Sigma_s u| \le \sqrt{e^\top Q_s e}\,\|u\|_{\Sigma_s},$$

using $Q_s\Sigma_s u = u$ and $Q_s\Sigma_s Q_s = Q_s$. Apply this with $e = v - v'$ and $u = \Delta w$; dividing by $\|\Delta w\|_{\Sigma_s}$, unless it is zero, proves Eq. (17). Finally, $\|\Delta w\|_1 \le \sqrt{n}\|\Delta w\|_2 \le \sqrt{n/\lambda_{\min}(\Sigma_s)}\|\Delta w\|_{\Sigma_s}$ gives Eq. (18). □

This stability statement is the quadratic, budget-projected form of the smoothness of the regularized prediction map (Blondel et al., 2020a, Propositions 2–3). For fixed pretrade holdings $\widetilde{w}$, the reverse triangle inequality gives

$$|\|\mathcal{A}_{s,h}(v;\gamma) - \widetilde{w}\|_1 - \|\mathcal{A}_{s,h}(v';\gamma) - \widetilde{w}\|_1| \le \|\Delta w\|_1.$$

Multiplying by the decimal fee c from Section 5 therefore bounds the change in the one-trade fee by $c\|\Delta w\|_1$. This comparison holds covariance and pretrade holdings fixed and compares the candidate allocations. Changing the execution-gate branch, later holding drift, covariance updates and cross-sleeve capital resets require their own path accounting.

Additional linear equalities. For m_{eq} linear equalities, let $B_{\text{eq}} \in \mathbb{R}^{m_{\text{eq}} \times n}$ have full row rank and $b_{\text{eq}} \in \mathbb{R}^{m_{\text{eq}}}$. The same arguments apply to a nonempty closed convex feasible set contained in $\{w : B_{\text{eq}} w = b_{\text{eq}}\}$ after replacing Q_s with

$$Q_s^{(B)} = \Sigma_s^{-1} - \Sigma_s^{-1} B_{\text{eq}}^\top (B_{\text{eq}} \Sigma_s^{-1} B_{\text{eq}}^\top)^{-1} B_{\text{eq}} \Sigma_s^{-1}.$$

Forecast shifts in the range of $B_{\text{eq}}^\top$ are then irrelevant to the action. This is the same null-space geometry used for equality constraints in Butler & Kwon (2023, Section 2.4).

Scope for delayed adaptation. The regret and stability bounds concern the action induced by a forecast within one fixed context. They do not imply that the surrogate gradient in Proposition 1 decreases the actual capped regret on every step. Componentwise clipping changes that direction, the affine parameter bounds restrict attainable forecasts, and the saved gradient is applied after the head has changed. Net wealth further depends on subsequent decisions and execution costs. The empirical role of the analysis is to separate target choice, constraint effects and action changes when testing utility-directed adaptation under the complete delayed trading protocol.

C Extended Related Work

C.1 Test-Time Adaptation and Label Timing Protocols

Test-time training (TTT) originally emerged in computer vision to combat distribution shift using self-supervised tasks (Sun et al., 2020; Wang et al., 2021; Gandelsman et al., 2022; Su et al., 2022). In time-series forecasting, TTT leverages sequentially observed targets. FSNet (Pham et al., 2023) uses fast adapters with associative memory, OneNet (Zhang et al., 2023) updates forecasters and combination weights jointly, and DoubleAdapt (Zhao et al., 2023) meta-learns data and model adapters for stock trend prediction.

To reduce computational overhead, a second line of research adapts only lightweight output components: SOLID (Chen et al., 2024) calibrates prediction layers, ELF (Lee et al., 2025b) blends online forecasters with frozen foundation models, and ORCA (Dai et al., 2026) employs black-box residual adapters. A crucial protocol challenge in online adaptation is label timing. TAFAS (Kim et al., 2025) and PETSA (Medeiros et al., 2025) calibrate using partial and full targets, Proceed (Zhao & Shen, 2025) compensates for label latency drift, and DSOF (Lau et al., 2025) highlights that standard online protocols often inadvertently evaluate on targets already used for training. Following Wang et al. (2026), DU-TTT strictly adheres to a realistic evaluation protocol where targets are consumed only after full maturation at their saved decision contexts.

