PETS: INFERENCE-TIME DIFFERENTIALLY PRIVATE SYNTHETIC TIME SERIES GENERATION

Anonymous authors
Paper under double-blind review

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 is not used to train generative models, but only to guide the selection of synthetic outputs at inference time, improving distributional fidelity while satisfying a privacy-budget constraint. PETS is modular by design, enabling modular adaptation without training on private data. Building on top of PE, we instantiate three concrete schemes, ChatTS+VAE, Diffusion-TS, and TimeCraft, each combining a generation module, a variation module, and a shared contrastive embedding for similarity-driven selection. Empirically, PETS achieves a mean $6.0\times$ lower C-FID than state-of-the-art (SOTA) DP methods across four public–private dataset pairs over 13 privacy budgets ($\varepsilon = 0.125$ to 32, plus ∞), with gains up to $17.6\times$ at $\varepsilon = 0.175$. For downstream tasks, PETS achieves a mean $1.4\times$ the activity classification accuracy of SOTA DP methods over these privacy budgets, showing PETS produces high-quality DP synthetic time series with strong utility–privacy trade-offs.

1 INTRODUCTION

Time series data is ubiquitous across numerous domains, capturing temporal patterns in healthcare, finance, and industrial processes (Fawaz et al., 2019). The recent rise of time series foundation models has further amplified the demand for diverse, high-quality datasets at scale, since model pre-training and fine-tuning hinges on broad and realistic data coverage. 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), leaving much of the most valuable pretraining data inaccessible.

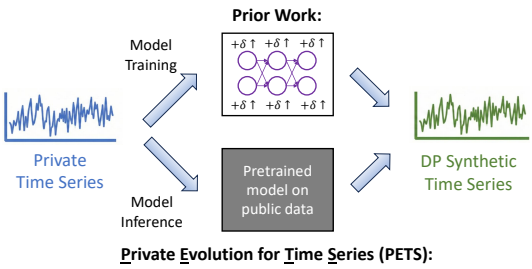


Figure 1: Comparison of prior work vs. PETS. **Top**: Prior work requires training or fine-tuning models on private data with DP-SGD. **Bottom**: PETS operates at inference time with blackbox access to pre-trained models, using private data only for DP-protected voting.

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. We provide a more comprehensive review of the landscape of DP time series in Appendix D.

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.

The community has proposed DP generation frameworks for images and text data that operate entirely at inference-time, such as Private Evolution (PE) (Lin et al., 2024; Xie et al., 2024; Lin et al., 2025). These methods leverage pre-trained foundation models, e.g., Stable Diffusion (Rombach et al., 2022), and skip the need for pre-training, circumventing the aforementioned issues. Unfortunately, the time series data modality still lacks comparable methods. High-quality API-based generative model for time-series are scarce, making it challenging to apply PE-style frameworks directly.

To overcome these limitations, we present **Private Evolution for Time Series (PETS)**, the first inference-time framework for DP synthetic time series generation. PETS does not require any model training or fine-tuning on private data. The framework leverages pre-trained generative models to generate synthetic DP time series candidates during inference. Figure 1 illustrates the idea. Private data is never shared with the models, thus it remains within the client boundaries. Clients use private data to locally drive a similarity-based selection of synthetic candidates, using Private Evolution. This paradigm offers three key advantages: (1) no expensive DP-constrained training is needed, (2) private data never enters the training pipeline, (3) the framework adapts to new domains by plugging in different modules without retraining. Table 1 summarizes how PETS combines a DP guarantee, no training on private data, and public-data priors with generation and variation modules designed for time-series synthesis.

Table 1: Comparison of synthesis paradigms. Method categories are detailed in Appendix F.5.

Methods	DP guarantee	No private-data training	Public-data priors	Time-series synthesis
Non-DP TS generators	✗	✗	✗	✓
DP-SGD TS generators	✓	✗	✗	✓
Public-pretrained DP-SGD	✓	✗	✓	✓
DPSDA (images/text)	✓	✓	✓	✗
PETS	✓	✓	✓	✓

PETS is based on three pluggable modules: a `GENERATION` module that produces an initial synthetic population, a `VARIATION` module that perturbs DP-selected candidates, and a `SIMILARITY` module that maps private and synthetic samples into an embedding space. PETS leverages Private Evolution (PE) (Lin et al., 2025) to iteratively bias the creation of synthetic candidates toward the private data distribution, while meeting privacy budget constraints.

We provide three concrete implementations for the `GENERATION`, `VARIATION`, and `SIMILARITY` modules, and compare performance. We evaluate the following schemes: *ChatTS+VAE* pairs a training-free rule-based generator (Xie et al., 2025) with a public-data VAE (Kingma & Welling, 2014) variation operator; *Diffusion-TS* (Yuan & Qiao, 2024) uses a single diffusion model for both initial sampling and SDEdit-style structure-preserving variation, with optional classifier guidance on labelled data; and *TimeCraft* (Microsoft Research Lab Asia, 2025) extends the diffusion scheme with TimeDP domain-prompt and class-label conditioning. We inherit PE's differential privacy mechanism, therefore our methodological contribution is to construct the generation–variation interface needed to make inference-time private evolution effective for time series, where comparable generative APIs are not readily available. Specifically, we equip a training-free rule-based generator with a public-data-trained VAE variation operator and adapt time-series diffusion models into structure-preserving variation operators through SDEdit with degree-to-timestep schedules.

Our main contributions include:

- 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 construct time-series generation and structure-preserving variation interfaces using rule-based generation, VAE perturbations, and diffusion-based editing, coupled with a fixed contrastive similarity encoder. All learned components are trained exclusively on public data.
- We demonstrate PETS' strong performance across four public–private dataset pairs against six DP and two non-DP baselines. Across 13 privacy budgets ($\varepsilon = 0.125$ to 32, plus ∞), PETS achieves a mean $6.0\times$ lower C-FID than SOTA DP methods, with gains up to $17.6\times$ at $\varepsilon = 0.175$. Independent metrics and domain-mismatch experiments assess robustness.
- We show that PETS delivers high downstream utility across imputation, classification, forecasting, and anomaly detection. Across 13 privacy budgets, PETS achieves a mean $1.4\times$ the activity classification accuracy of SOTA DP methods and a mean $3.4\times$ that of DP-SGD methods.

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 by the addition or removal of one record, and for any subset S of possible outputs, the following holds:

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

Throughout this paper, one record corresponds to one fixed-length time-series window, with an associated class label when labels are available.

Problem Formulation. Let $S_{\text{priv}} = \{(x_i, y_i)\}_{i=1}^{N_{\text{priv}}}$ denote a private time-series dataset, where $x_i \in \mathbb{R}^L$ for univariate windows or $x_i \in \mathbb{R}^{L \times D}$ for multivariate windows (with window length L and channel count D), and $y_i \in C$ is the optional, associated class label for the i^{th} sample. In the unconditional setting, we take $C = \{0\}$ and omit labels. The goal is to release a synthetic dataset $S_{\text{syn}} = \{(x'_j, y'_j)\}_{j=1}^{N_{\text{syn}}}$ whose distribution and downstream utility are close to those of S_{priv}, while satisfying (ε, δ)-DP with respect to S_{priv}.

Inference-time DP setting. PETS performs inference-time DP synthesis. It may use a public or proxy dataset S_{pub} to pretrain or configure the generation, variation, and similarity modules, but no private record is used to train or finetune these modules. Crucially, private data never leaves the client side, entering the process only through a DP-protected nearest-neighbor voting step: private windows vote for nearby synthetic candidates under a fixed embedding distance $d_\Phi(x, x') = \|\Phi(x) - \Phi(x')\|_2$, the resulting histogram is privatized, and the selected candidates are varied to form the next synthetic population. This makes the privacy guarantee depend on the histogram mechanism rather than on the internal training procedure or weights of the generative models. As a statistical objective, we seek to make the empirical distribution of S_{syn} close to that of S_{priv} under time-series-aware distances, such as the Wasserstein-p distance defined in Appendix C, and under downstream train-on-synthetic-test-on-real evaluation.

3 PETS DESIGN

Overview. Figure 2 illustrates the PETS pipeline. **Step 1**: An initial population of N_{syn} synthetic time series is produced by the `GENERATION` module (block Ⓘ in the figure). The population is then refined over T iterations. Let S_t denote the synthetic population after PE iteration t, with S_0 the initial population. At each iteration: **Step 2.1**, every private sample finds its nearest synthetic neighbor by computing distances in the `SIMILARITY` embedding space (Ⓘ) 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 the `VARIATION` module (Ⓘ) to produce the next-generation population. After T iterations, the final population S_T constitutes the DP synthetic dataset. The complete PETS algorithm is provided in Appendix A.

PETS draws inspiration from Private Evolution (PE) in other domains but critically adapts the methodology for time series via three modular components (`GENERATION`, `VARIATION`, and `SIMILARITY`) detailed below. `GENERATION` and `VARIATION` each admit several implementations: a PETS scheme is obtained by selecting one implementation per component. Table 2 lists the three schemes we instantiate in this paper. PETS supports both *unconditional* and *conditional* (per-class) generation: in the conditional setting, the `GENERATION` and `VARIATION` modules are made class-aware through scheme-specific mechanisms (classifier guidance for the DiffusionTS-based scheme, class-label embeddings for the TimeCraft-based scheme), and the PE loop is run independently per class (Algorithm 1).

3.1 `GENERATION` OF INITIAL POPULATION

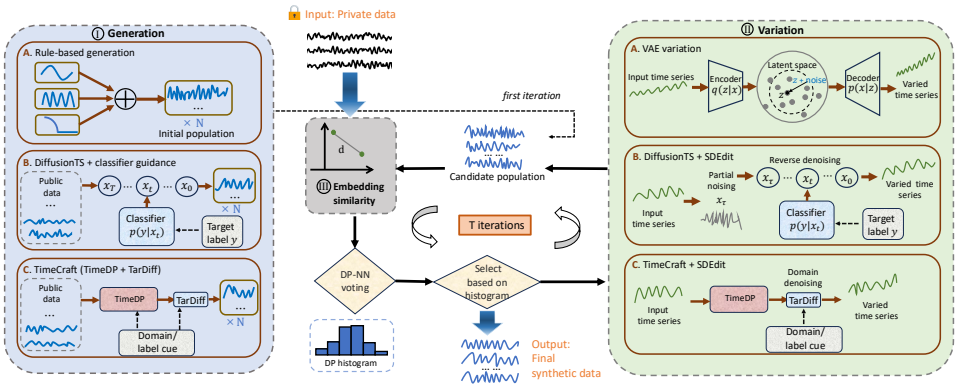


Figure 2: Overview of the PETS framework.

The GENERATION module constructs an initial synthetic population from public-data priors or rule-based templates.

Table 2: Module choices for the PETS schemes.

Scheme	GENERATION	VARIATION	SIMILARITY
ChatTS+VAE	ChatTS	VAE	TS2Vec
Diffusion-TS	DiffusionTS	DiffusionTS-SDEdit	TS2Vec
TimeCraft	TimeCraft	TimeCraft-SDEdit	TS2Vec

I-A. Rule-based generation. We use an LLM-free ChatTS generator (Xie et al., 2025) to produce diverse seed candidates from interpretable attributes, requiring no training data. Attributes include trend, seasonality, frequency, noise, and local-change:

$$x = \mathcal{G}(\theta), \qquad \theta = (\theta_{\text{tr}}, \theta_{\text{seas}}, \theta_{\text{freq}}, \theta_{\text{noise}}, \theta_{\text{loc}}). \tag{2}$$

In our implementation, seasonal patterns are sampled from no-periodic, sinusoidal, square, and triangular fluctuations; trend and noise types are drawn from the default ChatTS attribute set (a weighted vocabulary of categorical pattern types per attribute); and up to three local changes, such as spikes, shakes, or sudden level shifts, are inserted at random positions.

I-B. DiffusionTS-based generation. We train DiffusionTS (Yuan & Qiao, 2024) on public time-series data and generates samples from the learned public/proxy distribution. The diffusion model is optimized to predict the injected noise from noisy inputs x_t:

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I). \tag{3}$$

In the conditional setting, we apply classifier guidance: a classifier $p_\psi(y \mid x_t, t)$ is trained on noisy samples x_t at randomly-chosen timesteps t, and its score steers sampling toward the target class y:

$$\widetilde{s}(x_t, t, y) = s_\phi(x_t, t) + s\, \nabla_{x_t} \log p_\psi(y \mid x_t, t). \tag{4}$$

I-C. TimeCraft-based generation. We use a TimeCraft-style generator (Microsoft Research Lab Asia, 2025) trained on public data, combining TimeDP (Huang et al., 2025) and TarDiff (Deng et al., 2025). The model uses TimeDP/domain-prompt conditioning, with class-label embeddings added in the conditional setting. At each training step, a public time-series window is diffused to a randomly-chosen timestep, and the model is trained to predict the added noise under the corresponding domain (and class) conditions.

3.2 VARIATION MODULE: STRUCTURE-PRESERVING VARIATION

The VARIATION module refines a selected candidate x by applying controlled perturbations. A variation degree $\alpha \in [0, 100]$ controls the trade-off: larger α injects more diversity, smaller α preserves more of the input's temporal structure.

II-A. VAE-based variation. We use a Conv1D VAE pre-trained on public time-series windows. During training, each input is normalized at the series level and reconstructed through an encoder–decoder model with reconstruction, KL, frequency-domain, finite-difference, peak-matching, and curvature losses. At inference time, the VAE is kept fixed and only used as a public-data variation operator.

For a candidate x, we encode its standardized version into the latent space and generate a perturbed latent representation by adding posterior noise, mixing with a prior sample, and blending the decoded output back with the original input:

$$x' = (1 - \gamma_\alpha)\, p_\theta(z_\alpha) + \gamma_\alpha x. \tag{5}$$

Here z_α is obtained by latent-space perturbation, while the noise scale, prior-mixing ratio, and input-blending weight are deterministic functions of α. Larger α induces stronger latent perturbation and weaker input blending. We additionally apply mild amplitude, bias, and trend jitter with amplitude calibration to avoid cumulative scale drift.

II-B. DiffusionTS-based variation via SDEdit. We use the pretrained DiffusionTS model as an SDEdit variation operator. Rather than sampling from pure noise, we first normalize the candidate series and partially diffuse it to a timestep $\tau = \tau(\alpha)$:

$$x_\tau = \sqrt{\bar{\alpha}_\tau}\, x + \sqrt{1 - \bar{\alpha}_\tau}\, \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I). \tag{6}$$

The model then reverses the diffusion process from τ to 0 to obtain x', with classifier guidance applied in the conditional setting (reusing the classifier from I-B above). Thus, small α produces local perturbations, whereas large α approaches stronger regeneration.

II-C. TimeCraft-based variation via SDEdit. We use the TimeCraft model as an SDEdit variation operator, conditioned on a TimeDP/domain prompt computed from public or proxy domain samples. The candidate series is partially noised to $\tau = \tau(\alpha)$, and the reverse process is run with the domain prompt and, in the conditional setting, the class-label embedding y:

$$x' \sim p_\theta(x_0 \mid x_\tau, c_{\text{domain}}, y). \tag{7}$$

This produces target-aware variations that preserve the coarse structure of the candidate while adapting it toward the desired domain (and class).

3.3 SIMILARITY MODULE: LEARNED SIMILARITY VIA CONTRASTIVE EMBEDDINGS

For the SIMILARITY module, we use TS2Vec (Yue et al., 2022), a contrastive time series representation model pre-trained 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 $d_\Phi(x, x') = \|\Phi(x) - \Phi(x')\|_2$, where x is a private sample and x' is a synthetic candidate. Φ is pre-trained 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.

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

4 EXPERIMENTS

Datasets. We evaluate PETS on four domains, each pairing a public dataset (used to pretrain PETS' generation, variation, and similarity modules) with a disjoint private dataset (used for Private Evolution voting). The **traffic** pair uses PeMS-BAY (scchuy, 2022; Li et al., 2018) (public) and METR-LA (annnnguyen, 2024; Li et al., 2018) (private); the **crypto** pair uses BTC/USDT (Binance, 2021-2026a) (public) and ETH/USDT (Binance, 2021-2026b) (private); both are unlabelled. The **activity** pair uses WISDM (Weiss, 2019) (public) and UCI-HAR (Anguita et al., 2012; 2013) (private), with 4-class labels (*Walking, Stairs, Sitting, Standing*); the **ECG** pair uses MIT-BIH Arrhythmia (Moody & Mark, 2005; 2001) (public) and INCART (Yakushenko, 2008) (private), with AAMI (Association for the Advancement of Medical Instrumentation, 1999) 3-class beat labels: (*Normal, Supraventricular ectopic, Ventricular ectopic*). Across all domains, raw signals are converted into univariate fixed-length windows and run through a domain-specific standardisation, followed by a shared public+private percentile clip and min–max scaling to [0,1] within each pair. Pre-processing details are in Appendix F.1.

Fidelity metrics. We compare synthetic and private training windows using five lower-is-better metrics. **C-FID**, our primary fidelity metric, measures the Fréchet distance between distributions

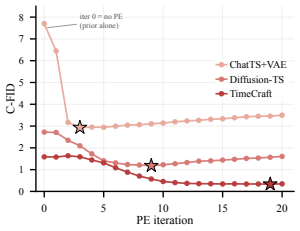


Figure 3: PETS C-FID trajectory on activity at $\varepsilon = 1$. ★ marks the C-FID-best iteration.

Dataset	PETS scheme	$\varepsilon = 0.5$			$\varepsilon = 1$			$\varepsilon = 2$			$\varepsilon = \infty$		
		C-FID↓	FID_M↓	ACD↓	C-FID↓	FID_M↓	ACD↓	C-FID↓	FID_M↓	ACD↓	C-FID↓	FID_M↓	ACD↓
Traffic	ChatTS+VAE	4.62	0.312	0.814	3.30	0.271	1.168	2.65	0.246	0.972	2.39	0.247	0.863
	Diffusion-TS	9.28	0.502	1.308	8.36	0.491	0.747	8.20	0.503	0.825	8.15	0.491	0.462
	TimeCraft	**2.00**	**0.213**	**0.379**	**1.62**	**0.194**	**0.310**	**1.35**	**0.192**	**0.391**	**1.18**	**0.178**	**0.261**
Activity	ChatTS+VAE	3.04	0.936	0.636	2.93	0.936	0.556	2.48	0.918	0.621	1.84	0.913	0.595
	Diffusion-TS	1.53	0.327	0.430	1.18	0.281	0.310	0.97	0.244	0.317	0.85	0.234	0.273
	TimeCraft	**0.45**	**0.148**	**0.226**	**0.33**	**0.140**	**0.153**	**0.30**	**0.122**	**0.134**	**0.26**	**0.121**	**0.167**
ECG	ChatTS+VAE	3.68	0.567	0.182	3.31	0.553	0.153	3.00	0.591	0.137	2.73	0.611	0.148
	Diffusion-TS	4.64	0.186	0.386	3.37	0.162	0.179	3.36	0.158	0.028	3.10	0.158	**0.030**
	TimeCraft	**2.47**	**0.136**	**0.032**	**1.73**	**0.140**	**0.038**	**1.46**	**0.147**	**0.021**	**1.17**	**0.142**	0.063
Crypto	ChatTS+VAE	0.76	0.535	0.361	0.58	0.515	0.373	**0.54**	0.500	0.353	0.61	0.502	0.357
	Diffusion-TS	1.67	0.089	0.097	1.05	0.084	0.099	0.74	0.073	0.096	**0.51**	0.058	0.101
	TimeCraft	**0.68**	**0.021**	**0.022**	**0.33**	**0.014**	**0.024**	0.66	**0.018**	**0.024**	0.80	**0.024**	**0.023**

Table 3: Performance of three PETS schemes on four time-series datasets. **Bold** indicates the best scheme for each <dataset, ε, metric> setting.

of TS2Vec embeddings (Jeha et al., 2022; Yue et al., 2022); **ACD** measures differences in temporal autocorrelation. $\textbf{FID}_\text{M}$ and $\textbf{FID}_\text{DAE}$ measure Fréchet distances in the embedding spaces of frozen MOMENT-1-large (Goswami et al., 2024) and a Conv1D denoising autoencoder (DAE) trained on each pair's public data, respectively. $\textbf{MMD}^2$ measures distributional differences directly on the preprocessed windows using a kernel two-sample statistic (Gretton et al., 2012). Definitions and computation details are in Appendix F.7.

