RESIDUAL JUMP FLOW MATCHING FOR EXTREME-EVENT TIME-SERIES GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

In linear flow matching, the large increments that define time-series extremes share one transport target with background variation and can be underrepresented by a finite velocity model. We introduce Residual Jump Flow Matching (RJFM), centered on a cumulative-residual probability path that separates large increments from a base trajectory and transports their persistent effect later in generative time. The path preserves both endpoints while changing the regression geometry; we quantify its maximum displacement from linear interpolation and its additional kinetic action. A structured temporal velocity with base and cumulative-increment branches provides an architectural refinement of this path design. Across nine real time series, RJFM improves tail and extreme-increment fidelity over capacity-matched flow baselines and diffusion models.

1 INTRODUCTION

Extreme events occupy a small fraction of a time series but can dominate the scenarios that a generator must reproduce. Electricity-price spikes, pollution episodes, and volatility bursts may have little effect on average statistics while determining the largest values and increments. A model that captures typical windows can therefore still misrepresent the tails.

A large increment has a distinctive geometry. It is local in the difference sequence but shifts every subsequent value in the original sequence. A generator must represent both within-regime variation and the persistent level change induced by an event.

Standard linear flow matching (FM) gives each noise–data pair a constant target velocity $x_1 - x_0$ (Lipman et al., 2023). On an event window, that target combines background variation and the persistent level shift throughout generative time. This does not limit the population FM construction, but it leaves a finite velocity network to learn a rare persistent shift alongside common variation.

This observation motivates Residual Jump Flow Matching (RJFM), a conditional path designed around the geometry of large sequence increments. RJFM uses a soft training-data threshold to extract large increments, accumulates their effect into a residual R, and defines the remaining target as a base signal $B = x_1 - R$. The base follows the linear schedule t, whereas the residual follows t^α. When $\alpha > 1$, the residual is transported later than the base while the path still reaches x_1 exactly. The term "jump" refers to a level change along the sequence coordinate; the generative dynamics remain a continuous ODE.

The path is the main contribution. A conventional temporal network can learn it; a two-head velocity field that mirrors the base–residual decomposition provides an architectural refinement. We characterize the path's maximum departure from linear interpolation and its added kinetic action, then evaluate path-only and structured models against linear FM controls and external diffusion baselines.

On nine real univariate series, RJFM lowers tail-99 and tail-95 error relative to a parameter-matched linear FM on eight series. It also has lower mean error than Diffusion-TS in 100 of 108 series–metric comparisons and leads DLPM on the 27 designated primary tail comparisons. The path-only control accounts for much of the gain; Stock and Energy experiments probe how the design behaves on multivariate windows.

Our contributions are:

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.

- An endpoint-preserving FM path that represents large increments as cumulative residuals and schedules them later than the base signal, together with a characterization of its delay and kinetic action.
- A structured temporal velocity model that mirrors the residual decomposition while retaining a standard FM regression objective and ODE sampler.
- A controlled evaluation on real univariate and multivariate series that separates path design from model capacity and compares against heavy-tailed and time-series diffusion baselines.

2 Background and Related Work

Flow matching and path design. Continuous normalizing flows transport a source distribution with a neural ODE (Chen et al., 2018). FM learns the velocity by regression along prescribed conditional paths (Lipman et al., 2023). Work on couplings, stochastic interpolants, and general path families shows that this choice changes the learning problem (Tong et al., 2024; Albergo et al., 2025; Shaul et al., 2025). RJFM specializes the path to time-series events: a large increment creates a persistent displacement across later sequence positions, and the path assigns that displacement its own schedule.

Time-series generation. Existing generators use adversarial objectives, causal transport, contrastive learning, or vector-quantized representations (Yoon et al., 2019; Jeon et al., 2022; Xu et al., 2020; Jarrett et al., 2021; Lee et al., 2023). Diffusion-TS combines a transformer, trend–season decomposition, sample reconstruction, and a Fourier loss (Yuan & Qiao, 2024). These approaches chiefly change the model or its objective. We instead change the conditional FM path and use a conventional dilated TCN (Bai et al., 2018) to measure the effect of that choice.

Heavy tails and jump dynamics. Gaussian diffusion and score-SDE models (Ho et al., 2020; Song et al., 2021) differ from LIM and DLPM, which use Lévy or stable-process constructions for heavy-tailed generation (Yoon et al., 2023; Shariatian et al., 2025). Classical jump-diffusion and neural jump models explicitly represent stochastic jumps (Merton, 1976; Applebaum, 2009; Jia & Benson, 2019; Gao et al., 2025). Generator Matching extends conditional-to-marginal training to Markov generators that include genuine jump processes (Holderrieth et al., 2025). RJFM makes no such process assumption: it uses observed increments to shape a continuous-ODE path without event labels.

3 Residual Jump Flow Matching

3.1 Overview and design principle

The central design principle is to align the conditional path with the cumulative geometry of abrupt sequence changes. RJFM implements this principle in three operations. It first identifies increments that are large relative to the training distribution. It then accumulates each selected increment over all subsequent positions, producing a step-shaped residual R and a base target $B = x_1 - R$. Finally, it transports B with schedule t and R with schedule t^α. Both schedules equal one at $t = 1$, so the decomposition changes the path but not its target endpoint.

Consider a window whose first half fluctuates near zero and whose second half fluctuates near level c. Its one large midpoint increment yields a residual that is near zero before the event and c afterward; the base keeps the within-level variation. Linear FM transports the full step from $t > 0$. With $\alpha > 1$, RJFM uses $t^\alpha < t$ for $0 < t < 1$, so the step develops later than the base. The model therefore sees ordinary variation and the persistent event on different temporal schedules while learning the same data endpoint. Figure 1 summarizes the construction, training objective, and ODE sampler.

Table 1 summarizes the notation used below.

3.2 Cumulative-residual probability path

Let $x_1 \in \mathbb{R}^d$ denote a target window and draw $x_0 \sim \mathcal{N}(0, I_d)$. Linear conditional FM defines

$$\bar{x}_t = (1-t)x_0 + tx_1, \qquad u^\star_{\text{lin}} = x_1 - x_0, \tag{1}$$

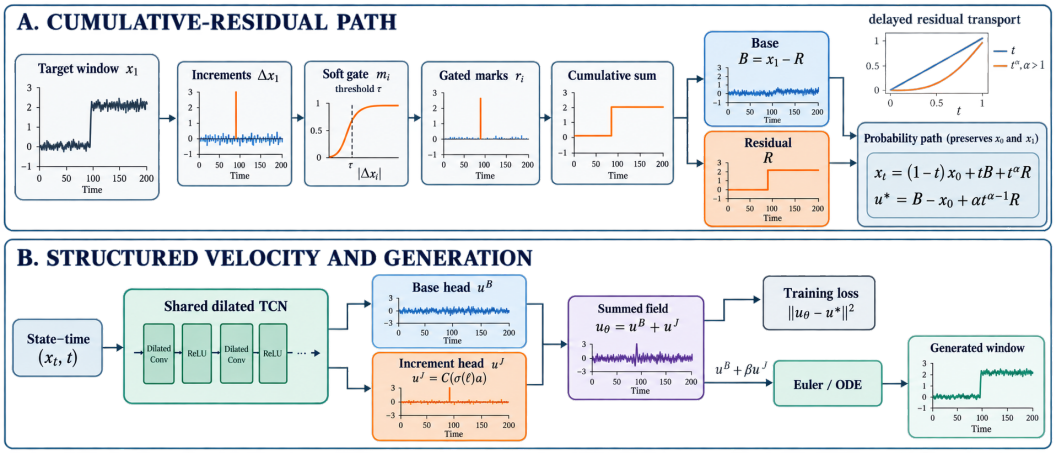


Figure 1: Visual overview of RJFM. The top row constructs the cumulative residual path: large increments are softly gated, accumulated into the residual R, and delayed through t^α while the base component B follows linear transport. The bottom row uses a shared dilated temporal network with base and increment heads, combines their velocity fields, and integrates the resulting ODE to generate a window. The diagrams describe shifts along sequence coordinates; generative time remains continuous.

