Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

SynForge-Cardio: A Unified Multimodal Synthetic Dataset Generation Framework for Privacy-Preserving Cardiovascular AI

Sreyas Sanil Kumar
Amrita School of Engineering, Amrita Vishwa Vidyapeetham
Kishore N
Amrita School of Engineering, Amrita Vishwa Vidyapeetham
Amirthalakshmi T M
tm_amirthalakshmi@ch.amrita.edu

Amrita School of Engineering, Amrita Vishwa Vidyapeetham
Rahul S G
Amrita School of Engineering, Amrita Vishwa Vidyapeetham

Article

Keywords: Multi-modality orchestration, cardiovascular AI, latent diffusion, 12-lead ECG synthesis, electronic health records, CTGAN, rule-based modality routing, privacy-risk screening, distance to closest record, prompt-driven orchestration

Posted Date: November 13th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11269482/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

SynForge-Cardio: A Unified Multimodal Synthetic Dataset Generation Framework for Privacy-Preserving Cardiovascular AI

Sreyas Sanil Kumar, Kishore N, Amirthalakshmi T M*, and Rahul S G

Department of Electronics and Communication Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Chennai 601103, Tamil Nadu, India

Abstract

Multimodal cardiovascular AI development is constrained by privacy regulation, institutional data silos and class imbalance, and open-source synthetic-data tools are largely modality-specific. We present SynForge-Cardio, an orchestration framework that coordinates established generators for chest X-ray (latent diffusion with optional LoRA adaptation), 12-lead ECG (parametric and bounded morphology-preserving augmentation, with an optional TimeGAN path) and tabular EHR data (size-aware routing among jitter augmentation, a Gaussian copula and CTGAN) behind a shared prompt interface, rule-based router and quality and privacy reporting layer. The modalities are generated independently from separate public datasets (NIH ChestX-ray14, PTB-XL, UCI Cleveland Heart Disease); no patient-level joint synthesis or cross-modality consistency is claimed. The router reached 94.6% modality-routing accuracy on a fixed prompt benchmark, and 1,105 automated tests passed with 3 skipped. In train-on-synthetic, test-on-real experiments with five seeds, augmenting a 25%-real training set changed held-out AUROC by −0.041 to +0.050 across three CXR tasks (none significant), by +0.019 for ECG MI-versus-NORM (from bounded real-waveform augmentation, not an independently learned generator) and by +0.017 for EHR heart-disease classification, which is not evidence of benefit given the 61-record hold-out. Privacy screening is empirical only; no formal privacy guarantee is established.

Keywords: Multi-modality orchestration; cardiovascular AI; latent diffusion; 12-lead ECG synthesis; electronic health records; CTGAN; rule-based modality routing; privacy-risk screening; distance to closest record; prompt-driven orchestration.

*Corresponding author: Amirthalakshmi T M. Email: tm_amirthalakshmi@ch.amrita.edu

1. Introduction

1.1. Clinical Motivation

Cardiovascular diseases remain a leading cause of global mortality [21]. Clinical assessment commonly combines complementary evidence: chest radiographs for structural signs such as cardiomegaly and pleural effusion; 12-lead ECG for electrical abnormalities including arrhythmia and ischemic patterns; and EHR variables for demographics, vitals, laboratory biomarkers, and comorbidities. Machine-learning models that use more than one of these streams are therefore attractive, but they require large, well-characterized multimodal cohorts.

Such cohorts are difficult to assemble. Privacy regulations restrict unrestricted sharing of protected health information; single-site collections are often limited in size and class balance; and cross-institution agreements can be slow. Synthetic data is one complementary strategy: generative models can produce additional samples for augmentation, stress testing, and method development while reducing direct exposure of raw patient records. Synthetic data does not remove the need for real clinical validation, but it can reduce friction during early model development when access to multi-site data is incomplete.

1.2. Tooling Fragmentation Across Modalities

A practical obstacle is that synthetic-data software is fragmented by modality. Tabular generators handle mixed-type EHR tables; diffusion models handle images; time-series methods handle waveforms. Each stack possesses distinct dependencies, configuration surface, and evaluation procedures. For a cardiovascular project that needs CXR, ECG, and EHR together, researchers typically assemble three toolchains by hand. That process is error-prone: preprocessing conventions diverge, validation is performed ad hoc, and there is no shared interface for stating a clinical generation request once and obtaining a consistent artifact package.

This paper focuses on that coordination problem. The underlying generators used in SynForge-Cardio are established methods. The contribution is a cardiovascular-oriented orchestration layer: a common natural-language clinical request interface, rule-based modality routing, size-aware EHR backend selection, independent modality pipelines behind common APIs, a common validation/reporting path, integrated privacy screening, and a reproducible artifact/provenance handling scheme, all in a workstation-scale full-stack implementation. We do not propose a new diffusion architecture, a new generative backbone, state-of-the-art generation, or universally superior synthetic-data quality.

1.3. Scope of SynForge-Cardio

SynForge-Cardio supports three modalities. For images, a latent diffusion pipeline (Stable Diffusion-class), optionally adapted with low-rank adapters (LoRA), synthesizes 512 × 512 chest radiographs conditioned on radiology-style text prompts; pathology conditioning is heterogeneous and, in the final product configuration, the supported CXR scope is restricted to Cardiomegaly and No-Finding (Section 10.2). For signals, the framework either synthesizes 12-lead ECG waveforms using a parametric P–QRS–T model or applies bounded morphology-preserving transforms to uploaded recordings; an optional TimeGAN path is retained for learned temporal modeling when sufficient seed data and compute are available. For tabular data, tables are routed by sample size to jitter augmentation, a Gaussian copula synthesizer, or CTGAN (size thresholds in Section 4.5). The system is exposed through a REST API and a browser interface.

The three modalities are generated independently from separate, modality-specific datasets. There is no shared patient latent, no cross-modal consistency loss, and no claim of anatomical or physiological consistency across a single virtual patient. SynForge-Cardio therefore provides multi-modality at the level of orchestration — one request interface, one router, independent pipelines, one reporting path — rather than patient-level joint synthesis.

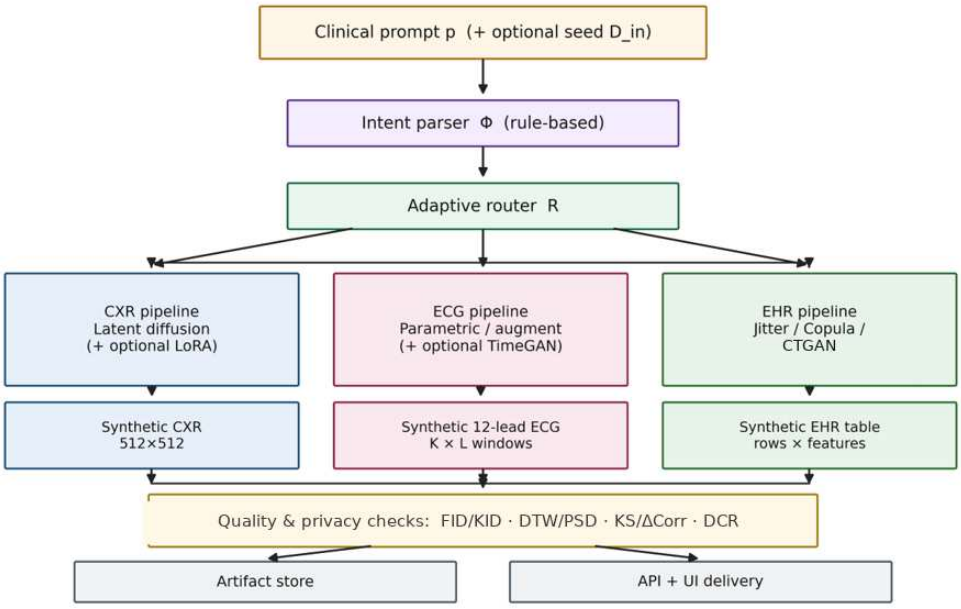


Figure 1: Overview of the SynForge-Cardio framework: a clinical prompt (with optional seed data) is parsed by a rule-based intent parser, routed by the rule-based router to the CXR, ECG, or EHR pipeline, and passed through a shared quality/privacy screening layer before artifact storage and API/UI delivery. The three pipelines are independent; the framework does not link the generated records into a single patient. The validation block reads FID/KID · DTW/PSD · KS/ΔCorr · DCR, consistent with the metric definitions in Section 7.

1.4. Contributions

The main contributions are as follows.

1. A multi-modality orchestration framework that coordinates independent CXR, 12-lead ECG, and EHR synthesis under one prompt-driven workflow for cardiovascular research use cases, without introducing a new generative backbone.

2. A rule-based clinical intent parser and modality router that map natural-language requests and optional file uploads to the appropriate generation pipeline. Routing is manually defined (hand-authored keyword and rule triggers), not learned; on 37 well-defined benchmark prompts it reaches 94.6% routing accuracy with EHR/CXR precision, recall and F1 of 1.0 and ECG F1 ≈ 0.941. A direct-backend-versus-routed ablation on the same prompts shows the routing layer is transparent to the underlying generation contract — 100% output equivalence across 50 valid modality-level comparisons, with real product backends producing byte-identical checksums whether called directly or dispatched through the router — while avoiding 59 unnecessary generator invocations relative to blind multi-backend execution at a mean routing overhead of 0.0025 ms per request; the router unit-test suite passes 31/31 and the ablation suite 18/18 (Sections 10.6–10.6.1).

3. A size-aware tabular routing rule selecting among jitter augmentation, Gaussian copula modeling, and CTGAN, adopted as a practical heuristic based on observed training stability and not claimed to be globally optimal.

4. A CXR synthesis configuration based on latent diffusion with optional LoRA adaptation and memory-saving inference settings suitable for a single workstation GPU.

5. A dual-mode ECG module that combines parametric P-QRS-T synthesis with bounded morphology-preserving augmentation of real recordings.

6. An integrated validation / privacy-screening layer combining standard modality-specific metrics and a distance-to-closest-record (DCR) nearest-neighbour screen for tabular outputs, attached automatically to every generated artifact (100% validation/reporting completeness on the benchmark).

7. A complete-stack reference implementation (FastAPI + React) and a validation strategy using standard datasets, including distributional checks and a four-regime TSTR utility analysis with five random seeds; 1108 automated tests are collected, of which 1105 pass, 3 are skipped and 0 fail in fresh chunked validation.

The rest of the paper describes related work (Section 2), lists research gaps (Section 3), formalizes the problem and generation task (Section 4), details architecture (Section 5) and methodologies (Section 6), defines validation metrics (Section 7), and presents implementation (Section 8) and experiments (Sections 9–11); limitations are presented in Section 12, and the conclusion in Section 13.

2. Related Work

2.1. Medical Image Synthesis

Medical image synthesis has evolved from GAN-based models [1] to latent diffusion models (LDMs) [2, 3]. Initial CNN-GANs tend to suffer from low diversity and unstable training on radiographic data. Cycle-consistent translation models enabled cross-modality mapping but were not natively text-conditional. Latent diffusion methods perform iterative denoising in a compressed latent space produced by an autoencoder, which reduces compute relative to pixel-space diffusion while retaining high visual fidelity.

Domain-adapted diffusion systems for chest radiography have been explored on public corpora such as NIH ChestX-ray14 [10], CheXpert [26], and MIMIC-CXR [27]. ControlNet-style spatial conditioning and parameter-efficient fine-tuning (e.g., LoRA) further improve controllability and adaptation cost [4, 5]. These systems address imaging well, but they do not by themselves provide ECG or EHR generation, routing, or shared privacy reporting. Classical image-augmentation surveys provide complementary context for non-generative transforms used alongside synthesis pipelines [19].

