PETS: Inference-Time Differentially Private Synthetic Time Series Generation

Yangzhixin Luo[1], **Haibo Wu**[1], **Alessandro Cornacchia**[1], **Chenxi Liu**[2,1], **Marco Canini**[1]
[1]KAUST, [2]HKISI-CAS
yangzhixin.luo@kaust.edu.sa

Abstract

Existing methods for differentially private (DP) synthetic time series generation inject privacy during model training via DP-SGD, requiring private data in the training phase, expensive hyperparameter tuning, and costly retraining for new domains. We propose **Private Evolution for Time Series (PETS)**, the first inference-time framework for DP synthetic time series generation via Private Evolution (PE). In this setting, private data are not used to train generative models, but only to guide the selection of synthetic outputs at inference time, to maximize fidelity and satisfy a privacy-budget constraint. Building on top of PE, we construct PETS through three specialized components: a rule-based generation module, a VAE-based structure-preserving variation module, and contrastive embeddings for similarity-driven selection. The framework is modular, enabling domain adaptation by swapping components with no retraining overhead. On the traffic benchmark (METR-LA) at ϵ=0.7, PETS achieves a C-FID of 3.38, reducing C-FID by 14$\times$ compared to the state-of-the-art method, and attains $\geq$27$\times$ lower forecasting RMSE, demonstrating strong utility-privacy trade-offs.

Track: Research

1 Introduction

Time series data is ubiquitous across numerous domains, capturing temporal patterns in healthcare, finance, and industrial processes (Fawaz et al., 2019). However, such data often contains sensitive information that organizations and individuals are reluctant to share due to privacy concerns and data leakage risks (Shokri et al., 2017).

Differential privacy (DP) (Dwork et al., 2006) provides formal guarantees that the inclusion or exclusion of any single individual's data has a bounded impact on the output. Generating a synthetic dataset that is statistically similar to the original data while ensuring DP enables privacy-preserving data sharing for downstream tasks and pretraining large models for time series (Liu et al., 2025).

Existing methods for DP synthetic time series generation inject privacy guarantees *during model training* via DP-SGD (Abadi et al., 2016) applied to generative models, adding calibrated noise to gradients during training (Frigerio et al., 2019). This paradigm has three limitations: (1) private data enters the training loop, raising deployment concerns; (2) DP training is computationally expensive and requires extensive tuning; (3) models lack transferability and require retraining for new domains.

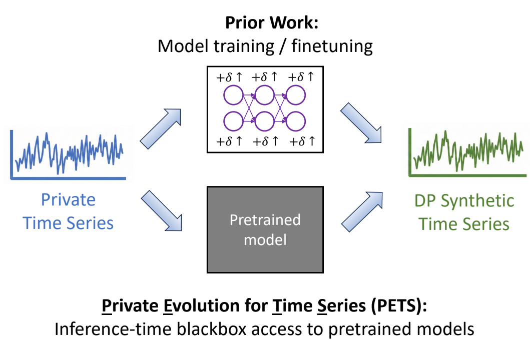


Figure 1: Comparison of prior work vs. PETS. **Top**: Prior work requires training or finetuning models on private data with DP-SGD. **Bottom**: PETS operates at inference time with blackbox access to pretrained models, using private data only for DP-protected voting.

To overcome these limitations, we present an *inference-time* framework for DP synthetic time series generation that does not require private data during model training or finetuning, as illustrated in Figure 1. The framework leverages pre-trained generative models, accessing private data only at inference for statistical selection. Private data remains on the client side and is only used to *vote* for synthetic candidates, with DP applied to voting counts. This paradigm offers key advantages: (1) private data never enters the training pipeline, (2) no expensive DP-constrained training is needed, and (3) the framework adapts to new domains by plugging in different modules without retraining.

However, while image and text domains have inference-time frameworks such as Private Evolution (PE) (Lin et al., 2024; Xie et al., 2024; Lin et al., 2025), which leverage pre-trained foundation models (e.g., Stable Diffusion (Rombach et al., 2022) for generation, CLIP (Radford et al., 2021) for embeddings), inference-time DP methods are largely absent in the time series domain. Time series lack comparable infrastructure: high-quality generation APIs and robust pretrained embedding models are scarce, making it challenging to apply PE-style frameworks directly.

We propose **Private Evolution for Time Series (PETS)**, the first inference-time framework for DP synthetic time series generation. PETS addresses the absence of inference-time DP methods in the time series domain by designing specialized components that respect temporal structure while enabling DP synthesis without training on private data. The generated synthetic data can serve as privacy-preserving training data for time series foundation models, addressing a key need in the era of large models where diverse, high-quality pretraining data is essential yet often privacy-sensitive.