C.2 Decision-Focused Learning and Surrogates

Decision-focused learning (DFL) trains prediction models by backpropagating through downstream optimization tasks (Donti et al., 2017; Elmachtoub & Grigas, 2022; Mandi et al., 2024; Sadana et al., 2025). Gradients are typically computed via differentiable surrogates (Blondel et al., 2020a; Huang & Gupta, 2024), learned local decision losses (Shah et al., 2022; Schutte et al., 2024), or implicit differentiation through KKT conditions (Agrawal et al., 2019; Tang & Khalil, 2024). For non-smooth or piecewise constant solution maps, perturbed optimizers (Berthet et al., 2020), black-box solvers (Vlastelica et al., 2020), or identity-with-projection surrogates (Sahoo et al., 2023) are widely adopted.

In quantitative portfolio management, mean–variance IPO connects forecast errors to asset allocation utility (Butler & Kwon, 2023; Lee et al., 2025a). Specialized DFL layers have been designed for distributionally robust portfolios (Costa & Iyengar, 2023), risk budgeting (Uysal et al., 2021), cardinality constraints (Anis & Kwon, 2025), and direct Sharpe ratio maximization (Zhang et al., 2020). Parametric portfolio policies (Brandt et al., 2009) directly map asset characteristics to portfolio weights. While Jeon et al. (2026) incorporate a smooth top-K operator in a tangency layer for offline training and Nosaka et al. (2026) embed KKT conditions for long-only programs, DU-TTT differentiates an equality-relaxed allocation map. This avoids vanishing gradients under binding position caps without backpropagating through the full solver.

C.3 Learning to Rank for Portfolio Selection

Learning-to-rank methods construct cross-sectional stock selection strategies by predicting relative asset orderings (Song et al., 2017; Feng et al., 2019; Poh et al., 2021; Zhang et al., 2022). Listwise and rank-correlation objectives (Xia et al., 2008; Lin et al., 2026) are made continuous via smooth sorting relaxations (Grover et al., 2019; Blondel et al., 2020b), optimal transport (Cuturi et al., 2019; Xie et al., 2020), sparse top-K operators (Sander et al., 2023), or limited multi-label projections (LML) (Amos et al., 2019). Although ranking surrogates perform well in predict-then-optimize settings (Mulamba et al., 2021; Mandi et al., 2022), they discard asset covariance and portfolio risk.

DU-TTT employs LML strictly as a boundary isolator to weight feedback on the active selection margin, while retaining utility as the primary update direction.

C.4 Online Learning with Delays and Financial Friction

Online learning under delayed feedback maps late-arriving losses back to historical decisions (Joulani et al., 2013; Quanrud & Khashabi, 2015). Regret bounds and stability have been refined via adaptive (McMahan & Streeter, 2014; Joulani et al., 2016), optimistic (Flaspohler et al., 2021), movement-cost-aware (Qiu et al., 2026), and scale-free normalized gradient methods (Hazan et al., 2015; Orabona & Pál, 2018). In online decision-making, IGT-OMD (Amoh et al., 2026) re-evaluates stale decision gradients, while ADOWIP (Wang, 2026) schedules updates using downstream losses.

In financial markets, trading friction fundamentally alters optimal execution: proportional costs create no-trade zones (Davis & Norman, 1990), and multi-period quadratic costs lead to dynamic aim portfolios (Gârleanu & Pedersen, 2013; Boyd et al., 2017). Because simple baseline portfolios (e.g., $1/N$) are notoriously hard to beat out-of-sample (DeMiguel et al., 2009), combining persistent signals lowers friction (DeMiguel et al., 2020). Existing DFL methods explicitly penalize turnover (Linghu et al., 2025; Wang & Hasuike, 2026) or ℓ_1 weight shifts (Das et al., 2013; Li et al., 2018). In contrast, DU-TTT leaves the allocator and cost model completely untouched; turnover reduction arises purely as an intrinsic byproduct of updating along the decision-utility gradient.

