Stiff deterministic and stochastic systems in physics and finance: Physics Informed neural networks and Onflow portfolio management

Gabriel Turinici

CEREMADE Université Paris Dauphine - PSL, Paris, France

DETERMINISTIC AND STOCHASTIC PROCESSES, Iasi, Feb. 25-28, 2026

Part I

Time sampling in Physics-Informed Neural Networks

Outline

What are PINNs (Physics-Informed Neural Networks) ?

• one of the most impactful applications of Neural Networks (cf; physics Nobel price 2024) in ... physics; huge literature, original 2019 paper [Raissi et al., 2019] $\geq$2k citations/y !

• **PINNs** combine deep learning techniques with physical laws described by partial differential equations (PDEs).

• They allow neural networks to learn solutions to PDEs, leveraging both data and known physical constraints.

• Unlike traditional neural networks, PINNs directly encode the physical knowledge into the loss function.

Applications of PINNs

• **Computational Fluid Dynamics**: Modeling airflow or fluid flow using Navier-Stokes equations.

• **Heat Transfer**: Solving heat equations with real-world constraints.

• **Wave Propagation**: Modeling sound, seismic waves, and other phenomena described by hyperbolic PDEs.

• **Quantum Mechanics**: Schrödinger's equation in high dimensions or for complex quantum systems.

• **...**

Outline

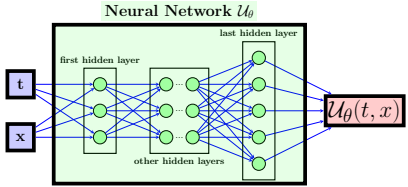
How do PINNs work? I

- consider some evolution equation:

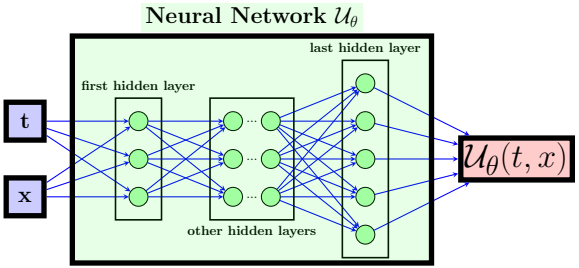
$$\partial_t u = \mathcal{G}_t u, \tag{1}$$
$$u(0, x) = u_0(x), \ \forall x \in \Omega \tag{2}$$
$$u(t, x) = u_{bc}(t, x), \ \forall t \in [0, T], x \in \partial\Omega. \tag{3}$$

$u(\cdot, \cdot)$ = solution, t=time, x=space, $\mathcal{G}_t$=operator (e.g., $\mathcal{G}_t = \partial_{xx}$ for the heat equation), $u_0(\cdot)$, $u_{bc}(\cdot, \cdot)$ = initial & boundary conditions.

- NN constructs an approximation $\mathcal{U}_\theta(t, x)$ of the solution $u(t, x)$



How do PINNs work? II



to solve $\partial_t u = \mathcal{G}_t u$ + IC + BC

- construct **automatically** the error $\boxed{w(t,x) = \partial_t \mathcal{U}_\theta(t,x) - \mathcal{G}_t \mathcal{U}_\theta(t,x)}$

- The neural network is trained to minimize loss : $\|w\|^2$ + fit to IC + fit to BC + fit any available data /measurements.

- Technical remark: constructing the loss and optimizing with gradient descent is a second/ third order auto-diff process (!) Special activations are necessary (RELU3, tanh, etc.)

Sampling time in PINN

- The NN is trained to minimize a loss functional; original [Raissi et al., 2019] is $\int_0^T \int_\Omega |w(t,x)|^2 dxdt$
- in fact any weighted ($\rho > 0$) loss has the same optimum $w = 0$:

$$\mathcal{L}(\theta) = \int_0^T \int_\Omega \rho(t)|w(t,x)|^2 dxdt, \tag{4}$$

- **weight $\rho(t) \geq 0$ are main object here; $\rho(t)$ is chosen to speed up the convergence and improve the quality of the output.**
- Computational cost assumption obtaining error w has numerical costs. If max computational cost is B then :

$$\int_0^T \frac{1}{\|w(t,\cdot)\|^2} dt \leq B. \tag{5}$$

- Research question : given max cost B what is the best weighting scheme ρ that makes final error $\|w(T,\cdot)\|$ minimal ? Similar for $\int_0^T \|w(t,x)\|^2 dt$.

Outline

Sampling time in PINN: intuition

$$\text{loss}: \quad \boxed{\mathcal{L}(\theta) = \int_0^T \int_\Omega \rho(t)|w(t,x)|^2 dxdt} \tag{6}$$

$$\text{error}: \quad \boxed{w(t,x) = \partial_t \mathcal{U}_\theta(t,x) - \mathcal{G}_t \mathcal{U}_\theta(t,x)} \tag{7}$$

- **Intuition**: errors will propagate in time, any initial error will amplify; this is the "causal" view [Wang et al., 2024]
- **Causal recommendation**: solve more precisely the initial points in time.

- **Question**: is this optimal, what "more precisely" means quantitatively ?

- **more general question**: are we going to oversample past, present or future ? Which one is more important ?