PETS adapts the Private Evolution (PE) framework (Lin et al., 2024) to time series with critical domain-specific innovations: (1) we utilize a rule-based generation method (Xie et al., 2025) that produces diverse initial candidates with controllable temporal patterns (periodicity, trends, noise characteristics) without requiring training, (2) we design a VAE-based (Kingma & Welling, 2014) variation mechanism pretrained on public data that preserves temporal dependencies (autocorrelation, seasonality) while introducing controlled diversity, and (3) we leverage TS2Vec embeddings (Yue et al., 2022), a contrastive representation pretrained on public data to define meaningful similarity robust to phase shifts and amplitude variations inherent in time series. Our framework is modular: users can substitute their own generation, variation, or similarity modules trained on domain-relevant public data, without retraining on private datasets.

Contributions. Our main contributions are:

- We introduce **PETS**, the first inference-time framework for DP synthetic time series generation, addressing a critical gap in privacy-preserving time series analysis. PETS accesses private data only at inference time, enabling privacy-preserving data sharing without the computational costs and privacy risks of training-based approaches.
- We design specialized components for time series DP synthesis, including generation, variation, and similarity measurement for selection, all respecting temporal structure and trained exclusively on public data for the inference-time DP setting.
- We demonstrate strong empirical performance: on z-score preprocessed traffic data (METR-LA), PETS achieves C-FID of 3.38 at $\epsilon = 0.7$, achieving a $14\times$ improvement over the state-of-the-art method. This indicates that PETS can generate high-quality DP synthetic time series and achieve strong utility-privacy trade-offs.

2 Problem Setup

Differential Privacy A randomized algorithm $\mathcal{M}$ satisfies (ϵ, δ)-differential privacy (DP) (Dwork et al., 2006) if for any two neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$ differing in at most one entry, and for any subset S of possible outputs, the following holds:

$$\Pr[\mathcal{M}(\mathcal{D}) \in S] \le e^{\epsilon} \Pr[\mathcal{M}(\mathcal{D}') \in S] + \delta. \tag{1}$$

Problem Formulation We aim to construct an (ϵ, δ)-DP algorithm that takes a private time series dataset $S_{\text{priv}} = \{x_i\}_{i=1}^{N_{\text{priv}}}$ as input and produces a synthetic dataset $S_{\text{syn}} = \{x'_j\}_{j=1}^{N_{\text{syn}}}$. Each time series x_i is univariate ($x_i \in \mathbb{R}^T$) or multivariate ($x_i \in \mathbb{R}^{T \times C}$) with length T. The objective is to minimize the Wasserstein-p distance $W_p(S_{\text{priv}}, S_{\text{syn}})$ with respect to a suitable distance function $d(\cdot, \cdot)$ between time series (formal definition in Appendix E).

3 PETS DESIGN

Figure 2 illustrates the PETS pipeline. **Step 1**: An initial population of N_{syn} synthetic time series is produced via rule-based generation (①). The population is then refined over T iterations. At each iteration: **Step 2.1**, every private sample finds its nearest synthetic neighbor by computing distances in TS2Vec embedding space (③) and votes for it; these votes form a histogram where each bin corresponds to a synthetic sample and its value is the number of votes received; the histogram is then privatized to DP by adding calibrated Gaussian noise; **Step 2.2**, synthetic candidates most similar to the private data are selected by resampling proportionally to the DP histogram; **Step 2.3**, the selected samples are passed through VAE-based variation (②) to produce the next-generation population. After T iterations, the final population constitutes the DP synthetic dataset.

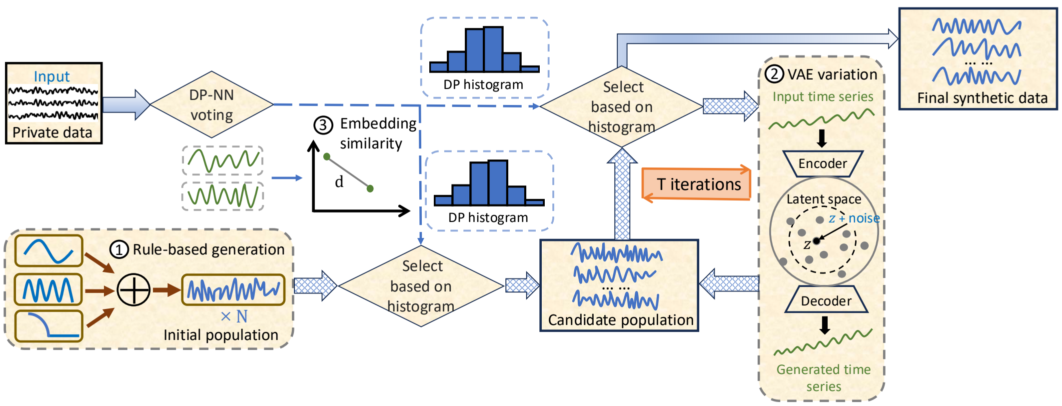


Figure 2: Overview of the PETS framework.