Table 1: Core notation for the residual path and structured velocity.

Symbol	Meaning	Symbol	Meaning
B	base target $x_1 - R$	R	cumulative gated residual
q	training quantile level	τ	threshold from training increments
α	residual path exponent	β	jump-branch sampling weight
m_i	target increment gate	ℓ_i, a_i	predicted gate logit and mark

for $t \sim \mathcal{U}[0,1]$. To expose persistent events, define the target increment $\Delta x_{1,i} = x_{1,i} - x_{1,i-1}$ for $i = 2, \ldots, d$, and let τ be the empirical q-quantile of $|\Delta x|$ computed only from training windows. The soft-gated increment is

$$m_i = \sigma\left(\frac{|\Delta x_{1,i}| - \tau}{\rho\tau}\right), \qquad r_i = m_i \Delta x_{1,i}, \tag{2}$$

with $\rho = 0.1$. The gate approaches zero for increments below the training threshold and one for increments above it. Accumulating the gated increments gives the residual R, and subtracting that residual gives the base target B:

$$R_1 = 0, \qquad R_k = \sum_{i=2}^{k} r_i, \qquad B = x_1 - R. \tag{3}$$

The residual conditional path and its velocity are

$$x_t = (1-t)x_0 + tB + t^\alpha R, \tag{4}$$

$$u^\star(x_t, t) = B - x_0 + \alpha t^{\alpha-1} R. \tag{5}$$

The path starts at x_0 and ends at $B + R = x_1$. Its linear component transports the base target, while $t^\alpha R$ restores the cumulative event residual. For $\alpha > 1$, residual displacement grows more slowly than linear interpolation for $0 < t < 1$, and the velocity factor $\alpha t^{\alpha-1}$ concentrates residual transport toward the endpoint. Thus, α controls when the event structure enters the regression target.

3.3 Structured velocity field and learning objective

A shared dilated TCN feeds a base head $u_\theta^B \in \mathbb{R}^d$ and a cumulative-increment head. The base head predicts coordinate-wise velocity. At increment position i, the second head predicts a gate logit ℓ_i and

a signed mark a_i; cumulative summation converts these local predictions into a persistent velocity contribution:

$$\widehat{r}_i^u = \sigma(\ell_i)a_i, \qquad u_{\theta,1}^J = 0, \qquad u_{\theta,k}^J = \sum_{i=2}^{k} \widehat{r}_i^u. \tag{6}$$

Training supervises only the summed field $u_\theta = u_\theta^B + u_\theta^J$:

$$\mathcal{L}_{\mathrm{RJFM}} = \mathbb{E}_{x_0,x_1,t}\big[\|u_\theta(x_t,t) - u^\star(x_t,t)\|_2^2\big]. \tag{7}$$

No branch-specific auxiliary loss is required. Consequently, the cumulative head should be interpreted as an architectural inductive bias rather than an identifiable estimate of a latent jump process. Euler sampling uses

$$x_{t+h} = x_t + h\big(u_\theta^B(x_t,t) + \beta u_\theta^J(x_t,t)\big). \tag{8}$$

where β controls the cumulative head at sampling time. Validation selects $q = 0.95$, $\alpha = 1.5$, $\beta = 0.5$, and 64 Euler steps. Algorithm 1 summarizes training and generation; the complete procedure is given in Appendix J, Algorithm J.1.

Algorithm 1 RJFM training and generation

Require: training threshold τ, path exponent α, sampling weight β, Euler steps N
1: Draw $x_1, x_0 \sim \mathcal{N}(0, I_d)$, and $t \sim \mathcal{U}[0,1]$
2: Compute R from the gated increments of x_1; set $B \leftarrow x_1 - R$
3: $x_t \leftarrow (1-t)x_0 + tB + t^\alpha R$; $u^\star \leftarrow B - x_0 + \alpha t^{\alpha-1}R$
4: Update θ using $\|u_\theta^B + u_\theta^J - u^\star\|_2^2$
5: Draw $x \sim \mathcal{N}(0, I_d)$; set $h \leftarrow 1/N$
6: **for** $n = 0, \ldots, N-1$ **do**
7: $\quad x \leftarrow x + h\left[u_\theta^B(x, nh) + \beta u_\theta^J(x, nh)\right]$
8: **end for**
9: **return** x

3.4 Properties of the residual path

The path geometry quantifies how residual scheduling departs from linear interpolation. Its population flow-matching interpretation follows from the standard conditional-FM argument.

Proposition 1 (Endpoint, delay, and action)**.** *For $\alpha \geq 1$, equation 4 satisfies $x_{t=0} = x_0$ and $x_{t=1} = x_1$. For $\alpha = 1$ it reduces to the linear path. For $\alpha > 1$,*

$$\max_{t\in[0,1]} \|x_t - \bar{x}_t\|_2 = (\alpha-1)\alpha^{-\alpha/(\alpha-1)}\|R\|_2, \tag{9}$$

$$\int_0^1 \|u^\star(x_t,t)\|_2^2 \,\mathrm{d}t = \|x_1 - x_0\|_2^2 + \frac{(\alpha-1)^2}{2\alpha-1}\|R\|_2^2. \tag{10}$$

Interpretation. At $\alpha = 1.5$, the maximum delay is $4\|R\|_2/27$ and the excess action is $\|R\|_2^2/8$. Proposition 1 therefore makes the tradeoff explicit: delaying the residual changes the regression geometry and increases kinetic action by a known amount.

Population flow-matching target. For the random training path X_t with $U_t = \dot{X}_t$, squared-loss regression yields $v_t^\star(x) = \mathbb{E}[U_t \mid X_t = x]$ (Lipman et al., 2023). Under the stated regularity conditions, this field satisfies the continuity equation, and its ODE transports p_0 to p_1. Residual scheduling thus preserves the ideal terminal distribution while changing the velocity target learned by a finite model. The formal statement and proof appear in Appendix I, Proposition I.2.

To make the edge indexing explicit, define $D : \mathbb{R}^d \to \mathbb{R}^{d-1}$ and $C : \mathbb{R}^{d-1} \to \mathbb{R}^d$ by

$$(Dx)_\ell = x_{\ell+1} - x_\ell, \qquad (Cz)_1 = 0, \qquad (Cz)_k = \sum_{\ell=1}^{k-1} z_\ell, \tag{11}$$

for $\ell = 1, \ldots, d-1$ and $k = 2, \ldots, d$. With $\widetilde{m}_\ell = m_{\ell+1}$, the decomposition satisfies

$$R = C(\widetilde{m} \odot Dx_1), \qquad D(Cz) = z. \tag{12}$$

A nonzero edge coordinate z_ℓ induces a level shift from sequence position $\ell + 1$ onward, which is precisely the cumulative geometry targeted by the method. For increment prediction error e, cumulative error obeys $\|Ce\|_2 \leq \sqrt{d(d-1)/2}\,\|e\|_2$; this bound motivates zero-centered initial predictions for the cumulative head. Complete proofs and the effect of $\beta \neq 1$ appear in Appendix I.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

The evaluation separates the effect of the probability path from model capacity and architectural structure. Linear-path TCN-FM provides the direct FM reference, and a width-matched version controls for parameter count. A single-head model trained on the residual path isolates the contribution of path design, while the full model measures the additional effect of the structured velocity. DLPM, Diffusion-TS, and two multivariate benchmarks place these comparisons against external generative models and standard time-series data.