2.2. Synthetic EHR and Tabular Generation

EHR tables mix continuous laboratory values with categorical diagnoses and demographics. The Synthetic Data Vault (SDV) popularized practical tabular synthesizers, including copula-based models, TVAE, and CTGAN [6, 7]. CTGAN uses mode-specific normalization for multimodal continuous features and conditional generation for discrete attributes. Gaussian copula models remain attractive for smaller tables because they avoid adversarial training and estimate dependence after rank-based marginal transforms. Privacy-oriented health-data generators explore additional constraints [14, 15], but most remain tabular-only, with model selection left to the user — which is problematic given that cohort sizes span orders of magnitude across clinical extracts. More general surveys of tabular and latent-space synthetic-data generation are also available [20].

2.3. ECG and Biomedical Signal Synthesis

ECG synthesis needs to preserve both the shape and frequencies of its waves. TimeGAN leverages supervised stepwise losses along with adversarial training in a latent space to preserve temporal dynamics [8]. Parametric dynamical models with Gaussian-shaped P–QRS–T morphology are easily manipulated by changes to the rate and timing of waves, offering an interpretable form of control. For the simple augmentation use case, noise, scaling, time shifts, and minor frequency-domain transformations remain popular as a means to grow a labeled set where there is no need to change the clinical class [9]. As with the imaging and tabular systems, signal algorithms tend to be available as separate modules rather than within larger frameworks.

2.4. Validation and Privacy Screening

Conventions for reporting quality vary across modalities. Image quality is commonly reported as Fréchet Inception Distance (FID) and structural similarity (SSIM). For ECG, dynamic time warping (DTW) and power spectral measures are frequently used. Marginal tests such as Kolmogorov–Smirnov (KS) statistics and dependence preservation via correlation structure are common in the tabular setting. Nearest-neighbor distances to real records (distance to closest record, or DCR) are used as a practical check against verbatim memorization in the synthetic tabular context [14]. These individual metrics are well understood; the

remaining problem is whether and how to run them consistently inside the same generation environment rather than as a disconnected offline process.

A complementary and adjacent line of work targets de-identification of existing clinical records rather than de novo synthesis: fully local multi-agent large-language-model systems have recently been proposed to remove protected health information from multimodal medical data [24]. That approach addresses a different privacy problem (anonymizing real records) and does not by itself provide the generative or orchestration capabilities considered here; SynForge-Cardio's privacy handling remains empirical screening, not a formal guarantee.

2.5. Position

We do not propose SynForge-Cardio as a tool that outperforms every single-modality tool on every benchmark. Its purpose is to consolidate a generation workflow: a single common interface, shared storage and routing, modality-appropriate backends, and one unified report. Existing single-modality tools remain useful complements and may be substituted in as backends behind the same interfaces.

3. Research Gaps

Based on the literature and our own experience assembling multimodal toolchains in practice, we identify the following gaps, which motivate the overall framework design.

G1. No established shared orchestration system across CXR, ECG, and EHR generators, with a common request and artifact interface.

G2. Manual model selection within tabular synthesis — including the choice between deep generative models and simpler alternatives — even though the appropriate choice depends heavily on dataset size.

G3. Weak coupling between generation and validation processes, which makes quality and privacy checks easy to skip or apply inconsistently.

G4. Risk of clinically nonsensical transforms, such as image generation without radiology-relevant prompting, or ECG augmentation that fundamentally alters diagnostic waveform morphology.

G5. Deployment and maintenance overhead for researchers who must manage separate software environments, scripts, and artifact formats per modality.

Table 1: Comparison of representative open-source tooling against framework-level requirements. The table names concrete tools and uses cautious wording where a capability was not directly verified; each entry summarizes common public tooling rather than every variant of the underlying papers.

Tool / class	Modalities	Shared NL request interface	Size-aware tabular routing	Built-in multi-metric validation	Privacy screen	Full-stack app
SDV (copula)	tabular	no	manual (user chooses model)	partial; external reporting commonly used	not built-in	library
CTGAN / TVAE	tabular	no	manual	external	not built-in	library
TimeGAN	signal (time series)	no	n/a	external	not built-in	reference code
Stable Diffusion / latent-diffusion image synthesis	image	prompt text only (no clinical routing)	n/a	external (FID/KID commonly reported separately)	not built-in	varies
SynForge-Cardio	CXR + ECG + EHR (independent)	yes (clinical prompt + rule-based router)	yes (Eq. 17 heuristic)	yes (attached to every artifact)	yes (DCR / nearest-neighbour screen; per-modality verdicts)	yes (FastAPI + React)

4. Problem Formulation

4.1. Data and Generation Map

SynForge-Cardio does not assume or use a patient-linked multimodal corpus. For each modality $M \in \{CXR, ECG, EHR\}$ there is a separate real corpus

$$D_real^M = \{ (x_i^M, y_i^M) \}, i = 1..N_M , \quad drawn\ from\ its\ own\ source\ distribution\ P_M(x^M, y^M), \tag{1}$$

where $x^{CXR} \in R^{(H \times W \times C)}$ is a chest radiograph ($H = W = 512$, $C = 1$ after preprocessing), $x^{ECG} \in R^{(K \times L)}$ is a $K = 12$-lead ECG window ($L = 1000$ for $f_s = 100$ Hz over 10 s), $x^{EHR} \in R^d$ is an encoded EHR feature vector, and y^M is the modality-specific label set. The three corpora are not patient-linked and are never combined into a single tuple.

For a natural-language prompt p, a target number of synthetic samples N_{syn}, an optional seed dataset D_{in}, and the routed modality m, SynForge-Cardio implements a per-modality generation map

$$G_\Theta^m(p, N_syn, D_in) = D_syn^m = \{ (\hat{x}_j^m, \hat{y}_j^m) \}, j = 1..N_syn . \tag{2}$$

A request naming several modalities returns the set $\{ D_{syn}^m \}$ of independently generated per-modality datasets — not linked tuples. Here Θ collects the backend-specific parameters (diffusion weights and LoRA adapters, ECG synthesis parameters, tabular model parameters). A synthetic dataset is operationally accepted when

$$Q(D_syn, D_ref) \geq \tau_q , \quad \Pi(D_syn, D_real) \geq \tau_\pi , \tag{3}$$

where D_{ref} is a reference set for fidelity comparison, Q aggregates quality scores, Π denotes the selected privacy screen (here, a DCR-based nearest-neighbour screen), and τ_q, τ_π are user-configured thresholds. Equation (3) is an operational rule for reporting and filtering; it does not imply the synthetic data is clinically identical to the real data.

4.2. Intent Parsing and Routing

Given a prompt p, an optional seed set D_in, and an optional UI-specified modality m_ui, the system forms an intent object using a rule-based parser Φ and routes the request via router R to a modality m ∈ M = {image, signal, tabular}:

$$I = \Phi(p, D_{in}, m_{ui}) , \quad m = \pi_m(I) , \quad G_m = R(m, I, D_{in}) . \tag{4}$$

The intent parser assigns priority among (i) file type uploaded, (ii) specific UI option selected, and (iii) keyword matches found in p. The routing rules are hand-authored, not learned, and there is no genuine reject/unsupported-modality outcome: a prompt matching none of the keyword triggers falls back to generating all three supported modalities (Section 10.6). Once the pipeline is determined, it generates data and, if a reference is available, returns the validation records:

$$D = G_m(p, D_{in}, I) , \quad Q = V_m(D, D_{ref}) , \quad O = (D, Q, R_{rep}, A) , \tag{5}$$

where V_m is the modality-specific quality evaluation protocol, R_rep a human-readable report, and A the persisted artifacts.

Algorithm 1: SynForge-Cardio request handling

Input: prompt p, optional seed D_in, request count N_syn, optional UI modality
m_ui Output: artifacts A, metrics Q (if reference available)
1: I ← (p, D_in, m_ui) via rule-based intent parser
2: m ← π_m(I) # modality ∈ {image, signal, tabular}
3: if m = tabular then
4: n ← number of rows in D_in
5: backend ← M(n) per Eq. (17)
6: else if m = signal then
7: backend ← Augment (if seed ECG exists) else Parametric (or TimeGAN)

8: else
9: backend ← Latent Diffusion (+ optional LoRA)
10: D ← Generate(backend, p, D_in, N_syn)
11: if reference available then Q ← V_m(D, D_ref) else Q ←
∅12: A ← Persist(D, Q, metadata); return (A, Q)

4.3. Latent Diffusion for Chest Radiographs

Following latent diffusion [3], an encoder E transforms an image into a latent code z_0 = E(I), reconstructed by a decoder Î = D(z). The forward noising process for t = 1, ..., T is

$$q(z_t \mid z_{t-1}) = N(z_t ; \sqrt{(1 - \beta_t)}\, z_{t-1} , \beta_t I) , \tag{6}$$

with variance schedule {β_t} ⊂ (0, 1). Equivalently, with α_t = 1 − β_t and ā_t = ∏ α_s,

$$z_t = \sqrt{\bar{a}_t}\, z_0 + \sqrt{(1 - \bar{a}_t)}\, \varepsilon , \quad \varepsilon \sim N(0, I) . \tag{7}$$

Text conditioning uses a frozen text encoder τ to embed the prompt, c = τ(p). A denoiser network ε_θ(z_t, t, c) is trained with the noise-matching objective

$$L_{diff}(\theta) = E[\| \varepsilon - \varepsilon_\theta(z_t, t, c) \|_2^2] . \tag{8}$$

At inference, sampling begins from z_T ~ N(0, I) (text-to-image) or from a partially noised encoding of a seed image (image-to-image). An N-step numerical solver produces z_0 from z_T; we use a DPM-Solver configuration with a reduced step count for interactive workstation use, then decode Î = D(z_0).

LoRA adaptation, when enabled, introduces low-rank updates ΔW = BA to selected projection matrices W ∈ R^(m×n) in the U-Net, with B ∈ R^(m×r), A ∈ R^(r×n), and rank r ≪ min(m, n) [5]. Only A and B are trained during domain adaptation, which reduces memory relative to full fine-tuning. Half-precision inference, VAE slicing/tiling, and optional CPU offload are implementation-level memory-management settings that do not alter Eqs. (6)–(8).

4.4. ECG: Parametric Synthesis, Bounded Augmentation, and Optional TimeGAN

Parametric synthesis. Individual morphologies of the P, Q, R, S, and T waves, plus a low-variance baseline, are modeled as Gaussian-shaped components:

$$x(t) = \Sigma_{w\in\{P,Q,R,S,T\}} a_w \exp(- (\theta(t) - \theta_w)^2 / (2 b_w^2)) + \eta(t) , \quad (9)$$

where $\theta(t) \in [0, 2\pi)$ is a cardiac phase variable, (a_w, θ_w, b_w) control the amplitude, center, and width of wave w, and $\eta(t)$ is low-variance baseline noise. Heart rate f_hr enters through $d\theta/dt = 2\pi f_{hr}$ (with optional R–R jitter). A 12-lead matrix $S \in R^{(12\times L)}$ is obtained by lead-wise parameters or linear projection of one or more dipole components onto the standard lead system. This model is interpretable and computationally light, offering direct control over rate and wave timing, but it is not a complete electrophysiological simulator and cannot capture more complex dynamic phenomena (e.g., Torsades de pointes morphology).

Bounded morphology-preserving augmentation. Given a real window $u \in R^L$ (or multi-lead window $U \in R^{(K\times L)}$), bounded transforms produce

$$\tilde{u} = (S_\alpha \circ Shift_\delta \circ F_\varphi)(u) , \quad (10)$$

where Shift_δ is a circular shift by a small integer offset δ, S_α multiplies amplitude by α close to 1, and F_φ applies a mild frequency-domain phase perturbation. Default ranges are small (a few samples for δ, and a few to about fifteen percent for amplitude), chosen to reduce – not eliminate – the risk of altering diagnostic morphology. A bounded-augmentation output is a deterministic, seed-parameterized transform of one specific real training window and inherits that window's label; it is not an independent generative-model sample and must not be read as one. Label retention should still be spot-checked for downstream tasks.