Downstream tasks. We evaluate forecasting, imputation, classification, and anomaly detection under the *train-on-synthetic, test-on-real* (TSTR) protocol. We reserve 30% of the private windows for testing and use the remaining 70% for synthesis. A *train-on-real, test-on-real* (RTR) reference is trained on the private training windows and evaluated on the same test split. We report RMSE for forecasting and imputation, accuracy for activity classification, macro-F_1 for ECG classification, and AUROC for anomaly detection. Task definitions and evaluation details are in Appendix F.8.

Baselines. We compare PETS against eight baselines spanning DP-SGD-trained, PE-based DP, and non-DP utility-oriented generation. To keep the comparison fair, all baselines consume the same PETS-processed windows and are evaluated through the same similarity and downstream-task pipeline. Our DP-SGD-trained baselines are **dp-GAN** (Frigerio et al., 2019), **GS-WGAN** (Chen et al., 2020; te Marvelde et al., 2021), **GlucoSynth** (Lamp et al., 2023) (ECG-only), and **Pub2Priv** (Liu et al., 2025). Our PE-based DP baselines are **DPSDA-image** (Lin et al., 2024) and **DPSDA-text** (Xie et al., 2024), both inference-time methods like PETS. We also evaluate two non-DP utility baselines: **TimeDiff** (Tian et al., 2024) and **TimeGAN** (meithnavlakha, 2024; Yoon et al., 2019). dp-GAN, Pub2Priv, and GlucoSynth do not have publicly available code, so we implemented them ourselves following their papers. Per-baseline descriptions are in Appendix F.5.

For PETS and the PE-based baselines, we report each fidelity metric and downstream task at its own best evaluated iteration $t \geq 1$, while PETS releases the final population S_T.

4.1 PETS Behavior and Scheme Comparison

This subsection analyzes the three PETS schemes (Table 2) along two axes: PE convergence dynamics (Figure 3) and synthesis-quality comparison across all four (public, private) dataset pairs (Table 3).

PETS converges within a small iteration budget for all three schemes. Figure 3 traces C-FID per PE iteration on activity at $\varepsilon = 1$: TimeCraft's C-FID drops from 1.59 at $t = 0$ to a best of 0.33 at $t = 19$; Diffusion-TS reaches its best at $t = 9$; ChatTS+VAE reaches its best earliest, at $t = 3$. This is intrinsic to PETS' loop: DP-vote-weighted `VARIATION` pulls the synthetic distribution toward the private one, while its decaying degree shrinks per-step movement close to zero. The small iteration budget makes per-iteration GPU time a useful runtime proxy (Figure 12).

Which scheme performs best in PETS? TimeCraft achieves the lowest score in 45 of the 48 <dataset, ε, metric> settings in Table 3. Across all 16 dataset–budget settings, its median margins over the better of the other two schemes are approximately $2.0\times$ for C-FID, $1.7\times$ for FID_M, and $2.3\times$ for ACD. At the tighter privacy budgets $\varepsilon \in \{0.5, 1\}$, TimeCraft leads on all three metrics across all four datasets, showing that its advantage persists under strong privacy constraints. The broad agreement on TimeCraft's lead across two embedding spaces and raw-window autocorrelation

Table 4: Synthesis-quality comparison of PETS with DP and non-DP baselines on four dataset pairs. Each ε block reports C-FID and the dataset's primary downstream metric. Grey "RTR" denotes the train-on-real reference for the downstream metric. The PETS row reports the per-cell best of the three schemes in subsection 4.1. For DPSDA, "uncond." and "cond." denote the unconditional and class-conditional versions. **Bold** indicates the best value per column.

Dataset & metric	Method	$\varepsilon = 0.5$		$\varepsilon = 1$		$\varepsilon = 2$		$\varepsilon = \infty$	
		C-FID↓	metric	C-FID↓	metric	C-FID↓	metric	C-FID↓	metric
Activity acc(%)↑ RTR = 82.2	dp-GAN	10.95	20.6	12.42	20.6	11.03	20.6	3.76	20.6
	GS-WGAN	45.44	20.6	50.32	20.6	45.44	20.6	2.48	20.6
	GlucoSynth	—	—	—	—	—	—	—	—
	Pub2Priv	3.16	20.6	3.41	20.6	3.17	20.6	1.68	20.6
	DPSDA-img (uncond.)	6.42	42.4	5.23	34.3	4.26	43.5	4.16	38.8
	DPSDA-img (cond.)	8.35	48.2	6.66	35.7	5.86	47.1	3.91	41.7
	DPSDA-txt (uncond.)	24.22	32.4	20.83	34.5	15.60	32.7	2.57	41.0
	DPSDA-txt (cond.)	25.16	47.0	21.66	51.2	18.50	46.4	6.30	57.6
	PETS	**0.45**	**69.3**	**0.33**	**69.0**	**0.30**	**69.7**	**0.26**	**73.8**
	TimeDiff	—	—	—	—	—	—	1.62	20.6
	TimeGAN	—	—	—	—	—	—	5.18	20.6
Traffic fcst-RMSE↓ RTR = 0.131	dp-GAN	20.53	0.259	18.36	0.670	15.10	0.199	13.27	0.206
	GS-WGAN	56.32	0.740	57.99	0.347	57.99	0.347	6.22	0.179
	GlucoSynth	—	—	—	—	—	—	—	—
	Pub2Priv	12.40	0.196	8.90	0.199	8.69	0.191	**0.52**	**0.149**
	DPSDA-img (uncond.)	16.85	0.215	15.81	0.212	13.74	0.215	14.04	0.219
	DPSDA-txt (uncond.)	37.73	0.216	35.59	0.215	32.22	0.212	6.66	0.203
	PETS	**2.00**	**0.166**	**1.62**	**0.171**	**1.35**	**0.166**	1.18	0.167
	TimeDiff	—	—	—	—	—	—	5.98	0.193
	TimeGAN	—	—	—	—	—	—	13.60	0.198

Dataset & metric	Method	$\varepsilon = 0.5$		$\varepsilon = 1$		$\varepsilon = 2$		$\varepsilon = \infty$	
		C-FID↓	metric	C-FID↓	metric	C-FID↓	metric	C-FID↓	metric
ECG AUROC↑ RTR = 0.953	dp-GAN	34.28	0.566	38.30	0.516	29.19	0.533	24.88	0.535
	GS-WGAN	63.97	0.751	55.03	0.798	63.97	0.751	9.60	0.890
	GlucoSynth	29.58	0.486	24.37	0.489	30.53	0.395	33.86	0.514
	Pub2Priv	27.43	0.565	26.46	0.515	25.84	0.551	1.67	0.952
	DPSDA-img (uncond.)	3.91	0.965	2.81	0.963	2.41	0.965	2.24	**0.964**
	DPSDA-img (cond.)	3.74	0.966	2.90	0.963	2.29	0.962	2.06	**0.964**
	DPSDA-txt (uncond.)	19.30	0.899	16.60	0.895	13.83	0.902	5.60	0.895
	DPSDA-txt (cond.)	20.22	0.906	16.44	0.899	15.07	0.905	4.95	0.890
	PETS	**2.47**	**0.967**	**1.73**	**0.966**	**1.46**	**0.966**	**1.17**	**0.964**
	TimeDiff	—	—	—	—	—	—	11.25	0.883
	TimeGAN	—	—	—	—	—	—	26.40	0.611
Crypto imp-RMSE↓ RTR = 0.110	dp-GAN	23.88	0.150	44.29	0.184	9.52	0.113	5.71	**0.110**
	GS-WGAN	62.63	0.513	62.63	0.513	62.63	0.513	3.66	0.128
	GlucoSynth	—	—	—	—	—	—	—	—
	Pub2Priv	8.95	0.116	6.92	0.112	5.09	0.112	0.59	0.111
	DPSDA-img (uncond.)	6.28	0.122	5.35	0.123	3.80	0.123	4.53	0.123
	DPSDA-txt (uncond.)	43.22	0.111	43.15	0.111	42.69	0.111	11.24	0.111
	PETS	**0.68**	**0.110**	**0.33**	**0.110**	**0.54**	**0.110**	**0.51**	**0.110**
	TimeDiff	—	—	—	—	—	—	2.22	**0.110**
	TimeGAN	—	—	—	—	—	—	5.63	0.120

suggests that its advantage is robust to metric choice. We therefore adopt TimeCraft as the default backend.

4.2 Comparison of PETS vs. DP and Non-DP Baselines

We compare PETS with eight baselines (six DP and two non-DP) on synthesis quality across four public–private dataset pairs (Table 4, Figure 4). For both DPSDA-image and DPSDA-text, we evaluate two versions: unconditional PE on the whole private set and class-conditional PE run separately within each class. The latter requires labels and is evaluated only on activity and ECG. Per-baseline configurations are in Appendix F.6.

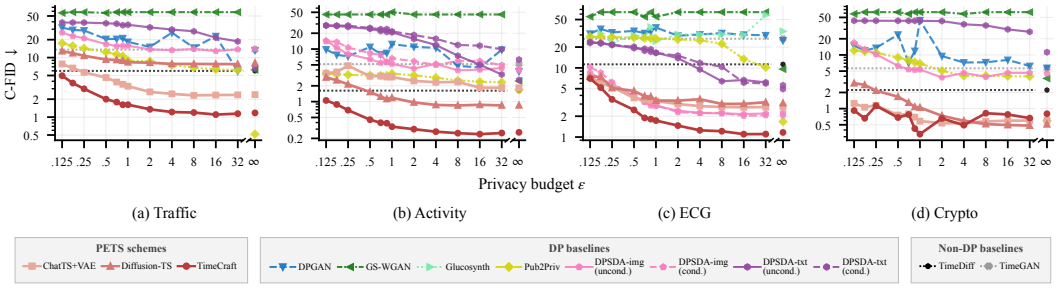


Figure 4: C-FID (lower is better) vs. privacy budget ε across all four datasets, comparing the three PETS schemes against six DP baselines and two non-DP references.

PETS achieves a mean 6.0× lower C-FID than the strongest DP baseline across the full privacy-budget sweep. Across four datasets and all 13 evaluated budgets ($\varepsilon = 0.125$ to 32, plus ∞), the best PETS scheme per setting outperforms every DP baseline in 51 of 52 settings (Figure 4), with a maximum margin of 17.6× on crypto at $\varepsilon = 0.175$. The median margin across these settings is 6.2×. At $\varepsilon = 1$, PETS achieves 5.5×, 10.2×, 1.6×, and 16.0× lower C-FID on traffic, activity, ECG, and crypto, respectively. Against the two non-DP baselines at $\varepsilon = \infty$, PETS wins on all four datasets, with a mean margin of 6.3× (Table 4). The activity samples also qualitatively resemble the real windows (Figure 5).

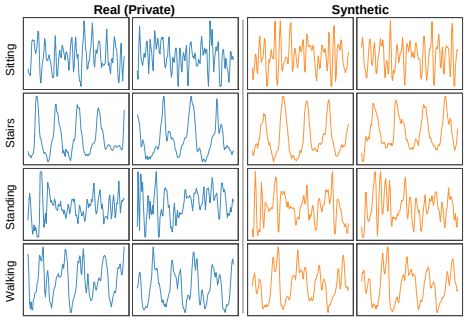


Figure 5: Real (private) vs. PETS-synthesized activity windows at $\varepsilon = 1$.

The lead persists with TimeCraft as the sole backend. Fixing TimeCraft across datasets and budgets, PETS still outperforms every DP baseline in all 48 finite-budget settings (4 datasets $\times$ 12 finite budgets; Figure 4). Its advantage therefore extends across tight and relaxed privacy budgets.

PETS significantly outperforms the SOTA for several reasons. (1) PETS adds DP noise to histogram votes, preserving public-pretrained weights, whereas DP-SGD perturbs gradients during training. (2) Same-domain public data provides temporal priors for initialization and structure-preserving refinement toward the private distribution; we examine the role of domain alignment in subsection 4.4. (3) Direct time-series generation and variation avoid the quantization and reconstruction errors that image/text conversion can introduce in the DPSDA adaptations. (4) Class-conditional generation and variation in TimeCraft and Diffusion-TS help retain class-specific modes. Its modular design further offers rule-based generation with VAE variation, classifier-guided diffusion, and domain- and class-conditional diffusion to suit different data characteristics.

The fidelity gains are consistent across evaluation spaces. With TimeCraft fixed, Table 5 compares C-FID with FID_M in frozen MOMENT embeddings, FID_DAE in public-data-trained DAE embeddings, and MMD^2 on raw windows. These three alternatives are not used for PE voting, yet each ranks TimeCraft first in $15/16$ settings, including every setting at $\varepsilon \in \{0.5, 1, 2\}$. Their mean margins at these finite budgets are $4.1\times$, at least $6.4\times$, and $5.7\times$, respectively. Agreement extends beyond the winning method: within each dataset–budget setting, method rankings under C-FID correlate positively with all three alternatives, with median Spearman correlations of 0.82, 0.90, and 0.93 across the 16 settings (Appendix H). This agreement across complementary metrics supports fidelity improvements beyond the TS2Vec voting space.

Table 5: Cross-metric comparison of PETS (TimeCraft) with baselines at $\varepsilon \in \{0.5, 1, 2, \infty\}$. Wins: settings beating all baselines (non-DP methods at ∞). Margins: mean strongest-DP-baseline/PETS ratios over the finite budgets; higher is better. $\geq$: noise-floor-limited ratios.

Metric	**C-FID**	**FID_M**	**FID_DAE**	**MMD^2**
Evaluation space	TS2Vec	MOMENT-1-large	Conv1D DAE	Raw windows
PE voting?	✓	✗	✗	✗
Wins	14/16	15/16	15/16	15/16
Mean margin over strongest DP baseline				
Traffic	6.0×	2.7×	4.2×	1.6×
Activity	9.2×	5.6×	7.8×	7.1×
ECG	1.6×	3.1×	$\geq$10.5×	3.6×
Crypto	10.3×	5.0×	3.1×	10.4×
Mean	**6.8×**	**4.1×**	**$\geq$6.4×**	**5.7×**

4.3 Performance on downstream tasks

We now compare PETS against baselines on downstream tasks under the TSTR protocol. The results are shown in Figure 6. Per-task details are in Appendix F.8.

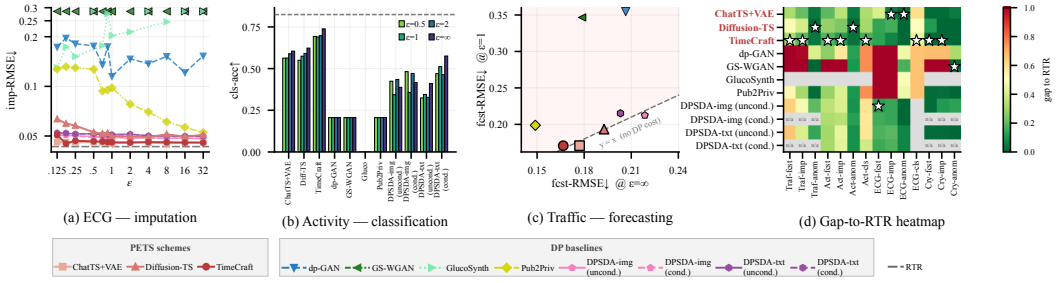


Figure 6: Downstream-task performance overview at $\varepsilon = 1$ unless stated. **(a)** ECG imputation RMSE↓ across the finite-budget ε sweep. **(b)** Activity classification accuracy↑ at $\varepsilon \in \{0.5, 1, 2, \infty\}$. **(c)** Traffic forecasting privacy–utility scatter: x =RMSE↓ at $\varepsilon = \infty$, y =RMSE↓ at $\varepsilon = 1$. **(d)** Gap-to-RTR heatmap at $\varepsilon = 1$ across all 14 (dataset, task) settings and 11 methods (including both DPSDA versions); ★ marks the column-best.

PETS delivers strong downstream utility, achieving a mean $1.4\times$ the activity classification accuracy of the strongest DP baseline and $3.4\times$ that of DP-SGD baselines across the full privacy-budget sweep. Across all 13 evaluated budgets ($\varepsilon = 0.125$ to 32, plus ∞), TimeCraft exceeds the strongest DP baseline at each budget by 15.8–23.0 percentage points (Appendix G). PETS' lead therefore persists when DPSDA also uses class labels (Appendix G.1). The gains extend beyond classification: at $\varepsilon = 1$, ChatTS+VAE reduces ECG imputation RMSE by 8.3%, while

TimeCraft reduces traffic forecasting RMSE by 14.1%, relative to the strongest DP baseline for each task (Figure 6a,c).

PETS is robust to privacy noise on downstream tasks, especially traffic forecasting, where it achieves the best utility at a small privacy cost. Moving from $\varepsilon = \infty$ to $\varepsilon = 1$, TimeCraft's traffic forecasting RMSE increases by only 2.4%, compared with 33.5% for Pub2Priv (Figure 6c). TimeCraft thus overtakes Pub2Priv and achieves the best utility at this budget. All three PETS schemes remain within 3.6% of their no-noise forecasting RMSE. This stability also holds across the finite-budget sweep: ChatTS+VAE's ECG imputation RMSE varies by less than 2% across all 12 budgets (Figure 6a).

PETS' downstream advantage holds across tasks and datasets. At $\varepsilon = 1$, PETS leads in 12 of 14 settings, with wins spanning all four task types and all four datasets (Figure 6d). All three PETS schemes also achieve at least $2.4\times$ lower mean normalized gaps to RTR than dp-GAN, GS-WGAN, and Pub2Priv (below 0.19 versus at least 0.45). Full results and further analyses are in Appendix G.

4.4 Effect of Public–Private Domain Mismatch

To examine PETS' dependence on public–private alignment, we fix MIT-BIH Arrhythmia (Moody & Mark, 2001) as the public source and evaluate all three schemes at $\varepsilon = 1$. The seven private cohorts range from closely related adult ECG to more distant signals, including fetal ECG, gestures, traffic, water-network chlorine concentrations, and engine noise. Table 6 reports the public–private *gap* (C-FID or FID_M computed between public and private training windows), with rows ordered by C-FID gap. Dataset sources and experimental details are in Appendix I.