C.5 Pretrained Time-Series and Financial Backbones

Pretrained foundation models and time-series large language models (Gruver et al., 2023; Jin et al., 2024; Ansari et al., 2024; Das et al., 2024; Woo et al., 2024) have recently been applied to financial market forecasting (Lopez-Lira & Tang, 2023; Zhang et al., 2024; Hwang et al., 2025). However, given the high noise-to-signal ratio of equity returns, their out-of-sample performance over simple random walks remains subtle and sparse (Noguer i Alonso & Franklin, 2026). In our unified framework, pretrained backbones serve as base feature extractors, demonstrating that DU-TTT delivers robust test-time calibration across both traditional statistical models and deep pretrained backbones.

D Margin components: detailed analysis

Table 6: Focus weight and temperature of DU-TTT, net of 25 bps.

	Fold 1			Fold 2		
Weighting	Cell SR ↑	CAGR % ↑	Turn./yr ↓	Cell SR ↑	CAGR % ↑	Turn./yr ↓
$\alpha = 0$	0.627	14.37	15.96	0.648	15.92	12.47
$\alpha = 0.25, \tau = 8$	0.660	15.11	14.27	0.695	17.08	10.99
$\alpha = 0.5, \tau = 8$	0.712	16.24	13.64	0.712	17.24	10.39
$\alpha = 0.75, \tau = 8$	0.695	15.83	13.06	0.723	17.53	9.78
$\alpha = 1, \tau = 8$	0.681	16.12	12.71	0.673	16.33	9.06
$\tau = 1, \alpha = 0.5$	0.621	14.59	16.14	0.642	16.15	12.85
$\tau = 2, \alpha = 0.5$	0.667	15.71	15.62	0.633	15.97	12.92
$\tau = 4, \alpha = 0.5$	0.686	15.93	14.66	0.651	16.13	11.76
$\tau = 8, \alpha = 0.5$	0.712	16.24	13.64	0.712	17.24	10.39
$\tau = 16, \alpha = 0.5$	0.718	16.49	13.54	0.729	17.43	9.61
$\tau = 32, \alpha = 0.5$	0.673	15.29	13.81	0.641	14.93	9.62
$\tau = 64, \alpha = 0.5$	0.563	13.35	16.33	0.625	14.19	12.43

At the selected settings every focus weight $\alpha > 0$ improves on $\alpha = 0$ in both folds, and the temperature has an interior optimum at $\tau \in \{8, 16\}$ in both folds. Every row keeps the fold's selected step, clip, horizon exponent and bias limit and changes only the weighting of Eq. (8); the shaded rows are the selected configuration ($\alpha = 0.5$, $\tau = 8$ in both folds). Cell SR is the mean over 327 cells; CAGR and turnover are the aggregate strategy.

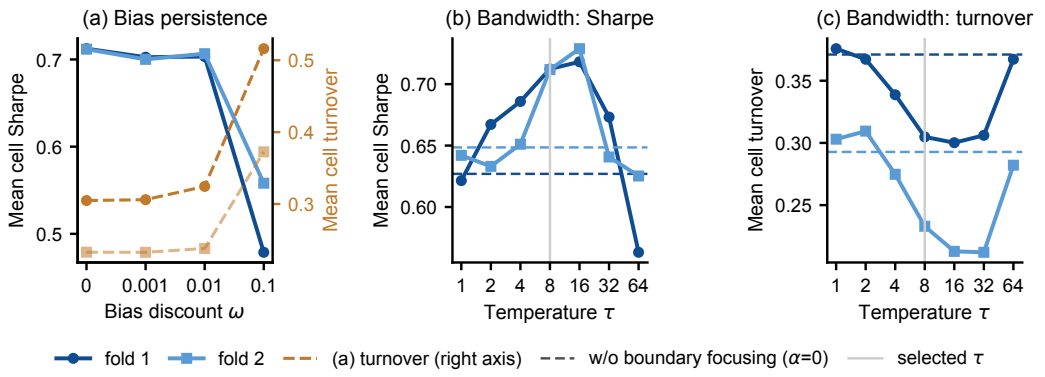


Figure 5: One-parameter sweeps of DU-TTT at 25 bps over the 327 cells, every other setting fixed. (a) Multiplying the accumulated bias by $1 - \omega$ before each step lowers Sharpe and raises turnover once ω reaches 0.1. (b, c) Mean cell Sharpe and turnover against the temperature τ; dashed lines are the ablation without boundary focusing ($\alpha = 0$) and the grey line the selected $\tau = 8$.