Optional TimeGAN. When a learned generator is enabled, TimeGAN-style training [8] uses embedding E, recovery R, generator G, and discriminator D:

$$h_t = E(s_t, h_{t-1}) , \quad \hat{s}_t = R(h_t) , \quad (11)$$

(14)

$$L_R = E[\Sigma_t \| s_t - \hat{s}_t \|_2] , \quad (12)$$

$$L_S = E[\Sigma_t \| h_t - G(h_{t-1}) \|_2] , \quad (13)$$

$$L_U = E[\log D(h_{1:L})] + E[\log(1 - D(\hat{h}_{1:L}))] ,$$

$$\min_{E,G,R} \max_D (\lambda_R L_R + \lambda_S L_S + L_U) . \quad (15)$$

In the reference implementation, parametric synthesis and bounded augmentation are the default interactive paths; TimeGAN is an optional backend when seed corpora and training time allow, since its training is comparatively computationally intensive.

4.5. Tabular EHR Models and Size-Aware Routing

CTGAN models heterogeneous tabular distributions using mode-specific normalization (TVAE, an alternative model in the same library, is not implemented in SynForge-Cardio). For a continuous feature V with a possibly multimodal density, modes are learned via a Gaussian mixture,

$$p(v) = \Sigma_{k=1}^{B} \pi_k N(v ; \mu_k, \sigma_k^2) , \quad (16)$$

and each observation is encoded by its most dominant mode together with a within-mode standardized value; categorical variables are handled via conditional generation. Let $n = |D_{in}|$ be the number of training rows.

We introduce a routing rule based on table size, using thresholds n_1, n_2:

$$M(n) = \text{Jitter if } n < n_1 ; \ \text{Gaussian Copula if } n_1 \le n < n_2 ; \ \text{CTGAN if } n \ge n_2 . \quad (17)$$

with default thresholds that differ between the two table generators in the public repository: the EHR clinical generator (the UCI Heart Disease-backed schema generator, app/generation/ehr/generator.py, used for all reported EHR experiments) uses (n_1, n_2) = (100, 1000), whereas the generic tabular generator used for

other uploaded tables (app/generators/tabular.py) uses (n_1, n_2) = (50, 200). Both are adopted as practical heuristics based on observed training stability and not claimed to be globally optimal values. A preliminary controlled matched-size ablation at the single n = 100 boundary of the EHR clinical generator has since been completed (Section 10.4.1); a comprehensive grid over table size and across multiple datasets has not, and is stated as a limitation (Section 12.2). When n is very small, we simply resample rows with added noise on continuous coordinates,

$$\hat{T} = T + \varepsilon , \quad \varepsilon \sim N(0, \sigma^2 I_{cont}) , \qquad (18)$$

applied to normalized continuous coordinates, with categorical coordinates resampled from empirical frequencies or left unchanged depending on configuration. For moderate n, a Gaussian copula estimates marginal CDFs F_j and a correlation structure on Gaussianized ranks; sampling draws latent Gaussians and applies F_j^(-1). For larger n, CTGAN is fit for a fixed epoch budget following the training objectives of the underlying tabular library.

Algorithm 2: Size-aware tabular synthesis

Input: table T_in with n rows, target rows N_syn, thresholds (n_1, n_2), jitter scale
σOutput: synthetic table T_syn
1: Validate schema; encode categoricals; impute or drop incomplete rows as configured
2: if n < n_1 then
3: T_syn ← sample rows with replacement and apply Eq. (18) on continuous fields 4:
else if n < n_2 then
5: fit Gaussian copula on T_in; T_syn ← sample N_syn rows
6: else
7: fit CTGAN on T_in for a fixed epoch budget; T_syn ← sample N_syn rows 8:
Optional: enforce simple range checks (e.g., DBP < SBP)
9: return T_syn

4.6. Design Principles

SynForge-Cardio aims to facilitate collaborative, workstation-scale use under limited compute. Toward this goal, the system is built around: clear and standardized interaction patterns across all modalities; explicitly defined routing and selection criteria between generative pipelines; the use of modality-standard metrics to promote objective comparison; and local, reproducible deployment. We do not claim that any deployed generator achieves new state-of-the-art results within its modality; the intent is to provide a practical, user-friendly solution for general-purpose medical data synthesis.

5. System Architecture

SynForge-Cardio operates on a client–server model, split between a web application (frontend) and a service layer (backend).

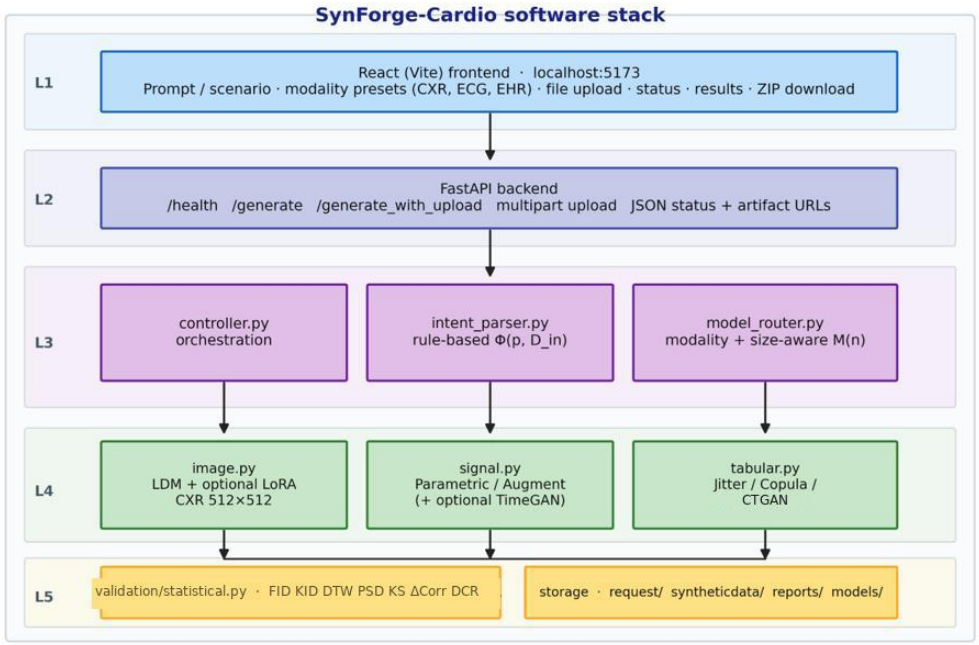


Figure 2: Software stack of SynForge-Cardio: a React (Vite) frontend (L1) communicates with the FastAPI backend (L2), which delegates to orchestration modules – controller, intent parser, and model router (L3) – that dispatch to the independent CXR, ECG, and EHR generators (L4), followed by shared validation and artifact storage (L5). The L5 validation module lists FID KID DTW PSD KS ΔCorr DCR, consistent with Section 7.

5.1. Components

Intent parser. Converts the prompt and any uploaded image, ECG, or table into structured fields – intent, synthesis type, modality-specific keywords, and desired sample count. It is rule-based, allowing for low computational overhead.

Adaptive router. Adopts a modular pipeline construction for all modalities, allowing users to explicitly choose components when needed, while applying hand-authored routing criteria to select which pipeline serves a given generative task – including size-aware routing (Eq. 17) for tabular jobs.

Generator services. Independent modules wrap the CXR diffusion pipeline, ECG synthesizers, and tabular backends. Isolation avoids loading heavy image models for table-only jobs and keeps the modalities decoupled.

Validation and reporting. After generation, available reference data trigger metric computation (Section 7); reports are stored alongside artifacts.

Storage. The inputs to the request, the generated synthetic data, and validation summaries are saved to a default directory tree (input data, synthetic data, validation reports, potential model weights), each keyed by a per-run identifier for provenance.

5.2. Control Flow and API

The request lifecycle:

$$(p, D_in) \rightarrow \Phi \rightarrow R \rightarrow G_m \rightarrow V_m \rightarrow (D, Q, A)\ . \qquad (19)$$

The HTTP surface provides standard health-check requests, prompt-only generation requests, and with-upload generation requests. Request failures (unsupported file type, empty table, invalid record count) respond with structured error formats. Jobs successfully completed respond with downloadable artifacts and a small summary in the UI.

6. Methodology

Each data modality applies the same external procedure – consume, dispatch, generate, and if applicable, validate – but can contain different generators.

6.1. Chest X-ray Pipeline

Radiographs are rescaled to 512 × 512 and normalized by intensity, and the prompts are modified with radiology-specific terms and negative prompts, then embedded via a CLIP prior. A fully trained LoRA adapter (rank r = 16, scale α = 16, applied to the U-Net's attention projection layers) on the public CXR dataset can be used for this step. Unless otherwise indicated, all generations are performed at a 7.5 CFG scale, using a 15-step DPM-Solver multistep solver with fp16 data type.

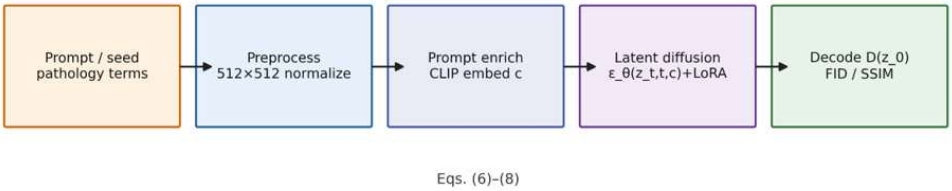


Figure 3: Generation pipeline for CXRs. The input prompt/seed is consumed, the radiograph image (rescaled to 512 × 512) is embedded into CLIP features, a latent diffusion model performs the generation with optional LoRA guidance, and the result is decoded. The generation is then scored with FID (SSIM is not applicable to text-conditioned generation and is reported only for image-to-image diagnostics; see Section 7.1).

For text-to-image generation the pipeline operates directly; when a seed image is supplied, it can instead run as image-to-image translation with limited deviation from the source layout. Because structural fidelity depends heavily on the backbone and conditioning, the automated metrics in Section 7 are required in addition to visual plausibility.

6.2. Twelve-Lead ECG Pipeline

Mode A (Parametric generation). Generates K-lead time series of length L via Eq. (9), allowing users to adjust heart rate and basic rhythm from the prompt or UI. Mode A is conditioned only on heart rate and has no disease/pathology parameter. Mode B (Augmented generation). The user provides ECG samples, which are segmented into windows of fixed length; transform Eq. (10) is then applied within configured bounds to produce a larger set of samples, each preserving the original clinical label. Mode B outputs are bounded perturbations of specific real training windows, not independently generated waveforms. Mode C (Optional learned time-series). This mode applies a pre-trained TimeGAN, whose sampled latent trajectories are then converted to waveforms by R. Being computationally expensive to train, Mode C is an optional generator for this pipeline.

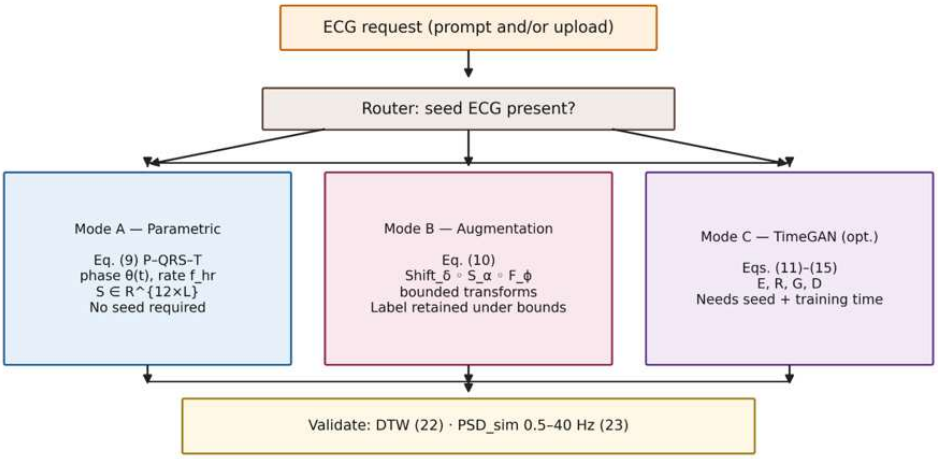


Figure 4: ECG synthesis and augmentation pathways: parametric synthesis (Mode A), bounded morphology-preserving augmentation (Mode B), and optional TimeGAN (Mode C), followed by DTW and PSD-similarity validation.