PETS draws inspiration from Private Evolution (PE) in other domains but critically adapts the methodology for time series with specialized generation, variation, and similarity components, as described below. The complete PETS algorithm is provided in Appendix A.

① **Rule-based generation.** For the `GENERATION` module, we adopt a rule-based ChatTS generator (Xie et al., 2025) that composes time series from interpretable attributes:

$$\mathcal{G}(\theta) = f_{\text{trend}}(\theta_{\text{tr}}) + f_{\text{seasonal}}(\theta_{\text{seas}}) + f_{\text{local}}(\theta_{\text{loc}}) + \eta, \tag{2}$$

where $\theta = (\theta_{\text{tr}}, \theta_{\text{seas}}, \theta_{\text{loc}})$ are randomly sampled attributes, f_{trend} models linear/exponential trends, f_{seasonal} models periodic patterns (sine/square/triangle with random amplitude/frequency), f_{local} injects spikes or step changes at random positions, and $\eta \sim \mathcal{N}(0, \sigma^2_{\text{noise}})$ adds Gaussian noise.

② **Structure-preserving variation via VAE.** For the `VARIATION` module, to maintain temporal structures (autocorrelation, spectral properties) while providing controlled perturbation strength, we pretrain a VAE (Kingma & Welling, 2014) on public time series datasets with Conv1D encoder $q_\phi(z|x)$ and decoder $p_\theta(x|z)$, where $x \in \mathbb{R}^T$ is an input time series and $z \in \mathbb{R}^{d_z}$ is the latent representation. The encoder outputs mean $\mu(x)$ and log-variance $\log \sigma^2(x)$; we sample $z = \mu(x) + \sigma(x) \odot \epsilon$ with $\epsilon \sim \mathcal{N}(0, I)$. The training loss includes reconstruction, KL divergence, and frequency-domain alignment:

$$\mathcal{L}_{\text{VAE}} = \|x - p_\theta(z)\|_2^2 + \beta \cdot \text{KL}(q_\phi(z|x) \| \mathcal{N}(0, I)) + \lambda_{\text{freq}} \|\mathcal{F}(x) - \mathcal{F}(p_\theta(z))\|_2^2 + \mathcal{L}_{\text{aux}}, \tag{3}$$

where $\beta, \lambda_{\text{freq}}$ are weighting coefficients, $\mathcal{F}(\cdot)$ denotes FFT, and $\mathcal{L}_{\text{aux}}$ includes auxiliary terms for preserving local features (finite-difference loss, peak-matching loss). The VAE is trained on *public* data only and kept fixed when running PETS on private data, preserving the inference-time privacy setting.

Given input x and `variation_degree` $\alpha \in [0, 100]$, we generate a variation x' via:

$$
\begin{aligned}
&\mu, \log \sigma^2 = q_\phi(\tilde{x}), && \epsilon_{\text{post}}, \epsilon_{\text{dir}}, \epsilon_{\text{prior}} \sim \mathcal{N}(0, I), \\
&\tilde{z} = \mu + s \cdot \sigma \cdot \epsilon_{\text{post}} + \kappa \cdot s \cdot \epsilon_{\text{dir}}, && z_{\text{var}} = (1-\rho) \cdot \tilde{z} + \rho \cdot \epsilon_{\text{prior}}, \\
&x_{\text{dec}} = p_\theta(z_{\text{var}}), && x' = (1-\gamma) \cdot x_{\text{dec}} + \gamma \cdot x,
\end{aligned} \tag{4}
$$

where $\tilde{x}$ is the per-series standardized input (matching VAE training), x_{dec} is rescaled back to the input mean/std before blending, κ is a direction noise scale that adds an extra random perturbation in latent space, and s (posterior noise scale), ρ (prior mixing ratio), γ (input blending ratio) are monotonically increasing/decreasing functions of α such that larger α yields stronger perturbations. This preserves long-range temporal dependencies while introducing controlled diversity.

③ **Learned similarity via contrastive embeddings.** For the `SIMILARITY` module, we use TS2Vec (Yue et al., 2022), a contrastive time series representation model pretrained on public datasets, as the embedding network Φ. Each series x is mapped to a fixed-dimensional vector $\Phi(x) \in \mathbb{R}^d$, and the distance for nearest-neighbor voting is $d(x, x') = \|\Phi(x) - \Phi(x')\|_2$, where x is a private sample and x' is a synthetic candidate. Φ is pretrained on public data and kept *fixed* during PETS. Beyond nearest-neighbor voting, the same TS2Vec embeddings are used to evaluate synthetic data quality via Context Fréchet Inception Distance (C-FID) (Jeha et al., 2022), which extends FID (Heusel et al., 2017) to time series by measuring the distributional similarity between real and synthetic samples in the learned embedding space.