Sampling time in PINN: formulation min/max, control

- loss : $\boxed{\mathcal{L}(\theta) = \int_0^T \int_\Omega \rho(t)|w(t,x)|^2 dxdt}$

- error : $\boxed{w(t,x) = \partial_t \mathcal{U}_\theta(t,x) - \mathcal{G}_t \mathcal{U}_\theta(t,x)}$

- **Intuition**: assuming NN is expressive, problem can be formalized as a random control problem or a min-max problem

$$\min_{\rho} \max_{\substack{\partial_t \mathcal{U}_\theta(t,x) = \mathcal{G}_t \mathcal{U}_\theta(t,x) + w(t,x), \\ \int_0^T 1/\|w(t,x)\|^2 dt \leq B}} \|\mathcal{U}_\theta(T,\cdot) - u(T,\cdot)\| \tag{8}$$

- **Alternative setting** [Turinici, 2024a]: $\min_\rho \max_{\substack{\partial_t \mathcal{U}_\theta(t,x) = \mathcal{G}_t \mathcal{U}_\theta(t,x) + w(t,x), \\ \int_0^T \rho(t)\|w(t,x)\|^2 dt \leq C_p}} \|\mathcal{U}_\theta(T,\cdot) - u(T,\cdot)\|$

Sampling time in PINN

Proposition ([Turinici, 2024b])

Consider $\mathcal{G}(u) = \lambda u$ and assume that (5) (computational cost hypothesis) holds true. Then under hypothesis ... the error $|\mathcal{U}_\theta(T) - u(T)|$ is optimal when $w(t)$ is proportional to $e^{-\lambda(T-t)/3}$ i.e., $d\rho(t)$ of the form $e^{-rt}dt$ for some $r = r(\lambda)$ (explicit).

• When $\lambda < 0$ future is oversampled not the past ! We follow regimes: chaotic for $Re(\lambda) > 0$, periodic $Re(\lambda) = 0$ or stable $Re(\lambda) < 0$.

• compare with [Wang et al., 2024] that uses a discrete form of $\rho(t) = e^{-\epsilon \int_0^t \|w(t,x)\|^2_{L^2_x}}$
Futher results : [Turinici, 2024a].

• follow-up question: what happens for other types of computational costs ?

Outline

Numerical results I

All tests reproducible using codes on Github

`https://github.com/gabriel-turinici/pinn_exponential_sampling` version August 31st 2024.

- situation: toy example $\mathcal{G}(u) = \lambda u$, equation $u' = \lambda u$

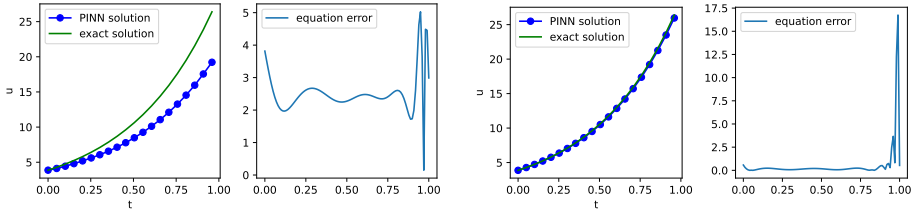


Figure: Results for $\lambda = 2.$ and sampling parameter $r = 2.0$. First two plots: the results for 500 epochs. Last two plots: results for 1500 epochs.

Numerical results II

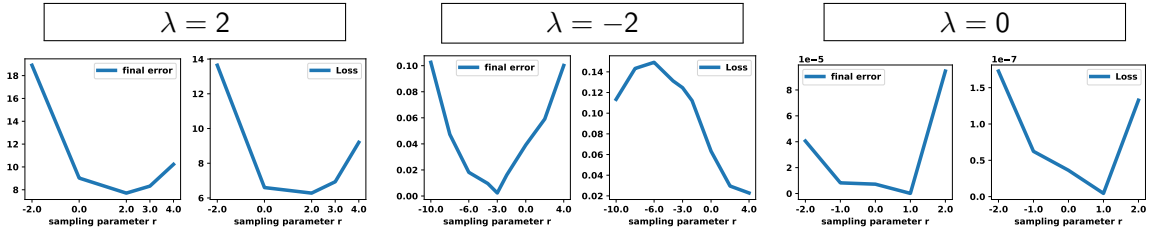


Figure: Sampling law influence for $\lambda = 2$ (from left to right, plots 1 and 2), $\lambda = -2$ (plots 3 and 4) and $\lambda = 0$ (plots 5 and 6). All sampling are done with law $\mathcal{E}^{0,T,r}$. Plots 1, 3 and 5 : the final error as a function of r. Plots 2,4 and 6 : the loss.

Numerical results III

- Burgers' equation ($u(x,t)$=velocity, $x \in [-1,1]$, $\nu = 0.01/\pi$=viscosity)

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}, \; u(0,x) = -\sin(\pi x), \; u(\pm 1, t) = 0 \; \forall t \in [0,1], \tag{9}$$

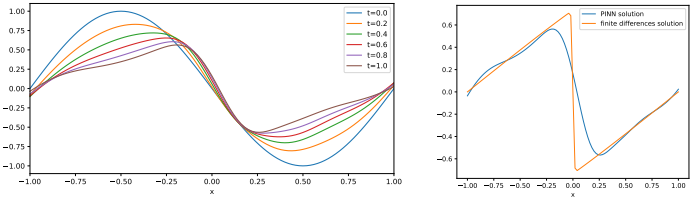


Figure: Burgers' equation sampling parameter $r = 0$ i.e., uniform law $\mathcal{E}^{0,T,0}$. Left plot: the solution at different times. Right plot: the comparison with a finite difference solution considered exact.

Numerical results IV

- Burgers' equation ($u(x,t)$=velocity, $x \in [-1,1]$, $\nu = 0.01/\pi$=viscosity)

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}, \ u(0,x) = -\sin(\pi x), \ u(\pm 1, t) = 0 \ \forall t \in [0,1], \tag{10}$$

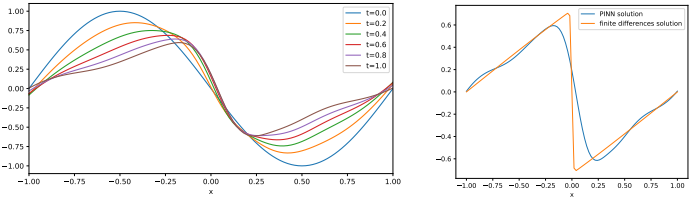


Figure: Burgers' equation sampling parameter $r = 1$ and law $\mathcal{E}^{0,T,r}$. Left plot: the solution at different times. Right plot: the comparison with a finite difference solution considered exact.