Table 6: Domain-mismatch results for the three PETS schemes. Grey columns show the public–private gap and train-on-real reference (RTR). **Bold** marks the best PETS scheme per row and metric.

	C-FID↓				FID_M ↓				Imputation RMSE↓			
Private cohort	Gap	ChatTS +VAE	Diffusion-TS	Time Craft	Gap	ChatTS +VAE	Diffusion-TS	Time Craft	RTR	ChatTS +VAE	Diffusion-TS	Time Craft
SVDB	2.07	3.30	2.64	**2.01**	0.134	0.224	0.175	**0.159**	0.099	0.110	0.112	**0.109**
NSRDB	6.14	4.18	3.03	**1.97**	0.246	0.253	0.224	**0.151**	0.088	0.118	0.115	**0.111**
UWaveGesture LibraryX	19.51	6.94	4.52	**2.60**	0.723	0.283	0.268	**0.220**	0.148	0.348	**0.260**	0.267
NonInvasiveFetal ECGThorax1	25.84	9.93	5.54	**3.44**	0.812	0.378	0.295	**0.268**	0.060	0.329	0.351	**0.168**
METR-LA	30.40	8.28	6.52	**4.09**	0.787	**0.676**	0.830	0.724	0.148	0.199	**0.189**	0.197
ChlorineConcentration	36.57	12.71	14.10	**8.89**	1.554	**1.230**	1.336	1.462	0.069	0.329	0.322	**0.318**
FordA	40.09	19.72	19.28	**12.47**	3.383	**3.343**	3.474	3.504	0.286	**0.357**	0.359	0.362

PETS makes effective use of domain-relevant public priors. All three schemes achieve their lowest C-FID on the two adult-ECG cohorts, with scores increasing as the public–private gap grows (Spearman $\rho = 0.96$–1.00). FID_M shows the same broad trend. TimeCraft's C-FID rises from approximately 2.0 on adult ECG to 12.47 on FordA. This consistent relationship supports PETS' use of related public data to guide synthesis toward the private domain.

PETS retains fidelity and utility gains even under substantial public–private mismatch. Across the cohorts beyond SVDB, all three schemes reduce C-FID relative to the public–private gap, with TimeCraft achieving a 3.1–$7.5\times$ reduction. These gains extend beyond the voting space: averaged across all seven cohorts, TimeCraft reduces FID_M by 24% relative to the corresponding gap and imputation RMSE by 12% compared with models trained on public windows.

5 Conclusion

We introduced PETS, a framework for differentially private time-series generation that adapts PE to a domain where pre-trained generative and representation infrastructure is less mature than in text or vision. PETS achieves high-quality synthesis without training generative models on private data. Our results across diverse domains show substantial gains in fidelity and downstream utility over SOTA DP methods, with limitations and future work discussed in Appendix E.

AI USE STATEMENT

In this work, we used generative AI tools for editing certain sections of the manuscript, proofreading, LaTeX formatting for mathematical equations, assistance for writing proofs and assistance with implementation and plotting code. We have not used generative AI tools for conducting a literature search, developing our conceptual framework, or for generating synthetic datasets for training. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our experiments use existing publicly available datasets, with designated datasets treated as private for evaluation. The privacy guarantee applies to the addition or removal of one preprocessed time-series window, but it does not directly provide patient- or user-level protection when multiple windows belong to the same individual. Deployment must also account for any additional access to private data during preprocessing, hyperparameter selection, or evaluation. Beyond privacy, synthetic data may inherit biases from public priors or inadequately represent rare patterns and minority classes. Differential privacy does not guarantee fairness, clinical validity, or downstream reliability under such conditions. Consequently, sensitive applications require validation for the intended population and task, particularly under public–private domain mismatch. We already explore this in subsection 4.4 and discuss further limitations in Appendix E.

REPRODUCIBILITY STATEMENT

We provide all code used to conduct our experiments in the supplementary material, together with implementation documentation and instructions for running the experiments. section 3 describes the PETS modules, Appendix A provides the algorithms, and Appendix B states the assumptions and derives the privacy guarantee. Appendix F documents the hardware, dataset preprocessing, training hyperparameters, PE inference configurations, baseline implementations, fidelity metrics, and downstream evaluation protocols. Appendix I provides the domain-mismatch setup and results.

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.

Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge Reyes-Ortiz. Human Activity Recognition Using Smartphones. UCI Machine Learning Repository, 2012.

Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, Jorge Luis Reyes-Ortiz, et al. A Public Domain Dataset for Human Activity Recognition Using Smartphones. In *The European Symposium on Artificial Neural Networks*, volume 3, pp. 3–4, 2013.

annnnguyen. METR-LA. Kaggle, 2024.

Supraventricular Arrhythmia. MIT-BIH Supraventricular Arrhythmia Database. PhysioNet, 1999. Version 1.0.0.

Association for the Advancement of Medical Instrumentation. *Testing and reporting performance results of cardiac rhythm and ST segment measurement algorithms*. Number ANSI/AAMI EC57:1998. AAMI, Arlington, VA, 1999. ISBN 1570201161.

Binance. Binance Public Data: BTC/USDT Spot Monthly Klines, 2021-2026a. Online archive of cryptocurrency market data.

Binance. Binance Public Data: ETH/USDT Spot Monthly Klines, 2021-2026b. Online archive of cryptocurrency market data.

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.

Min Jin Chong and David Forsyth. Effectively Unbiased FID and Inception Score and Where to Find Them. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6070–6079, 2020.

Hoang Anh Dau, Eamonn Keogh, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, and Shaghayegh Gharghabeghi. The UCR Time Series Classification Archive, 2018.

Bowen Deng, Chang Xu, Hao Li, Yu-hao Huang, Min Hou, and Jiang Bian. TarDiff: Target-Oriented Diffusion Guidance for Synthetic Electronic Health Record Time Series Generation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, pp. 474–485, 2025.

Jinshuo Dong, Aaron Roth, and Weijie J Su. Gaussian Differential Privacy. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 84(1):3–37, 2022.

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.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. MOMENT: A Family of Open Time-series Foundation Models. In *International Conference on Machine Learning*, 2024.

S. D. Greenwald. *Improved detection and classification of arrhythmias in noise-corrupted electrocardiograms using contextual information*. PhD thesis, Harvard-MIT Division of Health Sciences and Technology, 1990.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A Kernel Two-Sample Test. *Journal of Machine Learning Research*, 13(25):723–773, 2012.

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.

Yu-Hao Huang, Chang Xu, Yueying Wu, Wu-Jun Li, and Jiang Bian. TimeDP: Learning to Generate Multi-Domain Time Series with Domain Prompts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 17520–17527, 2025.

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.

Josephine Lamp, Mark Derdzinski, Christopher Hannemann, Joost Van der Linden, Lu Feng, Tianhao Wang, and David Evans. GlucoSynth: Generating Differentially-Private Synthetic Glucose Traces. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *International Conference on Learning Representations (ICLR)*, 2018.

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. *arXiv preprint arXiv:2502.05505*, 2025.

Penghang Liu, Haibei Zhu, Eleonora Kreacic, and Svitlana Vyetrenko. Privacy-Aware Time Series Synthesis via Public Knowledge Distillation. *Transactions on Machine Learning Research*, 2025.

meithnavlakha. TimeGAN with Differential Privacy. Kaggle, 2024.

Microsoft Research Lab Asia. TimeCraft: A universal framework for time-series generation, 2025.

George Moody. MIT-BIH Normal Sinus Rhythm Database. PhysioNet, 1999. Version 1.0.0.

George B. Moody and Roger G. Mark. The impact of the MIT-BIH Arrhythmia Database. *IEEE Engineering in Medicine and Biology Magazine*, 20(3):45–50, 2001.

George B. Moody and Roger G. Mark. MIT-BIH Arrhythmia Database. PhysioNet, 2005.

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.

scchuy. PEMS-BAY Traffic Dataset. Kaggle, 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.

Chang Wei Tan, Angus Dempster, Christoph Bergmeir, and Geoffrey I. Webb. MultiRocket: multiple pooling operators and transformations for fast and effective time series classification. *Data Mining and Knowledge Discovery*, 36(5):1623–1646, 2022.

Pepijn te Marvelde, Lydia Y. Chen, Aditya Kunar, and Zilong Zhao. Differentially Private GAN for Time Series, 2021.

Muhang Tian, Bernie Chen, Allan Guo, Shiyi Jiang, and Anru R Zhang. Reliable generation of privacy-preserving synthetic electronic health record time series via diffusion models. *Journal of the American Medical Informatics Association*, pp. ocae229, 2024.

Gary Weiss. WISDM Smartphone and Smartwatch Activity and Biometrics. UCI Machine Learning Repository, 2019.

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.

Evgeny Yakushenko. St Petersburg INCART 12-lead Arrhythmia Database. PhysioNet, 2008.

Jinsung Yoon, Daniel Jarrett, and Mihaela Van der Schaar. Time-series generative adversarial networks. *Advances in neural information processing systems*, 32, 2019.

Xinyu Yuan and Yan Qiao. Diffusion-TS: Interpretable Diffusion for General Time Series Generation. In *The Twelfth International Conference on Learning Representations*, 2024.

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 (Lin et al., 2024) with adaptations for time series data. Algorithm 1 handles both conditional ($|C| > 1$) and unconditional ($C = \{0\}$) generation by processing each class separately, which is equivalent to running unconditional PETS independently per class. Algorithm 2 describes the DP-protected nearest-neighbor histogram computation, including normalized multi-neighbor voting where each private record contributes weight $1/\sqrt{N_{\text{nn}}}$ to each of its N_{nn} distinct nearest synthetic candidates. The choice $N_{\text{nn}} = 1$ recovers single-neighbor voting, while our deployed configuration uses $N_{\text{nn}} = 7$.

Algorithm 1: Private Evolution for Time Series (PETS)

Input: Fixed, publicly specified class set 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 $N_{\text{syn}} \mod |C| = 0$)
Number of nearest neighbors N_{nn} where $1 \leq N_{\text{nn}} \leq N_{\text{syn}}/|C|$
Noise multiplier σ, Threshold H
Output: Synthetic dataset S_{syn}

$S_{\text{syn}} \leftarrow \emptyset$
for $c \in C$ **do**
  $private_samples \leftarrow \{x_i \mid (x_i, y_i) \in S_{\text{priv}} \text{ and } y_i = c\}$
  $S_0 \leftarrow$ `GENERATION`$(N_{\text{syn}}/|C|)$
  for $t = 1, \ldots, T$ **do**
    $h_t \leftarrow$ `DP_NN_HISTOGRAM`$(private_samples, S_{t-1}, N_{\text{nn}}, \sigma, H)$ `// 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} from one class
Synthetic candidates $S = (z_1, \ldots, z_n)$ for the same class
Number of nearest neighbors N_{nn} where $1 \leq N_{\text{nn}} \leq n$
Noise multiplier σ, Threshold H
Distance function $d(\cdot, \cdot)$ computed via `SIMILARITY` (e.g., L_2 in embedding space)
Output: DP nearest neighbors histogram on S

$h \leftarrow [0, \ldots, 0]$ (length n)
for $x_{\text{priv}} \in S_{\text{priv}}$ **do**
  $J \leftarrow$ indices of the N_{nn} nearest candidates to x_{priv} in S
  `// Distinct indices; break distance ties by candidate index`
  for $j \in J$ **do**
    $h[j] \leftarrow h[j] + 1/\sqrt{N_{\text{nn}}}$
  end
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 histogram-based DP mechanism. Its time-series generation, variation, and similarity modules change how synthetic candidates are constructed and compared, while preserving the sensitivity bound required for privacy accounting. Specifically, each private record contributes to N_{nn} distinct histogram bins with weight $1/\sqrt{N_{\text{nn}}}$ per bin, giving a contribution vector of unit L_2 norm. Under add/remove-one-record adjacency, the histogram therefore has sensitivity 1 for both single-neighbor and normalized multi-neighbor voting. The analysis applies to all three PETS schemes because private records enter the PE loop only through this voting mechanism.

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 participates in the selection process by contributing to a histogram, which is subsequently privatized. Our framework's modularity allows for different scheme instantiations without altering this fundamental privacy boundary:

- **Generation Modules** (ChatTS, Diffusion-TS, or TimeCraft): These provide the initial candidates. Since they are rule-based or pretrained on public data, they do not consume private records and thus do not impact the privacy budget.
- **Variation Modules** (VAE or SDEdit): These produce offspring from selected parents. In Diffusion-TS and TimeCraft, the SDEdit variation operates as a public-data-trained denoising process. Just as with the VAE in our initial scheme, these operators are fixed during inference and do not see the private data directly.
- **Similarity Function** (TS2Vec): The embedding network Φ is pretrained on public data. The choice of embedding space affects the utility of the selection but does not change the sensitivity of the voting mechanism.

B.2 DP MECHANISMS

The DP guarantee in PETS comes from two mechanisms in Algorithm 2: gaussian perturbation of the voting histogram, as well as thresholding and subsequent synthetic-data operations which preserve this guarantee through post-processing.

Gaussian noise: Independent Gaussian noise $\mathcal{N}(0, \sigma^2 I)$ is added to the weighted histogram. As established below, normalization by $1/\sqrt{N_{\text{nn}}}$ preserves an L_2 sensitivity of 1 under add/remove-one-record adjacency, including the deployed setting $N_{\text{nn}} = 7$. Consequently, the same Gaussian calibration applies for every valid choice of N_{nn}.

Thresholding (post-processing): Element-wise thresholding (controlled by H) sets low-count bins to zero. By the post-processing property of DP, this step incurs no additional privacy loss but improves fidelity by filtering out candidates that only received "votes" from the added Gaussian noise.

Lookahead (privacy-free preprocessing): Optionally, distances are computed based on the mean embedding of k candidate variations (the lookahead degree k is an inference hyperparameter; see subsection F.3). Since this pre-computation operates entirely on synthetic candidates and does not read private data, it does not enter the privacy accounting.

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 $\mathcal{M}(x) = f(x) + Z$ with $Z \sim \mathcal{N}(0, \sigma^2 I)$ satisfies (ε, δ)-DP if and only if*

$$\Phi_{\mathcal{N}}\left(\frac{\Delta}{2\sigma} - \frac{\varepsilon\sigma}{\Delta}\right) - e^{\varepsilon}\Phi_{\mathcal{N}}\left(-\frac{\Delta}{2\sigma} - \frac{\varepsilon\sigma}{\Delta}\right) \leq \delta, \tag{8}$$

where $\Phi_{\mathcal{N}}$ is the cumulative distribution function of the standard normal distribution.

Theorem 2 (Privacy Guarantee of PETS)**.** *Consider add/remove-one-record adjacency on the pre-processed private dataset. Assume that the class set, synthetic population sizes, iteration count, and inference hyperparameters are fixed independently of the private dataset, and that generation, variation, and similarity use only public information and previous DP outputs. For any valid $N_{\mathrm{nn}} \geq 1$, Algorithm 1, using normalized multi-neighbor voting in Algorithm 2 for T iterations with independent Gaussian noise of standard deviation $\sigma > 0$ per histogram bin, satisfies (ε, δ)-DP if*

$$\Phi_{\mathcal{N}}\left(\frac{\sqrt{T}}{2\sigma} - \frac{\varepsilon\sigma}{\sqrt{T}}\right) - e^{\varepsilon}\Phi_{\mathcal{N}}\left(-\frac{\sqrt{T}}{2\sigma} - \frac{\varepsilon\sigma}{\sqrt{T}}\right) \leq \delta. \tag{9}$$

Derivation. Fix a class and an iteration, and condition on the preceding DP outputs and the synthetic candidate construction. The candidate population and its similarity representations are then fixed for the sensitivity analysis.

Let $\mathrm{NN}_{N_{\mathrm{nn}}}(x; S)$ denote the N_{nn} distinct nearest candidate indices for record x, with ties broken by candidate index. Writing e_j for the jth standard basis vector, the record's contribution to the histogram is

$$v(x; S) = \frac{1}{\sqrt{N_{\mathrm{nn}}}} \sum_{j \in \mathrm{NN}_{N_{\mathrm{nn}}}(x;S)} e_j. \tag{10}$$

Because these indices are distinct,

$$\|v(x; S)\|_2^2 = N_{\mathrm{nn}} \cdot \frac{1}{N_{\mathrm{nn}}} = 1. \tag{11}$$

For neighboring class-specific datasets $D' = D \cup \{x\}$, the unperturbed histogram $h(D; S) = \sum_{u \in D} v(u; S)$ consequently satisfies

$$\|h(D'; S) - h(D; S)\|_2 = \|v(x; S)\|_2 = 1. \tag{12}$$

Thus, the histogram has L_2 sensitivity 1, independently of N_{nn}.

Adding Gaussian noise with standard deviation σ therefore makes each iteration $1/\sigma$-Gaussian differentially private. Adaptive composition over T iterations gives μ-Gaussian differential privacy with $\mu = \sqrt{T}/\sigma$ (Dong et al., 2022). Converting this guarantee to (ε, δ)-DP yields the stated inequality. Thresholding, resampling, variation, and release of the final synthetic population are post-processing and incur no additional privacy loss. The same guarantee applies jointly across classes by the parallel-composition argument below. □

Conditional generation: Let C be a fixed, publicly specified class set. Algorithm 1 partitions private records by label and runs a separate PE loop for each class, with its own synthetic candidate population. Adding or removing one labeled record (x, y) changes only the private input to the loop for class y. Although this record contributes to N_{nn} distinct bins within that class's histogram, its contribution has unit L_2 norm and it contributes nothing to any other class's histogram. Therefore, parallel composition across classes incurs no additional privacy loss: if each class loop satisfies the same (ε, δ) guarantee, their combined synthetic output also satisfies (ε, δ)-DP. More generally, class-specific guarantees $(\varepsilon_c, \delta_c)$ yield $(\max_{c \in C} \varepsilon_c, \max_{c \in C} \delta_c)$-DP. This argument uses add/remove adjacency; replacing a record with one bearing a different label can affect two class partitions and is not covered by this sensitivity argument.

C 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:

$$W_p(S_{\mathrm{priv}}, S_{\mathrm{syn}}) = \left(\inf_{\gamma \in \Pi(\hat{\mu}_{\mathrm{priv}}, \hat{\mu}_{\mathrm{syn}})} \mathbb{E}_{(x,x') \sim \gamma}[d(x, x')^p]\right)^{1/p}, \tag{13}$$

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

D Related Work

Differentially private time-series synthesis. Differential privacy has been widely used for synthetic data release by privatizing the training of generative models, most commonly through DP-SGD (Abadi et al., 2016). In the time-series setting, this paradigm has been instantiated with adversarial, diffusion, distillation, and motif-based generators. Early efforts like dp-GAN (Frigerio et al., 2019) focused on training the discriminator with DP-SGD to release the generator via post-processing, while recent methods like GS-WGAN (Chen et al., 2020; te Marvelde et al., 2021) introduced gradient sanitization and multiple discriminators to stabilize training. To leverage public knowledge, Pub2Priv (Liu et al., 2025) utilizes a public-data-trained context encoder to guide a private diffusion denoiser trained on sensitive records. Specialized architectures have also been proposed for specific structures, such as GlucoSynth (Lamp et al., 2023), which uses public motif codebooks combined with privatized transition statistics to preserve temporal morphology. These methods differ in architecture and domain assumptions, but share the same privacy boundary: private sequences enter the optimization loop, and privacy noise is injected into gradient updates or privatized training statistics. While effective in some settings, training-time DP typically requires expensive privacy-constrained optimization, careful hyperparameter tuning, and retraining when the private domain changes.