The PETS framework is modular by design: `GENERATION`, `VARIATION`, and `SIMILARITY` components can be replaced without modifying the core algorithm.

PETS inherits PE's DP properties and the DP mechanisms from PE (lookahead, Gaussian noise, thresholding). See Appendix B for detailed analysis.

4 EXPERIMENTS

Datasets. We evaluate PETS on two domain pairs, each consisting of an independent public dataset (for training VAE and TS2Vec) and a private dataset (for DP synthesis), with no data overlap between them. *Traffic (highway speed)*: PeMS-BAY (public)[1] and METR-LA (private),[2] both recording highway traffic speed from loop detectors in California. Each sensor is treated as a univariate time series, z-score normalized per sensor, and segmented into windows of length 576 (2 days of data sampled at 5-minute intervals). *Activity (human motion)*: WISDM (public)[3] and UCI-HAR (private),[4] both recording smartphone accelerometer data during daily activities (walking, sitting, etc.). We compute the acceleration magnitude, resample WISDM from 20 Hz to 50 Hz to match UCI-HAR, and extract per-window z-score normalized segments of length 128. Both experiments use unconditional generation.

Baseline. We compare against GS-WGAN (Gradient-Sanitized Wasserstein GAN) (Chen et al., 2020), regarded as the state-of-the-art open-source method for DP synthetic time series generation, a training-time DP method that applies gradient sanitization to a Wasserstein GAN. We use the time-series adaptation from (te Marvelde et al., 2021), which reshapes time series into image-like inputs for a ResNet generator. We use $\delta = 10^{-5}$, 500 training iterations, 100 discriminators, and batch size 32. Since GS-WGAN requires inputs in $[0, 1]$ (Sigmoid output layer), we apply a fixed, data-independent mapping $x_u = (\text{clip}(x_z, -K, K) + K)/(2K)$ with $K = 4.0$ before training, and the inverse $\hat{x}_z = 2K \cdot \text{clip}(\hat{x}_u, 0, 1) - K$ before evaluation, ensuring all metrics are computed on the same z-score scale as PETS.

Metrics. We report C-FID to assess distributional similarity. Lower C-FID indicates that the synthetic data distribution is closer to the real data distribution, suggesting higher fidelity. To evaluate the utility of synthetic data for downstream tasks, we train a simple forecasting model exclusively on synthetic time series and test it on held-out real data, following the Train on Synthetic, Test on Real (TSTR) paradigm. The forecasting model is a 3-layer MLP with hidden dimension 512 and

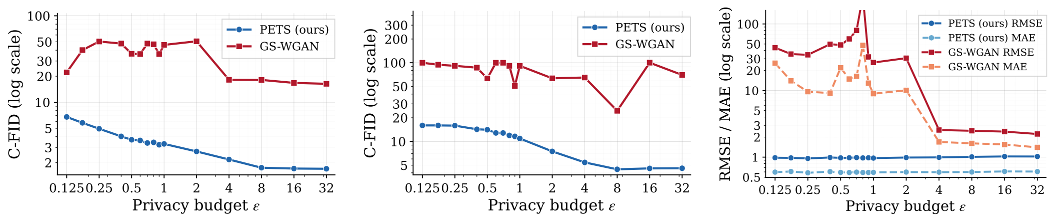
[1] `https://www.kaggle.com/datasets/scchuy/pemsbay`

[2] `https://www.kaggle.com/datasets/annnnguyen/metr-la-dataset`

[3] `https://archive.ics.uci.edu/dataset/507/wisdm+smartphone+and+smartwatch+activity+and+biometrics+dataset`

[4] `https://archive.ics.uci.edu/dataset/240/human+activity+recognition+using+smartphones`

dropout rate 0.1. We use the first 75% of each time series as input and the remaining 25% as the prediction target. The model is optimized with AdamW (learning rate 2×10^{-3}, weight decay 10^{-5}) using MSE loss, a batch size of 256, and training for up to 30 epochs with early stopping (patience of 6 epochs). We report RMSE and MAE on the real test set as measures of downstream forecasting quality.



(a) C-FID vs. privacy budget ϵ on METR-LA (traffic).

(b) C-FID vs. privacy budget ϵ on UCI-HAR (activity).

(c) Forecasting vs. privacy budget ϵ on METR-LA (traffic).

Figure 3: PETS vs. GS-WGAN across privacy budgets ϵ ($\delta = 10^{-5}$). (a)(b) C-FID (lower is better) on traffic and human activity private datasets. (c) Downstream forecasting RMSE/MAE trained on synthetic traffic data, evaluated on real test data (lower is better).