6.3. EHR Tabular Pipeline

Reference feature groups include demographics (age, sex), vitals (BMI, blood pressure, heart rate), representative laboratory markers (troponin, glucose), binary or categorical health history (hypertension, diabetes, smoking), and a diagnosis label. After type inference and minor cleaning, Algorithm 2 selects and runs the appropriate synthesizer. Optional deterministic checks can be configured to exclude clinically impossible numeric relations (e.g., diastolic blood pressure exceeding systolic).

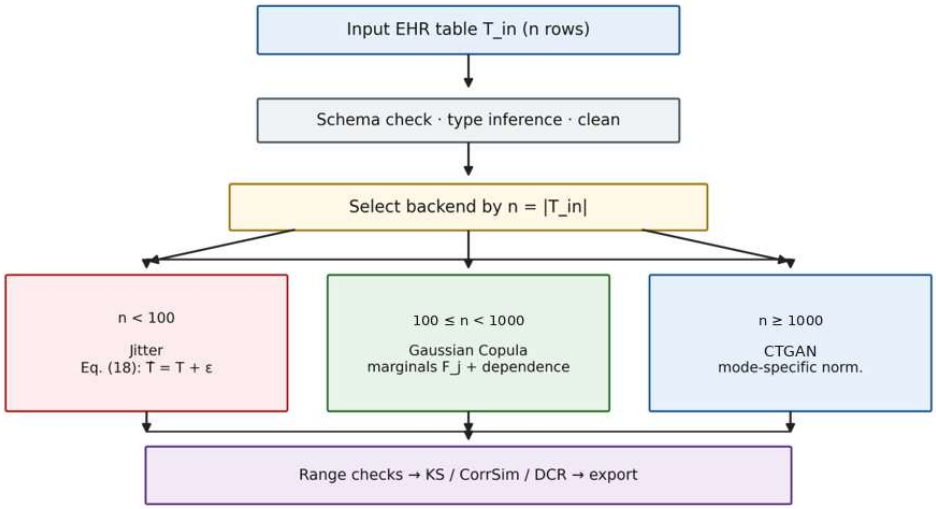


Figure 5: Size-aware tabular routing: input table size n determines whether jitter (Eq. 18), Gaussian copula, or CTGAN is used, followed by range checks and KS / correlation / DCR validation. Thresholds shown are those of the EHR clinical generator; the generic tabular generator uses $n_1 = 50$ and $n_2 = 200$.

Matched-size ablation protocol. To probe the n = 100 routing boundary empirically, a preliminary matched-size generator comparison was run (results in Section 10.4.1). A single 100-row training subset was drawn once from the UCI Cleveland Heart Disease [13] training split with seed 42 and used identically by every generator; the untouched 61-row hold-out was used only for downstream evaluation and never for fitting, tuning, or generation, and the frozen UCI train / hold-out split files were read-only. Each of the three size-regime backends — jitter, Gaussian copula, and CTGAN — was forced to run on that subset by temporarily narrowing the Eq. (17) thresholds (no new generator code), with three independent generation seeds (42, 43, 44) and 200 synthetic records per arm (identical for every arm). Evaluation reused the paper's own utilities: the random-forest downstream pipeline with hold-out AUROC,

the raw Frobenius correlation-matrix difference (ΔCorr) against a real-vs-real resampling reference, the deterministic constraint checker, and

generation wall-time. CTGAN is the deep-generative tabular backend wired into the pipeline; TVAE is not, and no TVAE substitution was made.

6.4. Complexity Considerations

Diffusion inference time scales with the number of denoising steps and latent spatial size. CTGAN training time scales with epochs, table rows, and feature dimensionality; copula fitting is dominated by marginal estimation and dependence estimation in transformed space. Parametric ECG synthesis runtime is linear in $K \cdot L$. The dominant factors of workstation wall-clock time are whether the image stack must be loaded and whether extensive tabular model training is required – one motivation for the router is to avoid unnecessary heavy model fits on small tables.

7. Quality and Privacy Metrics

Validation metrics are computed whenever reference data are available. These metrics are standard within their respective communities; SynForge-Cardio's role is to compute and attach them systematically to every generated artifact.

7.1. Image Metrics

FID evaluates the closeness of Gaussian fits to deep-feature representations of real and synthetic images, using feature means (μ_r, μ_s) and covariances (Σ_r, Σ_s):

$$FID(r, s) = \| \mu_r - \mu_s \|_2^2 + Tr(\Sigma_r + \Sigma_s - 2(\Sigma_r \Sigma_s)^{1/2}) . \quad (20)$$

FID absolute values depend on the exact feature extractor and preprocessing and must be interpreted only under a fixed evaluation configuration. The current evaluation script uses the canonical pytorch-fid Inception-v3 checkpoint (pt_inception-2015-12-05; Heusel et al. [22]), the 2048-dimensional pool3-equivalent feature block (output_blocks = [3]), single-channel radiographs replicated to three channels, internal bilinear resize to 299 × 299, the extractor's own (2x − 1) input normalization, empirical covariance via np.cov, and scipy.linalg.sqrtm with an imaginary-tolerance check. FID is computed between 1,038 held-out real images and 400–500 synthetic images (fixed seed 42); the real count is below 2,000, so FID here is a biased small-sample estimator and is reported as such. KID (unbiased polynomial-kernel MMD^2 [23]) is logged alongside FID.

SSIM measures similarity between images x and y based on local luminance, contrast, and structure:

$$SSIM(x, y) = [(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)] / [(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)] . \quad (21)$$

Higher SSIM and lower FID are preferred. SSIM requires a meaningful 1:1 pairing between a real and a synthetic image. Such a pairing exists only for image-to-image (same-seed / same-source) diagnostics, where compute_ssim_paired() is applied to explicit (real_seed_image, synthetic_output) pairs carrying seed-provenance metadata. For text-conditioned (text-to-image) CXR generation — the primary product path — no valid pairing exists, so SSIM is reported as NOT_APPLICABLE and no blind pairing is performed. Ordinary SSIM is therefore not used as the primary quality metric for unpaired text-to-image CXR generation in this work.

7.2. ECG Metrics

DTW aligns sequences u and v under a warping path $W = \{(i_k, j_k)\}$:

$$DTW(u, v) = \min_W \Sigma_k \| u_{i_k} - v_{j_k} \|_2 . \quad (22)$$

PSD cosine similarity compares spectral vectors P_u and P_v (e.g., Welch estimates over 0.5–40 Hz):

$$PSD_sim(u, v) = (P_u^T P_v) / (\| P_u \|_2 \| P_v \|_2) . \quad (23)$$

Higher PSD_sim and lower average DTW indicate closer spectral and temporal agreement under the selected alignment.

7.3. Tabular Metrics

For a scalar feature, the two-sample KS statistic between empirical CDFs F_n and G_m is

$$D_{n,m} = sup_x \mid F_n(x) - G_m(x) \mid . \qquad (24)$$

Dependence preservation compares Pearson correlation matrices R_real and R_syn on continuous columns. The evaluation reports the raw Frobenius norm of the correlation-matrix difference over numeric columns,

$$\Delta Corr = \| R_{real} - R_{syn} \|_F , \qquad (25)$$

and interprets it against a real-vs-real reference $\| R_{real} - R_{real}' \|_F$ obtained by resampling the real data at the same sample size (the sampling-noise floor). Lower ΔCorr is better, and a value close to the reference indicates dependence is preserved to within sampling noise. Secondary per-regime KL divergence on discretized marginals may also be logged. Mean KS across continuous columns is the primary marginal summary reported in our tables.

7.4. Tabular Privacy Screen: DCR

The distance from a synthetic point $\hat{t}$ to the real set is the Euclidean nearest-neighbour distance

$$DCR(\hat{t}) = min_{t \in D_{real}} \| \hat{t} - t \|_2 , \qquad (26)$$

computed after min–max scaling of each numeric column to [0, 1] using the real-data range (the SynForge evaluation additionally reports a standardized-feature variant). We summarize the empirical distribution of DCR($\hat{t}$) by quantiles – with particular attention to a lower-tail quantile such as the 5th percentile, denoted DCR_0.05 – and by the exact-match rate at numerical tolerance zero. A small lower-tail DCR may indicate data duplication or near-duplication and warrants closer inspection. DCR is an empirical nearest-neighbour privacy-risk screen, not a formal differential-privacy certificate, and does not by itself guarantee regulatory compliance.

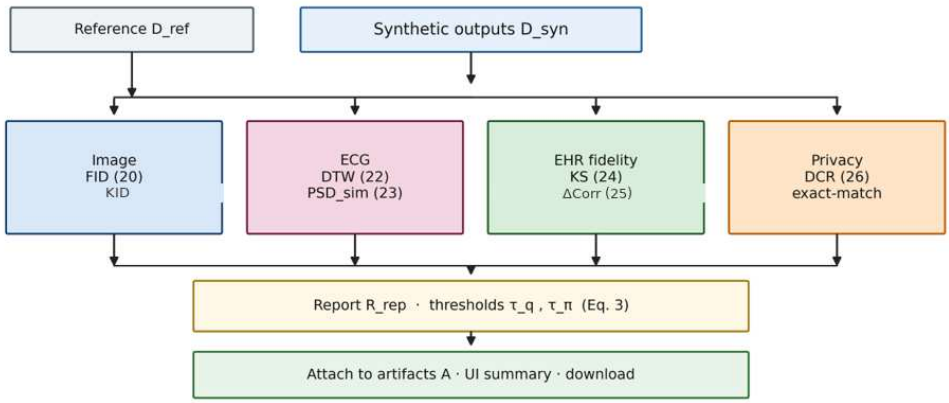


Figure 6: Unified validation and reporting flow: reference and synthetic outputs are scored per modality (image FID (Eq. 20) and KID; ECG DTW / PSD (Eqs. 22–23); EHR KS (Eq. 24) and ΔCorr (Eq. 25); privacy DCR (Eq. 26) / exact-match) and combined into a report attached to the generated artifacts. SSIM (Eq. 21) is applied only to paired image-to-image diagnostics and is not shown as a primary image-quality metric (Section 7.1).

8. Implementation

The proposed system architecture has an API orchestration module isolated from the generation backend. Within the API we have the controller, intent parser, model router, modality generators (image, signal, tabular), and statistics/validation modules. The UI is a React SPA served by Vite and communicates with the API. All user-adjustable parameters, including standard medical cardiology settings, are available in the UI, along with progressive result display in dedicated panels (image, waveform, table, and statistics panel).

The dependencies are the standard Python scientific libraries; one or more deep-learning frameworks required for the various diffusion-based generators; a library for tabular synthetic-data generation, with support for CTGAN, TVAE, copula, and more; and a FastAPI framework that delivers the HTTP API. A standard React application served with Vite is used as the UI. We provide simple scripts to run a single instance locally, but it is not intended for large-scale, distributed deployment.