Inference-time DP and time-series generative priors. Recent Private Evolution methods show that differentially private synthetic data can be produced with black-box access to pre-trained image or language models by alternating candidate generation, DP nearest-neighbor voting, and variation (Lin et al., 2024; Xie et al., 2024; Lin et al., 2025). These approaches rely on mature domain infrastructure, such as high-fidelity generative APIs and semantic embedding models, e.g., Stable Diffusion (Rombach et al., 2022) and CLIP (Radford et al., 2021). Such infrastructure is less developed for time series, where similarity must respect temporal structure, phase shifts, trends, local changes, and label or domain conditions. PETS connects inference-time DP with recent time-series generation and representation tools: rule-based generators provide controllable seed populations (Xie et al., 2025), public-data-trained variation modules preserve temporal dependencies (Kingma & Welling, 2014), and contrastive encoders define meaningful embedding-space similarity (Yue et al., 2022).

Non-DP benchmark models. To contextualize the utility-privacy trade-off, our work also benchmarks against state-of-the-art non-DP generative models. These include TimeGAN (Yoon et al., 2019), which utilizes a learned temporal embedding space to preserve distribution alignment, and TimeDiff (Tian et al., 2024), a diffusion-based approach designed for high-fidelity synthetic electronic health records. These baselines provide a non-private reference fidelity and downstream performance.

E Limitations and Future Work

E.1 Limitations

Despite the modular architecture and strong empirical results, PETS still has limitations. First, although PETS avoids private-data training, the framework remains constrained by the prior-knowledge gap between public and private domains. The pipeline depends on the quality of its public-data-trained components: if the public data is poorly matched or weakly correlated with the private domain, the generator may fail to produce candidates near the private distribution, the variation module (VAE-based or SDEdit-based) may distort important temporal structures, or the similarity encoder may emphasize irrelevant features. In such cases the DP voting step can only select among inadequate candidates, capping the final fidelity of the synthetic dataset. This issue is most pronounced in highly specialized or novel domains where public proxies and rule-based templates are scarce or unavailable (e.g., rare medical conditions, proprietary industrial telemetry, or highly localized sensor networks). Future work should investigate how much public–private mismatch PETS can tolerate, and whether self-supervised pretraining on broad public time-series corpora can reduce this dependence.

Second, while PETS satisfies a rigorous record-level DP guarantee, the interpretation of this safety varies by application. Because we treat each preprocessed window as an independent record (as

discussed in Appendix F.1), record-level DP is formally preserved regardless of window overlap. However, in domains where multiple windows originate from the same individual (e.g., heart rate monitors or personal mobility), future work should explore extending PETS to *user-level* or *entity-level* privacy models. Such models would ensure that the inclusion or exclusion of an entire individual's history, rather than just a single window, remains bounded. Additionally, further research is needed to determine "acceptable" ε values for time-series data that provide meaningful protection against record-level pattern leakage while maintaining downstream utility.

E.2 Future Work

PETS' modularity offers several directions for future work. Future systems could combine stronger public-pretrained components with inference-time private selection. Replacing individual modules with domain-specific alternatives, such as medical time-series encoders for ECG or more expressive simulators for scientific and industrial signals, may further improve fidelity and downstream utility. Another direction is extending PETS to irregularly sampled, partially observed, and heterogeneous multivariate time series. Finally, future work could evaluate whether a time-series foundation model pretrained exclusively on PETS-generated data from multiple sensitive sources can match the zero-shot or few-shot performance of one pretrained on the original private data.

F Detailed Experiment Configurations

Hardware: All experiments in this paper execute on a single NVIDIA A100 GPU with 80 GB of GPU memory per job, with 60–100 GB of host system RAM allocated via SLURM. All wall-clock comparisons reported in this paper are therefore made under matched compute conditions.

F.1 Dataset Details

Domain	Public Dataset	Private Dataset	Sequence length	Stride length
Traffic	PeMS-BAY	METR-LA	576	288
Activity	WISDM	UCI-HAR	128	64
ECG	MIT-BIH Arrhythmia	INCART	256	per-beat (R-peak centered)
Crypto	BTC/USDT	ETH/USDT	256	256

Table 7: Summary of the four public/private dataset pairs used in PETS.

As established in section 2, neighboring datasets are defined at the record level, where one record corresponds to a single fixed-length time-series window. Consequently, the window striding and overlapping policies utilized during preprocessing do not affect the formal (ε, δ)-DP guarantee. We regard the dataset produced *after* preprocessing as the original private input to the PETS framework. In this setting, each individual window is treated as an independent record. To demonstrate the generalizability of our approach across different temporal structures, we employ varying stride lengths for each dataset (summarized in Table 7).

There is no overlap between the two public/private datasets in a domain pair. The four pairs differ in the underlying variable (speed, acceleration magnitude, ECG voltage, or log-return), the window extraction policy, and whether labels are available for downstream tasks.

The *traffic* datasets are records of highway speed from loop detectors in California. Missing or invalid readings are imputed with the sensor mean, and each sensor stream is segmented into windows of length 576 with stride 288, corresponding to two days of 5-minute observations with 50% overlap. Following standard forecasting practice, each sensor is treated as an independent univariate time series. PETS then applies per-window z-score normalization, computes a unified percentile clip range from the combined public and private windows, and maps both datasets to $[0, 1]$.

Both *activity* datasets were derived from smartphone accelerometers during everyday activities. We reduce the raw tri-axial accelerometer to a single univariate signal by taking the acceleration magnitude $\|(a_x, a_y, a_z)\|$. WISDM is resampled from 20 Hz to 50 Hz to match UCI-HAR, and windows

of length 128 with stride 64 are extracted. To align the label spaces for downstream classification, we retain only the four activities shared across the two corpora: *Walking*, *Stairs*, *Sitting*, and *Standing*. In particular, UCI-HAR walking upstairs and downstairs are merged into a single *Stairs* class, and *Laying* is dropped. Each window is normalized by per-window z-score, after which a unified clip-and-rescale step maps both public and private windows to $[0, 1]$.

For the *ECG* datasets, heartbeat annotations are mapped to the AAMI EC57 three-class taxonomy: *Normal* (raw symbols `N, L, R, e, j`, normal-conducted beats), *Supraventricular ectopic* (`A, a, J, S`), and *Ventricular ectopic* (`V, E`). Fusion (`F`) and unknown/paced (`Q, /, f, ?`) symbols are dropped. We select lead II (preferring `MLII` when available, otherwise exact lead `II`) from each record, resample to 250 Hz, and perform *per-record* normalization over the full recording. For each retained beat we extract a 256-sample window centered around the R-peak, with the R-peak fixed at index 90 so that each window contains both pre- and post-beat context. Because the raw INCART side is much larger than the other domains, the private side is stratified down to 10,000 windows by default to keep PE and downstream evaluation computationally comparable, while the public MIT-BIH side is kept full. A unified percentile clip over the combined public/private windows is then followed by min–max scaling to $[0, 1]$.

For the *Crypto* datasets, rather than modeling raw prices, we compute 1-minute log-returns $r_t = \log(\text{close}_t/\text{close}_{t-1})$, which are more stationary and comparable across assets. BTC and ETH are each treated as a single long stream, standardized by per-stream z-score, and segmented into 256-sample windows with *non-overlapping* stride 256. We intentionally use non-overlap in this domain because financial series are strongly autocorrelated, and reduced overlap gives a more diverse window pool for PE voting. As with the other domains, a unified public+private percentile clip and min–max scaling maps both sides to $[0, 1]$.

F.2 Training-side Hyperparameter audit.

All learned components consumed by PETS are pre-trained on the *public* split of each dataset pair only, so no private data enters any training loop. Each trainable component (TS2Vec encoder, TimeCraft generator, Diffusion-TS generator, VAE variation module) for the traffic, activity, and crypto settings is selected via a 6-iteration PE audit on the activity training split (ECG training side hyperparameters are decided via an audit on ECG's public data at $\varepsilon = 1$). TS2Vec: lr $\in \{10^{-3}, 2 \times 10^{-3}, 5 \times 10^{-4}\}$ at epochs $\in \{1, 5\}$. TimeCraft: lr $\in \{10^{-3}, 3 \times 10^{-4}\}$ at steps $\in \{600, 1200, 1800\}$. Diffusion-TS: steps $\in \{1500, 4500\}$ at the YAML-default lr. ChatTS+VAE variation module: epochs $\in \{15, 30, 60, 90\}$ at lr$= 10^{-3}$. Figure 7 reports the audit C-FID for each candidate. We deploy the audit best for TS2Vec and Diffusion-TS, step 1620 for TimeCraft, and epoch 30 for the VAE (epoch 90 overfits on activity's small dataset under longer PE verification).

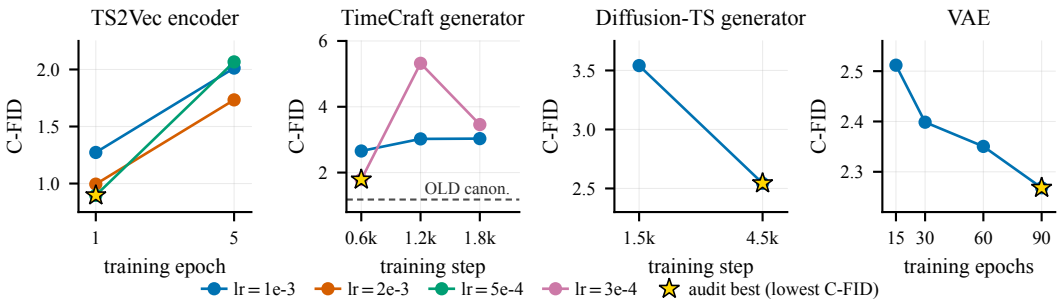


Figure 7: Training-side hyperparameter audit on activity (lower C-FID is better). Each panel sweeps one model's training HP via 6-iteration PE evaluation. $\star$ marks the deployed configuration; for TimeCraft, the dashed line shows the deployed step 1620 (outside the audit grid).

F.3 PE Inference Configuration

Hyperparameters: PETS exposes the following inference-side hyperparameters:

- δ: privacy parameter for approximate DP; smaller δ gives stronger guarantees.

- T: number of PE iterations.
- H (histogram threshold): post-processing threshold for the DP histogram; bins with count $< H$ are zeroed to reduce noise impact.
- N_{nn} (nearest neighbours): number of synthetic candidates each private sample votes for. The default 1 matches the standard PE form shown in Algorithm 2. For $N_{\text{nn}} > 1$, the implementation L_2-normalizes votes by $1/\sqrt{N_{\text{nn}}}$ to preserve sensitivity 1.
- k (lookahead degree): each candidate's embedding is averaged over k random variations before histogram voting; $k = 0$ uses the raw embedding. Lookahead is privacy-free since it operates only on synthetic candidates.
- s_f (sampling fraction): ratio of synthetic population size to private dataset size.
- Selection mode: `sample` (probability-weighted with replacement) or `rank` (deterministic top-voted).
- Variation-degree schedule: sequence $\{\alpha_t\}_{t=1}^{T}$ controlling per-iteration mutation strength; scheme-specific.

Deployed values: The three PETS schemes share the following PE-level hyperparameters on traffic, activity, and crypto at all 13 privacy budgets. ECG uses a retuned configuration, given after the following list.

- $\delta = 10^{-5}$.
- $T = 20$ DP voting rounds (`--num-iterations 21`, including the initial population at $t = 0$). Reporting rules are given in Appendix J.
- $H = 2$ (`--histogram-threshold 2`) unless otherwise stated for different tasks and domains.
- $N_{\text{nn}} = 7$ (`--num-nearest-neighbors 7`).
- $s_f = 0.5$ (`--num-samples-fraction 0.5`); $|S_{\text{syn}}| = s_f \times |S_{\text{priv,train}}|$.
- Selection mode: `sample`.
- Generation batch size: 64.
- Privacy budget grid: $\varepsilon \in \{0.125, 0.175, 0.25, 0.5, 0.7, 0.85, 1, 2, 4, 8, 16, 32, \infty\}$. The $\varepsilon = \infty$ ("oracle") setting sets noise multiplier $\sigma = 0$ via `--noise-multiplier 0`.

ECG configuration. On ECG, each scheme's generation/variation checkpoint and PE hyperparameters were retuned: the training configuration and checkpoint first, by the C-FID of PETS runs at $\varepsilon = 1$, then the PE hyperparameters on the chosen checkpoint, again at $\varepsilon = 1$ only. The chosen configuration is run unchanged at all 13 privacy budgets, with δ, T, the selection mode, the generation batch size, and the TS2Vec encoder as above:

- TimeCraft: generator trained with learning rate 3×10^{-4} and batch 128, checkpoint at step 4,000 of 16,000; schedule `[45,40,35,30,25,20,15,12,10,8, 6×11]`; $k = 0$, $H = 2$, $N_{\text{nn}} = 7$, $s_f = 0.5$ ($|S_{\text{syn}}| = 3{,}500$).
- Diffusion-TS: base learning rate 10^{-4}, batch 64, model width 64, 13,500 training steps (final checkpoint); schedule `[60,55,50,45,40,35,30,25,20,15, 10×11]`; $k = 8$, $H = 8$, $N_{\text{nn}} = 15$, $s_f = 0.25$ ($|S_{\text{syn}}| = 1{,}750$).
- ChatTS+VAE: VAE trained with learning rate 3×10^{-4}, latent dimension 32, and base width 64 for 90 epochs (final checkpoint); schedule `[45,40,35,30,25,20,15,12,10,8, 6×11]`; $k = 8$, $H = 1$, $N_{\text{nn}} = 11$, $s_f = 1.0$ ($|S_{\text{syn}}| = 7{,}000$).

F.4 PETS Hyperparameter Ablation

We ablate PETS' inference-time hyperparameters in Table 8, which is run on the activity dataset using the TimeCraft scheme at $\varepsilon = 1$. The default configuration is:

- Generation backbone: TimeCraft activity checkpoint (`--max-steps 1620`, YAML defaults: lr=10^{-3}, batch 256, seed 23).
- Embedding: TS2Vec activity checkpoint (`lr=5e-4`, epoch 1).
- PE: ε=1, δ=10^{-5}, T=20 DP voting rounds (21 snapshots), default schedule S3 `[40,35,30,25,20,15,10, 5×14]`, lookahead k=8, histogram threshold H=4,

Table 8: PETS hyperparameter ablation datasets on activity (TimeCraft, ε=1). Default configuration: δ=10^{-5}, H=4, k=8, N_{nn}=1, s_f=0.5, schedule S3, sample selection. **Bold** marks the best value within each ablation group (default included). Metrics: C-FID (fidelity ↓), acc (classification accuracy ↑), fcst (forecasting RMSE ↓), imp (imputation RMSE ↓), AUROC (anomaly detection ↑). Sweep values and downstream-model details in Appendix F.4, F.8.

Metric	default	δ		H				k				N_{nn}			s_f				schedule				selection	ε=∞
	—	10^{-4}	10^{-6}	2	8	12	16	0	4	12	16	3	5	7	0.75	1.0	1.25	1.5	S1	S5	S6	S7	rank	—
C-FID↓	**0.514**	0.524	0.570	**0.512**	0.558	0.565	0.561	0.504	0.528	0.554	**0.494**	0.485	0.453	**0.424**	0.551	0.517	0.560	0.626	0.546	0.548	**0.486**	0.487	1.384	0.331
acc↑	**0.688**	0.665	0.682	0.683	**0.699**	0.668	0.698	**0.708**	0.673	0.686	0.688	0.695	**0.713**	0.712	**0.730**	0.693	0.695	0.690	**0.720**	0.707	0.681	0.716	0.649	0.688
fcst↓	**0.1953**	**0.1919**	0.1924	**0.1908**	0.1919	0.1965	0.1977	0.1912	0.1944	**0.1911**	0.1913	0.1936	0.1897	**0.1865**	0.1911	0.1880	**0.1833**	0.1838	0.1955	0.1903	**0.1902**	0.1953	0.1886	0.1926
imp↓	**0.1606**	**0.1603**	0.1625	0.1618	0.1609	0.1608	0.1620	0.1598	0.1617	0.1626	**0.1597**	0.1607	0.1603	**0.1593**	**0.1600**	0.1617	0.1605	0.1633	0.1608	0.1620	0.1623	**0.1603**	0.1734	0.1582
AUROC↑	**0.911**	0.909	0.910	0.906	0.910	**0.912**	0.905	0.910	0.913	**0.914**	0.904	**0.914**	0.910	0.910	0.918	0.915	0.922	**0.933**	0.911	0.910	0.909	**0.913**	0.903	0.908

N_{nn}=1 (each private sample votes once), sampling fraction s_f=0.5 (yielding $|S_{\text{syn}}| = 2{,}924$), generation batch 64, selection mode `sample`.

- Privacy held-out: 30% of UCI-HAR private windows reserved for downstream evaluation; the remaining 70% (5,848 windows) is the PE-train split.

Swept values per axis (default in **bold**; see Appendix F.3 for HP definitions):

- $\delta \in \{10^{-4}, \mathbf{10^{-5}}, 10^{-6}\}$.
- $H \in \{2, \mathbf{4}, 8, 12, 16\}$.
- $k \in \{0, 4, \mathbf{8}, 12, 16\}$.
- $N_{\text{nn}} \in \{\mathbf{1}, 3, 5, 7\}$.
- $s_f \in \{\mathbf{0.5}, 0.75, 1.0, 1.25, 1.5\}$.
- schedule $\in$ {S1, S3, S5, S6, S7} (sequences listed below).
- selection $\in$ {sample, rank}.

The seven single-axis sweeps each vary one HP and hold the rest at default; total 24 unique runs (default + oracle + 22 single-axis variants). The five 21-iteration variation-degree schedules referenced in the schedule sweep are:

- S1: `[20,20,20, 15,15,15, 10,10, 5×13]`
- S3 (default): `[40,35,30,25,20,15,10, 5×14]`
- S5: `[50,45,40,35,30,25,20,15,10, 5×12]`
- S6: `[60,55,50,45,40,35,30,25,20,15, 10×11]`
- S7: `[95,85,75,65,55,45,35,25,20,15, 10×11]`

The number of nearest neighbours (N_{nn}) is the strongest tuning lever, reducing C-FID by 17% when increased from 1 to 7. *Monotonically*, $N_{\text{nn}} = 7$ reduces C-FID from 0.514 to 0.424 (−17%) and forecasting RMSE by 4.5%. N_{nn} spreads each private sample's vote across multiple synthetic candidates, providing a denser DP histogram.