Univariate data. The benchmark contains hourly electricity prices for Germany, Spain, Italy, and the United Kingdom from Ember (Ember, 2026); 2024 $PM_{2.5}$ records from OpenAQ locations 8118 and 8128 (OpenAQ, 2026); and five-minute realized variance for .SPX, .RUT, and .STOXX50E from Oxford-Man (Heber et al., 2013). Each timeline is split chronologically into 70/10/20 percent train, validation, and test segments, separated by a 64-point gap. Length-64 windows use training stride 8 and evaluation stride 1; all splits use training-only normalization.

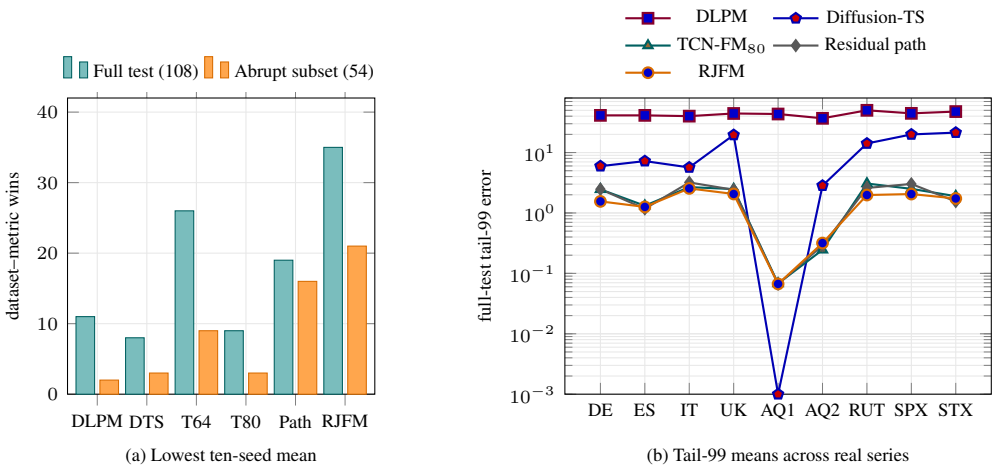
Baselines and controls. TCN-FM_{64} is the linear-path FM baseline. TCN-FM_{80} widens the same model to 165,201 parameters, close to RJFM's 156,867. Residual-$Path_{80}$ uses the proposed path with the wide single-head TCN, so it isolates path design from the two-branch network. DLPM follows the symmetric α-stable process of Shariatian et al. (2025); validation over $\alpha \in \{1.5, 1.8, 1.9\}$ selects $\alpha = 1.9$, with 100 steps and unclipped stable noise. Diffusion-TS uses its transformer, trend–season decomposition, reconstruction objective, and Fourier loss (Yuan & Qiao, 2024). We use two encoder and decoder layers, width 64, four attention heads, a 500-step cosine diffusion, 64-step DDIM sampling, and training-only min–max scaling. This model has 309,645 parameters. Validation selects all configurations before test evaluation.

Each model trains for 1,500 updates with batch size 256. The flow models and DLPM use learning rate 10^{-3}; Diffusion-TS uses linear warmup from 10^{-5} to 8×10^{-4} over 500 updates, Adam $(0.9, 0.96)$, gradient clipping at 1, and exponential moving average decay 0.995. Every run generates 2,048 samples. Reported results average ten seeds (100–109). Exact architectures, selection tables, and window counts appear in Appendices A–B and J.

Evaluation. The three primary metrics correspond directly to the problem statement. Tail-99 error measures the mismatch in extreme absolute values. Wasserstein-1 distance between per-window maximum absolute increments compares the distribution of each window's largest event. Expected-shortfall-95 error compares the mean increment magnitude beyond the reference 95th-percentile threshold. Nine secondary metrics measure bulk distribution, quantiles, jump rate and size, autocorrelation, exceedance clustering, and multiscale MMD (Gretton et al., 2012). Lower is better. The abrupt-window analysis selects the top quartile of reference and generated windows independently by maximum absolute increment; it is a marginal stress test, not conditional generation. Paired intervals use common seed labels and are reported without multiplicity adjustment as descriptive stability summaries.

4.2 EXTREME-EVENT FIDELITY ON REAL TIME SERIES

The residual path improves the primary tail metrics on most real series. RJFM reduces tail-99 and tail-95 errors relative to the parameter-matched TCN-FM_{80} on eight of nine series (Figure 2(b)). Tail-99 also improves over TCN-FM_{64} on eight series, with separated paired intervals on Germany, Spain, and the United Kingdom (Table 2). The same direction across both FM controls rules out network width as the sole explanation.



(a) Lowest ten-seed mean

(b) Tail-99 means across real series

Figure 2: Ten-seed real-data results. (a) Winner counts cover all 12 full-test metrics and the six abrupt-subset metrics; higher winner counts are better. (b) Tail-99 error is lower-is-better. DLPM uses the validation-selected $\alpha = 1.9$ configuration, and Diffusion-TS uses the fixed protocol-adapted configuration. Its exact zero tail-99 error on AQ1 is plotted at 10^{-3} for visibility on the logarithmic axis. T64/T80 denote TCN-FM widths; DE/ES/IT/UK denote Ember series, AQ1/AQ2 OpenAQ series, and STX denotes .STOXX50E.

Table 2: Directional primary-tail comparisons. Each entry is the number of series on which RJFM has the lower ten-seed mean; parentheses give the number with a separated unadjusted paired 95% interval. Bold marks the largest win count in each column, including ties.

Comparator	Tail-99	Max-increment W1	ES-95
Official DLPM	**9 (9)**	**9 (9)**	**9 (9)**
Diffusion-TS	8 (8)	**9 (9)**	8 (8)
TCN-FM$_{64}$	8 (3)	5 (0)	4 (0)
TCN-FM$_{80}$	8 (2)	6 (2)	4 (0)
Residual-Path$_{80}$	5 (0)	5 (2)	4 (0)

The abrupt-window stress test gives the same result. RJFM has the lowest tail-99 mean on seven series, while Residual-Path$_{80}$ leads on Spain and .STOXX50E. The gain is therefore concentrated where the target contains large realized increments.

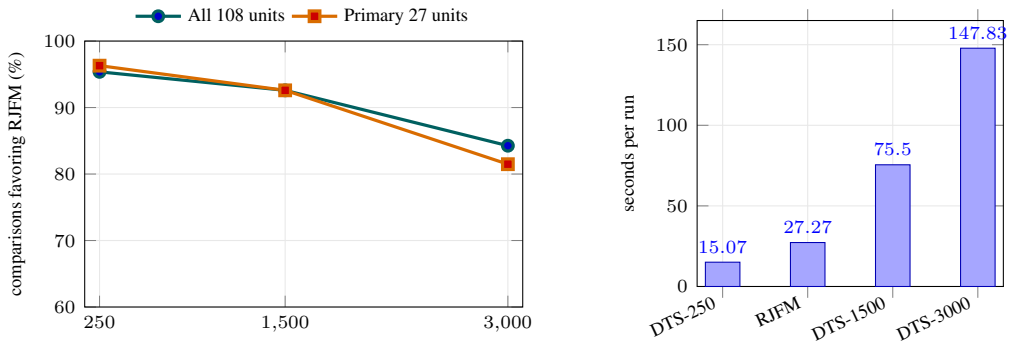
4.3 Path design accounts for most of the gain

Residual-Path$_{80}$ is the strongest ablation, attaining 19 of 108 full-test wins and 16 of 54 abrupt-subset wins. RJFM has the lower mean in 66/108 comparisons with this ablation, although most paired intervals overlap. Across all 12 metrics, RJFM records the lowest mean in 35/108 series–metric units, the highest count among the evaluated models (Figure 2(a)). The structured branch adds a smaller refinement after the path has already changed the learning target.