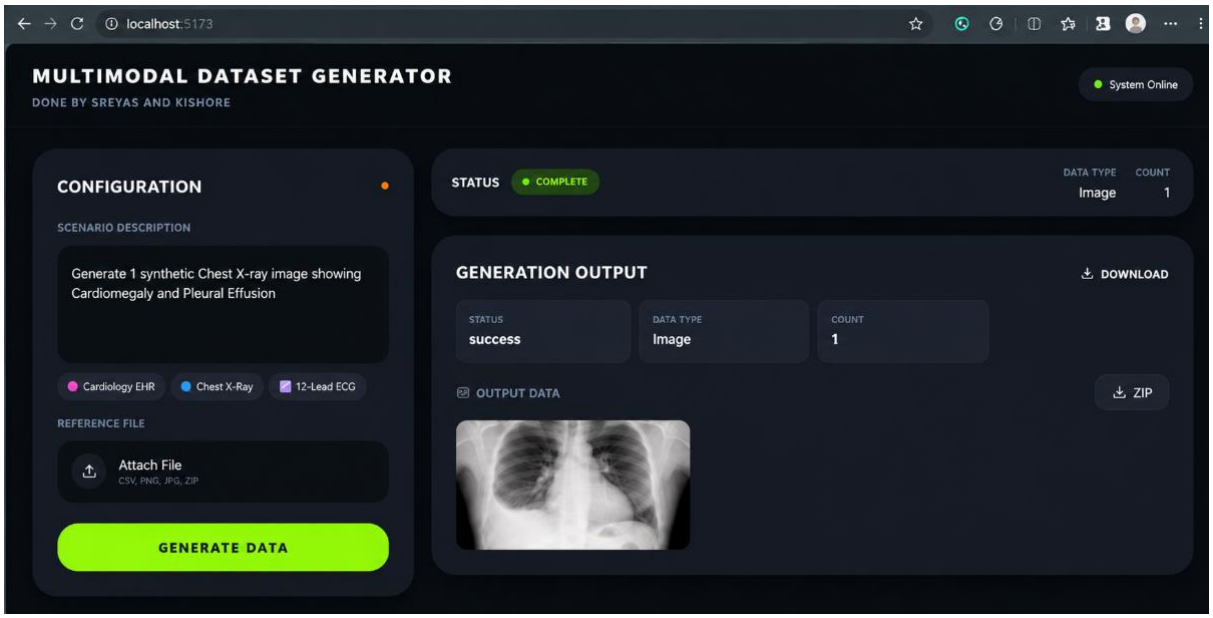


Figure 7: SynForge-Cardio web interface (React, localhost:5173), current build, General Data Studio tab, for a supported CXR request ("Generate 1 synthetic Chest X-ray image showing Cardiomegaly"). The configuration panel holds the clinical scenario prompt, modality presets, and an optional reference upload; the output panel shows generation status, run metadata (model, seed, synthetic ID, generator checkpoint), the decoded 512×512 image, a download control, and the generator's own validity-limitation note. The displayed image is the unedited current generator output and reflects the documented partial CXR fidelity (Section 10.2).

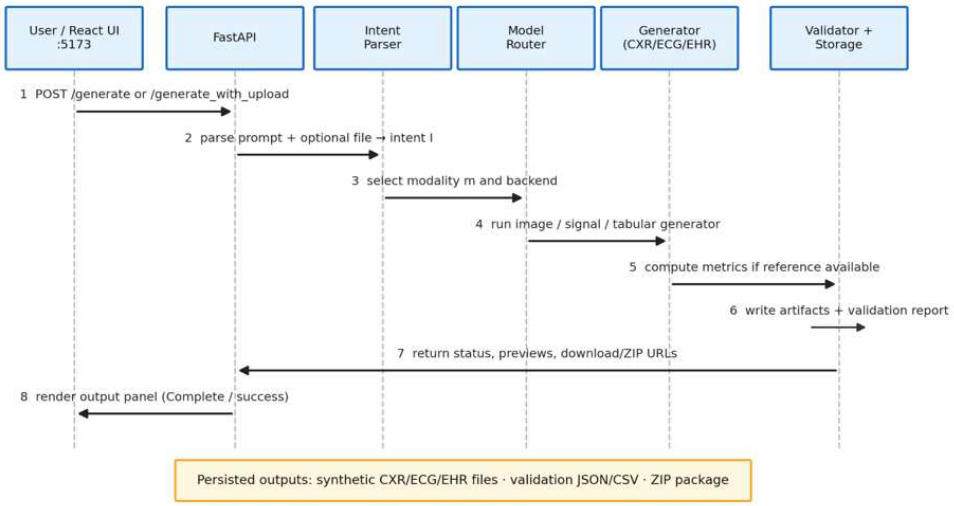


Figure 8: Control flow within the request-to-artifact system. The UI request reaches FastAPI, which is routed and processed by the intent parser and then by the model router to the proper generation backend. If applicable, validation is performed, artifacts are stored, and the corresponding records are returned to the UI for download.

9. Experimental Setup

9.1. Hardware and Software Environment

Experiments were conducted on a single workstation with an NVIDIA GeForce RTX 4060 laptop GPU (8 GB VRAM), 32 GB system RAM, and a modern multi-core CPU, run natively on Windows/Linux or via WSL where applicable. The software stack consists of common Python scientific libraries, a PyTorch-based deep-learning stack for image inference, and the SDV family of libraries for tabular data. The focus of these experiments is feasibility on modest hardware, not maximum throughput.

9.2. Datasets

CXR — NIH ChestX-ray14 [10], provided by the NIH Clinical Center (original download page: https://nihcc.app.box.com/v/ChestXray-NIHCC). The working corpus is the first NIH ChestX-ray14 image archive (images_001), obtained through the public Hugging Face repository alkzar90/NIH-Chest-X-ray-dataset together with NIH's Data_Entry_2017_v2020.csv metadata; all 4,999 images from 1,335 unique patients passed integrity checks and none was excluded. The images were converted to 512 × 512 single-channel format. The patient-disjoint split is 3,961 images / 1,068 patients for training and 1,038 images / 267 patients for hold-out, with zero patient overlap verified. Labels were used for pathology-conditioned prompting and downstream classification. The corpus is markedly class-imbalanced (No-Finding dominates; Cardiomegaly, Effusion and Edema are minority findings), which is relevant to the downstream results in Section 11.

ECG — PTB-XL [11], version 1.0.3 [25] (CC BY 4.0). 12-lead recordings sampled at 100 Hz (records100; the 500 Hz records500 waveforms were not used). The validated downstream task is MI vs NORM (myocardial infarction vs normal ECG), restricted to diagnostically pure records; PTB-XL has no "arrhythmia" superclass. The ECG split is the PTB-XL patient-level 70/15/15 partition (train / validation / test, seed 42) with zero patient overlap; the held-out evaluation set is the 15% test partition and the 15% validation partition is not used (fixed-epoch training, no tuning). On the diagnostically pure MI-vs-NORM subset this gives train 6,345 NORM / 1,774 MI and held-out test 1,336 NORM / 378 MI.

EHR — UCI Cleveland Heart Disease [13] (CC BY 4.0). The Cleveland file (processed.cleveland.data) contains 303 records; all 303 were retained and none was excluded. Six cells are missing in the source (four in the number of major vessels, two in thalassemia) and were left missing in the split files; the downstream random-forest classifier imputes numeric features by the median and categorical features by the most frequent value. The split is 80/20 (stratified by heart-disease status, seed 42): n_train = 242 and n_holdout = 61. MIMIC-IV [12] acquisition was not completed because credentialed access was unavailable; consequently, all reported EHR experimental results in this study are UCI-based, and no MIMIC table was constructed or used.

All datasets used are publicly described research corpora and should be handled under their respective licenses and institutional policies. Synthetic data generated from them in this study were used solely for methodological evaluation.

9.3. Protocol

For each modality we report distributional metrics (Section 7) and downstream performance. The downstream (TSTR) analysis uses four training regimes, all evaluated on held-out real data:

A — Real 25%: a stratified 25% subset of the real training split (low-real baseline);

B — Synthetic-only: synthetic samples only (train-on-synthetic / test-on-real);

C — Real 25% + Synthetic: the low-real subset augmented with synthetic samples;

D — Real 100%: the full real training split (upper reference).

Classifiers are standard, reproducible baselines: a DenseNet-121 multi-label network for CXR tasks, a from-scratch ResNet-1D for ECG, and a tree-ensemble model (random forest) for tabular prediction [16, 17, 18]. Hyperparameters are fixed across compared training sets within a task to isolate data effects; we report AUROC. The CXR and EHR tasks are split at the patient level using an 80/20 train/hold-out

boundary; the ECG task uses the PTB-XL 70/15/15 patient-level split (train / validation / test), with the 15% test partition as the held-out evaluation set and the 15% validation partition unused (fixed-epoch training, no tuning). All splits are defined with seed 42. The major downstream TSTR analyses were run with five classifier-training seeds (42–46) and are reported as mean ± standard deviation; per-seed values are provided in the public repository artifacts. The split definition and the single-snapshot distributional metrics use seed 42 only. The CTGAN fits in the reported matched-size ablation (Table 3d) use 100 epochs (SDV default batch size); TVAE is not run in the reported experiments.

9.4. Reproducibility Configuration Summary

Table 2 reports the major settings used in our workstation runs. These reflect the defaults of the reference implementation rather than global parameter tuning for state-of-the-art performance.

Table 2: Key configuration settings for the experiments conducted on the workstation.

Item	Value / configuration used
Hardware	RTX 4060 laptop GPU (8 GB VRAM); 32 GB system RAM; multi-core CPU; Windows/Linux/WSL
CXR resolution	512 × 512, single-channel radiometric representation
LoRA	rank r = 16, α = 16, applied to U-Net attention projection layers
Diffusion inference	DPM-Solver, 15 steps; guidance scale 7.5; fp16
FID evaluation	1,038 real hold-out images / 400–500 synthetic; canonical pytorch-fid Inception-v3 (pt_inception-2015-12-05); SSIM NOT_APPLICABLE for text-to-image
ECG windows	12-lead, 100 Hz sampling frequency; task MI vs NORM (PTB-XL pure records)
Tabular routing thresholds	EHR: n_1 = 100, n_2 = 1000; generic: 50, 200 (Eq. 17); heuristics
CTGAN budget	100 epochs (separate CTGAN ablation, Table 3d); main EHR: Gaussian copula
EHR data	UCI Cleveland Heart Disease: 303 records (none excluded); 242 train / 61 hold-out. MIMIC-IV blocked (credentialed) — not used
Train/hold-out split	CXR and EHR: 80/20 patient-level (seed 42). ECG: PTB-XL 70/15/15 patient-level (train / validation / test); 15% test = held-out set, 15% validation unused. Low-real regime: stratified 25% sample (seed 42)
Downstream models	DenseNet-121 (CXR); ResNet-1D (ECG); random forest (EHR)
Reported statistics	Five-seed mean ± SD (seeds 42–46) for downstream TSTR; single seed (42) for split definition and distributional snapshots

10. Results

This section reports outcomes obtained on the workstation configuration described above. Values should be interpreted relative to the stated preprocessing, splits, and backends, rather than as universally valid constants of the method class.

10.1. Cross-Modality Summary

Table 3: Summary of distributional and privacy-screening metrics across modalities. Values are specific to the stated evaluation configuration.

Modality	Primary metric(s)	Observed value (this study)	Notes
CXR	FID ↓	FID ≈ 212.0 (F4 checkpoint, reproduced); Stage-A domain adaptation 240.96; 500-image Cardiomegaly product run ≈ 250	Canonical pytorch-fid Inception-v3; 1,038 real / 400–500 synthetic. SSIM not applicable (text-to-image). Generative status: partial (Section 10.2)

ECG	condition fidelity; spectral agreement	Independent-generator classifier AUROC on generated samples ≈ 0.9985; spectral/temporal agreement assessed qualitatively (Figure 10)	PSD/DTW point values are not reported
EHR	ΔCorr ↓; constraint validity ↑	Correlation-matrix Frobenius difference 0.616 (real-vs-real reference 0.531); constraint satisfaction 100%	Numeric columns; GaussianCopula backend, n = 242 real training rows
EHR regimes	backend execution	Jitter, Gaussian copula and CTGAN each were each forced and executed successfully at n = 100 (jitter and CTGAN out-of-	Preliminary single-boundary ablation only (n = 100, 3 seeds, UCI
		matched-size comparison at n = 100 is reported in Section 10.4.1	Cleveland); a full n-grid over table size was not validated (Section 12.2)
Privacy (EHR)	NN disclosure; MIA; attribute inference	NN distance median 2.16 (5th pct 1.50); exact-match rate 0%; membership-inference AUC 0.581; attribute-inference accuracy 0.656 vs 0.541 baseline	Empirical screening; attribute inference is a concern (Section 10.5)

10.2. Image Synthesis

Under the fixed configuration of Section 7.1, the reproducible FID of the frozen F4 checkpoint is approximately 212.0; the Stage-A domain-adaptation checkpoint (no pathology conditioning) scores 240.96, and the 500-image Cardiomegaly product run scores about 250. These are within-pipeline comparisons (raw Stable Diffusion 1.5 → Stage-A LoRA → F4); no external published diffusion baseline (e.g., a RoentGen-class model) was run, and FID at this sample size is a biased estimator. We make no claim of state-of-the-art generation.