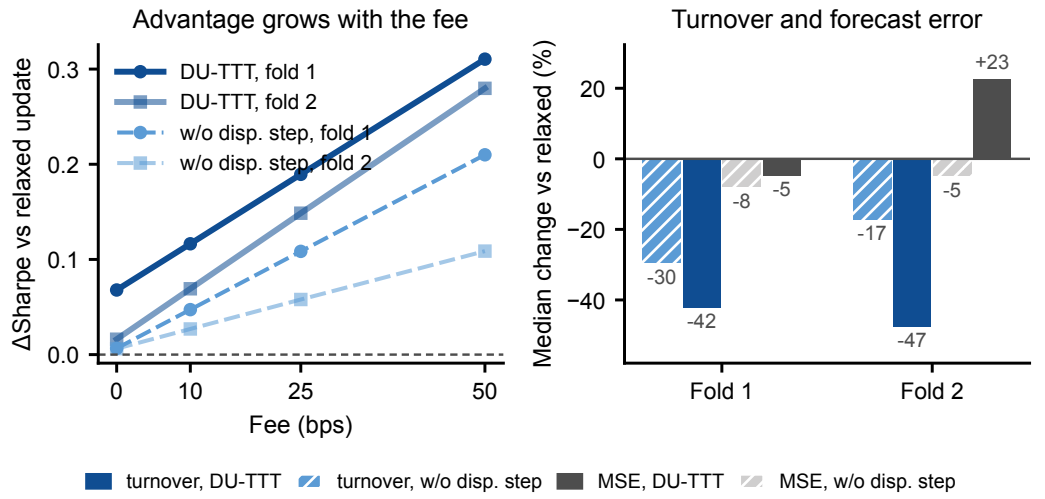


Figure 6: The margin components across trading fees. Left: mean cell Sharpe of DU-TTT and of its ablation without the dispersion-scaled step, minus the relaxed update, on the same 327 cells at each fee; the advantage grows with the fee. Right: median per-cell change against the relaxed update at 25 bps in turnover and in squared forecast error.

Boundary focusing and the temperature. As shown in Table 6 and Figure 5(b, c): ❶ every focus weight $\alpha > 0$ improves on $\alpha = 0$ in both folds, with mean cell Sharpe rising from 0.627 and 0.648 to 0.66–0.72; ❷ the temperature, which sets how many names the membership treats as marginal (Eq. (10)), has an interior optimum at $\tau = 8$–16 in both folds, where the effective support in a 15-name universe is 3.75 to 2.73, near the three names the cap holds; ❸ turnover is lowest at $\tau = 16$ in fold 1 and $\tau = 16$–32 in fold 2, so the criterion $m_s(\tau) \approx K$ locates a region and validation selects within it.

Effect of trading fees. Figure 6 scores every cell at each fee. The ablation without the dispersion-scaled step shows the same monotone growth of the advantage with the fee as DU-TTT, with a smaller gain.

E EVALUATION PROTOCOL DETAILS

This appendix completes Section 5: the feature set and risk model, the close-level order of operations, the aggregate strategy's accounting, the role of each control, and the language-model protocol.

Table 7: Chronological splits. Validation-selected configurations are fixed for each fold's later evaluation; fold 3 has no validation pass and uses the fold-2 configuration.

Fold	Pass	Fit ends	Scored period
1	Validation	2023-06-30	2023-07-03 to 2023-11-15
1	Evaluation	2023-12-31	2024-01-02 to 2025-01-31
2	Validation	2024-12-31	2025-01-02 to 2025-04-30
2	Evaluation	2025-04-30	2025-06-02 to 2026-04-15
3	Held-out	2026-04-15	2026-05-15 to 2026-09-21

Selection and the held-out period. Design choices were screened on validation windows; the final configuration was then chosen among twelve variants by a pre-registered rule on the fold-1 and fold-2 evaluation windows, which are therefore not fully held out. Eleven of the variants were opened in earlier rounds, and the pre-registered rule governed the last round. Eleven diagnostic arms and three matched-control families were also scored on these windows, one of them a member of the final family. The sealed period (App. G) was not used during design and was evaluated once with fixed fold-2 settings. Its panel was built after the method was fixed and extends the same adjusted-price construction; the frozen forecaster is refitted through 2026-04-15 with labels purged at the cutoff.