Sample (probability-weighted) candidate selection is critical, with 2.7× better C-FID than rank-based selection. Replacing sample selection with deterministic rank-based selection collapses C-FID from 0.514 to 1.384. Rank-based selection deterministically picks the top-voted candidate every iteration, so the synthetic population collapses to copies of a single prototype within a few iterations. Instead, sample selection mixes in lower-ranked candidates and preserves diversity.

PETS is insensitive to most hyperparameters, making practical use easy without fine-grained tuning. Across the δ, H, k, s_f, and schedule sweeps, every downstream task metric stays within $\sim 6\%$ of the default value. This robustness lets practitioners use the default configuration on new datasets without per-axis hyperparameter search.

F.5 Baseline Details

In Table 1, TimeGAN and TimeDiff represent non-DP time-series generators, dp-GAN and GS-WGAN represent DP-SGD generators, and Pub2Priv represents public-pretrained DP-SGD. The DPSDA row refers to the original image/text PE methods. Their time-series adaptations are described below.

GS-WGAN (Chen et al., 2020; te Marvelde et al., 2021) is an open-source DP adversarial baseline. It combines Wasserstein GAN training with gradient sanitization and multiple discriminators, and uses the time-series adaptation that reshapes each window into an image-like square input for convolutional processing. Non-square sequence lengths are padded internally for the convolutional architecture and cropped back to the original length after generation. The training pipeline remains two-stage: non-private discriminator pretraining followed by DP generator training, with privacy controlled by the noise multiplier and reported using the method's accountant.

dp-GAN (Frigerio et al., 2019) is a training-time DP baseline in which only the discriminator is trained with DP-SGD, while the generator is released through DP post-processing of the privatized discriminator updates. Our implementation uses an LSTM generator and an MLP discriminator.

Pub2Priv (Liu et al., 2025) is a DP conditional diffusion baseline that distills non-private information from public data into a context encoder and trains the private denoiser with DP-SGD. The original method assumes paired public metadata, which PETS does not provide, so our implementation replaces that conditioning signal with deterministic public-window context features derived solely from the public dataset. This preserves the method's public-to-private knowledge-transfer structure while keeping the privacy boundary explicit.

GlucoSynth (Lamp et al., 2023) is a motif-aware DP generative baseline. Its core idea is to preserve higher-level temporal structure by learning a public motif codebook and regularizing generation against privatized motif-transition statistics. In our PETS adaptation, the public dataset of each pair is used to construct the motif codebook, while the private motif statistics and neural components are privatized. Because the original method was introduced for glucose traces, this baseline should be understood as a PETS adaptation of the paper's motif-level DP idea rather than a dataset-identical reproduction.

DPSDA-image (Lin et al., 2024) and **DPSDA-text** (Xie et al., 2024) are PE-based inference-time DP baselines, both implemented on top of the official `microsoft/DPSDA` repository. DPSDA-image renders each time-series window as a grayscale image, performs nearest-neighbor histogram selection with Gaussian noise in the image domain, and decodes the selected outputs back into time series. DPSDA-text uses the analogous text-style PE loop, converting each window into a reversible structured text representation before privatized selection and variation. In both cases, the final time-series outputs are mapped back to the PETS scale and evaluated identically to the other baselines.

Finally, we include two non-DP utility baselines. **TimeDiff** (Tian et al., 2024) is a diffusion-based time-series generator and serves as a strong non-private upper-bound baseline. It uses its standard internal diffusion transform but is trained and exported on PETS windows. **TimeGAN** (meithnavlakha, 2024; Yoon et al., 2019) is included via the widely used `ydata-synthetic` implementation. Although the source notebook mentions privacy in a downstream evaluator, the generator itself is trained non-privately, so we treat it strictly as a non-DP baseline. These two methods help contextualize the utility gap between PETS and unconstrained non-private generation.

F.6 Baseline Configurations

All baselines consume the same PETS-prepared $[0, 1]$-normalized private and public windows used by the PETS schemes (section 4). Synthetic outputs that fall outside $[0, 1]$ (e.g., from Pub2Priv at low ε or dp-GAN) are clipped to $[0, 1]$ before being passed to the C-FID and downstream pipelines, ensuring all methods are scored under identical input ranges. For the privacy sweep we cover $\varepsilon \in \{0.125, 0.175, 0.25, 0.5, 0.7, 0.85, 1, 2, 4, 8, 16, 32, \infty\}$.

For both PE-based DP baselines (DPSDA-image and DPSDA-text), the PE algorithm is run from the official `microsoft/DPSDA` library (Lin et al., 2024; Xie et al., 2024), with a TS↔image / TS↔text codec layer that bridges the time-series window format to each foundation model's expected input. To match PETS, the private samples vote in PETS' TS2Vec embedding space, with $T = 20$ DP rounds, $N_{\text{nn}} = 7$, lookahead 0, $\delta = 10^{-5}$, and the synthetic-set size of PETS at $s_f = 0.5$. DPSDA-image varies 64×64 line-plot renderings with the ImageNet 64×64 Improved-Diffusion 270M model. DPSDA-text varies a 64-bin text serialization with GPT-2 through LLM-based augmentation.

F.7 FIDELITY METRICS

All five fidelity metrics compare synthetic windows with private training windows, excluding the downstream test split, and are lower-is-better. "Raw windows" denotes PETS-preprocessed values without an embedding transform.

C-FID. Context Fréchet Inception Distance (Jeha et al., 2022) compares whole-window TS2Vec embeddings (Yue et al., 2022). Each pair uses a 320-dimensional encoder trained on public data only and fixed for PE voting and evaluation of all methods. Given embedding means μ_r, μ_s and covariances Σ_r, Σ_s, the Fréchet statistic (Heusel et al., 2017) is

$$F(r,s) = \|\mu_r - \mu_s\|_2^2 + \mathrm{Tr}\Big(\Sigma_r + \Sigma_s - 2(\Sigma_r\Sigma_s)^{1/2}\Big). \tag{14}$$

C-FID uses all available embeddings without sample-size extrapolation.

ACD. Autocorrelation difference measures temporal dependence in the window values. Reference and synthetic datasets are each centered by their pooled mean, with lagged products normalized by their pooled variance. Let $a_r(\ell)$ and $a_s(\ell)$ be the resulting lag-ℓ autocorrelations, averaged over windows and valid time positions. For our univariate windows of length L,

$$\mathrm{ACD}(r,s) = \left[\sum_{\ell=0}^{K-1}\big(a_r(\ell) - a_s(\ell)\big)^2\right]^{1/2}, \qquad K = \min(64, L). \tag{15}$$

FID$_{\mathrm{M}}$. We compute the Fréchet statistic in frozen MOMENT-1-large's 1024-dimensional embedding space (Goswami et al., 2024), using one checkpoint for all datasets and methods. Reference and synthetic windows of length 128 or 256 are zero-padded to 512 with an input mask; length-576 windows are linearly resampled to 512. MOMENT is used only for evaluation; its per-window normalization suppresses absolute offset and amplitude information.

FID$_{\mathrm{DAE}}$. We compute the Fréchet statistic using 128-dimensional embeddings from a Conv1D denoising autoencoder (DAE), trained on each pair's public windows with a fixed recipe and frozen at the final checkpoint. Its encoder has channels $1 \to 32 \to 64 \to 128$, kernel size 5, stride 2, batch normalization and GELU activations, followed by average pooling and a linear projection; a mirrored decoder reconstructs the input. Training minimizes reconstruction MSE with Gaussian input noise of standard deviation 0.1, using Adam at learning rate 10^{-3}, batch size 128, 30 epochs, and seed 0. Native window lengths and preprocessed amplitudes are retained without per-window normalization. The DAE is used only for evaluation.

MMD^2. The unbiased squared maximum mean discrepancy (Gretton et al., 2012) compares raw windows without an encoder. For reference windows $r_1, \ldots, r_m$ and synthetic windows $s_1, \ldots, s_n$,

$$\widehat{\mathrm{MMD}}^2 = \frac{\sum_{i\neq j} k(r_i, r_j)}{m(m-1)} + \frac{\sum_{i\neq j} k(s_i, s_j)}{n(n-1)} - \frac{2\sum_{i,j} k(r_i, s_j)}{mn}. \tag{16}$$

The kernel averages five Gaussian kernels, $\exp(-\|x - y\|_2^2/(2b^2))$, with bandwidths $\{1/4, 1/2, 1, 2, 4\}$ times the median pairwise distance among at most 2,000 reference windows. These bandwidths are shared across methods. Scoring samples at most 4,000 windows per side (seed 0).

Sample-size correction and noise floors. For FID$_{\mathrm{M}}$ and FID$_{\mathrm{DAE}}$, we reduce finite-sample bias by extrapolating the Fréchet statistic linearly in $1/n$ (Chong & Forsyth, 2020). For the smaller set size m, both sets are independently subsampled without replacement at fractions 0.25, 0.5, 0.75, and 1 of m, with three draws at each of the first three fractions and one at the last (seed 0). We report the least-squares intercept at $n \to \infty$. These metrics and MMD^2 share the PETS private training reference across methods. A random half-split provides an empirical noise floor; Table 12 reports the largest absolute estimate across scoring hosts. Negative estimates are retained. For multiplicative margins, values below the noise floor are replaced by it; margins with a floor-limited PETS denominator are marked as lower bounds.

Evaluation-encoder pretraining. Our audit of MOMENT's Time-series Pile (Goswami et al., 2024) found none of the four main private datasets but identified five of the seven domain-mismatch cohorts (Table 16), limiting its interpretation as an unseen-domain evaluator there. TS2Vec and DAE use only each pair's public data, while MMD^2 uses no encoder.

F.8 Downstream Evaluation Tasks

Task	Traffic	Activity	ECG	Crypto	Notes
Forecasting	•	•	•	•	next-horizon prediction; no labels needed
Classification	○	•	•	○	only on labelled datasets
Imputation	•	•	•	•	reconstruct masked subsequences
Anomaly detection	•	•	•	•	test-time point / shift / swap perturbations

Table 9: Downstream tasks used in PETS. Task availability depends on whether a domain has per-window labels.

Protocol. Private windows are split $70/30$ for synthesis and testing (seed 0, stratified when labels are available). TSTR trains a task model on synthetic windows; RTR trains the same model on private training windows. Both use the held-out real test split. Neural models reserve 10% of their respective training data for validation and early stopping. Forecasting and imputation errors are reported on the PETS preprocessing scale after undoing the task model's training-set standardization. Task coverage is summarized in Table 9.

Forecasting. A three-layer MLP (hidden width 512, dropout 0.1) predicts the final 25% of each window from its first 75%: horizons are 144, 32, 64, and 64 for traffic, activity, ECG, and crypto. Training uses AdamW at learning rate 2×10^{-3}, batch size 256, and at most 30 epochs with patience 6. The primary score is RMSE over all predicted test values.

Imputation. A two-layer bidirectional LSTM (hidden width 128, dropout 0.1) reconstructs a contiguous masked block covering 25% of each window, given the masked values and observation mask. Loss and test RMSE use only masked positions. Training masks are resampled per batch; validation and test masks use fixed seeds. Training uses AdamW at learning rate 10^{-3}, batch size 128, and at most 60 epochs with patience 10.

Classification. MultiRocket features (Tan et al., 2022) from PETS-preprocessed windows feed a standardized ridge classifier with cross-validated regularization; the shared implementation uses an MLP fallback if MultiRocket is unavailable. TSTR uses generated labels and RTR uses private training labels. The primary score is accuracy for the four activity classes and macro-F_1 for the three imbalanced ECG classes.

Anomaly detection. A temporal-convolutional autoencoder learns from unperturbed training windows and scores anomalies by mean reconstruction error per window. Its encoder and decoder each have four dilated-convolution blocks (width 64, kernel size 3, dropout 0.1). Training uses AdamW at learning rate 10^{-3}, batch size 128, and at most 60 epochs with patience 10. Test data contain equal numbers of original real windows and perturbed copies. After standardization by training-set statistics, each copy receives one random perturbation: adding ± 6 at one point, adding ± 3 over a 10% segment, or swapping a segment of that length with one from another test window. The primary score is AUROC.

G Additional Downstream Task Results

This appendix collects the full per-dataset ε-sweeps (Figure 8–11) and four cross-cutting analyses (Figure 13–16) that complement the main-text Figure 6. The results span four datasets, 14

downstream-task settings, and 13 privacy budgets, with method coverage depending on task applicability and result availability.

For the aggregate comparison in Figure 6d, mean normalized gaps measure the relative utility loss against each task's median RTR reference, clipped to $[0, 1.5]$ and averaged over all 14 settings.

All figures use the canonical paper-wide color palette: PETS three in red shades (TimeCraft ■, Diffusion-TS ■, ChatTS+VAE ■), DP baselines in cool tones (dp-GAN ■, GS-WGAN ■, GlucoSynth ■, Pub2Priv ■, DPSDA-image ■, DPSDA-text ■), and non-DP baselines in darker colors (TimeDiff ■, TimeGAN ■).

G.1 Class conditioning in DPSDA

We evaluate unconditional and class-conditional DPSDA to assess whether PETS' advantage on labelled tasks persists when the baselines also use class labels. The unconditional version runs PE on the whole private set. The conditional version runs PE separately within each class, using a separate noisy vote histogram and assigning the class label to its selected outputs. This changes selection and output labelling while retaining the same image or text variation model.

Across $\varepsilon \in \{0.5, 1, 2, \infty\}$ in Table 4, conditioning improves activity accuracy by 1.3–5.7 percentage points for DPSDA-image and 13.6–16.7 points for DPSDA-text, computed from unrounded scores. TimeCraft still exceeds the better conditional baseline by 16.3–22.6 points. Meanwhile, activity C-FID increases in seven of the eight method–budget comparisons. The exception is DPSDA-image at $\varepsilon = \infty$, where it decreases from 4.16 to 3.91. Thus, class conditioning improves label-dependent utility without consistently improving distributional fidelity, and PETS' classification advantage persists when compared with these conditional baselines.

G.2 Per-dataset full-grid views

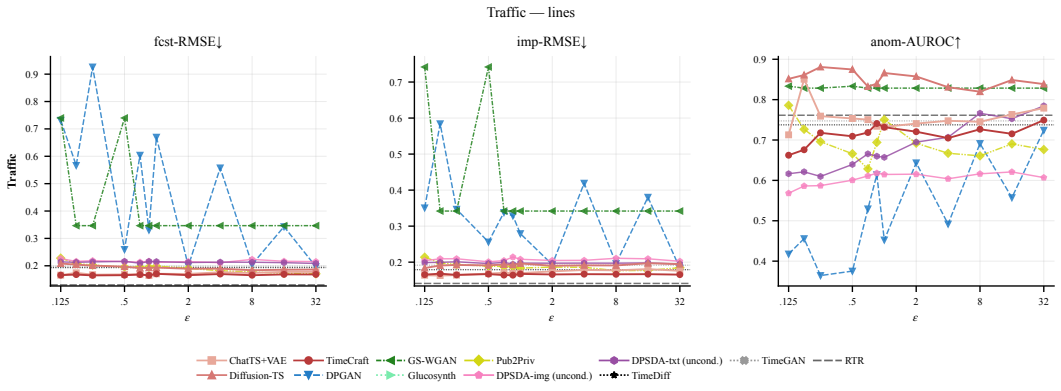


Figure 8: **Traffic.** Task metric ↓ / ↑ (x: $\log \varepsilon$ on 13-point grid; y: per-task metric) for forecasting RMSE, imputation RMSE, and anomaly-detection AUROC. Train-on-real (RTR) reference shown as dashed grey line.

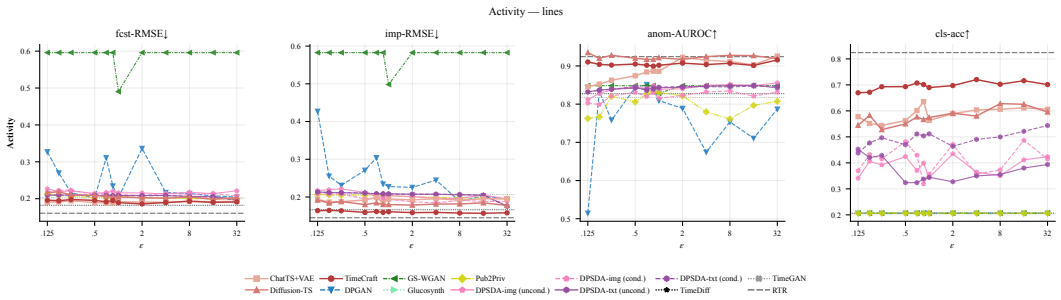


Figure 9: **Activity.** Forecasting RMSE, imputation RMSE, anomaly-detection AUROC, and classification accuracy (only Activity-cls is shown in main Figure 6b).

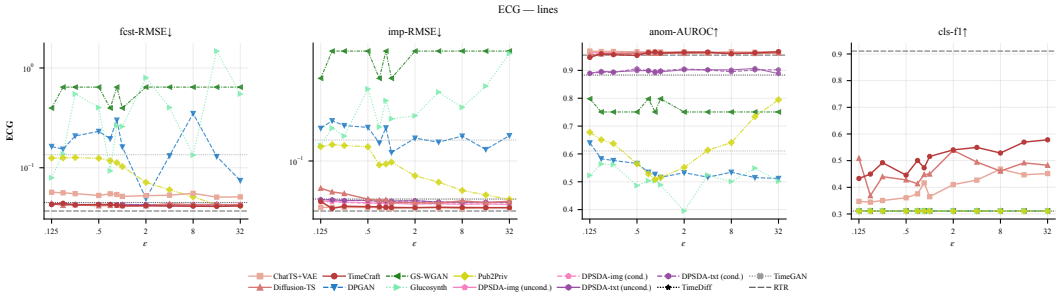


Figure 10: **ECG.** Forecasting RMSE, imputation RMSE, anomaly-detection AUROC, and classification F1 (only ECG-imp is shown in main Figure 6a).

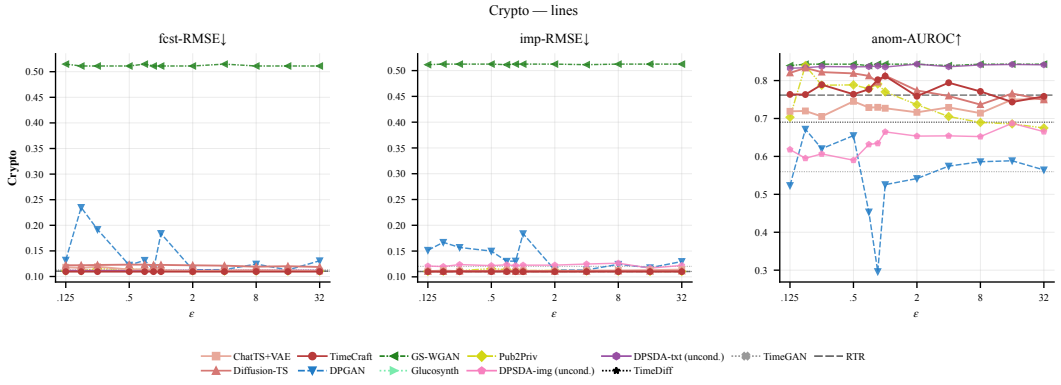


Figure 11: **Crypto.** Forecasting RMSE, imputation RMSE, anomaly-detection AUROC. Crypto only appears via the heatmap in main Figure 6d, with this figure giving the per-task expansion.