Pathology-conditioning fidelity was heterogeneous. On real hold-out data the independent evaluator discriminates Cardiomegaly at AUROC ≈ 0.90, Effusion ≈ 0.76 and Edema ≈ 0.77 — these establish the evaluator's real-data discrimination and are not a measure of synthetic-image quality. On generated images, the classifier response to the intended condition is AUROC ≈ 0.67–0.72 for Cardiomegaly — a reproducible positive signal across the conditioning diagnostics (Mann–Whitney U, one-sided, $p < 0.02$, $n = 25$ per arm) that also holds on the final 500-image product run (AUROC 0.63, 95% bootstrap CI [0.60, 0.67]; Mann–Whitney U, one-sided, $p \approx 1.8 \times 10^{-13}$) — but only ≈ 0.38–0.44 for Effusion and Edema (weak or inverted signal). The inversion persisted across architecturally unrelated conditioning mechanisms and was not explained by label mapping, sampler bugs, classifier breakage, or simple Effusion data scarcity. Cardiomegaly shows measurable positive condition signal; Effusion and Edema exhibit weak or inverted classifier responses, indicating limitations in pathology-specific semantic control. The final product scope is therefore frozen to Cardiomegaly and No-Finding; Effusion and Edema are investigated-but-unsupported.

Validity / decoder diagnostics. A residual colour/tint contamination in a minority of decoded images was investigated with four controlled experiments: (i) the appended 78th pathology-token is not the source of the validity floor; (ii) raw Stable Diffusion 1.5 already contains the residual colour/tint floor before any Stage-A adaptation; (iii) fp16 versus fp32 VAE decoding of the same 50 latents has no meaningful effect (same-latent SSIM ≈ 0.9998); (iv) swapping the SD1.5 VAE for sd-vae-ft-mse gives a small colourfulness reduction but no validity improvement, with the same 2/50 catastrophic seeds and 100% agreement on the affected seeds. The residual contamination was therefore traced to the diffusion latent under the evaluated SD1.5 + DPM-Solver configuration rather than to VAE precision or decoder weights. This diagnosis does not make the CXR generator clinically valid.

Optional post-hoc luminance projection. The product includes an optional inference-time BT.601 grayscale projection (default off). On the 50-latent diagnostic panel it moves the grayscale rate from 80% to 100%, the tint rate from 16% to 0%, catastrophic colour cases from 2/50 to 0/50, and mean colourfulness from 16.4

to ≈ 0, while leaving luma structure unchanged. The known catastrophic images remain structurally defective after projection. Post-hoc luminance projection provides product-level grayscale normalization but does not repair underlying generative defects or pathology fidelity, and is not a generative improvement.

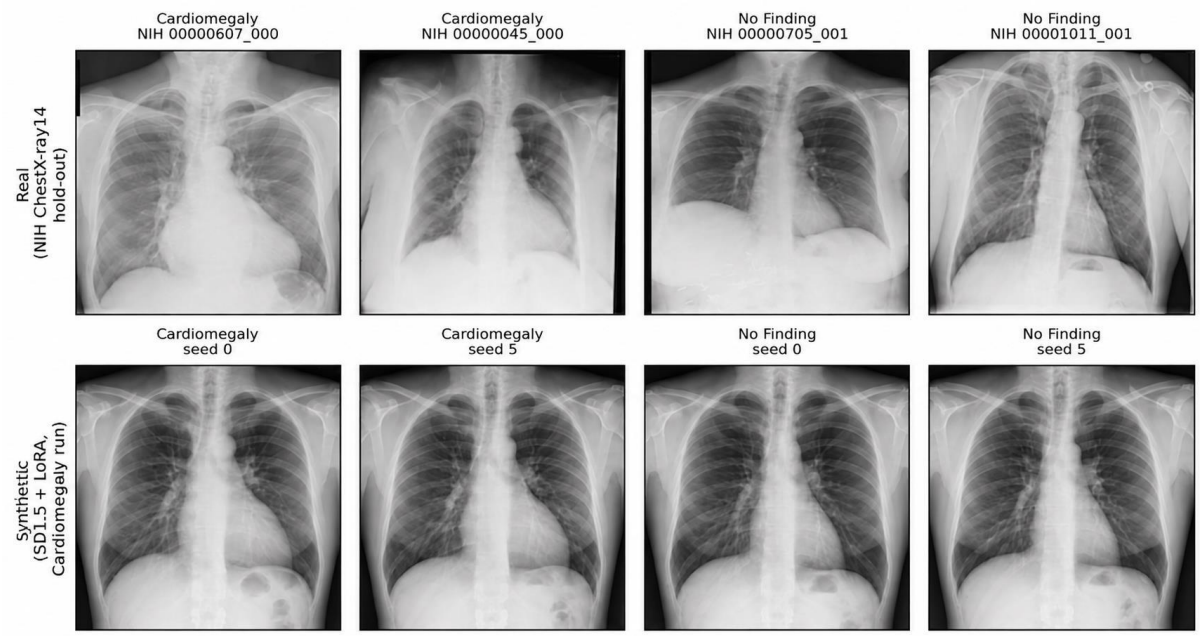


Figure 9: Representative real and synthetic chest radiographs for the supported CXR scope (Cardiomegaly, No-Finding). Top row: genuine, unaltered NIH ChestX-ray14 hold-out images, identified by dataset image index (00000607_000, 00000045_000: Cardiomegaly; 00000705_001, 00001011_001: No-Finding). Bottom row: unedited outputs of the Stable Diffusion 1.5 + LoRA Cardiomegaly/No-Finding product runs using seeds 0 and 5. Real radiographs: NIH Chest X-ray dataset (ChestX-ray14), provided by the NIH Clinical Center (original download page: https://nihcc.app.box.com/v/ChestXray-NIHCC); dataset paper [10]. The synthetic panels are shown without cherry-picking and illustrate the documented partial generative fidelity: gross thoracic layout is present but fine anatomical structure is not reliably reproduced. No synthetic image is presented as real.

10.3. ECG Synthesis and Augmentation

The downstream ECG task is MI vs NORM (Section 9.2); PTB-XL has no arrhythmia superclass, so no arrhythmia-based ECG task is evaluated. The synthetic pool used for the ECG TSTR analysis (Section 11) is produced by Mode B — bounded real-waveform augmentation — not by an independently learned generative ECG model; Mode A cannot be class-conditioned, and Mode C (TimeGAN) was not used for this analysis. The paper therefore does not claim that synthetic generative ECGs reproduce MI morphology. Over five seeds the Mode-B augmentation raises held-out MI-vs-NORM AUROC from 0.9047 ± 0.0130 (regime A) to 0.9236 ± 0.0054 (regime C), Δ = +0.0190, with regime D (Real 100%) at 0.9478 ± 0.0057 (Table 4).

A separate, independent conditional ECG generator was also characterized. Against an independent real-data classifier its condition-fidelity AUROC on generated samples is ≈ 0.9985, with mean P(MI) ≈ 0.02 for NORM-conditioned and ≈ 0.76 for MI-conditioned output; the exact duplicate rate is 0%. However, the MI near-duplicate rate to real training data is ≈ 17%, Mode-B source recovery is ≈ 42.7–43.4% (Section 10.5), the MI mean beat-template correlation to real MI signals is −0.27, and MI morphology is applied uniformly across all 12 leads and lacks infarct-territory localization. These are disclosed limitations of the ECG pipeline.

Interpreting condition-fidelity AUROC alongside morphological correlation. The independent-generator condition-fidelity AUROC of ≈ 0.9985 measures how often an external classifier's NORM/MI decision agrees with the requested conditioning label on generated samples; it is a measure of classifier agreement with the intended label, not a measure of morphological fidelity to real MI waveforms. The mean beat-template correlation of −0.27 between MI-conditioned generated waveforms and the real MI beat-template reference is negative, indicating that generated MI beats do not track the shape of genuine MI morphology;

agreement and morphological correlation are therefore measuring different things and need not move together. Taken together, near-perfect classifier agreement (AUROC ≈ 0.9985) alongside negative morphological correlation (−0.27) raises the possibility that the fidelity classifier is partly exploiting synthetic-specific artifacts or distributional shortcuts particular to this generator, rather than relying solely on clinically meaningful MI morphology, to reach its decision. We report this as a plausible interpretation

consistent with the two measurements, not as a demonstrated causal finding: no dedicated shortcut-attribution study (e.g., saliency analysis, or a classifier trained explicitly to separate generator artifacts from morphology) was run to establish it directly. Separately, Mode-B source recovery of ≈ 42.7–43.4% (Section 10.5) — the rate at which a generated Mode-B sample can be matched back to the one real recording it was derived from — indicates that Mode-B outputs remain relatively close to their specific real source record. Mode B should therefore continue to be read as bounded perturbation / augmentation of real waveforms, consistent with its description elsewhere in this paper (Sections 4.4, 6.2), and not characterized as fully independent ECG generation.

Parametric and bounded-augmentation outputs preserved spectral composition and gross morphology in qualitative comparison (Figure 10). These observations do not provide complete information about diagnostic equivalence across every electrophysiological subclass, but they indicate first-order fidelity useful for augmentation studies.

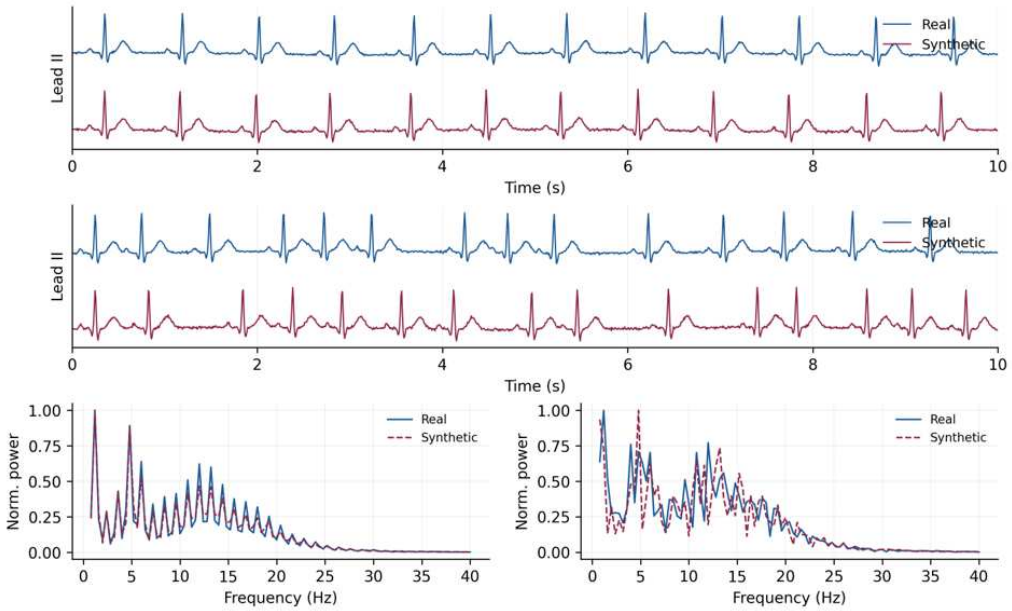


Figure 10: Example real (blue) and synthetic (red, dashed) Lead-II ECG waveforms over a 10-second window (top two panels), with corresponding power spectral density comparisons over 0.5–40 Hz (bottom panels).