Distributional quality (C-FID). As shown in Figure 3a and Figure 3b, PETS consistently outperforms GS-WGAN across all privacy budgets on both datasets. On METR-LA, PETS achieves a C-FID of 3.38 at $\epsilon = 0.7$, compared to 47.72 for GS-WGAN, achieving $14\times$ lower C-FID. Even at a strict privacy budget of $\epsilon = 0.125$, PETS attains 6.77 versus 22.15 for GS-WGAN. As ϵ increases, PETS C-FID generally decreases from 6.77 ($\epsilon = 0.125$) to 1.70 ($\epsilon = 32$), consistent with the expected privacy-utility trade-off. In contrast, GS-WGAN exhibits high variance and does not improve consistently with larger ϵ, suggesting training instability under DP constraints.

On UCI-HAR, PETS also demonstrates substantial improvements. At $\epsilon = 0.7$, PETS achieves a C-FID of 12.82 while GS-WGAN scores 99.89, with PETS achieving nearly $8\times$ lower C-FID. PETS C-FID again shows a generally decreasing trend from 15.99 ($\epsilon = 0.125$) to 4.57 ($\epsilon = 32$).

Downstream forecasting. Figure 3c evaluates downstream utility by training an MLP forecaster on synthetic METR-LA data and testing on real data. PETS achieves remarkably stable forecasting performance: RMSE remains near $\sim$0.98 and MAE near $\sim$0.59 across all ϵ values, indicating that PETS-generated data preserves temporal patterns useful for forecasting even under strict privacy. At $\epsilon = 1$, PETS achieves approximately $27\times$ lower RMSE than GS-WGAN.

Hyperparameter sensitivity. We conduct ablation studies on METR-LA to assess the robustness and sensitivity of PETS to key hyperparameters, as presented in Table 1. See Appendix D for complete hyperparameter descriptions and variation degree schedules.

Table 1: Hyperparameter sensitivity on METR-LA (traffic, $\epsilon = 5.0$). Bold values achieve the best C-FID among the values tested. For variation schedules, see Appendix D for complete sequences.

Hyperparameter	Values Tested	Best C-FID ↓	RMSE ↓	MAE ↓
Privacy δ	$\mathbf{10^{-4}}, 10^{-5}, 10^{-6}$	1.77	0.964	0.599
Threshold H	$2, 4, 8, \mathbf{12}, 16$	1.90	0.953	0.594
Lookahead k	$0, 4, 8, 12, \mathbf{16}$	1.74	0.979	0.598
# Nearest neighbors	$\mathbf{1}, 3, 5, 7$	1.85	0.952	0.592
Sampling fraction	$0.5, 0.75, 1.0, 1.25, \mathbf{1.5}$	1.73	0.944	0.573
Variation schedule	S1, **S2**, S3, S4, S5, S6, S7	1.90	1.007	0.605

Summary. The experimental results demonstrate that PETS produces high-quality synthetic time series data across both distributional similarity and downstream task performance, indicating that the inference-time DP approach is highly promising for privacy-preserving time series generation. Qualitative examples of PETS-generated traffic time series are provided in Appendix F.

REFERENCES

Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pp. 308–318, 2016.

Dingfan Chen, Tribhuvanesh Orekondy, and Mario Fritz. GS-WGAN: A gradient-sanitized approach for learning differentially private generators. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, pp. 265–284, 2006.

Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data Min. Knowl. Discov.*, 33(4): 917–963, 2019.

Lorenzo Frigerio, Anderson Santana de Oliveira, Laurent Gomez, and Patrick Duverger. Differentially private generative adversarial networks for time series, continuous, and discrete open data. In *ICT Systems Security and Privacy Protection - 34th IFIP TC 11 International Conference, SEC 2019, Lisbon, Portugal, June 25-27, 2019, Proceedings*, pp. 151–164, 2019.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 6626–6637, 2017.

Paul Jeha, Michael Bohlke-Schneider, Pedro Mercado, Shubham Kapoor, Rajbir-Singh Nirwan, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. PSA-GAN: progressive self attention GANs for synthetic time series. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

Zinan Lin, Sivakanth Gopi, Janardhan Kulkarni, Harsha Nori, and Sergey Yekhanin. Differentially private synthetic data via foundation model APIs 1: Images. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, 2024.

Zinan Lin, Tadas Baltrusaitis, Wenyu Wang, and Sergey Yekhanin. Differentially private synthetic data via APIs 3: Using simulators instead of foundation model, 2025.

Penghang Liu, Haibei Zhu, Eleonora Kreacic, and Svitlana Vyetrenko. Privacy-aware time series synthesis via public knowledge distillation. *Trans. Mach. Learn. Res.*, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, pp. 8748–8763, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 10674–10685, 2022.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pp. 3–18, 2017.

Pepijn te Marvelde, Lydia Y. Chen, Aditya Kunar, and Zilong Zhao. Differentially private GAN for time series, 2021. URL `https://resolver.tudelft.nl/uuid:8c4171d0-db68-4235-badb-6e57953162b8`.