Features and risk. The eight features are 5/21/63-day momentum, 21/63-day realized volatility, price relative to its 50-day moving average, the fraction of positive returns over 14 days, and drawdown from a 126-day high, each using prices through the decision close. The ridge forecaster fixes their training means and standard deviations and uses penalty 10. LightGBM uses the same raw features with 200 trees, learning rate 0.05, 15 leaves, a minimum of 40 child samples and seed zero. The allocator uses $\gamma = 2$ and $w_{\max} = 0.35$; its sample covariance uses the preceding 125 return observations excluding the current close, annualized by 252, with at least 40 complete rows required, and the solver adds 10^{-10} to the diagonal. Basket construction uses 2022–2023 statistics from a current index roster with complete price histories, so the panel is retrospectively available and the construction period extends beyond the first fold's validation start.

Close-level chronology. Each universe–horizon sleeve decides on Fridays present in the trading index, with an additional initial investment and no trade at the final close, and starts in cash. At close t its previous holdings first earn $r_t = P_t \oslash P_{t-1} - \mathbf{1}$ and drift to the pre-trade vector $\widetilde{w}_t$; matured labels are processed next, followed by any new forecast and trade. With w_t the post-close holdings, the daily net return is

$$R_t^{\text{net}}(c) = w_{t-1}^{\top} r_t - c \, \|w_t - \widetilde{w}_t\|_1, \qquad c = c_{\text{bps}}/10^4,$$

with $c_{\text{bps}} \in \{0, 10, 25, 50\}$. L1 turnover includes buys and sells and is zero on a day without a trade; the cap is imposed on target allocations and drift can move weights between decisions. Fees are an additive debit on a fixed gross-return and turnover path and do not model cost-dependent sizing or impact.

Aggregate strategy accounting. The combined policy allocates equal capital to the 327 sleeves, restores the sleeve allocations daily and can therefore trade between the underlying Friday decision dates. Asset trades are charged against the strategy's own drifted holdings, including capital transfers between sleeves and netting opposing trades in shared assets. Its gross return equals the mean sleeve gross return, but its net return need not equal the mean sleeve net return, which is retained as an accounting diagnostic. The equal-weight comparator combines the same baskets' equal-weight policies and is not a uniformly weighted portfolio of the whole panel.

Uncertainty. Intervals for differences in mean cell Sharpe use a cluster bootstrap over the 109 universes, so the three horizons of a universe are resampled together. Intervals for paired excess

returns of the aggregate strategy use 10,000 circular-block bootstrap resamples of the synchronized daily return differences and are reported as 95% simultaneous intervals. Both condition on the selected configurations.

Execution gate. In the matched-execution study, the gate trades only when the predicted five-day utility improvement exceeds $c_{\text{int}}\|w_t^\star - \widetilde{w}_t\|_1$, where c_{int} is a fixed internal decimal fee and $w_t^\star$ the candidate allocation. The five-day utility is $(5/h)U_{t,h}$, an explicit horizon-scaling approximation; initial investment and repairs of infeasible holdings are executed; keeping the current holdings means their Euclidean projection onto $\mathcal{W}$, and the internal gate fee is held fixed across the evaluated cost ladder.

Role of each control. Table 8 identifies the information used by each rule and the explanation it tests. The affine comparison holds the forecast representation fixed and changes the online calibration rule; gated minimum variance supplies a risk-and-execution control with no learned return mean.

Table 8: Learning and execution controls and the explanation each tests.

Rule	Learning signal or adapted object	Explanation tested
Frozen / static rescaling	Fixed features and coefficients; static scale	Whether signal rescaling suffices without test-time learning.
MSE / RLS	Return error; affine coefficients	Whether ordinary supervised tracking explains the improvement.
Gated minimum variance	Covariance and utility-based trade acceptance	Whether risk allocation and turnover control suffice without a learned mean.
DU-TTT	Covariance-weighted utility direction, focused on and scaled at the selection margin; affine coefficients	Whether task geometry improves the same calibration head.