10.4. EHR Synthesis

All EHR results are on the UCI Cleveland Heart Disease dataset [13] (242 train / 61 hold-out); MIMIC-IV was not acquired and is not reported. Routing based on size avoids unstable deep fitting of small tables and allows deeper fitting on larger records; the three backends each execute at their respective regime boundary. Dependence fidelity: the correlation-matrix Frobenius difference between real and synthetic numeric features is 0.616, against a real-vs-real resampling reference of 0.531; 100% of synthetic rows satisfy all applicable range/consistency constraints. Invalid-row checking eliminates simple inconsistencies, but not all clinical constraints are encoded.

10.4.1. Matched-size generator ablation at the n = 100 routing boundary

This preliminary ablation (protocol in Section 6.3) probes a single point — the n = 100 boundary between the jitter and Gaussian-copula regimes of Eq. (17) (EHR clinical generator thresholds $n_1 = 100$, $n_2 = 1000$) — on the UCI Cleveland dataset only, with three generation seeds. All nine runs (three generators × three seeds) completed successfully; no TVAE substitution was required. The 100-row training subset (seed 42)

and the 61-row hold-out were identical for every arm, and every arm produced 200 synthetic rows. Because the router itself selects the Gaussian copula at n = 100 (under the EHR-generator thresholds and under the generic-generator thresholds of 50 and 200), the jitter and CTGAN arms were forced by temporarily narrowing the thresholds and are therefore out-of-regime comparisons; this configuration is documented in the saved experiment configuration file, whereas the code revision used for the run is unrecoverable (the saved provenance records no commit; it records Python 3.13.7, numpy 2.5.2, pandas 2.3.3, scikit-learn 1.7.2 and SDV 1.32.0). Because the thresholds were set explicitly for each arm, the arms do not depend on the default thresholds of any code revision; the default thresholds quoted in this paper describe the current public repository and are not claimed to be those of the historical run. The results are retained from the saved outputs (configuration, sampling manifest, per-run results, summary table), which are in the public repository, with this reproducibility limitation; the ablation was not re-run.

Table 3d: Preliminary matched-size EHR generator ablation at n = 100 (UCI Cleveland Heart Disease; identical 100-row seed-42 training subset and identical untouched 61-row hold-out for every arm; 3 generation seeds 42–44; 200 synthetic rows per arm; mean ± SD). This is not a generator leaderboard: a single boundary point, three seeds, one dataset, no significance test.

Backend (size regime)	Downstream AUROC (hold-out)	ΔCorr Frobenius (raw)	Constraint validity	Generation time (s)
Jitter ($n < n_1$; forced at n = 100, out-of-regime)	0.930 ± 0.015	0.385 ± 0.075	100.0 ± 0.0%	0.074 ± 0.002
Gaussian copula ($n_1 \leq n < n_2$)	0.797 ± 0.033	0.630 ± 0.089	100.0 ± 0.0%	3.982 ± 1.764
CTGAN ($n \geq n_2$; forced at n = 100, out-of-regime stress test)	0.416 ± 0.099	1.572 ± 0.100	100.0 ± 0.0%	22.472 ± 1.180
Real-100 reference (RF trained on the 100 matched real rows)	0.944 ± 0.003	—	100.0%	0

The real-vs-real ΔCorr reference (resampling noise floor at this n) is 0.8788; each backend's ΔCorr should be read against that value, not against zero.

Interpretation. This is a preliminary matched-size comparison at one routing boundary, not a significance test (three seeds; no hypothesis test was performed). At exactly n = 100 the current public-repository router (Eq. 17; $n_1 = 100$ for the EHR generator, $n_1 = 50$ and $n_2 = 200$ for the generic tabular generator) selects the Gaussian copula. In this comparison, however, jitter (forced, although the router would not select it at this size) was closer to the real-100 downstream AUROC reference (0.930 vs 0.797; reference 0.944) and preserved correlation structure better (ΔCorr 0.385 vs 0.630, both below the 0.8788 real-vs-real floor). The experiment therefore does not validate the Gaussian copula as globally optimal at n = 100, and it does not establish that the current routing thresholds are globally optimal; it suggests the n = 100 boundary may warrant further investigation with a broader n-grid and additional datasets. CTGAN at n = 100 is well outside its intended larger-table regime and is included only as a stress-test reference — its near-chance AUROC here is not a fair indication of its general performance. The 100% constraint validity is non-discriminative in this setting because the shared post-processing step that clips values to the training range (_clip_to_training_bounds) is applied to every backend before constraint checking. Full configuration, sampling manifest, per-run results, and provenance are in data/processed/orchestration/ehr_matched_size_ablation/.

10.5. Privacy Screening

We describe SynForge-Cardio as privacy-audited / privacy-conscious rather than privacy-preserving: the evaluations below provide empirical privacy-risk screening rather than a formal differential-privacy guarantee, and no (ε, δ)-differential privacy was trained for.

Table 3b: Per-modality privacy-screening verdicts.

Modality	Screen	Result	Verdict
CXR	Membership inference (loss-based, AUC = 0.51)	advantage ≈ 0.02	pass
CXR	Attribute inference (gender)	attack acc 0.385 vs chance 0.541	pass
CXR	Nearest-neighbour / memorization / re-identification	feature-space NN median 15.42; 2% flagged for inspection	preliminary screening only
ECG	Source recovery (Mode B)	top-1 42.7% (NORM) / 43.4% (MI); mean correlation to true source ≈ 0.42; MI near-duplicate ≈ 17%	concern
ECG	Membership inference	no learned model to attack	not applicable
EHR	Membership inference (DCR-based)	AUC 0.581	pass (threshold 0.60)
EHR	Attribute inference (heart disease)	attack acc 0.656 vs 0.541 baseline	concern
EHR	Nearest-neighbour disclosure	median distance 2.16; exact-match 0%	pass

For the tested tabular extracts, no exact memorization of the training set was observed at zero tolerance, and the lower-tail nearest-neighbour distance stayed above the internal alert level (Figure 11). The EHR screening values are those in Table 3b. The ECG source-recovery result and the EHR attribute-inference result are genuine concerns and are reported as such rather than understated as strong protection.

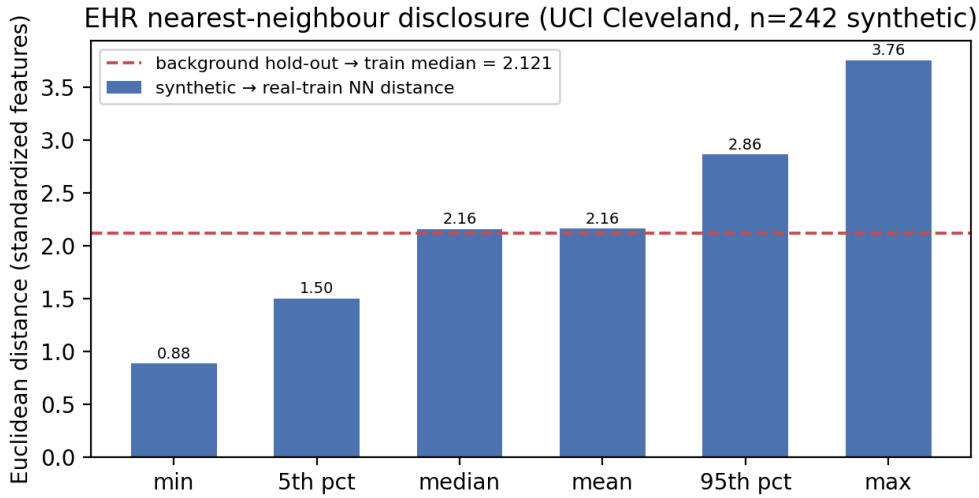


Figure 11: Nearest-neighbour disclosure screen for the EHR synthetic pool (UCI Cleveland, n = 242 synthetic vs 242 real training rows): distance-to-closest-real-record summary statistics in the standardized feature space, with the background hold-out→train median as reference. Exact-match rate is 0%. Generated from the privacy screening artifact.

10.6. Router / Orchestration Evaluation

SynForge-Cardio's multi-modality contribution is a software-level orchestration layer over independent modality-specific generators, not a joint multimodal model. This layer was evaluated on an internal, author-constructed benchmark of 52 prompts (37 well-defined, single correct modality set; 15 ambiguous / unsupported / malformed), written by the authors to exercise the router's keyword-matching rules rather than drawn from an external prompt corpus or independent clinical users. Given its small absolute size, the reported percentages describe this framework's internal routing accuracy on its own benchmark, not a generalizable estimate of routing accuracy on arbitrary clinical text.

Routing accuracy on the 37 well-defined prompts is 94.6% (35/37 exact modality-set matches). Per-modality precision / recall / F1 are 1.0 / 1.0 / 1.0 for EHR and CXR and 0.889 / 1.0 / 0.941 for ECG. Direct-versus-routed generation was 100% successful on both paths, with 100% output equivalence on the modalities both paths activated, and the router avoided 59 unnecessary generator calls across the well-defined subset. Schema-valid output, validation execution, and report generation were each 100%, with 0% missing metadata (validation/reporting completeness = 100%). The router unit-test suite passes 31/31. Routing is manually defined (hand-authored rules), not learned.

Representative prompts (verbatim from the benchmark): “Generate a chest X-ray showing cardiomegaly” → CXR; “Create an EKG waveform with heart rate 88 bpm” → ECG; “Generate an EHR record for a 55 year old with hypertension” → EHR; “Generate an EHR record for a patient with heart failure” → EHR + CXR (rule); “Create tabular data for a patient with atrial fibrillation” → EHR + ECG (rule).

Failure cases. The two routing errors are the “arrhythmia” prompts (“Generate a patient record for someone with arrhythmia”; “Create tabular data for a patient with cardiac arrhythmia”), both expected to route to EHR only. The keyword matcher matches the substring “rhythm” inside “arrhythmia” and additionally fires the ECG branch — two ECG false positives. More broadly, the router has no genuine reject outcome: all 15 ambiguous / unsupported-modality / malformed prompts fell back to generating all three supported modalities (none crashed). Ambiguity and keyword-less prompt handling remains a limitation. The router's demonstrated contribution is operational dispatch, efficiency and standardization; it does not improve medical generation quality.

10.6.1. Direct-backend versus routed orchestration ablation

To isolate the value of the routing layer we ran a controlled ablation over the same 37 well-defined benchmark prompts. Arm A (direct backend) supplies the correct modality set for each prompt and invokes those backend(s) directly, without consulting the router; Arm B (SynForge routed) sends the same prompt

through the shared clinical-prompt interface and the rule-based router, then calls the identical per-modality generation code. Prompt text, request count, seed (42), ECG mode (parametric), and validation configuration are the same in both arms; the only intended difference is direct backend invocation versus router-based dispatch. A blind reference arm (no router; always invoke all three modalities) is used only to quantify wasted calls.

Both arms completed generation, validation, per-modality metadata, and artifact assembly for all 37 prompts. Arm A selected the intended backend for all 37 prompts by construction; the router (Arm B) reproduced the intended set for 35/37 (94.6%), the two misses being the arrhythmia substring over-inclusions described above (an extra ECG call, not a wrong or missing primary modality). For every modality that both arms invoked (50 modality-level comparisons) the generated output was identical under deterministic evaluation: exact ECG waveform equality and exact CXR/EHR condition-contract equality, for an output-equivalence rate of 100%. The same transparency held for the real product backends reached through the model router: the GaussianCopula EHR generator and the LoRA-adapted CXR diffusion generator produced byte-identical output checksums whether called directly or dispatched through the router, and the model router returns exactly one generator for each known request type (raising an error for an unknown type rather than fanning out).

Table 3c: Direct-backend versus SynForge-routed orchestration on the 37-prompt benchmark. Not a generation-quality leaderboard: the underlying generators are identical in both arms.