4.4 Comparison with external generative baselines

The advantage on extreme-event metrics persists against both external baselines. On the three primary diagnostics, all 27 per-series comparisons with DLPM favor RJFM in mean direction and unadjusted paired interval. Median error reductions are 96.2% for tail-99, 96.0% for maximum-increment W1, and 99.2% for expected-shortfall-95. DLPM is better on selected MMD statistics, so tail calibration and bulk kernel fit do not produce the same ranking.

Against Diffusion-TS, RJFM has the lower mean on 100/108 full-test series–metric units and a separated favorable paired interval on 99. It leads on 25/27 primary comparisons, including all nine maximum-increment W1 comparisons. Suite-averaged tail-99 errors for Diffusion-TS versus



(a) Robustness vs. Diffusion-TS training updates

(b) End-to-end runtime

Figure 3: Compute-aware Diffusion-TS stress test. (a) Percentage of comparisons whose ten-seed mean favors RJFM as the Diffusion-TS training budget grows; the conclusion remains stable at 3,000 updates. (b) Mean training-and-evaluation time on one RTX 5090. Every checkpoint is evaluated from the same 3,000-update Diffusion-TS trajectory; RJFM uses 1,500 updates.

Dataset	Model	SWD	Corr.	Tail co-ex.	Inc. $q_{.99}$	Disc.	Pred.
Energy	TCN-FM$_{\text{match}}$	**1.134±0.103**	0.183±0.009	**0.413±0.007**	**0.325±0.020**	**0.491±0.012**	0.618±0.016
	Residual-Path$_{\text{match}}$	1.138±0.108	**0.182±0.013**	0.422±0.009	0.337±0.022	**0.491±0.012**	**0.617±0.020**
	RJFM	1.214±0.102	0.191±0.008	0.444±0.009	0.376±0.012	0.499±0.001	0.799±0.034
	Diffusion-TS	1.415±0.130	0.364±0.017	0.821±0.018	3.371±0.257	0.499±0.003	1.493±0.139
Stock	TCN-FM$_{\text{match}}$	4.146±0.497	0.158±0.008	**0.052±0.001**	0.641±0.048	**0.499±0.003**	2.250±0.185
	Residual-Path$_{\text{match}}$	4.140±0.492	0.155±0.008	**0.052±0.001**	0.633±0.073	**0.499±0.002**	**2.243±0.187**
	RJFM	**4.064±0.440**	**0.145±0.013**	**0.052±0.001**	**0.549±0.051**	0.500±0.000	2.535±0.236
	Diffusion-TS	4.121±0.488	0.526±0.082	0.970±0.000	3.375±0.396	0.500±0.000	4.306±0.524

Table 3: Standard multivariate generation on the TimeGAN Energy and Stock benchmarks (mean ± standard deviation over ten seeds; lower is better). SWD denotes sliced Wasserstein distance, Corr. cross-feature correlation error, Tail co-ex. feature co-exceedance error, and Disc./Pred. the standard post-hoc discriminative and synthetic-to-real predictive scores. Bold marks the lowest displayed mean within each dataset and metric; rounded ties are all marked.

RJFM are 9.605 versus 1.856 on Ember, 1.417 versus 0.192 on OpenAQ, and 18.490 versus 1.925 on Oxford-Man.

Longer Diffusion-TS training improves 88/108 means between 1,500 and 3,000 updates. At 3,000 updates, RJFM still has lower means on 91/108 comparisons, including 22/27 primary diagnostics, and 81 favorable separated intervals (Figure 3(a)). The longer checkpoint averages 147.83 seconds per run, versus 27.27 seconds for RJFM. More optimization narrows the gap but does not change the ranking on the primary tail metrics.

4.5 Multivariate Generation

We extend the same protocol to the standard TimeGAN Stock and Energy data (Yoon et al., 2019): Stock contains 6 variables and 3,685 timestamps; Energy contains 28 variables and 19,735 timestamps. Raw timelines are split before constructing length-24 windows, with a 24-step gap and training-only per-feature standardization. Every model uses the same 1,500 updates, 64 sampling steps, 2,048 generated samples, and ten seeds. The flow models use multichannel temporal convolutions rather than flattening variables.

On Stock, RJFM gives the lowest sliced Wasserstein, correlation, and increment-$q_{.99}$ errors, improving over both temporal FM controls on dependency and extreme-increment diagnostics (Table 3). On Energy, the linear and path-only controls attain the lowest displayed means. RJFM nevertheless has lower displayed means than Diffusion-TS on five of the six reported metrics; their discriminative

scores coincide at the shown precision. Thus the structured model improves several Stock diagnostics but does not lead on Energy.

4.6 Efficiency

On one RTX 5090, RJFM averages 27.27 seconds for training and evaluation, compared with 14.99 seconds for TCN-FM$_{80}$, 8.10 seconds for DLPM, and 73.03 seconds for Diffusion-TS at 1,500 updates (Figure 3(b)). The residual construction costs more than linear FM, but RJFM remains 2.68 times faster than Diffusion-TS under the primary budget.

5 Discussion

Path design and velocity structure. The path-only model captures much of the improvement over linear FM; the structured velocity field adds a further gain. The geometric analysis explains the intervention: residual transport enters later in generative time, changing the regression target and adding kinetic action. The univariate experiments connect this change to improved extreme-event fidelity.

Beyond univariate windows. The Stock experiment shows that the same path design can improve multivariate dependency and extreme-increment diagnostics. On Energy, RJFM has lower displayed means than Diffusion-TS on five of six metrics. Input-dependent residual schedules and conditional generation provide natural extensions of the cumulative-path construction.

6 Conclusion

Residual Jump Flow Matching represents large time-series increments as cumulative residuals in a conditional FM path. Delaying residual transport preserves the prescribed endpoints but changes the regression target. Controlled experiments identify this path as the main source of the tail-fidelity improvement on the tested univariate series. The Stock results extend the evidence to multivariate dependency and extreme increments.

AI use statement

In this work, we used generative AI tools for language editing and polishing of the manuscript. We have not used generative AI tools for generating experimental results, theoretical proofs, or core research ideas. Additionally, we used generative AI tools for code debugging assistance during the implementation of baseline methods. We have reviewed all AI-assisted work—all AI-assisted text was verified for factual accuracy by the authors, and AI-assisted code was verified and tested for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Ethics statement

The study uses public aggregate environmental, energy-market, and financial index data and does not involve human subjects or individual-level personal records. Synthetic tail estimates should not be treated as guarantees in safety- or finance-critical decisions without domain-specific validation.

Reproducibility statement

The appendix specifies data sources, splits, normalization, architectures, selection rules, seeds, samplers, metrics, and theoretical proofs. Result archives contain per-seed summaries, aggregate outputs, logs, and analysis scripts for the univariate, convergence, and multivariate experiments.

References

Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

David Applebaum. *Lévy Processes and Stochastic Calculus*. Cambridge University Press, 2 edition, 2009.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K. Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Ember. European wholesale electricity price data. `https://ember-energy.org/data/european-wholesale-electricity-price-data/`, 2026. Accessed September 2026.