Numerical results V

• situation: Lorenz system (chaotic) $x'(t) = \sigma(y - x)$, $y'(t) = x(\rho - z) - y$, $z'(t) = xy - \beta z$. Here x, y, z= state variables, $\sigma = 10$, $\rho = 28$, $\beta = 8/3$=parameters , initial state $(1, 1, 1)$.

$r = 0$	$r = -10$

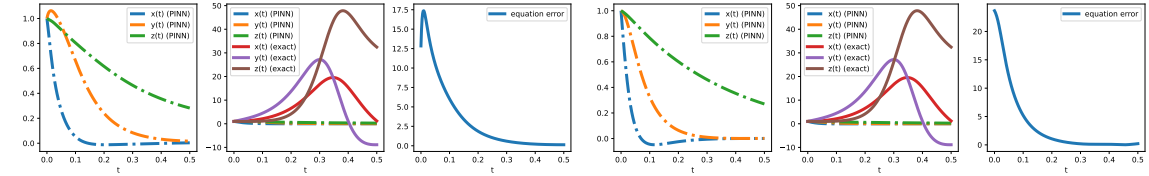


Figure: Lorenz system with weight parameters $r = 0$ and $r = -10$. The solution quality is not good.

Numerical results VI

• situation: Lorenz system (chaotic) $x'(t) = \sigma(y - x)$, $y'(t) = x(\rho - z) - y$, $z'(t) = xy - \beta z$. Here x, y, z= state variables, $\sigma = 10$, $\rho = 28$, $\beta = 8/3$=parameters , initial state $(1, 1, 1)$.

$r = 10$

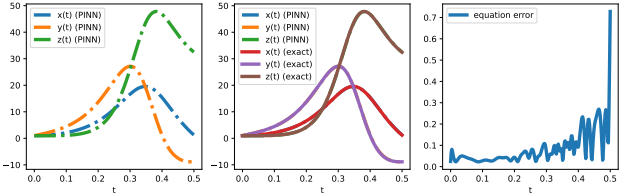


Figure: Lorenz system with weight parameter $r = 10$; the numerical and exact solutions are superposed and indistinguishable graphically.

Conclusions on time sampling in PINN

- adapted sampling in time is important in many time evolution PINNs
- the optimal sampling is of exponential type
- numerical results are positive
- the optimal parameter is not explicit but alternatives exist cf. [Turinici, 2024a]. Further work required.
- past is not always the one that is oversampled !

Part II

Stochastic dynamics: portfolio allocation with Onflow, joint work with P. Brugière (CEREMADE)

Disclamer

What follows is a scientific presentation and not an invitation to use one approach or another in a professional or personal framework, the reader is encouraged to use her/his common sense and critical views.
In particular past performances does not guarantee future performance.
Moreover, the result can depend on hyper-parameters and their robustness should be investigated in practice.

Outline

Onflow Summary

- Onflow: RL-based portfolio allocation; gradient flows; softmax parameterization; ODE updates; **transaction costs included**
- Stochastic optimization framework; evaluation vs. log-normal theory + old NYSE benchmarks
- Theory: log-normal, zero costs $\rightarrow$ recovers Markowitz optimum; optimal allocation
- Empirics (old NYSE): zero costs $\rightarrow$ comparable to Universal Portfolio + multiplicative updates
- High costs $\rightarrow$ outperforms existing dynamic methods; remains effective when others fail
- Model-free: uses only observed prices; no distributional assumptions; robust to model risk; practical for real trading
- Friction makes things more deterministic ...

Outline

Motivation and Literature Review for online porfolio management I

Task: allocate wealth in some investment portfolio across assets: π repartition.

Classical setting

- MPT; mean–variance; need **future** moments ([Markowitz, 1952]): $\max_{Var \leq c} \langle \pi, \mathbb{E}[\text{mean return}] \rangle$; but high uncertainty $\rightarrow$ degraded performance

Early alternatives

- [Kelly Jr, 1956]: optimal bet sizing; [Black and Litterman, 1990]: Bayesian returns; [Cover, 1991]: Universal Portfolio (model-free, hindsight-competitive): example $(1/2, 1/2)$ bet on asset couple: (constant, $1 \rightarrow 2 \rightarrow 1 \rightarrow ...$).

Online / model-free approaches

- Reinforcement-learning viewpoint (data $\rightarrow$ decisions)
- [Helmbold et al., 1998]: multiplicative updates; [Li et al., 2015]: mean-reversion; [Blum and Kalai, 1997]: transaction costs; [Kirby and Ostdiek, 2012]: low turnover; [Borodin et al., 2003]: Anticor (correlation)

Motivation and Literature Review for online porfolio management II

Surveys / resources [Li and Hoi, 2014], [Sato, 2019], [Sun et al., 2023]; ML-oriented: [Cesa-Bianchi and Lugosi, 2006], [Györfi et al., 2012, Ch. 3]; Toolbox: [Li et al., 2016]

Recent developments

- [Zhang et al., 2021b]: expert combination; [He and Li, 2024]: DTW-Anticor; [Zhang et al., 2021a]: policy-gradient RL; [Ngo et al., 2023]: RL vs deep learning

Our positioning

- Online, no-hindsight, model-free RL
- Gradient flows (continuous-time updates), softmax parameterization
- Natural handling of transaction costs

Outline

Market Model and Portfolio Dynamics

- Market setup: K assets; $t \in \{1, \dots, T\}$; prices $S_0^k = 1$, $S_t^k > 0$; price relatives $f_t^k = S_t^k / S_{t-1}^k$, $f_t = (f_t^1, \dots, f_t^K) \in (\mathbb{R}_+^*)^K$

- Portfolios: weights $\pi = (\pi(1), \dots, \pi(K))$; constraints $\pi(k) \geq 0$, $\sum_k \pi(k) = 1$; simplex $\mathcal{S}_K$, interior $\mathring{\mathcal{S}}_K$; no short selling; CRP = constant π with rebalancing