G.3 PETS EXECUTION SPEED ON GPU

All PETS schemes are 11–216× faster per iteration than the other PE-based DP baselines. Figure 12 compares per-iter runtime on a single NVIDIA A100 GPU. The gap reflects PETS' lightweight time-series-specific models versus DPSDA's large general-purpose foundation models (subsection F.6). PETS' per-iter time is stable across ε (within $\sim 4\%$) because DP noise enters only the histogram-vote step and leaves the model forward passes unchanged, while DPSDA bars vary by $\sim 53\%$ across the same ε range.

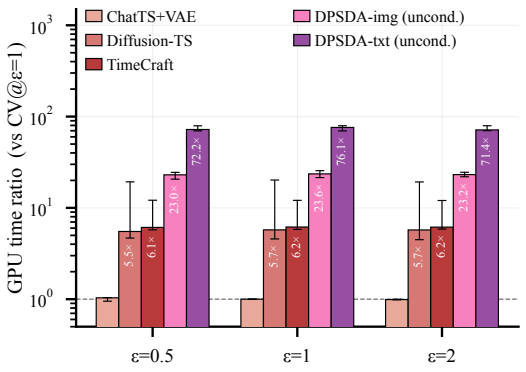


Figure 12: Per-iteration GPU time (log). Bars represent median over 4 datasets.

G.4 Cross-method analysis

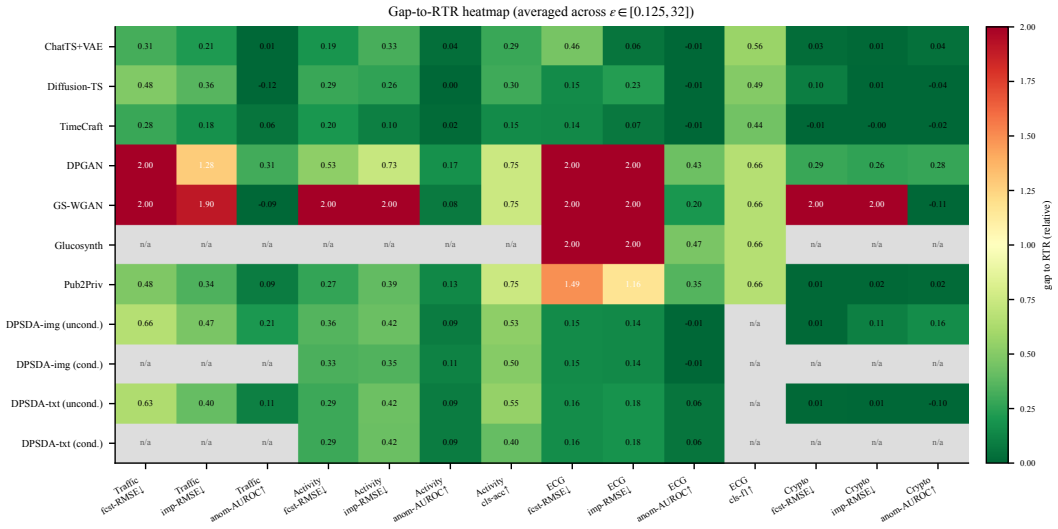


Figure 13: **Gap-to-RTR heatmap, averaged across ε.** 9 DP methods $\times$ 14 (dataset, task) settings. Compared with main Figure 6d which fixes $\varepsilon = 1$, this view averages over $\varepsilon \in [0.125, 32]$ to surface the privacy-budget-robust ranking. PETS three remain top across all settings.

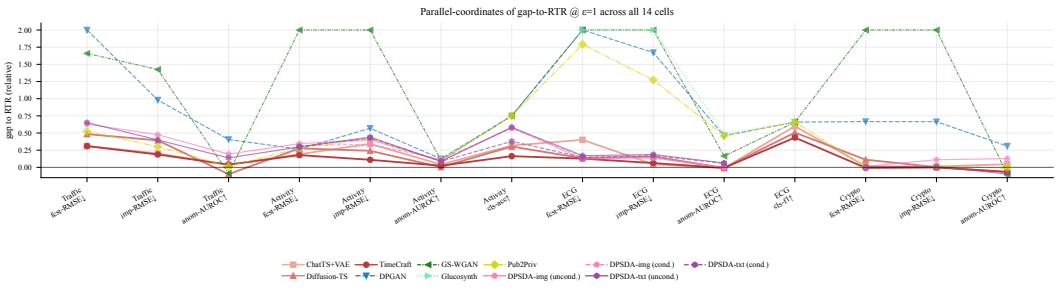


Figure 14: **Parallel-coordinates view at $\varepsilon = 1$.** Each method = one polyline across all 14 (dataset, task) settings. PETS three lie uniformly low (close to RTR floor) on every axis, while DP-SGD baselines spike on classification settings.

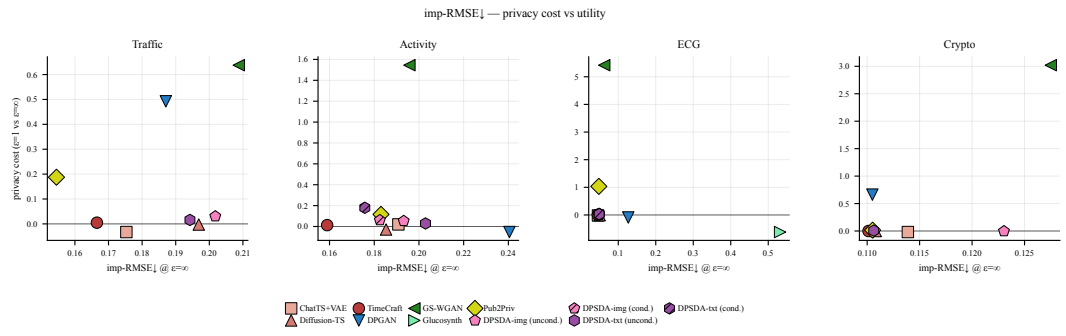


Figure 15: **Privacy cost vs. no-DP utility, imputation task on all 4 datasets.** x: imputation RMSE at $\varepsilon = \infty$ (no-DP floor); y: relative DP cost $(\text{RMSE}_{\varepsilon=1} - \text{RMSE}_{\varepsilon=\infty})/\text{RMSE}_{\varepsilon=\infty}$. PETS three cluster in the bottom-left (low floor, near-zero DP cost). The same mechanism as the main-text traffic-fcst scatter (Figure 6c) generalizes to imputation.

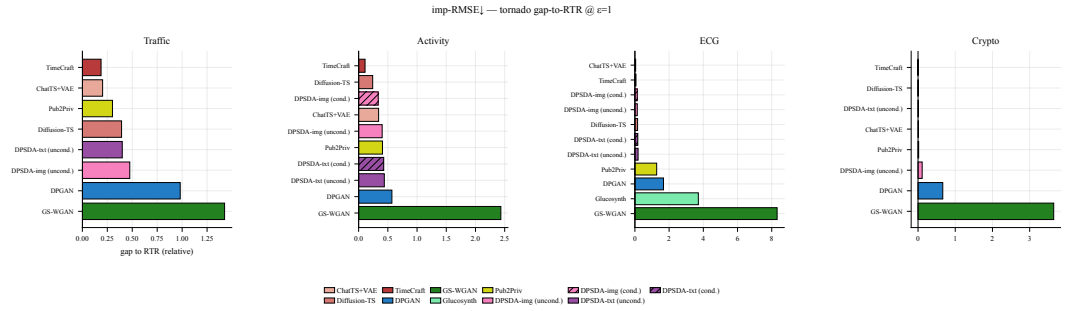


Figure 16: **Tornado plot (ranked bar chart) of gap-to-RTR for imputation at $\varepsilon = 1$.** Methods sorted by gap (smallest at top); PETS three (TimeCraft, Diffusion-TS, ChatTS+VAE) hold the top three positions on all four datasets.

H Encoder-Independent Comparison with Baselines

This section expands Table 5 across four datasets and $\varepsilon \in \{0.5, 1, 2, \infty\}$ (16 settings). The separate full-sweep C-FID summary below covers all 13 budgets. Metric definitions are in Appendix F.7.

Evaluation protocol. Runs, private PE-train references, and applicable baselines match Table 4. PETS and DPSDA use each metric's best evaluated iteration $t \geq 1$, while other baselines use their single output. ECG uses the PETS configurations of Table 3. Summaries fix TimeCraft unless explicitly comparing the per-setting best or worst scheme. Full-score tables include all three schemes.

Summary conventions. Table 10 compares TimeCraft with the best and worst schemes for each dataset, budget, and metric. Wins count settings where PETS beats every applicable baseline out of 16. Mean margins average strongest-DP-baseline/PETS ratios over the 12 finite-budget settings. Table 11 lists individual ratios, including comparisons with all baselines at $\varepsilon = \infty$. Tables 13–15 provide the full scores.

Noise floors. Table 12 reports the largest absolute score across scoring hosts when comparing two random halves of the private reference. Multiplicative margins replace scores below this empirical floor with the floor. A floor-limited PETS denominator is marked $\geq$, as for FID_{DAE} on ECG. Wins and raw-score tables retain the original scores. FID_{M}-PCA128 projects MOMENT embeddings onto 128 principal components fitted on the private reference and serves only as a covariance-estimation diagnostic.

Agreement between metric rankings. Within each setting, we correlate method rankings under C-FID with each alternative metric, using TimeCraft and all applicable baselines. Both DPSDA versions enter where applicable, and non-DP methods enter only at $\varepsilon = \infty$. Spearman correlations

use unrounded scores, average ranks for ties, and no floor clipping. All correlations are positive. For $\mathrm{FID_M}$, $\mathrm{FID_{DAE}}$, and MMD^2, their medians are 0.82, 0.90, and 0.93, with ranges 0.60–0.90, 0.70–1.00, and 0.40–1.00. Computing correlations within settings avoids pooling datasets with different score scales.

Full-sweep C-FID summary. The $6.0\times$ margin in subsection 4.2 averages 52 ratios over four datasets and all 13 budgets, including $\varepsilon = \infty$. Each divides the lowest DP-baseline C-FID by the best PETS scheme's C-FID, using unrounded scores and both DPSDA versions where applicable. The 48 finite-budget settings alone average $6.4\times$. The non-DP comparison averages four ratios at $\varepsilon = \infty$, using the better of TimeDiff and TimeGAN. All means are arithmetic.

Table 10: Wins and mean margin (as in Table 5) for three definitions of the PETS row.

	TimeCraft (default)		**Per-cell best scheme**		**Per-cell worst scheme**	
Metric	Wins	Mean margin	Wins	Mean margin	Wins	Mean margin
C-FID	14/16	6.8×	15/16	6.9×	9/16	1.9×
$\mathrm{FID_M}$	15/16	4.1×	15/16	4.1×	3/16	0.7×
$\mathrm{FID_{DAE}}$	15/16	≥6.4×	15/16	≥8.2×	6/16	1.2×
MMD^2	15/16	5.7×	15/16	5.7×	1/16	0.5×
$\mathrm{FID_M}$-PCA128	15/16	4.2×	15/16	4.2×	3/16	0.7×

Table 11: Strongest-baseline/TimeCraft ratios, with the baseline in parentheses. Finite budgets compare against DP baselines; $\varepsilon = \infty$ includes non-DP baselines. Values below $1\times$ favour the baseline. Ratios use the noise-floor convention above, with $\geq$ marking a floor-limited TimeCraft denominator.

Dataset	**Metric**	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = \infty$
Traffic	C-FID	6.2× (Pub2Priv)	5.5× (Pub2Priv)	6.4× (Pub2Priv)	0.4× (Pub2Priv)
	$\mathrm{FID_M}$	2.6× (DPSDA-img (uncond.))	2.7× (DPSDA-img (uncond.))	2.7× (DPSDA-img (uncond.))	0.2× (Pub2Priv)
	$\mathrm{FID_{DAE}}$	5.4× (Pub2Priv)	3.8× (Pub2Priv)	3.6× (Pub2Priv)	0.2× (Pub2Priv)
	MMD^2	1.7× (Pub2Priv)	1.6× (Pub2Priv)	1.6× (Pub2Priv)	0.2× (Pub2Priv)
Activity	C-FID	7.0× (Pub2Priv)	10.2× (Pub2Priv)	10.6× (Pub2Priv)	6.2× (TimeDiff)
	$\mathrm{FID_M}$	5.6× (DPSDA-img (cond.))	5.4× (DPSDA-img (cond.))	5.6× (DPSDA-img (cond.))	1.8× (Pub2Priv)
	$\mathrm{FID_{DAE}}$	11.7× (Pub2Priv)	5.8× (Pub2Priv)	6.0× (Pub2Priv)	1.3× (Pub2Priv)
	MMD^2	13.4× (Pub2Priv)	4.7× (Pub2Priv)	3.1× (Pub2Priv)	2.2× (Pub2Priv)
ECG	C-FID	1.5× (DPSDA-img (cond.))	1.6× (DPSDA-img (uncond.))	1.6× (DPSDA-img (cond.))	1.4× (Pub2Priv)
	$\mathrm{FID_M}$	3.5× (DPSDA-img (uncond.))	3.1× (DPSDA-img (uncond.))	2.7× (DPSDA-img (uncond.))	1.3× (Pub2Priv)
	$\mathrm{FID_{DAE}}$	18.0× (DPSDA-img (uncond.))	2.8× (DPSDA-img (uncond.))	≥10.7× (DPSDA-img (uncond.))	5.4× (DPSDA-img (uncond.))
	MMD^2	4.0× (DPSDA-img (uncond.))	2.7× (DPSDA-img (uncond.))	4.1× (DPSDA-img (uncond.))	4.5× (DPSDA-img (uncond.))
Crypto	C-FID	9.2× (DPSDA-img (uncond.))	16.0× (DPSDA-img (uncond.))	5.8× (DPSDA-img (uncond.))	0.7× (Pub2Priv)
	$\mathrm{FID_M}$	4.0× (Pub2Priv)	6.4× (Pub2Priv)	4.6× (Pub2Priv)	1.1× (TimeDiff)
	$\mathrm{FID_{DAE}}$	2.6× (DPSDA-img (uncond.))	4.1× (DPSDA-img (uncond.))	2.5× (DPSDA-img (uncond.))	1.9× (GS-WGAN)
	MMD^2	6.9× (DPSDA-img (uncond.))	17.1× (DPSDA-img (uncond.))	7.1× (DPSDA-img (uncond.))	3.2× (TimeDiff)

Table 12: Empirical noise floors: the largest absolute metric estimate across scoring hosts when comparing two random halves of the private reference.

Dataset	$\mathrm{FID_M}$	$\mathrm{FID_{DAE}}$	MMD^2	$\mathrm{FID_M}$-PCA128
Traffic	1.9×10^{-3}	5.9×10^{-4}	4.7×10^{-6}	3.5×10^{-4}
Activity	4.8×10^{-4}	3.5×10^{-4}	4.2×10^{-5}	6.5×10^{-4}
ECG	3.1×10^{-4}	1.5×10^{-2}	8.8×10^{-5}	3.1×10^{-4}
Crypto	2.4×10^{-3}	9.2×10^{-2}	4.9×10^{-6}	7.8×10^{-4}

Table 13: $\text{FID}_{\text{M}}\downarrow$ per setting (own best iteration for PE-based methods). **Bold**: lowest in each dataset and ε.

Dataset	Method	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = \infty$	Dataset	Method	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = \infty$
Activity	dp-GAN	2.026	1.952	1.456	0.812	ECG	dp-GAN	2.766	1.449	0.916	2.412
	GS-WGAN	3.030	3.030	3.030	0.429		GS-WGAN	1.810	1.810	1.810	0.500
	Pub2Priv	1.608	1.443	1.094	0.213		GlucoSynth	0.839	0.655	1.084	1.081
	DPSDA-img (uncond.)	0.835	0.856	0.804	0.590		Pub2Priv	2.126	2.086	1.915	0.178
	DPSDA-img (cond.)	0.830	0.760	0.685	0.646		DPSDA-img (uncond.)	0.480	0.428	0.403	0.393
	DPSDA-txt (uncond.)	1.509	1.484	1.469	1.497		DPSDA-img (cond.)	0.487	0.444	0.404	0.395
	DPSDA-txt (cond.)	1.504	1.520	1.484	1.377		DPSDA-txt (uncond.)	1.008	1.016	1.030	1.042
	PETS (ChatTS+VAE)	0.936	0.936	0.918	0.913		DPSDA-txt (cond.)	1.018	1.028	1.045	1.060
	PETS (Diffusion-TS)	0.327	0.281	0.244	0.234		PETS (ChatTS+VAE)	0.567	0.553	0.591	0.611
	PETS (TimeCraft)	**0.148**	**0.140**	**0.122**	**0.121**		PETS (Diffusion-TS)	0.186	0.162	0.158	0.158
	TimeDiff	–	–	–	0.365		**PETS (TimeCraft)**	**0.136**	**0.140**	**0.147**	**0.142**
	TimeGAN	–	–	–	4.338		*TimeDiff*	–	–	–	0.627
Traffic	dp-GAN	1.748	1.540	1.971	1.157		*TimeGAN*	–	–	–	3.865
	GS-WGAN	2.328	2.328	2.328	0.379	Crypto	dp-GAN	2.635	2.390	0.174	0.586
	Pub2Priv	1.307	0.947	0.734	**0.031**		GS-WGAN	3.089	3.089	3.089	0.584
	DPSDA-img (uncond.)	0.556	0.518	0.519	0.494		Pub2Priv	0.086	0.089	0.081	0.029
	DPSDA-txt (uncond.)	1.204	1.173	1.141	0.998		DPSDA-img (uncond.)	2.067	2.048	2.044	2.088
	PETS (ChatTS+VAE)	0.312	0.271	0.246	0.247		DPSDA-txt (uncond.)	1.378	1.342	1.308	0.259
	PETS (Diffusion-TS)	0.502	0.491	0.503	0.491		PETS (ChatTS+VAE)	0.535	0.515	0.500	0.502
	PETS (TimeCraft)	**0.213**	**0.194**	**0.192**	0.178		PETS (Diffusion-TS)	0.089	0.084	0.073	0.058
	TimeDiff	–	–	–	0.447		**PETS (TimeCraft)**	**0.021**	**0.014**	**0.018**	**0.024**
	TimeGAN	–	–	–	2.588		*TimeDiff*	–	–	–	0.026
							TimeGAN	–	–	–	2.414

Table 14: $\text{FID}_{\text{DAE}}\downarrow$ per setting (own best iteration for PE-based methods). **Bold**: lowest in each dataset and ε; see Table 12 for the noise floor.