Yuanpei Gao, Qi Yan, Yan Leng, and Renjie Liao. Neural MJD: Neural non-stationary merton jump diffusion for time series prediction. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, 2012.

Gerd Heber, Asger Lunde, Neil Shephard, and Kevin Sheppard. Oxford-man institute's realized library, 2013. Library version 0.2, Oxford-Man Institute of Quantitative Finance.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Peter Holderrieth, Marton Havasi, Jason Yim, Neta Shaul, Itai Gat, Tommi Jaakkola, Brian Karrer, Ricky T. Q. Chen, and Yaron Lipman. Generator matching: Generative modeling with arbitrary markov processes. In *International Conference on Learning Representations*, 2025.

Daniel Jarrett, Ioana Bica, and Mihaela van der Schaar. Time-series generation by contrastive imitation. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Jinsung Jeon, Jeonghak Kim, Haryong Song, Seunghyeon Cho, and Noseong Park. GT-GAN: General purpose time series synthesis with generative adversarial networks. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Junteng Jia and Austin R. Benson. Neural jump stochastic differential equations. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Daesoo Lee, Sara Malacarne, and Erlend Aune. Vector quantized time series generation with a bidirectional prior model. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 7665–7693, 2023.

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

Robert C. Merton. Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics*, 3(1–2):125–144, 1976.

OpenAQ. OpenAQ global air quality data archive. `https://docs.openaq.org/aws/about`, 2026. Accessed September 2026.

Dario Shariatian, Umut Simsekli, and Alain Durmus. Denoising lévy probabilistic models. In *International Conference on Learning Representations*, 2025.

Neta Shaul, Itai Gat, Marton Havasi, Daniel Severo, Anuroop Sriram, Peter Holderrieth, Brian Karrer, Yaron Lipman, and Ricky T. Q. Chen. Flow matching with general discrete paths: A kinetic-optimal perspective. In *International Conference on Learning Representations*, 2025.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

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

Tianlin Xu, Li Kevin Wenliang, Michael Munn, and Beatrice Acciaio. COT-GAN: Generating sequential data via causal optimal transport. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Eun Bi Yoon, Keehun Park, Sungwoong Kim, and Sungbin Lim. Score-based generative models with lévy processes. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar. Time-series generative adversarial networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Xinyu Yuan and Yan Qiao. Diffusion-TS: Interpretable diffusion for general time series generation. In *International Conference on Learning Representations*, 2024.

APPENDIX

This appendix reports validation selection, data splits, full benchmark results, paired comparisons, proofs, and implementation details.

A VALIDATION-ONLY SELECTION

A.1 RJFM CONFIGURATION

Eight RJFM configurations were trained with seeds 42 and 43. Selection minimized average rank over six validation metrics and all nine real series. Test, synthetic, and abrupt-subset ranks were excluded from the decision.

Configuration	q	α	β	Base w.	Cls. w.	Reg. w.	Val. rank	Val. tail/jump	Test rank	Synth. rank
Combined, no auxiliary	.950	1.50	.50	.000	.000	.000	**3.667**	3.625	3.042	3.000
Combined, low auxiliary	.950	1.50	.50	.100	.025	.025	3.750	3.812	4.375	6.917
$\beta = .5$	.900	2.00	.50	.100	.050	.050	4.000	**3.438**	4.500	7.500
$q = .975$	.975	2.00	1.00	.100	.050	.050	4.292	4.688	4.042	3.167
Initial V3	.900	2.00	1.00	.100	.050	.050	4.708	4.812	4.375	5.000
$\alpha = 1.5$	.900	1.50	1.00	.100	.050	.050	4.917	5.125	5.458	3.167
$q = .95$	.950	2.00	1.00	.100	.050	.050	5.042	4.500	4.917	3.250
$\alpha = 1$	.900	1.00	1.00	.100	.050	.050	5.625	6.000	5.292	4.000

Table A.1: RJFM configuration selection. Selection uses the average validation rank; test and synthetic ranks are reported for completeness. Bold marks the lowest validation rank in each validation column.

A.2 DLPM CONFIGURATION

The DLPM validation sweep varies the stable index while fixing learning rate 10^{-3}, 100 diffusion steps, and unclipped stable variables. Germany, OpenAQ 8118, and .SPX provide one validation series per domain; each configuration uses seeds 42 and 43. The score averages ranks over six metrics and the three series, without using test results.

Configuration	α	Learning rate	Steps	Avg. rank	Tail rank
DLPM-1.9	1.9	10^{-3}	100	**1.111**	**1.000**
DLPM-1.8	1.8	10^{-3}	100	2.000	2.000
DLPM-1.5	1.5	10^{-3}	100	2.889	3.000

Table A.2: Validation-only DLPM selection. DLPM-1.9 is used in every reported test comparison.

A.3 DIFFUSION-TS CONFIGURATION

Diffusion-TS uses the authors' model implementation under the shared benchmark protocol. We retain the stock two-layer encoder, two-layer decoder, width 64, four-head transformer, cosine diffusion, L1 reconstruction, Fourier loss, Adam $(0.9, 0.96)$, unit gradient clipping, and EMA decay 0.995. The input and output dimensions are adapted to univariate length-64 windows. A training-only min–max transform maps inputs to $[-1, 1]$; generated windows are inverted before evaluation. The model trains for the shared 1,500-update, batch-256 budget, warming the learning rate from 10^{-5} to 8×10^{-4} over 500 updates. Training uses 500 diffusion steps and evaluation uses deterministic 64-step DDIM sampling. This fixed configuration has 309,645 parameters and is denoted Diffusion-TS throughout.

B REAL-DATA SPLITS

Table B.1 reports exact window counts. Training uses stride 8, validation and test use stride 1, and 64-point gaps prevent overlap in raw observations between splits.

Suite	Series	Raw points	Train windows	Validation	Test
Ember	Germany	102,428	8,944	10,167	20,397
Ember	Spain	102,428	8,944	10,167	20,397
Ember	Italy	101,654	8,876	10,089	20,243
Ember	United Kingdom	89,232	7,789	8,847	17,759
OpenAQ	8118 (New Delhi)	8,636	737	787	1,640
OpenAQ	8128	4,350	362	359	782
Oxford-Man	.SPX	4,641	387	388	840
Oxford-Man	.RUT	4,636	387	387	840
Oxford-Man	.STOXX50E	4,713	394	395	855

Table B.1: Raw series lengths and window counts in the real-data benchmark.

C Complete Full-Test Winner Counts

Table C.1 counts which model has the lowest ten-seed mean for each metric and series. Counts in each row sum to nine.

Metric	DLPM	Diffusion-TS	TCN_{64}	TCN_{80}	Residual path	RJFM
RBF-MMD	2	1	2	0	1	**3**
Tail-95 error	0	1	1	0	1	**7**
Tail-99 error	0	1	1	1	2	**5**
Jump-rate error	2	0	**3**	0	2	2
Jump-size error	0	1	**3**	**3**	1	1
Lag-1 error	2	0	**3**	0	**3**	1
Maximum-increment W1	0	0	2	0	3	**4**
Maximum-increment $q_{.95}$ error	1	0	**4**	0	1	3
Expected-shortfall-95 error	0	1	**3**	**3**	1	1
Exceedance ACF-1 error	1	2	2	1	0	**3**
Cluster-length error	0	0	2	1	**3**	**3**
Multiscale MMD	**3**	1	2	0	1	2
Total	11	8	26	9	19	**35**

Table C.1: Full-test dataset-level winner counts over 12 metrics. Bold marks the largest count in each row; ties are all bolded.