- Wealth dynamics: $\mathcal{V}_0(\pi) = 1$; $\mathcal{V}_t(\pi) = \prod_{s=1}^t \langle \pi_s, f_s \rangle$; $\ln \mathcal{V}_t(\pi) = \sum_{s=1}^t \ln \langle \pi_s, f_s \rangle$

- RL interpretation: $\ln \mathcal{V}_t = t \cdot \frac{1}{t} \sum_{s=1}^t \ln \langle \pi_s, f_s \rangle$; $\frac{1}{t} \sum_{s=1}^t \ln \langle \pi_s, f_s \rangle \approx \mathbb{E}[\ln \langle \pi, f \rangle]$; objective: maximize expected log-reward

Reinforcement Learning Framework

- RL setting: repeated decisions; model-free; examples: games, robotics, autonomous control
- Time instants: $\mathcal{T}$
- State at time t: allocation $\pi_t \in \mathcal{S}_K$; portfolio value $\mathcal{V}_t(\pi)$
- Actions: next allocation $\pi_{t+1} \in \mathcal{S}_K$
- Rewards: r_t depending on previous actions; not necessarily deterministic
- Strategy: probability law on $\mathcal{S}_K$ (policy)
- RL objective: choose π_{t+1} to maximize $\mathbb{E}[r_t]$
- Policy-gradient viewpoint ([Sutton and Barto, 2018]); No discounting of rewards
- Natural reward choice: $r_t = \ln\langle \pi_t, f_t \rangle$ (log-return)
- Literature link: [Helmbold et al., 1998] uses multiplicative updates to maximize expected log-return
- Our extension: add transaction-cost term to r_t

Softmax Parameterization and Rewards

- Iterative update: from $(\pi_t, \mathcal{M}_t)$ to π_{t+1} with improved expected reward
- Proximity constraint: π_{t+1} close to π_t (gradient-flow rationale)
- Softmax map: $S(H)(k) = \frac{e^{H_k}}{\sum_\ell e^{H_\ell}}$, $H \in \mathbb{R}^K$
- Possible short selling: $\pi_\lambda = (1+\lambda)S(H) - \lambda/K$
- Reward function: $F_t(H) = \ln\langle S(H), f_t\rangle$
- Transaction-fee motivation: H_{t+1} close to H_t

Transaction Costs and Drifted Allocation

- Proportional fee: $\xi V \sum_k |\tilde{\pi}(k) - \pi(k)|$
- Drift after price move: $\pi_{t+} = \frac{\pi_t \odot f_t}{\langle \pi_t, f_t \rangle}$
- Rebalancing effect: multiply wealth by $\exp(-\xi \sum_k |\pi_{t+}(k) - \pi(k)|)$
- Smooth penalty: pseudo-Huber $\sqrt{x^2 + a^2} - a$ (stability, differentiability)
- Fee term: $G_t(H) = \xi \sum_k \left(\sqrt{(S(H)(k) - \pi_{t+}(k))^2 + a^2} - a \right)$
- Objective: minimize $\mathcal{F}_t(H) = G_t(H) - F_t(H)$

Gradient-Flow ODE and Onflow Update I

General idea to minimize $F(x)$ from x_n:

• **Taylor/EE** $F(x+\delta x) = F(x) + \delta x \cdot \nabla F(x) + o(\|\delta x\|)$;

best decrease for a given effort (norm): $\delta x = -h\nabla F(x)$, obtain $x_{n+1} = x_n - h\nabla F(x_n)$, explicit Euler scheme for $x' = -\nabla F(x)$

• **gradient flow view / IE:** $F : \mathbb{R}^d \to \mathbb{R}$ = a smooth convex function, $\bar{x} \in \mathbb{R}^d$; gradient flow from $\bar{x}$ = a curve $(x_t)_{t\geq 0}$: $x_t' = -\nabla F(x_t)$ for $t > 0$, $x_0 = \bar{x}$.

• Polish metric space $(\mathcal{X}, d)$, functional $F : (\mathcal{X}, d) \to \mathbb{R} \cup \{+\infty\}$: non-trivial definition, huge literature (cf. books by Ambrosio et al. , Villani, Santambroggio)

[Villani, 2009, Ambrosio and Gigli, 2013]...

$\mathcal{X}$= $\mathcal{P}_2(\mathbb{R})$ (the set of probability measures on $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$ with finite second-order moment, endowed with the Wasserstein distance $\mathcal{W}_2$)

• **Gradient flows: the JKO scheme**

Gradient-Flow ODE and Onflow Update II

• Jordan, Kinderlehrer and Otto '98, (JKO) numerical scheme: time step $= \tau > 0$, $x_0^\tau = \bar{x} \in \mathcal{X}$, by recurrence $x_{n+1}^\tau =$ a minimizer of the functional

$$x \mapsto P_F^{JKO}(x; x_n^\tau, \tau) := \frac{1}{2\tau}\, d^2(x_n^\tau, x) + F(x). \tag{11}$$

• If $\mathcal{X}=$ Hilbert, $F=$ smooth, JKO $=$ implicit Euler (IE) scheme, i.e., $\frac{x_{n+1}^\tau - x_n^\tau}{\tau} = -\nabla F(x_{n+1}^\tau)$.
• JKO scheme was initially used theoretically to prove the existence of a gradient flow (see [Legendre and Turinici, 2017, Laguzet, 2018] for higher order schemes).
implicit Euler scheme for $x' = -\nabla F(x)$
• **Conclusion:** both EE / IE-JKO converge to solution of $x' = -\nabla F(x)$ when $h \to 0$ / $\tau \to 0$;
the implicit gradient flow is sometimes preferred when geometry is not linear