Chulin Xie, Zinan Lin, Arturs Backurs, Sivakanth Gopi, Da Yu, Huseyin A. Inan, Harsha Nori, Haotian Jiang, Huishuai Zhang, Yin Tat Lee, Bo Li, and Sergey Yekhanin. Differentially private synthetic data via foundation model APIs 2: Text. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, pp. 54531–54560, 2024.

Zhe Xie, Zeyan Li, Xiao He, Longlong Xu, Xidao Wen, Tieying Zhang, Jianjun Chen, Rui Shi, and Dan Pei. ChatTS: aligning time series with LLMs via synthetic data for enhanced understanding and reasoning. *Proc. VLDB Endow.*, 18(8):2385–2398, 2025.

Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. TS2Vec: Towards universal representation of time series. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pp. 8980–8987, 2022.

A Algorithms

We present the Private Evolution for Time Series (PETS) algorithms, reproduced from Private Evolution (PE) in the DPSDA framework (Abadi et al., 2016) with adaptations for time series data. Algorithm 1 handles both conditional ($|C| > 1$) and unconditional ($C = \{0\}$) generation by processing each class separately—equivalent to running unconditional PETS independently per class. Algorithm 2 describes the DP-protected nearest-neighbor histogram computation.

Algorithm 1: Private Evolution for Time Series (PETS)

Input: Set of private classes C ($C = \{0\}$ for unconditional, $|C| > 1$ for conditional)
Private samples $S_{\text{priv}} = \{(x_i, y_i)\}_{i=1}^{N_{\text{priv}}}$ where x_i is a time series and $y_i \in C$
Number of iterations T
Number of synthetic samples N_{syn} (assume $N_{\text{syn}} \mod |C| = 0$)
Noise multiplier σ, Threshold H
Output: Synthetic dataset S_{syn}

$S_{\text{syn}} \leftarrow \emptyset$
for $c \in C$ **do**
  $private\_samples \leftarrow \{x_i \mid (x_i, y_i) \in S_{\text{priv}}$ and $y_i = c\}$
  $S_0 \leftarrow$ `GENERATION`$(N_{\text{syn}}/|C|)$
  for $t = 1, \ldots, T$ **do**
    $h_t \leftarrow$ `DP_NN_HISTOGRAM`$(private\_samples, S_{t-1}, \sigma, H)$ `// algorithm 2`
    $P_t \leftarrow h_t/\text{sum}(h_t)$ `// Normalize to distribution`
    $S'_t \leftarrow$ sample $N_{\text{syn}}/|C|$ samples with replacement from P_t
    $S_t \leftarrow$ `VARIATION`(S'_t)
  end
  $S_{\text{syn}} \leftarrow S_{\text{syn}} \cup \{(x, c) \mid x \in S_T\}$
end
return S_{syn}

Algorithm 2: DP Nearest Neighbors Histogram

Input: Private time series samples S_{priv}
Generated time series samples $S = \{z_i\}_{i=1}^{n}$
Noise multiplier σ, Threshold H
Distance function $d(\cdot, \cdot)$ computed via `SIMILARITY` (e.g., L2 in embedding space)
Output: DP nearest neighbors histogram on S

$h \leftarrow [0, \ldots, 0]$ (length n)
for $x_{priv} \in S_{priv}$ **do**
  $i \leftarrow \arg\min_{j \in [n]} d(x_{\text{priv}}, z_j)$
  $h[i] \leftarrow h[i] + 1$
end
$h \leftarrow h + \mathcal{N}(0, \sigma^2 I_n)$ `// Add Gaussian noise for DP`
$h \leftarrow \max(h - H, 0)$ `// Element-wise thresholding`
return h

B Privacy Analysis

PETS leverages Private Evolution (PE) and inherits its DP properties. The key insight is that PETS uses the same DP-protected nearest-neighbor histogram mechanism as PE, but differs in the data modality (time series instead of images/text) and the specific choices of generation, variation, and similarity modules. Since these module choices do not affect the histogram voting structure—where each private sample contributes exactly one vote—the privacy analysis remains identical. The core mechanism—DP-protected nearest-neighbor histogram voting—remains unchanged; PETS differs in data modality (time series) and module instantiations (ChatTS, VAE, TS2Vec), which do not affect the voting structure or sensitivity.

B.1 Why PE's Privacy Analysis Applies to PETS

The privacy analysis of PE relies solely on the structure of Algorithm 2: each private sample votes for its nearest neighbor, contributing exactly one count to the histogram, which is then privatized via Gaussian noise. PETS employs this identical mechanism. The specific choices of:

- **Generation method** (ChatTS vs. Stable Diffusion): affects candidate quality, not histogram structure
- **Variation method** (VAE vs. diffusion-based variation): affects offspring distribution, not voting
- **Similarity function** (TS2Vec vs. CLIP): affects nearest-neighbor assignments, but not sensitivity (still one vote per sample)

Since the histogram mechanism is data-modality-agnostic and the sensitivity calculation depends only on the voting rule (not on data type or similarity choice), PE's privacy analysis directly applies to PETS.