All policies share return-before-trade timing, the target-position cap and evaluated turnover costs. An execution-gate fee is distinct from a fee differentiated through a learning objective.

Language-model backbone protocol. Each frozen Qwen checkpoint maps the eight anonymous price features to 1-, 5- and 20-day return forecasts. The frozen, MSE and DU-TTT arms replay the same cached forecasts, and MSE and DU-TTT are selected on the same 60-setting grid per backbone and fold. Appendix F gives the prompt, checkpoints and deployment.

F Language-model protocol and deployment

This appendix documents how the frozen language-model backbones produce base forecasts: the checkpoints and their provenance, the exact input protocol, caching and coverage and the hardware and software stack.

F.1 Checkpoint and input provenance

The transformer remains frozen throughout forecasting and portfolio evaluation. Qwen's native multimodal checkpoints are used in text-only mode, with image and video inputs disabled. A separate head with $2n$ parameters is reset for every fold and universe–horizon sleeve, and no validation head state carries into the historical evaluation.

The comparison holds nominal model size fixed at 27B across two recent checkpoints. Both releases postdate the last date scored with these backbones, 2026-04-15 (end of fold 2), and the model cards do not establish a market-data training cutoff that excludes these outcomes. We therefore retain the historical qualification for the entire comparison.

Exact system instruction. Every model receives the following system text:

Table 9: Checkpoint provenance.

Checkpoint	Official release	Revision prefix	BF16 weights
Qwen3.8-27B	2026-08-14	`1d4bf0f2ff60`	55.6 GB
Qwen3.6-27B	2026-04-22	`6a9e13bd6fc8`	55.6 GB

Official repositories are `Qwen/Qwen3.8-27B` and the correspondingly named `Qwen/` repositories on Hugging Face. GB denotes decimal storage, not active parameter count or GPU allocation. Release dates follow the official model announcements, not repository creation. Full revisions and per-file SHA256 checks are retained in the experiment manifests.

> You forecast an anonymous asset's future returns using only the supplied technical indicators observed at the current market close. All returns and ratios use decimal units: 0.01 means 1 percent. Trading days index the prices. For k equal to 5, 21 or 63, momentum_kd is P[t]/P[t-k]-1: the cumulative return over all k past trading days, not a one-day return. Its equivalent constant one-day rate is (1+momentum_kd)^(1/k)-1. daily_volatility_21d and daily_volatility_63d are sample standard deviations of one-day returns over the stated past windows, not annualized or cumulative volatility. price_over_50d_average_minus_one is the current price divided by the past 50-day average price minus 1. positive_daily_return_fraction_14d is the fraction of the past 14 one-day returns that are positive. drawdown_from_126d_high is the current price divided by the highest price over the past 126 trading days minus 1. Estimate the expected future cumulative returns P[t+h]/P[t]-1 for h equal to 1, 5 and 20. The three outputs must use these future horizons, not the historical feature windows. Return only a JSON array of three decimal return estimates in the order [1-day, 5-day, 20-day], without explanation.

The user message begins with "Features (all values in decimal units):" and gives one named value per line: 5-, 21- and 63-day momentum; 21- and 63-day daily volatility; price over its 50-day average minus one; the fraction of positive daily returns over 14 days; and drawdown from a 126-day high. Values are formatted to six decimal places before tokenization. The official checkpoint chat template is used with `enable_thinking=False`. No return target enters the prompt.

Caching and coverage. Request identifiers hash the complete anonymous message pair. The manifest records the input panel, feature order, prompt, tokenizer and template hashes, checkpoint revision and runtime configuration. The prepared requests cover 476 distinct assets in the 109 specified universes. A complete eligibility replay requires 44,799 vector forecast calls, or 696,084 scalar lookups, with none missing. Of the 64,951 supplied asset–date links, 410 are unused because the original complete-history filter excludes the corresponding asset from its scored sleeve. These links do not enter portfolio performance; they are kept so that input preparation and evaluation remain separately auditable.