Gradient-Flow ODE and Onflow Update

- Gradient-flow step: solve $\frac{d}{du}\mathcal{H}(u) = -\nabla_H \mathcal{F}_t(\mathcal{H}(u))$, $\mathcal{H}(0) = H_t$
- Update: $H_{t+1} = \mathcal{H}(\tau)$, $\pi_{t+1} = S(H_{t+1})$
- Softmax derivative: $\partial_{H_b} S(H)(k) = S(H)(k)(\mathbf{1}_{k=b} - S(H)(b))$
- Full ODE: drift term $S(\mathcal{H}) \odot f_t / \langle S(\mathcal{H}), f_t \rangle - S(\mathcal{H})$
- Algorithm: read f_t; compute π_{t+}; solve ODE; set H_{t+1}; output π_{t+1}

Theoretical Convergence Results — Setup

- Continuous limit: $\mathcal{T} = \mathbb{R}_+$; no transaction fees ($\xi = 0$)
- Log-normal dynamics: $\frac{dS_t^k}{S_t^k} = \mu_k\, dt + \sum_\ell \sigma_{k\ell}\, dz_\ell(t)$, Covariance: $\Sigma = \sigma^T \sigma$
- Objective: maximize $\frac{d}{dt}\mathbb{E}[\ln \mathcal{V}_t(\pi)]$
- Reward functional: $R(\pi) = \langle \mu, \pi \rangle - \frac{1}{2}\, \pi^T \Sigma \pi$
- Softmax policy: $\pi_t = S(H_t)$, Gradient-flow ODE: $\dot{H}(t) = \nabla_H R(S(H(t)))$, output allocation: $\pi_t = S(H(t))$
- Goal: find minimum w/o knowing statistics of the return time series

Theoretical Convergence Results — Convergence

Proposition

Proposition

Assume that Σ is non-singular. Then :

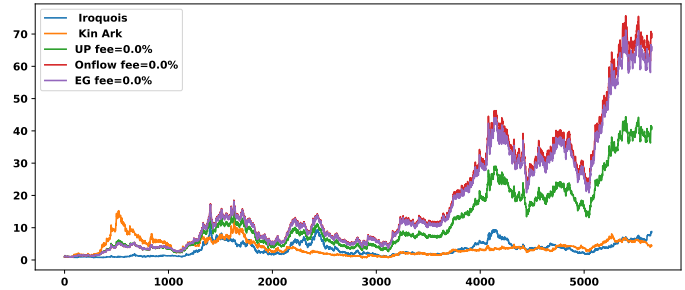
1. *maximization problem of $R(\pi)$ has a unique solution $\pi^\star \in \mathcal{S}_K$;*
2. *the reward $R_t = R(\pi_t)$ is monotonically increasing;*
3. *The output allocation $(\pi_t)_{t\geq 0}$ converges, we denote $\pi^\infty := \lim_{t\to\infty} \pi_t$*
4. *Local convergence: $\exists c_\epsilon = c_\epsilon(\Sigma, \mu)$ such that if $\|\pi_0 - \pi^\star\|_\Sigma \leq c_\epsilon$ then $\lim_{t\to\infty} \pi_t = \pi^\star$;*
5. *Global convergence: for general π_0 if $\pi^\infty \in \mathring{\mathcal{S}}_K$ then $\pi^\star \in \mathring{\mathcal{S}}_K$ and $\lim_{t\to\infty} \pi_t = \pi^\star$. Exponential convergence : $\exists c_0, c_1 > 0$ such that : $\|\pi_t - \pi^\star\| \leq c_0 e^{-c_1 t}$, $\forall t \geq 0$.*

Implementation Setup

- Inputs: dataset; fee level ξ; parameter τ; ODE solver (SciPy `odeint`; default settings; robust)
- Dataset: Old NYSE (36 stocks, 1965–1987, 5651 prices)
- Test pairs: four asset pairs
- Data features: heavy tails; positive excess kurtosis (p-value ≤ 0.001)
- Fee scenarios: $\xi = 0$ and $\xi = 2\%$
- Compared portfolios: individual assets; UP; EG ($\eta = 0.05$); Onflow
- Onflow parameters: $\tau = 0.05$ for $\xi = 0$; $\tau = 1$ for $\xi = 2\%$

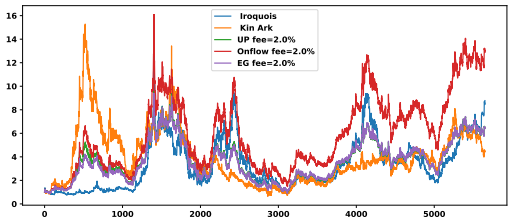
Irocquois–Kin Ark (Fee 0%)

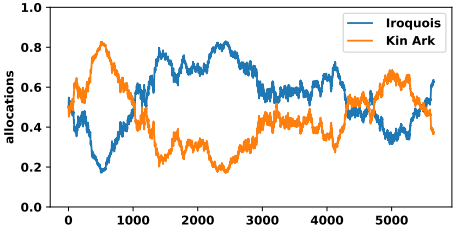
- Individual growth: 8.92 and 4.13
- UP: $\sim 40\times$ initial wealth; EG and Onflow: $\sim 70\times$ initial wealth
- Interpretation: dynamic strategies outperform UP and individual assets



Irocquois–Kin Ark (Fee 2%)

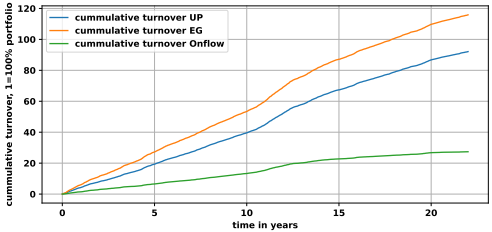
- UP and EG collapse under fees; near buy-and-hold performance
- Onflow remains profitable; stable under high transaction costs
- Parameter: $\tau = 1$