Dataset	Method	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = \infty$	Dataset	Method	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = \infty$
Activity	dp-GAN	1.120	1.825	1.405	3.770	ECG	dp-GAN	17.413	10.882	22.189	10.826
	GS-WGAN	34.017	21.703	34.017	0.149		GS-WGAN	153.004	146.983	153.004	4.171
	Pub2Priv	0.463	0.195	0.181	0.044		GlucoSynth	49.622	59.078	21.943	111.731
	DPSDA-img (uncond.)	0.704	0.653	0.584	0.484		Pub2Priv	10.538	8.420	8.156	1.041
	DPSDA-img (cond.)	0.615	0.563	0.542	0.478		DPSDA-img (uncond.)	0.500	0.192	0.164	0.130
	DPSDA-txt (uncond.)	8.522	7.083	5.485	0.302		DPSDA-img (cond.)	0.639	0.205	0.208	0.183
	DPSDA-txt (cond.)	8.699	7.823	6.659	2.219		DPSDA-txt (uncond.)	38.258	31.633	25.224	1.607
	PETS (ChatTS+VAE)	0.321	0.280	0.226	0.141		DPSDA-txt (cond.)	39.382	30.268	25.161	0.936
	PETS (Diffusion-TS)	0.253	0.212	0.142	0.136		PETS (ChatTS+VAE)	0.382	0.489	0.299	0.304
	PETS (TimeCraft)	**0.040**	**0.034**	**0.030**	**0.034**		PETS (Diffusion-TS)	0.161	0.330	0.323	0.091
	TimeDiff	–	–	–	0.149		**PETS (TimeCraft)**	**0.028**	**0.068**	**-0.003**	**0.024**
	TimeGAN	–	–	–	0.388		*TimeDiff*	–	–	–	3.735
Traffic	dp-GAN	2.797	0.920	0.564	1.132		*TimeGAN*	–	–	–	10.089
	GS-WGAN	127.986	5.669	5.669	0.053	Crypto	dp-GAN	15.769	22.932	19.976	11.914
	Pub2Priv	0.328	0.223	0.200	**0.009**		GS-WGAN	1883.540	1883.540	1883.540	0.767
	DPSDA-img (uncond.)	2.530	2.638	1.777	1.563		Pub2Priv	24.389	6.416	6.644	6.729
	DPSDA-txt (uncond.)	71.219	67.422	62.430	5.324		DPSDA-img (uncond.)	1.719	1.244	1.472	1.553
	PETS (ChatTS+VAE)	0.062	**0.058**	**0.056**	0.040		DPSDA-txt (uncond.)	877.212	876.279	871.845	187.703
	PETS (Diffusion-TS)	1.942	1.209	1.137	0.563		PETS (ChatTS+VAE)	0.433	0.311	0.773	0.709
	PETS (TimeCraft)	**0.061**	**0.058**	**0.056**	0.052		PETS (Diffusion-TS)	**0.186**	**0.060**	**0.191**	**0.049**
	TimeDiff	–	–	–	0.157		**PETS (TimeCraft)**	0.672	0.304	0.585	0.402
	TimeGAN	–	–	–	0.330		*TimeDiff*	–	–	–	7.934
							TimeGAN	–	–	–	12.521

Table 15: $\text{MMD}^2\downarrow$ on raw windows per setting (own best iteration for PE-based methods). **Bold**: lowest in each dataset and ε.

Dataset	Method	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = \infty$	Dataset	Method	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = \infty$
Activity	dp-GAN	0.3538	0.4804	0.3652	0.1685	ECG	dp-GAN	0.6587	0.3162	0.6671	0.1007
	GS-WGAN	0.8762	0.7912	0.8762	0.0141		GS-WGAN	1.3095	1.0470	1.3095	0.0346
	Pub2Priv	0.0178	0.0059	0.0040	0.0037		GlucoSynth	0.5612	0.5661	0.5714	1.0161
	DPSDA-img (uncond.)	0.0271	0.0363	0.0344	0.0240		Pub2Priv	0.2022	0.1153	0.0757	0.0146
	DPSDA-img (cond.)	0.0317	0.0310	0.0312	0.0245		DPSDA-img (uncond.)	0.0194	0.0091	0.0084	0.0082
	DPSDA-txt (uncond.)	0.2131	0.1642	0.1056	0.0113		DPSDA-img (cond.)	0.0219	0.0092	0.0084	0.0084
	DPSDA-txt (cond.)	0.2237	0.1820	0.1477	0.0286		DPSDA-txt (uncond.)	0.2343	0.1941	0.1486	0.0230
	PETS (ChatTS+VAE)	0.0124	0.0099	0.0083	0.0093		DPSDA-txt (cond.)	0.2410	0.1803	0.1494	0.0191
	PETS (Diffusion-TS)	0.0053	0.0046	0.0042	0.0042		PETS (ChatTS+VAE)	0.0242	0.0220	0.0209	0.0220
	PETS (TimeCraft)	**0.0013**	**0.0013**	**0.0013**	**0.0017**		PETS (Diffusion-TS)	0.0358	0.0202	0.0200	0.0139
	TimeDiff	–	–	–	0.0051		**PETS (TimeCraft)**	**0.0048**	**0.0034**	**0.0020**	**0.0018**
	TimeGAN	–	–	–	0.0045		*TimeDiff*	–	–	–	0.0489
Traffic	dp-GAN	0.2133	0.5709	0.0804	0.0617		*TimeGAN*	–	–	–	0.0766
	GS-WGAN	1.0721	0.6396	0.6396	0.0372	Crypto	dp-GAN	0.4463	0.5287	0.1701	0.0058
	Pub2Priv	0.0213	0.0220	0.0224	**0.0031**		GS-WGAN	1.2245	1.2245	1.2245	0.0448
	DPSDA-img (uncond.)	0.0707	0.0508	0.0517	0.0503		Pub2Priv	0.0844	0.0414	0.0149	0.0061
	DPSDA-txt (uncond.)	0.5022	0.4611	0.3839	0.0271		DPSDA-img (uncond.)	0.0045	0.0054	0.0062	0.0041
	PETS (ChatTS+VAE)	0.0248	0.0255	0.0252	0.0270		DPSDA-txt (uncond.)	0.4443	0.4390	0.4245	0.1136
	PETS (Diffusion-TS)	0.0648	0.0490	0.0456	0.0565		PETS (ChatTS+VAE)	0.0033	0.0038	0.0038	0.0037
	PETS (TimeCraft)	**0.0125**	**0.0137**	**0.0142**	0.0143		PETS (Diffusion-TS)	0.0248	0.0227	0.0218	0.0197
	TimeDiff	–	–	–	0.0267		**PETS (TimeCraft)**	**0.0007**	**0.0003**	**0.0009**	**0.0013**
	TimeGAN	–	–	–	0.0200		*TimeDiff*	–	–	–	0.0040
							TimeGAN	–	–	–	0.0059

I PUBLIC–PRIVATE DOMAIN MISMATCH: SETUP AND FULL RESULTS

This section supplements Table 6 with the experimental setup, public-data and Pub2Priv references at matched sample sizes, and additional fidelity and downstream results.

Data sources. MIT-BIH Arrhythmia (Moody & Mark, 2001) is the public source. Private cohorts are MIT-BIH Supraventricular Arrhythmia (SVDB) (Arrhythmia, 1999; Greenwald, 1990), MIT-BIH Normal Sinus Rhythm (NSRDB) (Moody, 1999), METR-LA (Li et al., 2018), and UCR's UWaveGestureLibraryX, NonInvasiveFetalECGThorax1, ChlorineConcentration, and FordA (Dau et al., 2018). They represent adult and fetal ECG, gestures, traffic, simulated water-network chlorine concentrations, and engine noise. Rows follow increasing C-FID gap. Changing private targets means that these trends reflect both domain alignment and target-data characteristics.

Preprocessing and splits. SVDB and NSRDB use PhysioNet lead `ECG1`, resampled to 250 Hz and cut into beat windows. UCR datasets pool TRAIN and TEST splits, remove labels, and interpolate to length 256. METR-LA uses the PETS traffic pipeline followed by the same interpolation. Each cohort samples 3,765 windows without replacement (seed 0). Arrays in $[0, 1]$ retain their scale; others undergo per-window z-scoring, clipping, and min-max scaling with the ECG pipeline's joint public/private 0.5/99.5 percentiles. All labels become one class. The split contains 2,635 PE-train and 1,130 test windows (seed 0).

PETS. $\varepsilon = 1$, $\delta = 10^{-5}$, $T = 20$ DP iterations (21 snapshots), generation batch 64, selection mode `sample`, seed 0. Each scheme uses its ECG checkpoint and PE configuration from Table 3:

- TimeCraft: generator trained at lr 3×10^{-4}, batch 128, step 4,000; schedule `[45,40,35,30,25,20,15,12,10,8, 6×11]`, $k = 0$, $H = 2$, $N_{\text{nn}} = 7$, $s_f = 0.5$ (1,318 samples).
- Diffusion-TS: generator at step 13,500; schedule `[60,55,50,45,40,35,30,25,20,15, 10×11]`, $k = 8$, $H = 8$, $N_{\text{nn}} = 15$, $s_f = 0.25$ (659 samples).
- ChatTS+VAE: VAE trained at lr 3×10^{-4} with latent dimension 32 for 90 epochs (final checkpoint); schedule `[45,40,35,30,25,20,15,12,10,8, 6×11]`, $k = 8$, $H = 1$, $N_{\text{nn}} = 11$, $s_f = 1.0$ (2,635 samples).

The three schemes retain these configurations and their different output sizes in the main table. The comparisons below match each reference arm to the corresponding scheme's output size n.

Reference arms. Public samples n unchanged MIT-BIH windows. Pub2Priv trains on the same private PE-train split with MIT-BIH public context, targeting $\varepsilon = 1$ (accounted 0.994, $\delta = 10^{-5}$, noise multiplier 14.06). We draw n of its 3,500 outputs. Both references use sampling without replacement (seed 0), with n matched to each PETS scheme. Implementation details are in Appendix F.6.

Fidelity metrics and gap. All scores use the 2,635 private PE-train windows as reference. C-FID uses the public MIT-BIH TS2Vec encoder, FID_{M} uses frozen MOMENT, and MMD^2 uses raw windows (Appendix F.7). The main-table Gap scores 1,318 unchanged public windows against this reference; it is distinct from PETS' generated S_0. The full fidelity table uses each scheme's n for Public. Noise floors compare two random halves of the private reference (1,317 windows each). PETS uses each metric's best evaluated iteration $t \geq 1$.

Downstream evaluation. Single-class cohorts support forecasting, imputation, and injected-anomaly detection. Task models train on synthetic or public windows and evaluate on private test windows. RTR instead trains on the private training split. PETS uses the best score at $t = 3, 6, \ldots, 18$. All arms use the same callbacks, with identical forecasting and imputation RTR scores. Injected-anomaly RTR varies by up to 0.032 AUROC between calls, limiting interpretation of small differences.

The main-text mean FID_{M} and imputation reductions average per-cohort percentage changes over all seven cohorts, including increases, relative to the Gap and public-trained imputation references, respectively. Both references use 1,318 public windows, matching TimeCraft's output size.

Full results. Table 16 gives gaps, floors, selected iterations, and MOMENT pretraining coverage. Five cohorts occur in MOMENT's pretraining corpus, so FID_{M} is independent of PE voting but not

an unseen-domain evaluation there. Tables 17 and 18 report fidelity and utility at matched sample sizes.

Table 16: Domain mismatch: gap (the public data scored as an output, $n = 1{,}318$) and noise floor per metric, the C-FID-selected iteration of each PETS scheme, and whether the cohort is part of MOMENT's pretraining corpus (Time-series Pile).

	Gap (public, n=1,318)			**Noise floor** (half-split)			**C-FID-selected iterate**			**In MOMENT**
Private cohort	C-FID	MMD^2	FID_M	C-FID	MMD^2	FID_M	ChatTS+VAE	Diffusion-TS	TimeCraft	**corpus**
SVDB	2.07	0.061	0.134	0.455	1.2×10^{-4}	0.0028	t=6	t=5	t=20	yes
NSRDB	6.14	0.103	0.246	0.334	3.6×10^{-5}	0.0025	t=10	t=9	t=13	no
UWaveGestureLibraryX	19.51	0.242	0.723	0.336	-3.7×10^{-5}	0.0020	t=6	t=8	t=8	yes
NonInvasiveFetalECGThorax1	25.84	0.592	0.812	0.148	-8.5×10^{-5}	-0.0013	t=10	t=7	t=20	yes
METR-LA	30.40	0.483	0.787	0.249	2.0×10^{-4}	0.0088	t=15	t=6	t=20	no
ChlorineConcentration	36.57	0.464	1.554	0.089	-1.2×10^{-4}	-0.0012	t=11	t=16	t=20	yes
FordA	40.09	0.221	3.383	0.138	-1.7×10^{-5}	0.0031	t=13	t=10	t=19	yes

Table 17: Domain mismatch, fidelity at matched sample sizes: Public and Pub2Priv drawn at each PETS scheme's n, next to that scheme. PETS values at each metric's own best iteration. **Bold**: best of the three in each group.

	$n = 659$			$n = 1{,}318$			$n = 2{,}635$		
Private cohort	Public	Pub2Priv	Diffusion-TS	Public	Pub2Priv	TimeCraft	Public	Pub2Priv	ChatTS+VAE
C-FID↓									
SVDB	**2.38**	25.47	2.64	2.07	25.33	**2.01**	**1.94**	25.36	3.30
NSRDB	6.27	26.36	**3.03**	6.14	26.22	**1.97**	6.12	26.24	**4.18**
UWaveGestureLibraryX	19.69	15.57	**4.52**	19.51	15.45	**2.60**	19.40	15.37	**6.94**
NonInvasiveFetalECGThorax1	25.96	12.70	**5.54**	25.84	12.68	**3.44**	25.72	12.69	**9.93**
METR-LA	30.44	**5.17**	6.52	30.40	5.06	**4.09**	30.31	**5.01**	8.28
ChlorineConcentration	36.68	**5.85**	14.10	36.57	**5.76**	8.89	36.50	**5.73**	12.71
FordA	40.20	**4.66**	19.28	40.09	**4.54**	12.47	40.04	**4.49**	19.72
FID_M ↓									
SVDB	**0.128**	2.398	0.175	**0.134**	2.409	0.159	**0.124**	2.413	0.224
NSRDB	0.246	2.462	**0.224**	0.246	2.511	**0.151**	**0.235**	2.516	0.253
UWaveGestureLibraryX	0.731	2.351	**0.268**	0.723	2.370	**0.220**	0.735	2.358	**0.283**
NonInvasiveFetalECGThorax1	0.805	1.608	**0.295**	0.812	1.615	**0.268**	0.815	1.625	**0.378**
METR-LA	**0.787**	1.683	0.830	0.787	1.651	**0.724**	0.779	1.649	**0.676**
ChlorineConcentration	1.552	1.871	**1.336**	1.554	1.880	**1.462**	1.542	1.867	**1.230**
FordA	3.396	**2.552**	3.474	3.383	**2.531**	3.504	3.366	**2.515**	3.343
MMD^2 ↓									
SVDB	0.065	0.113	**0.061**	0.061	0.113	**0.049**	0.060	0.114	**0.024**
NSRDB	**0.104**	0.192	0.135	**0.103**	0.193	0.129	0.102	0.193	**0.094**
UWaveGestureLibraryX	0.244	**0.029**	0.106	0.242	**0.029**	0.041	0.241	**0.029**	0.124
NonInvasiveFetalECGThorax1	0.595	**0.332**	0.455	0.592	**0.332**	0.427	0.592	**0.333**	0.445
METR-LA	0.485	**0.041**	0.197	0.483	**0.041**	0.107	0.480	**0.041**	0.154
ChlorineConcentration	0.468	**0.221**	0.457	0.464	**0.220**	0.445	0.462	**0.220**	0.321
FordA	0.223	**0.009**	0.103	0.221	**0.009**	0.112	0.220	**0.009**	0.127

Window-level diagnostics. Pub2Priv's median mean absolute step change is 0.27–0.31, versus 0.012–0.149 for private windows (Table 19). ChlorineConcentration and FordA have the greatest private-data variability, which may partly explain Pub2Priv's relative C-FID advantage. TimeCraft remains smoother (0.015–0.019), consistent with limited adaptation to rapidly varying signals. ChatTS+VAE's repeated medians reflect concentrated window-level means and standard deviations. These are descriptive observations, not causal tests.

Table 18: Domain mismatch, downstream utility at matched sample sizes (TSTR on the private test windows; PETS at the best evaluated iteration). RTR in grey; for injected anomalies it is the TimeCraft run's value (see text). **Bold**: best of the three in each group.

		$n = 659$			$n = 1{,}318$			$n = 2{,}635$		
Private cohort	RTR	Public	Pub2Priv	Diffusion-TS	Public	Pub2Priv	TimeCraft	Public	Pub2Priv	ChatTS+VAE
Imputation RMSE↓										
SVDB	0.099	**0.112**	0.123	**0.112**	**0.108**	0.122	0.109	**0.106**	0.122	0.110
NSRDB	0.088	0.119	0.129	**0.115**	0.116	0.128	**0.111**	**0.098**	0.127	0.118
UWaveGestureLibraryX	0.148	0.329	0.357	**0.260**	0.310	0.359	**0.267**	**0.298**	0.361	0.348
NonInvasiveFetalECGThorax1	0.060	0.297	**0.264**	0.351	0.283	0.257	**0.168**	**0.249**	0.269	0.329
METR-LA	0.148	0.296	**0.185**	0.189	0.231	**0.184**	0.197	0.232	**0.183**	0.199
ChlorineConcentration	0.069	0.330	0.325	**0.322**	0.319	0.323	**0.318**	**0.318**	0.321	0.329
FordA	0.286	0.430	**0.349**	0.359	0.416	**0.349**	0.362	0.405	**0.348**	0.357
Forecasting RMSE↓										
SVDB	0.087	**0.100**	0.106	0.106	**0.105**	**0.105**	0.109	**0.100**	0.104	0.110
NSRDB	0.087	0.095	0.112	**0.094**	**0.099**	0.112	0.102	**0.097**	0.111	0.117
UWaveGestureLibraryX	0.169	0.369	**0.357**	0.375	0.362	**0.357**	0.384	0.352	0.360	**0.348**
NonInvasiveFetalECGThorax1	0.059	0.247	0.244	**0.239**	0.249	**0.241**	0.326	0.300	0.238	**0.175**
METR-LA	0.140	0.244	**0.201**	0.218	0.258	**0.202**	0.227	0.237	**0.202**	0.231
ChlorineConcentration	0.024	0.347	**0.336**	0.383	0.512	0.335	**0.330**	0.411	**0.329**	0.358
FordA	0.326	0.354	**0.348**	0.355	0.374	**0.347**	0.365	0.367	**0.349**	0.357
Injected-anomaly AUROC↑										
SVDB	0.964	**0.891**	0.401	0.879	**0.917**	0.496	0.886	**0.922**	0.519	0.918
NSRDB	0.924	**0.880**	0.424	0.859	**0.904**	0.488	0.878	0.914	0.588	**0.917**
UWaveGestureLibraryX	0.980	0.752	0.669	**0.965**	0.917	0.672	**0.963**	0.959	0.694	**0.980**
NonInvasiveFetalECGThorax1	0.951	0.813	0.722	**0.939**	0.883	0.761	**0.948**	0.900	0.775	**0.947**
METR-LA	0.859	0.625	0.554	**0.852**	0.567	0.517	**0.751**	0.553	0.587	**0.776**
ChlorineConcentration	0.892	0.562	**0.699**	0.612	0.567	**0.708**	0.684	0.569	0.738	**0.823**
FordA	0.906	0.583	**0.782**	0.652	0.578	**0.794**	0.761	0.576	0.796	**0.831**

Table 19: Median per-window mean, standard deviation, and mean absolute step change $|\Delta| = \frac{1}{L-1}\sum_{t=2}^{L}|x_t - x_{t-1}|$. Private uses the 2,635 PE-train windows, Public and Pub2Priv use 1,318 windows, and each PETS scheme uses its C-FID-selected population.