Briefly: the histogram has L_2 sensitivity 1 (each private sample contributes exactly one vote); each iteration applies a Gaussian mechanism with noise multiplier σ; across T iterations, adaptive composition yields an effective noise multiplier of $\sigma/\sqrt{T}$; and final (ϵ, δ) parameters follow standard Gaussian mechanism formulas.

B.2 DP Mechanisms

PETS adopts the same DP mechanisms as the PE framework: (1) *lookahead* to compute distances based on averaged embeddings of k variations, improving selection stability; (2) *Gaussian noise* addition ($\mathcal{N}(0, \sigma^2 I)$) to histogram counts for DP protection; and (3) *thresholding* to improve signal-to-noise ratio.

B.3 Formal Privacy Guarantee

The privacy analysis of PETS follows the Private Evolution framework.

Theorem 1 (Gaussian Mechanism for DP). *Let $f : \mathbb{X} \to \mathbb{R}^d$ be a query function with global L_2 sensitivity Δ. For any $\varepsilon \geq 0$ and $\delta \in [0, 1]$, the Gaussian mechanism $M(x) = f(x) + Z$ with $Z \sim \mathcal{N}(0, \sigma^2 I)$ satisfies (ε, δ)-DP if and only if*

$$\Phi\left(\frac{\Delta}{2\sigma} - \frac{\varepsilon\sigma}{\Delta}\right) - e^{\varepsilon}\Phi\left(-\frac{\Delta}{2\sigma} - \frac{\varepsilon\sigma}{\Delta}\right) \leq \delta,$$

where Φ is the cumulative distribution function of the standard normal distribution.

Theorem 2 (Privacy Guarantee of PETS). *Algorithm 1 executed for T iterations with noise multiplier σ (added to each histogram bin) satisfies (ε, δ)-DP if and only if*

$$\Phi\left(\frac{\sqrt{T}}{2\sigma} - \frac{\varepsilon\sigma}{\sqrt{T}}\right) - e^{\varepsilon}\Phi\left(-\frac{\sqrt{T}}{2\sigma} - \frac{\varepsilon\sigma}{\sqrt{T}}\right) \leq \delta.$$

Derivation. The L_2 sensitivity of the histogram computed in algorithm 2 is $\Delta = 1$, as each private sample contributes exactly one vote. With Gaussian noise of scale σ added per bin, each iteration constitutes a Gaussian mechanism with sensitivity 1 and noise multiplier σ. Over T iterations, algorithm 1 applies this mechanism adaptively, where each iteration's outcome depends on the previous synthetic dataset. By the composition theorem for Gaussian mechanisms under adaptive composition, the cumulative privacy cost is equivalent to that of a single Gaussian mechanism with sensitivity 1 and effective noise scale $\sigma/\sqrt{T}$. Applying Theorem 1 with $\Delta = 1$ and $\sigma \leftarrow \sigma/\sqrt{T}$ yields the stated condition. □

Conditional generation. For the conditional version (Algorithm 1 with $|C| > 1$), each class is processed independently. Adding/removing one sample (x, y) affects only class y's histogram, with bounded impact as above. Thus, the conditional version satisfies the same (ε, δ)-DP guarantee.

C Detailed Experiment Configurations

PETS (Traffic). $\delta = 10^{-5}$, $T = 61$ iterations, histogram threshold $H = 8.0$, lookahead degree $k = 4$, variation degrees: $[40, 35, 30, 25, 20, 15, 15, 10, 10, 10, 5 \times 51]$ (61 total).

PETS (Human activity). $\delta = 10^{-5}$, $T = 20$ iterations, histogram threshold $H = 4.0$, lookahead degree $k = 8$, variation degrees: $[25, 20, 20, 15, 15, 15, 10, 10, 10, 10, 5 \times 11]$ (21 total).

D Ablation Study: Hyperparameter Sensitivity

Default configuration for all ablation experiments. Unless explicitly varied, all experiments use: dataset METR-LA (traffic), $\delta = 10^{-5}$, $\epsilon = 5.0$, $T = 21$ iterations (or 11 for certain variation schedules), lookahead degree $k = 4$, histogram threshold $H = 8.0$, nearest neighbors $= 1$, sampling fraction $= 1.0$, variation schedule S3, rescale policy: public generation only, variation spike preservation $= 0.0$. 30% of private data is reserved for downstream forecasting evaluation.

Hyperparameter descriptions.