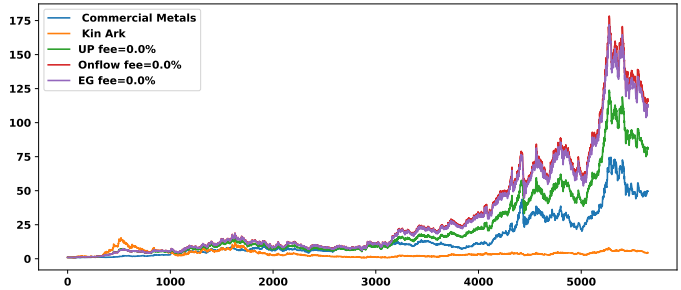
Turnover Analysis

- Turnover metric: $\sum_t |\pi_{t+1} - \pi_{t+}|$
- Fee 0%: UP, EG, Onflow $\sim$ 2% daily turnover
- Fee 2%: UP and EG unchanged; Onflow reduces to $\sim$ 0.5%
- Annualized: 2% daily $\approx$ 500% yearly; 0.5% daily $\approx$ 125%
- Total (22 years): UP/EG $\sim 100\times$ portfolio value; Onflow $\sim 25\times$
- Explanation: lower turnover $\rightarrow$ lower fees $\rightarrow$ preserved performance



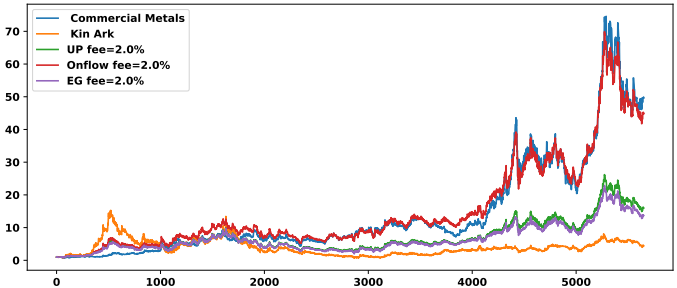
Commercial Metals – Kin Ark (Fee 0%)

- Individual assets: $< 50\times$ initial wealth;
- UP: $\sim 80\times$; EG and Onflow: $\sim 110\times$
- Interpretation: dynamic strategies outperform UP and individuals



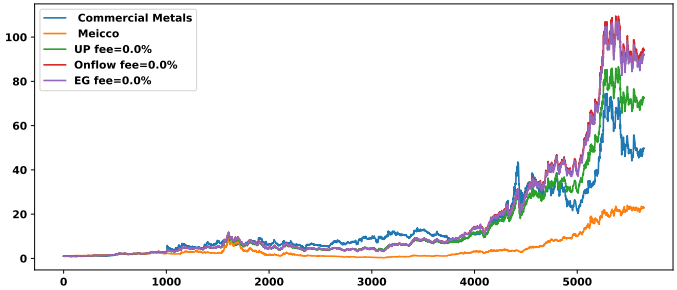
Commercial Metals – Kin Ark (Fee 2%)

- UP and EG degrade to $\sim 15\times$
- Onflow retains $\sim 50\times$ initial wealth
- Reason: Commercial Metals dominates $\rightarrow$ Onflow tracks it
- Allocation: post-$t \approx 1000$, weight on Commercial Metals often $> 80\%$



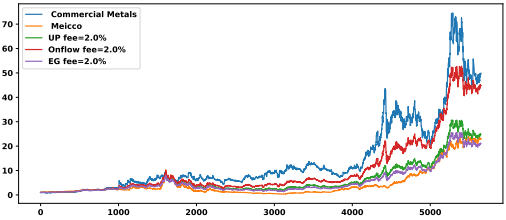
Commercial Metals – Meicco (Fee 0%)

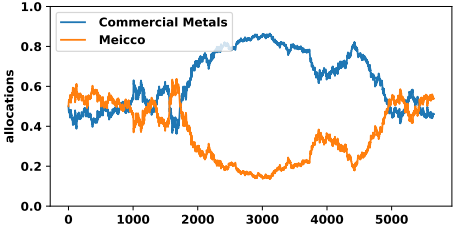
- Commercial Metals dominates the pair
- Limited room for improvement over buy-and-hold
- Onflow remains competitive with UP and EG



Commercial Metals – Meicco (Fee 2%)

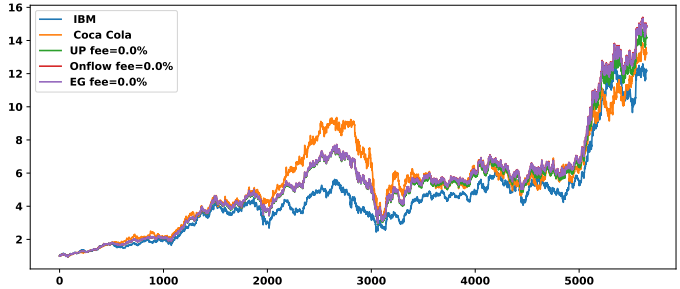
- UP and EG degrade under fees
- Onflow remains competitive due to low turnover
- Dominance of Commercial Metals limits dynamic gains





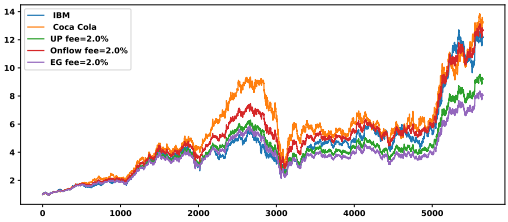
IBM – Coca Cola (Fee 0%)

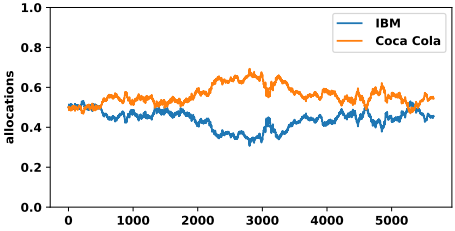
- Dynamic strategies offer no advantage
- All portfolios comparable to individual assets
- Low volatility + high correlation → limited rebalancing gains