Parsing and export. Export reparses each complete raw completion and rejects duplicate requests, including duplicates with identical values. The semantic parser accepts JSON numbers, or strings consisting entirely of a finite JSON decimal-number literal, and preserves their decimal value before conversion to binary64. Booleans, nonfinite values, overflow, nonzero underflow and commentary are rejected. A finite forecast below -1 remains visible in the diagnostics and is never clipped or replaced. A completion must stop normally before its output cap. The exporter verifies complete request coverage, the canonical template hash, and actual input tokens against independently tokenized inputs, and all policies for a checkpoint share the exported bytes. For parallel inference, fixed disjoint request shards are merged only after every shard completes under the same configuration; no output is selected from repeated generations.

F.2 Hardware and software

Stack. All large-model inference runs on a host with eight NVIDIA H800 80 GB GPUs, with tensor parallelism 2 per replica. The driver is 570.172.08, supporting CUDA 12.8. The models use PyTorch 2.10.0+cu128, vLLM 0.19.1 and Transformers 5.16.1, and a workspace-local CUDA 12.8.1 compiler and headers support the Qwen kernel compilation. Isolated environments and all temporary, compiler and model caches reside in the experiment workspace; the system driver and shared Python environments are unchanged.

Inference settings. Inference uses BF16, a maximum model length of 2,048 tokens, greedy decoding at temperature zero, seed 20260910 and a 96-token output budget. Prefix caching and constrained JSON decoding are disabled. The outer batch contains 32 requests, with at most 32 concurrent sequences and 8,192 scheduled tokens. The profile uses eager execution with asynchronous scheduling disabled, and final partial batches are retained and recorded. The manifests report actual GPU assignments, replica counts and memory allocations. Portfolio calibration is CPU work: cached predictions are reused across the full validation grid and the historical replays, which separates the one-time inference cost from the inexpensive delayed-head updates.

G HELD-OUT CONFIRMATION

The main text reports the two evaluation periods, folds 1 and 2, which informed the design and the choice of the final configuration. This appendix reports a third, sealed period that did not: it was specified, with its arms, before any policy was run on it, has no validation window, runs every policy at its fold-2 selection, and was evaluated once (Appendix E). It covers 2026-05-15 to 2026-09-21, 88 trading days, on the same 109 universes and three horizons at 25 bps.

Table 10: Held-out period: ablations of DU-TTT and paired excess returns.

Variant	Cell SR ↑	CAGR % ↑	Sharpe ↑	Turn. ↓	Δ cell SR, DU-TTT minus row [95%]
DU-TTT	**0.562**	**19.58**	**1.574**	**16.59**	reference
w/o dispersion-scaled step, own selection	0.405	15.91	1.323	20.97	+0.158 [+0.044, +0.272]
relaxed update (own selection)	0.333	12.81	1.110	22.83	+0.229 [+0.126, +0.332]
affine MSE (own selection)	0.072	11.82	0.897	38.96	+0.490 [+0.349, +0.634]

2026-05-15 to 2026-09-21, evaluated once at the fold-2 selections. Cell SR is the mean over 327 universe–horizon cells; CAGR, Sharpe and annualized turnover are the aggregate strategy. The interval is a cluster bootstrap over the 109 universes (10,000 replicates). Paired excess returns of the aggregate strategy, with 95% simultaneous circular-block intervals, are: over affine MSE $+6.51$ [-15.37, +28.39]; over the ablation without the dispersion-scaled step $+3.13$ [-4.53, +10.80]; over the relaxed update $+5.88$ [-2.53, +14.29].

The dispersion-scaled step. The comparison the period was set up to test is DU-TTT against its ablation without the dispersion-scaled step, each at its own selection (Table 10). DU-TTT raises mean cell Sharpe from 0.405 to 0.562, a difference of 0.158 with a universe-cluster interval of $[+0.044, +0.272]$, and aggregate Sharpe from 1.323 to 1.574. Against the relaxed update the difference is 0.229 $[+0.126, +0.332]$, and against affine MSE 0.490 $[+0.349, +0.634]$.

Squared-error calibration. On the aggregate strategy DU-TTT again improves on affine MSE, with CAGR 19.58% against 11.82%, Sharpe 1.57 against 0.90 and less than half the turnover.