D Tail-99 Results by Series

Series	DLPM	Diffusion-TS	TCN_{64}	TCN_{80}	Residual path	RJFM
Germany	41.2658	5.9547	2.4163	2.4562	2.4697	**1.5548**
Spain	41.2606	7.2214	1.8860	1.3172	**1.1741**	1.2594
Italy	40.0687	5.6716	2.8144	2.6791	3.2020	**2.5486**
United Kingdom	44.3537	19.5741	2.6439	2.4832	2.4193	**2.0624**
OpenAQ 8118	43.4636	**0.0000**	0.0531	0.0680	0.0666	0.0669
OpenAQ 8128	36.8950	2.8345	0.3952	**0.2453**	0.2899	0.3174
.RUT	50.0462	14.1322	2.3375	3.0710	2.5943	**1.9808**
.SPX	44.5738	19.9494	2.0634	2.5081	3.0233	**2.0565**
.STOXX50E	47.6810	21.3893	1.8501	1.8948	**1.5514**	1.7370

Table D.1: Full-test tail-99 error, averaged over seeds 100–109. Lower is better; the best mean in each row is bolded.

Series	DLPM	Diffusion-TS	TCN_{64}	TCN_{80}	Residual path	RJFM
Germany	223.4050	5.1377	2.5771	2.5882	2.6003	**1.6422**
Spain	229.0477	7.5292	2.8688	2.0906	**1.7616**	1.8006
Italy	224.8631	5.7590	3.7457	3.6103	4.0709	**3.4622**
United Kingdom	234.7717	20.3522	3.3015	2.9454	2.9801	**2.6110**
OpenAQ 8118	243.2057	0.0860	0.0253	0.0308	0.0286	**0.0229**
OpenAQ 8128	245.0880	2.7370	0.7322	0.5631	0.4997	**0.2172**
.RUT	260.5074	14.4689	4.8399	5.9931	5.2246	**4.4922**
.SPX	250.5840	20.8951	3.7681	4.5076	5.4855	**3.6251**
.STOXX50E	247.9802	22.7208	5.2407	5.4914	**4.5764**	5.0100

Table D.2: Tail-99 error on independently selected top-quartile abrupt windows, averaged over ten seeds. This is a marginal stress test, not conditional generation.

E Abrupt-Subset Winner Counts

Metric	DLPM	Diffusion-TS	TCN_{64}	TCN_{80}	Residual path	RJFM
RBF-MMD	0	2	1	0	**4**	2
Tail-95 error	0	1	1	1	2	**4**
Tail-99 error	0	0	0	0	2	**7**
Jump-rate error	0	0	**5**	0	1	3
Jump-size error	0	0	2	1	**5**	1
Lag-1 error	2	0	0	1	2	**4**
Total	2	3	9	3	16	**21**

Table E.1: Winner counts on independently selected abrupt windows.

F Paired Comparisons

For each model, series, and seed, the analysis stores the six standard metrics and six diagnostics. Direct comparison forms ten paired differences using identical seed labels. Table F.1 reports direction counts for the three designated primary metrics. Unadjusted interval counts summarize seed-level stability; they are descriptive and are not used as multiple-testing significance claims.

Comparator	Metric	RJFM lower mean	RJFM interval win
DLPM	Tail-99	9/9	9/9
DLPM	Maximum-increment W1	9/9	9/9
DLPM	Expected-shortfall-95	9/9	9/9
Diffusion-TS	Tail-99	8/9	8/9
Diffusion-TS	Maximum-increment W1	9/9	9/9
Diffusion-TS	Expected-shortfall-95	8/9	8/9
$TCN\text{-}FM_{64}$	Tail-99	8/9	3/9
$TCN\text{-}FM_{64}$	Maximum-increment W1	5/9	0/9
$TCN\text{-}FM_{64}$	Expected-shortfall-95	4/9	0/9
$TCN\text{-}FM_{80}$	Tail-99	8/9	2/9
$TCN\text{-}FM_{80}$	Maximum-increment W1	6/9	2/9
$TCN\text{-}FM_{80}$	Expected-shortfall-95	4/9	0/9
$Residual\text{-}Path_{80}$	Tail-99	5/9	0/9
$Residual\text{-}Path_{80}$	Maximum-increment W1	5/9	2/9
$Residual\text{-}Path_{80}$	Expected-shortfall-95	4/9	0/9

Table F.1: Directional and interval-separated results for primary metrics.

G DIFFUSION-TS BUDGET STRESS TEST

The primary Diffusion-TS comparison uses 1,500 updates. A separate convergence experiment trains each of the 90 series–seed trajectories for 3,000 updates and evaluates EMA checkpoints at 250, 1,500, and 3,000 updates. Each checkpoint restores the random-number-generator state saved at that update before drawing 2,048 samples, matching the sampling semantics of a run stopped at the same point. The checkpoint grid diagnoses optimization behavior and is not used for model selection.

		All 108 series–metric units			27 primary units		
Updates	Sec.	RJFM mean	RJFM int.	DTS int.	RJFM mean	RJFM int.	DTS int.
250	15.07	103	98	3	26	25	1
1,500	75.50	100	99	6	25	25	2
3,000	147.83	91	81	5	22	18	0

Table G.1: Checkpoint-wise Diffusion-TS comparison. "Mean" counts units on which RJFM has the lower ten-seed mean. "Int." counts separated paired 95% intervals in the indicated direction. The unreported interval counts are overlaps: 7, 3, and 22 over all metrics, and 1, 0, and 9 over the primary metrics, respectively. Runtime is the mean end-to-end seconds per trajectory.

Extending Diffusion-TS from 1,500 to 3,000 updates improves its mean error on 88/108 units. At the 3,000-update checkpoint, RJFM has lower means on 91/108 units and separated paired intervals on 81; five separated intervals favor Diffusion-TS. Table G.2 shows the corresponding per-metric directions.

	Lower mean		Separated interval	
Metric	RJFM	DTS-3k	RJFM	DTS-3k
RBF-MMD	**8/9**	1/9	**8/9**	0/9
Tail-95 error	**7/9**	2/9	**5/9**	1/9
Tail-99 error	**7/9**	2/9	**5/9**	0/9
Jump-rate error	**9/9**	0/9	**9/9**	0/9
Jump-size error	**7/9**	2/9	**5/9**	0/9
Lag-1 error	**9/9**	0/9	**9/9**	0/9
Maximum-increment W1	**8/9**	1/9	**8/9**	0/9
Maximum-increment $q_{.95}$	**5/9**	4/9	**4/9**	2/9
Expected-shortfall-95	**7/9**	2/9	**5/9**	0/9
Exceedance ACF	**7/9**	2/9	**7/9**	2/9
Cluster-length error	**9/9**	0/9	**8/9**	0/9
Multiscale MMD	**8/9**	1/9	**8/9**	0/9

Table G.2: Per-metric directions at the 3,000-update Diffusion-TS checkpoint. Each denominator contains the nine real series. Interval columns omit overlaps. Bold marks the larger directional count in each pair.

Series	RJFM	DTS-250	DTS-1.5k	DTS-3k
Germany	1.555	7.014	5.935	**1.317**
Italy	2.549	6.462	5.688	**1.880**
Spain	**1.259**	8.911	7.112	1.468
United Kingdom	**2.062**	24.458	19.471	5.681
OpenAQ 8118	**0.067**	0.000	0.000	0.145
OpenAQ 8128	**0.317**	3.180	2.841	1.274
.RUT	**1.981**	17.249	14.559	5.260
.SPX	**2.056**	25.558	19.888	6.016
.STOXX50E	**1.737**	27.895	21.297	5.724

Table G.3: Mean tail-99 error along the Diffusion-TS training trajectory. Bold marks the lower value between RJFM and the 3,000-update checkpoint; earlier checkpoints are shown as convergence diagnostics.