IBM – Coca Cola (Fee 2%)

- UP and EG fall below individual assets
- Onflow remains comparable to buy-and-hold
- Interpretation: Onflow's low turnover protects performance



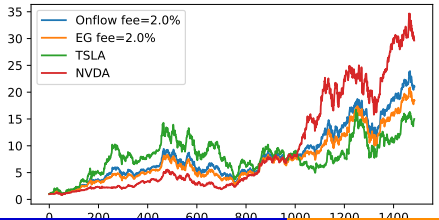


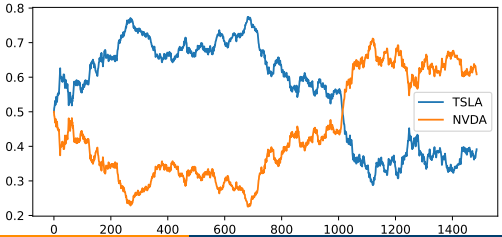
Remark on Kurtosis

- Tests performed on real (non log-normal) data
- Heavy tails present in all pairs
- Empirical finding: kurtosis variations do not bias allocation
- Over-allocation driven by long-term performance, not kurtosis

High-Volatility Assets: NVDA – TSLA

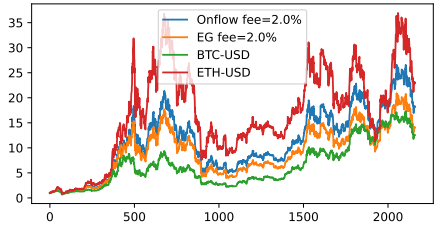
- Dataset: Jan 2020 – Nov 2025; daily quotes; $\xi = 2\%$, $\tau = 1$
- Assets: Nvidia (NVDA), Tesla (TSLA)
- Sharpe ratios: Onflow 1.21; EG 1.19; TSLA 0.99; NVDA 1.32
- NVDA dominates long-term; TSLA stronger early (up to day 1000)
- Onflow tracks regime shifts: TSLA $\rightarrow$ NVDA as NVDA surges
- Fee impact: Onflow 22.32 $\rightarrow$ 21.04; EG 27.67 $\rightarrow$ 18.46
- Turnover: Onflow 0.2% daily; EG 1.36% daily





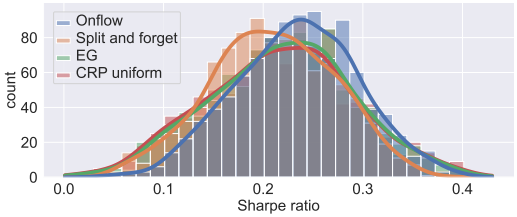
High-Volatility Assets: BTC – ETH

- Assets: Bitcoin (BTC-USD), Ethereum (ETH-USD), 2020–2025; $\xi = 2\%$, $\tau = 1$
- Sharpe ratios: Onflow 0.86; EG 0.82; BTC 0.83; ETH 0.87
- Onflow performance: 18.67 $\rightarrow$ 18.19 under fees; EG performance: 19.86 $\rightarrow$ 14.04 under fees
- Turnover: Onflow 0.06% daily; EG 0.8% daily
- Allocation behavior: Onflow under-allocates ETH late in period due to rapid decay
- Interpretation: protective shift in highly volatile regime



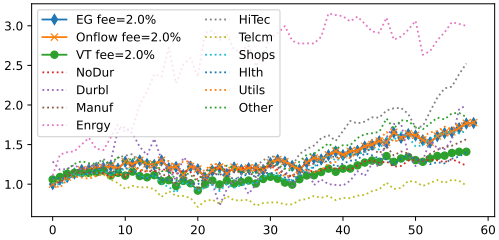
Sharpe Ratio Tests with 10-Asset Samples

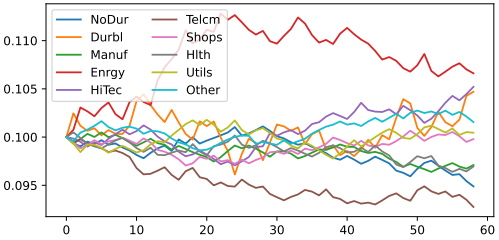
- Dataset: same universe as before; random samples of 10 assets; 1000 independent draws; $\xi = 1\%$
- Strategies: Onflow $\tau = 2.5$; EG; uniform CRP; Split-and-Forget
- Sharpe ratio: mean return minus risk-free rate (8%) over std. dev.
- Goal: compare risk-adjusted performance across strategies
- Result: Onflow not always best per sample but statistical test: Wilcoxon $\rightarrow$ Onflow (median 0.237) significantly superior (p-value $< 10^{-3}$) than EG: 0.221, CRP uniform: 0.218, Split-and-Forget: 0.209; performance dispersion increases with more assets



Comparison with Low-Turnover Approaches

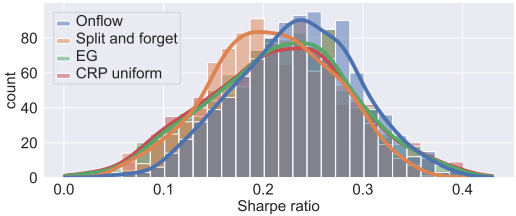
- Benchmark: Volatility Timing strategy from [Kirby and Ostdiek, 2012]
- Dataset: 10 Fama–French industry portfolios, 2020–2025, monthly data, $\xi = 2\%$
- Goal: compare low-turnover mean–variance methods with RL-based Onflow
- Onflow ($\tau = 1$) competitive Sharpe ratio; not beating best sector; monthly data scarcity limits dynamic strategies; VT strategy could improve with tuned η; default $\eta = 1$ used
- Turnover: Onflow low; comparable to other low-turnover methods