	Private (ref.)			Public			Pub2Priv			ChatTS+VAE			Diffusion-TS			TimeCraft		
Private cohort	mean	std	$\|\Delta\|$	mean	std	$\|\Delta\|$	mean	std	$\|\Delta\|$	mean	std	$\|\Delta\|$	mean	std	$\|\Delta\|$	mean	std	$\|\Delta\|$
SVDB	0.379	0.106	0.012	0.376	0.105	0.011	0.397	0.254	0.285	0.382	0.109	0.009	0.379	0.101	0.007	0.382	0.105	0.015
NSRDB	0.355	0.110	0.012	0.376	0.105	0.011	0.389	0.247	0.275	0.360	0.122	0.011	0.354	0.106	0.010	0.350	0.105	0.016
UWaveGestureLibraryX	0.492	0.358	0.013	0.376	0.105	0.011	0.508	0.285	0.311	0.420	0.223	0.014	0.412	0.337	0.016	0.495	0.373	0.019
NonInvasiveFetalECGThorax1	0.375	0.351	0.025	0.376	0.105	0.011	0.413	0.281	0.268	0.420	0.307	0.016	0.399	0.367	0.017	0.359	0.322	0.018
METR-LA	0.721	0.189	0.059	0.376	0.105	0.011	0.674	0.250	0.276	0.642	0.111	0.012	0.459	0.397	0.014	0.748	0.146	0.015
ChlorineConcentration	0.387	0.347	0.105	0.376	0.105	0.011	0.434	0.281	0.304	0.420	0.307	0.014	0.247	0.191	0.010	0.194	0.167	0.018
FordA	0.500	0.348	0.149	0.376	0.105	0.011	0.514	0.283	0.310	0.420	0.307	0.014	0.524	0.136	0.010	0.971	0.212	0.018

J BEST-ITERATION EVALUATION AND FINAL RELEASE

For PETS and DPSDA, each fidelity metric and downstream task is reported at its own best evaluated iteration $t \geq 1$, using the primary task scores defined in Appendix F.8. Table 4 reports the best result across the three PETS schemes in each setting, while Table 5 uses TimeCraft as the PETS backend.

PETS releases the final population S_T at the predetermined iteration $T = 20$, under the fixed-configuration privacy guarantee in Appendix B. Table 20 compares the reported best-iteration results with those on S_T. Tables 21–24 provide the corresponding per-setting values.

On S_T, TimeCraft still outperforms every DP baseline in all 12 settings with $\varepsilon \in \{0.5, 1, 2\}$ under C-FID, FID_M, and FID_DAE. Its MMD^2 wins decrease from $12/12$ to $7/12$; in the domain-mismatch experiment, the number of cohorts with MMD^2 below the public-data gap falls from $6/7$ to $3/7$. Thus, the best-iteration MMD^2 advantage does not consistently carry over to the final release.

Table 20: Main comparisons at the best reported iteration and on the released set S_T ($t = T = 20$). PETS is the fixed TimeCraft backend except in the comparison of the three schemes. Wins count the $\langle$dataset, $\varepsilon\rangle$ settings, out of 16 (12 at finite ε), in which TimeCraft is better than every baseline; margins are those of Table 5 ($\varepsilon \leq 2$). In Table 6, the gap is the public data scored as an output.

Table	**Comparison**	**Best iteration (reported)**	**Released S_T**
Table 3	TimeCraft lowest C-FID of the three schemes	14/16	15/16
	TimeCraft lowest ACD of the three schemes	15/16	16/16
	TimeCraft lowest FID_M of the three schemes	16/16	16/16
Table 4, Table 5	C-FID: wins (at finite ε); mean margin; smallest margin	14/16 (12/12); 6.8×; 1.51×	14/16 (12/12); 6.9×; 1.68×
	FID_M: wins (at finite ε); mean margin; smallest margin	15/16 (12/12); 4.1×; 2.61×	13/16 (12/12); 3.5×; 2.17×
	FID_DAE: wins (at finite ε); mean margin; smallest margin	15/16 (12/12); ≥6.4×; 2.52×	13/16 (12/12); ≥5.7×; 1.80×
	MMD^2: wins (at finite ε); mean margin; smallest margin	15/16 (12/12); 5.7×; 1.58×	9/16 (7/12); 4.1×; 0.41×
Table 6	TimeCraft below the gap, C-FID	7/7	7/7
	TimeCraft below the gap, FID_M	5/7	5/7
	TimeCraft below the gap, MMD^2	6/7	3/7

Table 21: C-FID↓ of every PE-based method: best iteration $t \geq 1$ / released S_T ($t = 20$). The other baselines release a single output, so their values are those of Table 4.

Dataset	**Method**	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = \infty$	**Dataset**	**Method**	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = \infty$
Activity	PETS (TimeCraft)	0.45 / 0.45	0.33 / 0.35	0.30 / 0.31	0.26 / 0.26	ECG	PETS (TimeCraft)	2.47 / 2.47	1.73 / 1.73	1.46 / 1.46	1.17 / 1.17
	PETS (Diffusion-TS)	1.53 / 1.56	1.18 / 1.61	0.97 / 1.55	0.85 / 1.32		PETS (Diffusion-TS)	4.64 / 5.25	3.37 / 4.93	3.36 / 5.16	3.10 / 4.19
	PETS (ChatTS+VAE)	3.04 / 3.39	2.93 / 3.50	2.48 / 3.47	1.84 / 3.40		PETS (ChatTS+VAE)	3.68 / 3.73	3.31 / 3.63	3.00 / 3.43	2.73 / 3.04
	DPSDA-img (uncond.)	6.42 / 7.86	5.23 / 6.00	4.26 / 4.53	4.16 / 4.16		DPSDA-img (uncond.)	3.91 / 4.16	2.81 / 3.21	2.41 / 2.89	2.24 / 2.74
	DPSDA-img (cond.)	8.35 / 8.35	6.66 / 6.66	5.86 / 5.91	3.91 / 3.91		DPSDA-img (cond.)	3.74 / 4.28	2.90 / 3.60	2.29 / 2.86	2.06 / 3.08
	DPSDA-txt (uncond.)	24.22 / 24.22	20.83 / 20.83	15.60 / 15.60	2.57 / 2.57		DPSDA-txt (uncond.)	19.30 / 19.30	16.60 / 16.87	13.83 / 13.83	5.60 / 5.60
	DPSDA-txt (cond.)	25.16 / 25.16	21.66 / 21.85	18.50 / 18.50	6.30 / 6.30		DPSDA-txt (cond.)	20.22 / 20.22	16.44 / 16.44	15.07 / 15.07	4.95 / 4.95
Traffic	PETS (TimeCraft)	2.00 / 2.00	1.62 / 1.62	1.35 / 1.35	1.18 / 1.18	Crypto	PETS (TimeCraft)	0.68 / 0.79	0.33 / 0.35	0.66 / 0.69	0.80 / 0.83
	PETS (Diffusion-TS)	9.28 / 9.28	8.36 / 8.63	8.20 / 8.28	8.15 / 8.23		PETS (Diffusion-TS)	1.67 / 1.85	1.05 / 1.31	0.74 / 0.77	0.51 / 0.55
	PETS (ChatTS+VAE)	4.62 / 5.88	3.30 / 6.27	2.65 / 6.01	2.39 / 5.84		PETS (ChatTS+VAE)	0.76 / 3.98	0.58 / 3.79	0.54 / 4.86	0.61 / 4.91
	DPSDA-img (uncond.)	16.85 / 17.82	15.81 / 16.74	13.74 / 15.07	14.04 / 14.04		DPSDA-img (uncond.)	6.28 / 8.43	5.35 / 5.52	3.80 / 4.07	4.53 / 6.13
	DPSDA-txt (uncond.)	37.73 / 37.90	35.59 / 35.59	32.22 / 32.22	6.66 / 6.66		DPSDA-txt (uncond.)	43.22 / 44.69	43.15 / 43.72	42.69 / 42.69	11.24 / 11.24

Table 22: FID_M↓ of every PE-based method: best iteration $t \geq 1$ / released S_T ($t = 20$). The other baselines release a single output, so their values are those of Table 13.

Dataset	**Method**	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = \infty$	**Dataset**	**Method**	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = \infty$
Activity	PETS (TimeCraft)	0.148 / 0.148	0.140 / 0.146	0.122 / 0.126	0.121 / 0.126	ECG	PETS (TimeCraft)	0.136 / 0.183	0.140 / 0.183	0.147 / 0.201	0.142 / 0.199
	PETS (Diffusion-TS)	0.327 / 0.358	0.281 / 0.396	0.244 / 0.373	0.234 / 0.341		PETS (Diffusion-TS)	0.186 / 0.316	0.162 / 0.374	0.158 / 0.370	0.158 / 0.343
	PETS (ChatTS+VAE)	0.936 / 0.958	0.936 / 1.011	0.918 / 1.051	0.913 / 0.987		PETS (ChatTS+VAE)	0.567 / 0.670	0.553 / 0.656	0.591 / 0.655	0.611 / 0.624
	DPSDA-img (uncond.)	0.835 / 0.835	0.856 / 0.856	0.804 / 0.804	0.590 / 0.590		DPSDA-img (uncond.)	0.480 / 0.512	0.428 / 0.468	0.403 / 0.447	0.393 / 0.447
	DPSDA-img (cond.)	0.830 / 0.830	0.760 / 0.760	0.685 / 0.685	0.646 / 0.646		DPSDA-img (cond.)	0.487 / 0.503	0.444 / 0.485	0.404 / 0.452	0.395 / 0.446
	DPSDA-txt (uncond.)	1.509 / 1.545	1.484 / 1.514	1.469 / 1.469	1.497 / 1.573		DPSDA-txt (uncond.)	1.008 / 1.044	1.016 / 1.045	1.030 / 1.075	1.042 / 1.109
	DPSDA-txt (cond.)	1.504 / 1.556	1.520 / 1.552	1.484 / 1.484	1.377 / 1.377		DPSDA-txt (cond.)	1.018 / 1.103	1.028 / 1.055	1.045 / 1.120	1.060 / 1.169
Traffic	PETS (TimeCraft)	0.213 / 0.247	0.194 / 0.210	0.192 / 0.241	0.178 / 0.194	Crypto	PETS (TimeCraft)	0.021 / 0.035	0.014 / 0.017	0.018 / 0.021	0.024 / 0.031
	PETS (Diffusion-TS)	0.502 / 0.516	0.491 / 0.491	0.503 / 0.558	0.491 / 0.570		PETS (Diffusion-TS)	0.089 / 0.120	0.084 / 0.124	0.073 / 0.110	0.058 / 0.110
	PETS (ChatTS+VAE)	0.312 / 0.397	0.271 / 0.468	0.246 / 0.465	0.247 / 0.458		PETS (ChatTS+VAE)	0.535 / 0.929	0.515 / 1.127	0.500 / 1.126	0.502 / 1.205
	DPSDA-img (uncond.)	0.556 / 0.571	0.518 / 0.518	0.519 / 0.523	0.494 / 0.494		DPSDA-img (uncond.)	2.067 / 2.115	2.048 / 2.146	2.044 / 2.164	2.088 / 2.170
	DPSDA-txt (uncond.)	1.204 / 1.211	1.173 / 1.180	1.141 / 1.144	0.998 / 1.006		DPSDA-txt (uncond.)	1.378 / 1.409	1.342 / 1.342	1.308 / 1.367	0.259 / 0.259

Table 23: $\text{FID}_{\text{DAE}}\downarrow$ of every PE-based method: best iteration $t \geq 1$ / released S_T ($t = 20$). The other baselines release a single output, so their values are those of Table 14.

Dataset	Method	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = \infty$	Dataset	Method	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = \infty$
Activity	PETS (TimeCraft)	0.040 / 0.098	0.034 / 0.082	0.030 / 0.085	0.034 / 0.077	ECG	PETS (TimeCraft)	0.028 / 0.114	0.068 / 0.068	-0.003 / 0.014	0.024 / 0.024
	PETS (Diffusion-TS)	0.253 / 0.292	0.212 / 0.293	0.142 / 0.279	0.136 / 0.241		PETS (Diffusion-TS)	0.161 / 0.586	0.330 / 0.815	0.323 / 0.577	0.091 / 0.708
	PETS (ChatTS+VAE)	0.321 / 0.454	0.280 / 0.424	0.226 / 0.464	0.141 / 0.414		PETS (ChatTS+VAE)	0.382 / 0.420	0.489 / 0.556	0.299 / 0.322	0.304 / 0.398
	DPSDA-img (uncond.)	0.704 / 0.746	0.653 / 0.687	0.584 / 0.593	0.484 / 0.484		DPSDA-img (uncond.)	0.500 / 0.564	0.192 / 0.647	0.164 / 0.359	0.130 / 0.251
	DPSDA-img (cond.)	0.615 / 0.654	0.563 / 0.564	0.542 / 0.542	0.478 / 0.484		DPSDA-img (cond.)	0.639 / 1.267	0.205 / 0.489	0.208 / 0.532	0.183 / 0.422
	DPSDA-txt (uncond.)	8.522 / 8.642	7.083 / 7.083	5.485 / 5.635	0.302 / 0.302		DPSDA-txt (uncond.)	38.258 / 38.258	31.633 / 32.392	25.224 / 25.224	1.607 / 1.607
	DPSDA-txt (cond.)	8.699 / 8.699	7.823 / 8.010	6.659 / 6.659	2.219 / 2.223		DPSDA-txt (cond.)	39.382 / 39.382	30.268 / 30.268	25.161 / 25.161	0.936 / 0.936
Traffic	PETS (TimeCraft)	0.061 / 0.071	0.058 / 0.067	0.056 / 0.064	0.052 / 0.055	Crypto	PETS (TimeCraft)	0.672 / 1.295	0.304 / 0.551	0.585 / 0.585	0.402 / 0.821
	PETS (Diffusion-TS)	1.942 / 1.942	1.209 / 1.426	1.137 / 1.293	0.563 / 0.756		PETS (Diffusion-TS)	0.186 / 23.287	0.060 / 4.685	0.191 / 0.644	0.049 / 0.167
	PETS (ChatTS+VAE)	0.062 / 0.064	0.058 / 0.064	0.056 / 0.067	0.040 / 0.068		PETS (ChatTS+VAE)	0.433 / 1.209	0.311 / 0.676	0.773 / 3.708	0.709 / 5.399
	DPSDA-img (uncond.)	2.530 / 2.530	2.638 / 2.638	1.777 / 1.777	1.563 / 1.563		DPSDA-img (uncond.)	1.719 / 2.332	1.244 / 2.894	1.472 / 3.287	1.553 / 1.553
	DPSDA-txt (uncond.)	71.219 / 71.850	67.422 / 67.422	62.430 / 62.430	5.324 / 5.324		DPSDA-txt (uncond.)	877.212 / 891.917	876.279 / 883.691	871.845 / 871.845	187.703 / 187.703

Table 24: MMD^2 on raw windows$\downarrow$ of every PE-based method: best iteration $t \geq 1$ / released S_T ($t = 20$). The other baselines release a single output, so their values are those of Table 15.

Dataset	Method	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = \infty$	Dataset	Method	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = \infty$
Activity	PETS (TimeCraft)	0.0013 / 0.0076	0.0013 / 0.0091	0.0013 / 0.0097	0.0017 / 0.0155	ECG	PETS (TimeCraft)	0.0048 / 0.0048	0.0034 / 0.0034	0.0020 / 0.0020	0.0018 / 0.0022
	PETS (Diffusion-TS)	0.0053 / 0.0227	0.0046 / 0.0434	0.0042 / 0.0253	0.0042 / 0.0282		PETS (Diffusion-TS)	0.0358 / 0.0438	0.0202 / 0.0542	0.0200 / 0.0518	0.0139 / 0.0418
	PETS (ChatTS+VAE)	0.0124 / 0.0541	0.0099 / 0.0551	0.0083 / 0.0568	0.0093 / 0.0507		PETS (ChatTS+VAE)	0.0242 / 0.0243	0.0220 / 0.0253	0.0209 / 0.0209	0.0220 / 0.0235
	DPSDA-img (uncond.)	0.0271 / 0.0271	0.0363 / 0.0422	0.0344 / 0.0344	0.0240 / 0.0240		DPSDA-img (uncond.)	0.0194 / 0.0311	0.0091 / 0.0206	0.0084 / 0.0153	0.0082 / 0.0118
	DPSDA-img (cond.)	0.0317 / 0.0317	0.0310 / 0.0310	0.0312 / 0.0313	0.0245 / 0.0260		DPSDA-img (cond.)	0.0219 / 0.0364	0.0092 / 0.0229	0.0084 / 0.0138	0.0084 / 0.0121
	DPSDA-txt (uncond.)	0.2131 / 0.2131	0.1642 / 0.1642	0.1056 / 0.1056	0.0113 / 0.0113		DPSDA-txt (uncond.)	0.2343 / 0.2343	0.1941 / 0.1941	0.1486 / 0.1486	0.0230 / 0.0230
	DPSDA-txt (cond.)	0.2237 / 0.2237	0.1820 / 0.1838	0.1477 / 0.1477	0.0286 / 0.0286		DPSDA-txt (cond.)	0.2410 / 0.2410	0.1803 / 0.1803	0.1494 / 0.1494	0.0191 / 0.0191
Traffic	PETS (TimeCraft)	0.0125 / 0.0274	0.0137 / 0.0534	0.0142 / 0.0382	0.0143 / 0.0239	Crypto	PETS (TimeCraft)	0.0007 / 0.0014	0.0003 / 0.0004	0.0009 / 0.0011	0.0013 / 0.0017
	PETS (Diffusion-TS)	0.0648 / 0.0648	0.0490 / 0.0498	0.0456 / 0.0456	0.0565 / 0.0565		PETS (Diffusion-TS)	0.0248 / 0.0490	0.0227 / 0.0455	0.0218 / 0.0430	0.0197 / 0.0470
	PETS (ChatTS+VAE)	0.0248 / 0.0773	0.0255 / 0.0748	0.0252 / 0.0727	0.0270 / 0.0988		PETS (ChatTS+VAE)	0.0033 / 0.0990	0.0038 / 0.0757	0.0038 / 0.1436	0.0037 / 0.1047
	DPSDA-img (uncond.)	0.0707 / 0.0707	0.0508 / 0.0508	0.0517 / 0.0517	0.0503 / 0.0507		DPSDA-img (uncond.)	0.0045 / 0.0059	0.0054 / 0.0060	0.0062 / 0.0064	0.0041 / 0.0041
	DPSDA-txt (uncond.)	0.5022 / 0.5054	0.4611 / 0.4611	0.3839 / 0.3839	0.0271 / 0.0271		DPSDA-txt (uncond.)	0.4443 / 0.4443	0.4390 / 0.4390	0.4245 / 0.4245	0.1136 / 0.1136