- δ: Privacy parameter for approximate DP. Smaller δ provides stronger privacy guarantees.
- *Histogram threshold* H: Post-processing threshold for the DP histogram. Bins with counts below H are set to zero, reducing noise impact.
- *Lookahead degree* k: Number of top-k nearest neighbors considered for selection. Higher k smooths selection but may reduce diversity.
- *Number of nearest neighbors*: Number of nearest neighbors each private sample votes for (default 1 for standard PE).
- *Sampling fraction*: Ratio of synthetic population size to private dataset size. Values > 1 increase population diversity.
- *Variation schedule*: Sequence of variation degrees α across T iterations, controlling the mutation strength at each step.

Variation degree schedules.

- S1: $[20 \times 3, 15 \times 3, 10 \times 2, 5 \times 3]$ (11 iterations)
- S2: $[25, 25, 20, 20, 15, 15, 10, 10, 8, 6, 5]$ (11 iterations)
- S3: $[40, 35, 30, 25, 20, 15, 10, 5 \times 4]$ (11 iterations)
- S4: $[45, 40, 35, 30, 25, 20, 15, 10, 10, 5, 5]$ (11 iterations)
- S5: $[50, 45, 40, 35, 30, 25, 20, 15, 10, 5, 5]$ (11 iterations)
- S6: $[60, 55, 50, 45, 40, 35, 30, 25, 20, 15, 10]$ (11 iterations)
- S7: $[95, 85, 75, 65, 55, 45, 35, 25, 20, 15, 10]$ (11 iterations)

E Wasserstein Distance Definition

Given discrete point sets S_{priv} and S_{syn}, the Wasserstein-p distance between the uniform distributions on these sets is defined as:

$$W_p(S_{\text{priv}}, S_{\text{syn}}) = \left(\inf_{\gamma \in \Pi(\hat{\mu}_{\text{priv}}, \hat{\mu}_{\text{syn}})} \mathbb{E}_{(x,x') \sim \gamma}[d(x, x')^p] \right)^{1/p}, \tag{5}$$

where $\hat{\mu}_{\text{priv}} = \frac{1}{N_{\text{priv}}} \sum_{i=1}^{N_{\text{priv}}} \delta_{x_i}$ and $\hat{\mu}_{\text{syn}} = \frac{1}{N_{\text{syn}}} \sum_{j=1}^{N_{\text{syn}}} \delta_{x'_j}$ are empirical distributions, $\Pi(\hat{\mu}_{\text{priv}}, \hat{\mu}_{\text{syn}})$ denotes the set of all couplings with these marginals, and $d(\cdot, \cdot)$ is a distance function between samples. By minimizing $W_p(S_{\text{priv}}, S_{\text{syn}})$, we ensure that S_{syn} captures the distribution of S_{priv} while maintaining temporal properties such as autocorrelation and seasonality.

F Qualitative Examples

Figure 4 presents a qualitative comparison of real private samples from the METR-LA dataset and synthetic samples generated by PETS. Each plot shows a randomly selected time series segment, with the x-axis representing time. The visualization demonstrates that PETS-generated synthetic time series preserve temporal patterns and statistical properties of the original data at the distributional level. Note that the synthetic samples are not one-to-one matches with specific real samples; rather, they reflect the overall distribution and characteristics of the private dataset.

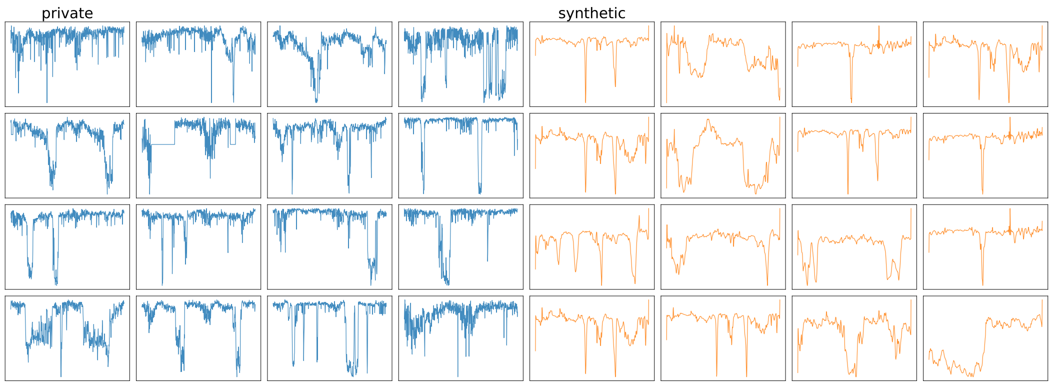


Figure 4: Qualitative comparison of real private data (METR-LA) and PETS-generated synthetic samples at $\epsilon = 1.0$ ($\delta = 10^{-5}$). Samples are randomly selected from each dataset.