Performance Metrics: 10 FF Industries

- Sector Sharpe ratios: range 0.25 (Telcm) to 4.55 (HiTec)
- Sector turnover: 0% (buy-and-hold)
- Volatility Timing: performance 1.59; Sharpe 2.34; turnover 10.68%
- EG: performance 1.84; Sharpe 3.99; turnover 3.21%
- Onflow: performance 1.84; Sharpe 4.01; turnover 2.87%
- Interpretation: Onflow competitive with EG; both outperform VT after fees
- Note: Onflow does not beat best individual sector but maintains low turnover



Conclusions

- Onflow: online, model-free portfolio allocation; RL-style updates
- Softmax parameterization; gradient-flow ODE; transaction-cost term
- Theory (log-normal, continuous time, true gradients): convergence to optimal allocation
- Empirics: competitive with UP and EG at $\xi = 0$; significantly better under high fees (2%)
- Strengths: robustness to model risk; low turnover; strong performance under costs
- Limitations / future work: short positions; comparison with ML predictors (LSTM); other markets and periods

Ambrosio, L. and Gigli, N. (2013).
Modelling and Optimisation of Flows on Networks: Cetraro, Italy 2009, Editors: Benedetto Piccoli, Michel Rascle, chapter A User's Guide to Optimal Transport, pages 1–155.
Springer Berlin Heidelberg, Berlin, Heidelberg.

Black, F. and Litterman, R. (1990).
Asset allocation: combining investor views with market equilibrium.
Goldman Sachs Fixed Income Research, 115(1):7–18.

Blum, A. and Kalai, A. (1997).
Universal portfolios with and without transaction costs.
In *Proceedings of the Tenth Annual Conference on Computational Learning Theory*, pages 309–313.

Borodin, A., El-Yaniv, R., and Gogan, V. (2003).
Can we learn to beat the best stock.
In Thrun, S., Saul, L., and Schölkopf, B., editors, *Advances in Neural Information Processing Systems*, volume 16, pages 344–352. MIT Press.

Cesa-Bianchi, N. and Lugosi, G. (2006).
Prediction, learning, and games.
Cambridge University Press.

Cover, T. M. (1991).
Universal portfolios.
Mathematical Finance, 1(1):1–29.

Györfi, L., Ottucsák, G., and Walk, H. (2012).
Machine Learning for Financial Engineering.
Imperial College Press.

He, H. and Li, H. (2024).
A New Boosting Algorithm for Online Portfolio Selection Based on dynamic Time Warping and Anti-correlation.
Computational Economics, 63(5):1777–1803.

Helmbold, D. P., Schapire, R. E., Singer, Y., and Warmuth, M. K. (1998).
On-line portfolio selection using multiplicative updates.
Mathematical Finance, 8(4):325–347.

Kelly Jr, J. L. (1956).
A new interpretation of information rate.
Bell System Technical Journal, 35(4):917–926.

Kirby, C. and Ostdiek, B. (2012).
It's All in the Timing: Simple Active Portfolio Strategies that Outperform Naïve Diversification.
The Journal of Financial and Quantitative Analysis, 47(2):437–467.
Publisher: Cambridge University Press.

Laguzet, L. (2018).
High order variational numerical schemes with application to Nash–MFG vaccination games.
Ricerche di Matematica, 67:247–269.

Legendre, G. and Turinici, G. (2017).
Second-order in time schemes for gradient flows in wasserstein and geodesic metric spaces.
Comptes Rendus Mathematique, 355(3):345–353.

Li, B. and Hoi, S. C. (2014).
Online portfolio selection: A survey.
ACM Computing Surveys (CSUR), 46(3):1–36.

Li, B., Hoi, S. C. H., Sahoo, D., and Liu, Z.-Y. (2015).
Moving average reversion strategy for on-line portfolio selection.
Artificial Intelligence, 222:104–123.

Li, B., Sahoo, D., and Hoi, S. C. (2016).
Olps: a toolbox for on-line portfolio selection.
The Journal of Machine Learning Research, 17(1):1242–1246.

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

Ngo, V. M., Nguyen, H. H., and Nguyen, P. V. (2023).
Does reinforcement learning outperform deep learning and traditional portfolio optimization models in frontier and developed financial markets?
Research in International Business and Finance, 65:101936.

Raissi, M., Perdikaris, P., and Karniadakis, G. (2019).
Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations.
Journal of Computational Physics, 378:686–707.

Sato, Y. (2019).
Model-free reinforcement learning for financial portfolios: A brief survey.
arXiv:1904.04973.

Sun, S., Wang, R., and An, B. (2023).
Reinforcement Learning for Quantitative Trading.
ACM Trans. Intell. Syst. Technol., 14(3):1–29.
Place: New York, NY, USA Publisher: Association for Computing Machinery.

Sutton, R. S. and Barto, A. G. (2018).
Reinforcement learning. An introduction.
Adapt. Comput. Mach. Learn. Cambridge, MA: MIT Press, 2nd expanded and updated edition edition.

Turinici, G. (2024a).
Lyapunov weights to convey the meaning of time in physics-informed neural networks.
arxiv:2407.21642.

Turinici, G. (2024b).
Optimal time sampling in physics-informed neural networks.
arxiv:2404.18780, ICPR 2024 accepted paper.

Villani, C. (2009).

Optimal transport, Old and new, volume 338.
Springer-Verlag, Berlin.

Wang, S., Sankaran, S., and Perdikaris, P. (2024).
Respecting causality for training physics-informed neural networks.
Computer Methods in Applied Mechanics and Engineering, 421:116813.
arXiv preprint arXiv:2203.07404.

Zhang, H., Jiang, Z., and Su, J. (2021a).
A Deep Deterministic Policy Gradient-based Strategy for Stocks Portfolio Management.
In *2021 IEEE 6th International Conference on Big Data Analytics (ICBDA)*, pages 230–238.

Zhang, Y., Lin, H., Yang, X., and Long, W. (2021b).
Combining expert weights for online portfolio selection based on the gradient descent algorithm.
Knowledge-Based Systems, 234:107533.