At 3,000 updates, Diffusion-TS has lower tail-99 means on Germany and Italy, while RJFM has lower means on the other seven series; the paired intervals for Germany and Italy overlap zero. The last-100-update training loss decreases from 0.0679 at 1,500 updates to 0.0656 at 3,000, a further 3.3% reduction. Evaluation metrics also continue to improve, so the checkpoint study does not establish asymptotic convergence. The 3,000-update trajectory averages 147.83 seconds, compared with 27.27 seconds for RJFM.

The 1,500-update checkpoint in the longer convergence trajectories averages 75.50 seconds. The independently executed primary 1,500-update runs average 73.03 seconds, which is the value used in the main efficiency comparison.

H Additional Aggregate Comparisons

Comparator	RJFM lower mean	RJFM interval	Comparator interval	Overlap
Official DLPM	95/108	86	7	15
Diffusion-TS	100/108	99	6	3
TCN-FM$_{64}$	61/108	8	2	98
TCN-FM$_{80}$	74/108	14	0	94
Residual-Path$_{80}$	66/108	14	3	91

Table H.1: Direct ten-seed comparison over nine series and 12 metrics. Interval columns count separated unadjusted paired 95% intervals; overlap denotes an interval containing zero.

Model	Parameters	Sampling steps	Seconds/run
Official DLPM	153,792	100	**8.10**
Diffusion-TS	309,645	64	73.03
TCN-FM$_{64}$	**106,561**	64	14.51
TCN-FM$_{80}$	165,201	64	14.99
Residual-Path$_{80}$	165,201	64	15.74
RJFM	156,867	64	27.27

Table H.2: Model size and mean end-to-end time for the independently executed primary real-data runs on one RTX 5090. Bold marks the smallest parameter count and shortest measured runtime; sampling steps are protocol settings.

I Theoretical Analysis

This section characterizes the geometry and population target of the prescribed residual path. Finite-sample performance is evaluated empirically.

I.1 Endpoint, delay, and kinetic action

Let $x_1 = B + R$ and define

$$x_t = (1-t)x_0 + tB + t^\alpha R, \qquad u_t = B - x_0 + \alpha t^{\alpha-1} R. \tag{I.1}$$

Proposition I.1 (Endpoint, delay, and action). *For $\alpha \geq 1$, equation I.1 starts at x_0 and ends at x_1. When $\alpha = 1$, it is the linear interpolation $\bar{x}_t = (1-t)x_0 + tx_1$. When $\alpha > 1$,*

$$\max_{t\in[0,1]} \|x_t - \bar{x}_t\|_2 = (\alpha-1)\alpha^{-\alpha/(\alpha-1)}\|R\|_2, \tag{I.2}$$

$$\int_0^1 \|u_t\|_2^2 \,\mathrm{d}t = \|x_1 - x_0\|_2^2 + \frac{(\alpha-1)^2}{2\alpha-1}\|R\|_2^2. \tag{I.3}$$

Proof. Substitution at $t = 0$ and $t = 1$ gives the two endpoints. Since $B = x_1 - R$,

$$x_t - \bar{x}_t = (t^\alpha - t)R. \tag{I.4}$$

For $\alpha > 1$, $g(t) = t - t^\alpha$ is nonnegative and is maximized at $t_\star = \alpha^{-1/(\alpha-1)}$. Evaluating $g(t_\star)$ gives $(\alpha-1)\alpha^{-\alpha/(\alpha-1)}$, proving Equation I.2. For the action, set $A = B - x_0$ and integrate:

$$\int_0^1 \|A + \alpha t^{\alpha-1}R\|_2^2 \,\mathrm{d}t = \|A\|_2^2 + 2\langle A, R\rangle + \frac{\alpha^2}{2\alpha-1}\|R\|_2^2 \tag{I.5}$$

$$= \|A+R\|_2^2 + \left(\frac{\alpha^2}{2\alpha-1} - 1\right)\|R\|_2^2. \tag{I.6}$$

The identities $A + R = x_1 - x_0$ and $\alpha^2/(2\alpha-1) - 1 = (\alpha-1)^2/(2\alpha-1)$ complete the proof. □

For $\alpha = 1.5$, the maximal path displacement from linear interpolation is $4/27 \approx 0.148$ times $\|R\|_2$, and the excess action is $\|R\|_2^2/8$. Thus the selected schedule moves residual transport toward later generative time and increases the conditional-path action by the stated amount.

I.2 Population flow-matching target

Let (X_0, X_1) follow the training coupling, let X_t be the random path in equation I.1, and write $U_t = \dot{X}_t$. Denote the law of X_t by p_t.

Proposition I.2 (Population flow-matching target). *Assume $\mathbb{E}\|U_t\|_2^2 < \infty$ and define $v_t^\star(x) = \mathbb{E}[U_t \mid X_t = x]$. Then:*

1. *$v_t^\star$ minimizes $\mathbb{E}\|v(X_t,t) - U_t\|_2^2$ over measurable velocity fields with finite second moment;*
2. *for every smooth compactly supported test function φ,*

$$\frac{\mathrm{d}}{\mathrm{d}t}\int \varphi(x)\, p_t(\mathrm{d}x) = \int \nabla\varphi(x)^\top v_t^\star(x)\, p_t(\mathrm{d}x); \tag{I.7}$$

3. *if the continuity equation and ODE induced by $v_t^\star$ are well posed, the ODE flow has marginals p_t and transports the Gaussian p_0 to the data distribution p_1.*

Proof. Conditional-expectation orthogonality gives

$$\mathbb{E}\|v(X_t,t) - U_t\|_2^2 = \mathbb{E}\|v(X_t,t) - v_t^\star(X_t)\|_2^2 + \mathbb{E}\|U_t - v_t^\star(X_t)\|_2^2, \tag{I.8}$$

which proves the first claim. Differentiating a test function along the path and conditioning on X_t yields

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[\varphi(X_t)] = \mathbb{E}[\nabla\varphi(X_t)^\top U_t] \tag{I.9}$$

$$= \mathbb{E}[\nabla\varphi(X_t)^\top v_t^\star(X_t)], \tag{I.10}$$

which is equation I.7. The final statement follows from uniqueness of the corresponding continuity equation under the stated regularity condition. The endpoint identities give the required p_0 and p_1. □

The proposition gives the population-level interpretation: changing from the linear path to the residual path preserves the ideal terminal distribution but changes the regression target, and hence the inductive bias seen by a finite model.

I.3 CUMULATIVE JUMP GEOMETRY AND ERROR AMPLIFICATION

Define the first-difference and cumulative-sum operators by

$$(Dx)_\ell = x_{\ell+1} - x_\ell, \qquad (Cz)_1 = 0, \qquad (Cz)_k = \sum_{\ell=1}^{k-1} z_\ell, \tag{I.11}$$

where $D : \mathbb{R}^d \to \mathbb{R}^{d-1}$, $C : \mathbb{R}^{d-1} \to \mathbb{R}^d$, $\ell \in \{1, \dots, d-1\}$, and $k \in \{2, \dots, d\}$. Thus coordinate ℓ of an increment vector corresponds to the edge between sequence positions ℓ and $\ell + 1$. With $\widetilde{m}_\ell = m_{\ell+1}$, the target residual is $R = C(\widetilde{m} \odot Dx_1)$, and the jump head applies the same operator C to its predicted increment velocities.

Proposition I.3 (Cumulative residual operator)**.** *For every $z \in \mathbb{R}^{d-1}$, $D(Cz) = z$. If $z_\ell = \delta$ and all other coordinates of z are zero, then $(Cz)_k = 0$ for $1 \le k \le \ell$ and $(Cz)_k = \delta$ for $\ell + 1 \le k \le d$. Moreover, for any increment error e,*

$$\|Ce\|_2 \le \|C\|_F \|e\|_2 = \sqrt{\frac{d(d-1)}{2}}\, \|e\|_2. \tag{I.12}$$

Proof. For $\ell = 1, \dots, d-1$, $(D(Cz))_\ell = (Cz)_{\ell+1} - (Cz)_\ell = z_\ell$, proving $D(Cz) = z$. The single-increment claim follows directly from the upper summation limit in the definition of C. The matrix of C is lower triangular below its first row and contains $d(d-1)/2$ unit entries. Hence $\|C\|_2 \le \|C\|_F = \sqrt{d(d-1)/2}$, which gives Equation I.12. □

The bound shows that an error made in increment space can persist across later sequence positions and be amplified by the cumulative map by at most a factor $\sqrt{d(d-1)/2}$. This accumulation motivates initializing the mark head at zero before learning persistent level shifts.

The sigmoid gate in the target construction satisfies $m_i > 1/2$ exactly when $|(Dx_1)_{i-1}| > \tau$ for $i = 2, \dots, d$. In the absence of quantile ties, choosing the empirical q-quantile therefore places approximately $1 - q$ of hard increment labels above the midpoint; for $q = 0.95$, this is approximately 5%. The gate remains soft, so neither the target residual nor the predicted branch is exactly sparse.

I.4 BRANCH IDENTIFIABILITY AND TEMPERED SAMPLING

The selected model is trained only on the total velocity. For any field h, the transformation

$$(u_\theta^B, u_\theta^J) \mapsto (u_\theta^B + h, u_\theta^J - h) \tag{I.13}$$

leaves both $u_\theta^B + u_\theta^J$ and the reported training loss unchanged. The two branches are therefore not statistically identifiable from this loss alone; their learned values also depend on architecture and initialization.

The sampling coefficient β tempers the learned jump field. The following bound quantifies the resulting perturbation without assuming that the learned branches recover the target decomposition.

Proposition I.4 (Tempered-field perturbation)**.** *Let $b, j : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ and define $f_\gamma(x,t) = b(x,t) + \gamma j(x,t)$. Fix $\beta \in \mathbb{R}$, and let X_t^1 and X_t^β be absolutely continuous solutions of $\dot{X}_t = f_1(X_t, t)$ and $\dot{X}_t = f_\beta(X_t, t)$, respectively, from the same initial state. Suppose $f_1(\cdot, t)$ is L-Lipschitz in its first argument for almost every t, and suppose an integrable function $J_\beta : [0,1] \to [0, \infty)$ satisfies $\|j(X_t^\beta, t)\|_2 \le J_\beta(t)$ for almost every t. Then*

$$\|X_1^\beta - X_1^1\|_2 \le |1 - \beta| \int_0^1 e^{L(1-s)} J_\beta(s)\, \mathrm{d}s. \tag{I.14}$$

Proof. Set $\Delta_t = X_t^\beta - X_t^1$. The exact decomposition

$$\dot{\Delta}_t = f_\beta(X_t^\beta, t) - f_1(X_t^1, t) \tag{I.15}$$

$$= f_1(X_t^\beta, t) - f_1(X_t^1, t) + (\beta - 1) j(X_t^\beta, t) \tag{I.16}$$

uses the Lipschitz property only for f_1 and evaluates the perturbation on the X^β trajectory. Therefore, for almost every t, the upper Dini derivative of $\|\Delta_t\|_2$ satisfies

$$D^+\|\Delta_t\|_2 \le L\|\Delta_t\|_2 + |1-\beta|J_\beta(t). \tag{I.17}$$

Since $\Delta_0 = 0$, Gronwall's inequality yields equation I.14. □

The proposition bounds the continuous-time trajectory change induced by $\beta \neq 1$ in terms of the jump field encountered along X^β. The bound concerns the continuous-time trajectory; attenuation and Euler step count remain empirical choices.

An oracle calculation makes the distinction from path endpoint consistency explicit. If the predicted branches exactly equal the target components in equation I.1, N left-Euler steps produce

$$x_N = B + \beta S_{N,\alpha} R, \qquad S_{N,\alpha} = \frac{\alpha}{N}\sum_{k=0}^{N-1}\left(\frac{k}{N}\right)^{\alpha-1}. \tag{I.18}$$

For $\alpha > 1$, the integrand is increasing, so the left Riemann sum satisfies $S_{N,\alpha} < 1$. Exact recovery of $x_1 = B + R$ would require $\beta S_{N,\alpha} = 1$ in this oracle setting. The selected $\beta = 0.5$ is instead a validation-chosen attenuation. Endpoint consistency therefore describes the training path, whereas the performance of the tempered finite-step sampler is determined empirically.

J Implementation and Reproducibility Details

The residual threshold is computed once from training windows. Jump-head logits are initialized with bias -2.2, and mark-head weights and bias are initialized to zero so early cumulative marks do not dominate the base velocity. TCN-FM and RJFM use four dilated residual blocks with group normalization and SiLU activations. RJFM splits the blocks into a shared trunk and two branches. TCN-FM$_{80}$ and Residual-Path$_{80}$ use width 80; TCN-FM$_{64}$ and RJFM use width 64. Euler sampling begins from a standard Gaussian window. DLPM reverse sampling begins from isotropic symmetric α-stable noise. Diffusion-TS uses the authors' decomposition transformer and direct reconstruction loss with a Fourier-domain term. Its EMA model is sampled with deterministic 64-step DDIM and transformed back from the training-only $[-1, 1]$ scale before computing the shared metrics.

The experiments used PyTorch 2.12.1 with CUDA 13.0 on one NVIDIA RTX 5090. The result archive contains 468 seed-level summaries: 18 DLPM tuning jobs, 90 selected-DLPM confirmation jobs, and 360 flow-model ablation jobs. The independent Diffusion-TS confirmation archive adds 90 jobs with the same series and seed labels, giving complete six-model coverage. The budget experiment adds 90 training trajectories and 270 checkpoint evaluations.

J.1 Full training and sampling procedure

Algorithm J.1 Full RJFM training and Euler sampling procedure

Require: training threshold τ, exponent α, sampling weight β, and number of Euler steps N

Training step

Sample x_1, $x_0 \sim \mathcal{N}(0, I_d)$, and $t \sim \mathcal{U}[0,1]$
$\Delta x_{1,i} \leftarrow x_{1,i} - x_{1,i-1}$ and $m_i \leftarrow \sigma((|\Delta x_{1,i}| - \tau)/(0.1\tau))$
$r_i \leftarrow m_i \Delta x_{1,i}$; $R_k \leftarrow \sum_{i=2}^{k} r_i$; $B \leftarrow x_1 - R$
$x_t \leftarrow (1-t)x_0 + tB + t^\alpha R$; $u^\star \leftarrow B - x_0 + \alpha t^{\alpha-1} R$
$(u_\theta^B, u_\theta^J) \leftarrow \textsc{TemporalVelocity}_\theta(x_t, t)$
Update θ using $\|u_\theta^B + u_\theta^J - u^\star\|_2^2$

Euler sampling

$x \leftarrow x_0 \sim \mathcal{N}(0, I_d)$ and $h \leftarrow 1/N$
for $n = 0, \ldots, N-1$ **do**
    $(u^B, u^J) \leftarrow \textsc{TemporalVelocity}_\theta(x, nh)$
    $x \leftarrow x + h(u^B + \beta u^J)$
end for
return x