Metric	Direct backend	SynForge routed
Backend selection	37/37 (operator-supplied)	35/37 (94.6%, automatic)
Generation success	100%	100%
Validation success	100%	100%
Metadata completeness	100%	100%
Artifact completeness	100%	100%
Mean latency	0.21 ms	0.28 ms
Unnecessary calls vs. intended backend set	0	2

Against blind multi-backend execution, the router avoided 59 unnecessary generator invocations across the 37-prompt benchmark. The direct arm is faster only because the operator already supplies the correct backend; the router introduced a mean dispatch overhead of 0.0025 ms per request while avoiding those 59

invocations relative to blind multi-backend execution. Frozen-artifact integrity was verified before and after the ablation (45 tracked checkpoints and result files byte-identical); no training was run and no model weights, seeds, thresholds, or TSTR artifacts were changed. The supporting test suite (tests/test_direct_vs_routed_baseline.py) passes 18/18.

The direct-versus-routed ablation indicates that routing is transparent to the underlying generation contract: all 50 valid modality-level comparisons were byte-identical or contract-identical under deterministic evaluation. This equivalence is expected by construction, because both arms ultimately invoke the same backend generator; it shows that routing does not alter the generated output, not that routing improves it. The routed path incurred a small dispatch overhead (0.0025 ms per request) but substantially reduced unnecessary backend execution relative to blind multi-backend invocation. These results support an operational rather than generative interpretation of the orchestration contribution: SynForge-Cardio does not improve the underlying generator's image, signal, or tabular generation quality through routing; it standardizes dispatch, validation, artifact handling, and modality selection. A head-to-head generation-quality comparison against external, independently published specialist frameworks was not performed and is stated as a limitation (Section 12.2).

10.7. Final Testing Summary

The automated test suite collects 1108 tests. In fresh chunked execution (each chunk in its own process, required by the 8 GB GPU's CUDA-context constraint) the aggregate is 1105 passed, 3 skipped, 0 failed. These totals are the authors' confirmed results from that chunked run. The original run log or CI report could not be located, and no new test run was performed for this revision, so the totals have not been independently reproduced.

The 3 skipped tests are GPU-gated CXR product-contract tests; executed separately with the GPU flag enabled they pass 3/3. Earlier single-process ~1000-test CUDA failures were environmental CUDA-context degradation and all affected suites passed in fresh processes.

11. Downstream Utility Experiments

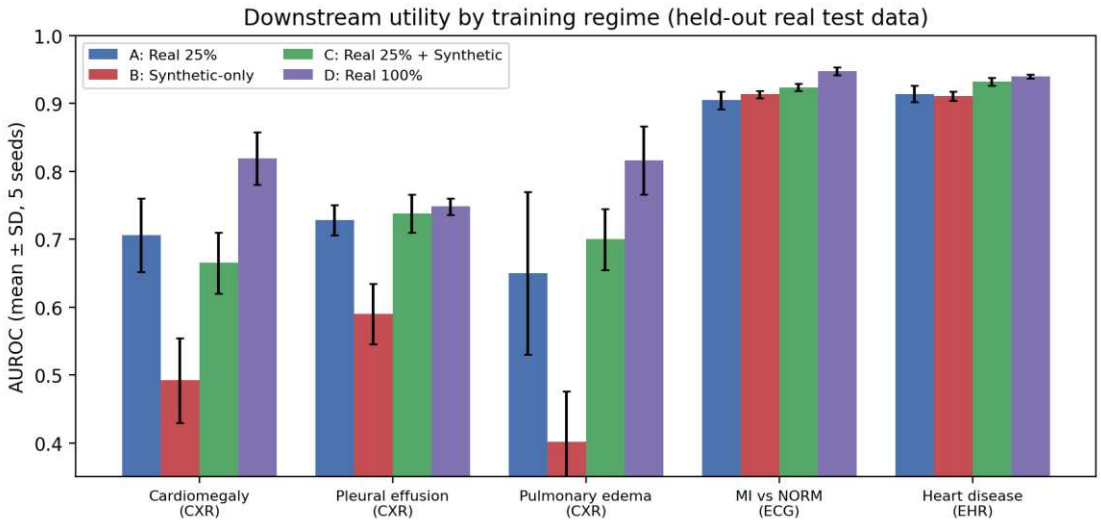
Utility experiments test whether synthetic data can preserve a certain degree of clinical utility for a standard baseline task, rather than the readiness of synthetic data for clinical application.

Table 4: Downstream utility under the four training regimes, five seeds (42–46), mean ± SD, evaluated on held-out real data. A = Real 25% (low-real baseline), B = Synthetic-only, C = Real 25% + Synthetic, D = Real 100%. Δ = C − A (improvement over the low-real baseline; not C − D).

Task	Model	A: Real 25%	B: Synthetic-only	C: Real 25% + Syn	D: Real 100%	Δ = C − A
Cardiomegaly (CXR)	DenseNet-121	0.706 ± 0.054	0.492 ± 0.062	0.665 ± 0.045	0.819 ± 0.039	−0.041
Pleural effusion (CXR)	DenseNet-121	0.728 ± 0.022	0.590 ± 0.044	0.738 ± 0.028	0.748 ± 0.012	+0.010
Pulmonary edema (CXR)	DenseNet-121	0.650 ± 0.120	0.402 ± 0.074	0.700 ± 0.045	0.816 ± 0.050	+0.050
MI vs NORM (ECG)	ResNet-1D	0.9047 ± 0.0130	0.9132 ± 0.0051	0.9236 ± 0.0054	0.9478 ± 0.0057	+0.0190
Heart disease (EHR, UCI)	Random forest	0.914 ± 0.012	0.911 ± 0.007	0.932 ± 0.006	0.940 ± 0.003	+0.017

All values are AUROC. Interpretation: for the three CXR tasks the change from augmenting the low-real baseline with synthetic data is −0.041 (Cardiomegaly), +0.010 (Effusion) and +0.050 (Edema); none of the three is statistically distinguishable from zero at the current five-seed scale (the 95% t-intervals of the paired per-seed differences overlap zero; Section 11.1), so the existing results do not demonstrate a reliable downstream gain from CXR synthetic augmentation. Synthetic-only training (regime B) is consistently worse than the real-data regimes for CXR and is near or below chance for two of the three CXR tasks. For ECG MI-vs-NORM the effect is directionally positive (Δ = +0.019 AUROC; the paired per-seed differences are all positive, +0.009 to +0.035; Section 11.1) but reflects bounded real-waveform augmentation acting as a data-augmentation technique, not evidence that a learned generative model

captures MI morphology. For EHR heart-disease classification Δ = +0.017 AUROC (positive at all five seeds, +0.001 to +0.027, which varies only the classifier seed); the EHR held-out set has only 61 records, so this estimate is highly sample-size-sensitive and noisy, and +0.017 should not be read as evidence of general clinical utility. Overall: synthetic augmentation improved some low-real downstream conditions, while effects were mixed across CXR tasks and no regime substitutes for real data.



11.1. Scope of the Uncertainty Estimates

The five-seed mean ± SD in Table 4 quantifies uncertainty from classifier-training-seed variation only. It does not capture sampling variability of the held-out test sets, which is large for the 61-record EHR hold-out and for the CXR tasks (1,038 hold-out images, with 26 positives for cardiomegaly), nor generator variability: the frozen synthetic pools were reused as-is and no repeated-generation experiment (regenerating the synthetic pool and retraining) was run, so the reported spreads understate total uncertainty.

Paired per-seed differences (C − A), computed from the saved per-seed results (seeds 42–46): ECG MI-vs-NORM +0.023, +0.019, +0.035, +0.009, +0.009; EHR +0.019, +0.027, +0.024, +0.001, +0.015; CXR cardiomegaly +0.033, +0.024, −0.120, −0.113, −0.027; pleural effusion +0.067, −0.003, −0.001, −0.044, +0.030; pulmonary edema +0.147, +0.043, −0.186, +0.034, +0.211. The 95% t-intervals of these paired differences are [+0.005, +0.033] for ECG and [+0.005, +0.030] for EHR, and overlap zero for all three CXR tasks. These are seed-level intervals only. The ECG gain comes from bounded real-waveform augmentation (Section 10.3), and the EHR result rests on a 61-record hold-out, so neither should be read as evidence of general clinical utility.

A test-set bootstrap analysis over held-out records and a simple real-data ECG reference baseline were drafted earlier, but their scripts, outputs and test files could not be recovered, so no results from them are reported in this paper. Both remain future work.

Figure 12: Downstream AUROC by training regime across five diagnostic tasks (five seeds, mean ± SD). A: Real 25%; B: Synthetic-only; C: Real 25% + Synthetic; D: Real 100%. Generated from the five-seed artifacts; the EHR bar is UCI Cleveland heart-disease classification (no MIMIC-IV).

12. Discussion and Limitations

12.1. Claims Boundary

SynForge-Cardio attempts to reduce the fragmentation of multimodal synthetic-data generation workflows within cardiovascular research. The chief claim is that CXR, ECG, and EHR generation can leverage a common prompt interface, router, validation path, and local deployment model. The empirical claim: on the public-data protocol, the defaults run end-to-end and produce measurable framework-level behaviour

(94.6% routing accuracy, 100% direct/routed equivalence on evaluated cases, 59 unnecessary calls avoided, 100% validation/reporting completeness), while downstream-utility gains are small and mixed.

The SynForge-Cardio framework does not claim that: (i) it achieves state-of-the-art results in any one modality generative model; (ii) its output is legally sanctioned for HIPAA/GDPR data sharing and clinical use; (iii) synthesized data are clinically equivalent in diagnostics; (iv) a single virtual patient's image, ECG, and tabular representation is anatomically or physiologically consistent; or (v) it offers any guarantee of formal differential privacy.

CXR, ECG, and EHR outputs are independently generated from modality-specific datasets. The framework therefore provides multimodality at the orchestration level rather than patient-level joint synthesis, and does not claim anatomical or physiological consistency between independently generated records. No expert radiologist or cardiologist review of the synthetic outputs was performed.

The framework-level evidence shows that the orchestration layer preserves the selected backend's output while providing automatic dispatch and reducing unnecessary multi-backend execution (Section 10.6.1); it does not improve image, ECG, or EHR fidelity through routing.

12.2. Limitations

1. At the time of reported use, CXR is 512 × 512 in resolution to fit GPU memory constraints.

2. The CXR generative status is partial: pathology-conditioning fidelity is heterogeneous (reproducible for Cardiomegaly, weak or inverted for Effusion and Edema), and a residual colour/tint contamination is intrinsic to the SD1.5 diffusion latent under the evaluated configuration. The supported CXR scope is frozen to Cardiomegaly and No-Finding.

3. The data do not adequately capture rare morphology or rare diseases; average metrics will not catch all such failures.

4. The prompt parser is rule-based and can struggle with intent clarification; the router has no genuine reject outcome and falls back to generating all three modalities for ambiguous or unsupported-modality prompts.

5. The thresholds in Eq. (17) are heuristics. A preliminary matched-size ablation at the single n = 100 boundary was completed (Section 10.4.1): on UCI Cleveland, with three generation seeds, jitter was closer to the real-data downstream reference than the Gaussian copula at exactly n = 100, so the experiment does not establish that the current routing thresholds are globally optimal and indicates the boundary warrants further study. Its scope is narrow: (i) only one boundary point was tested; (ii) only three generation seeds, and no significance test was run; (iii) only the UCI Cleveland dataset (242 training rows); (iv) the jitter and CTGAN arms were forced out-of-regime at n = 100 (the router selects the Gaussian copula there); CTGAN in particular is outside its intended larger-table regime and was included only as a stress-test reference; (v) the 100% constraint validity is non-discriminative because a shared clip-to-training-range post-processing step is applied to every backend before constraint checking; and (vi) a comprehensive grid over table size and across multiple datasets remains future work.

6. DCR and the other privacy screens are empirical evidence, not a formal privacy guarantee. ECG Mode-B synthetic samples can be matched to their true real source patient at top-1 ≈ 43% (a genuine privacy concern), and EHR attribute-inference accuracy (0.656) exceeds the majority-class baseline (0.541).

7. MIMIC-IV was not acquired (credentialed access unavailable); all EHR results are UCI-based and the EHR sample size (n = 303) is small.

8. A controlled direct-backend-versus-routed orchestration ablation was completed (Section 10.6.1), showing the routing layer preserves the backend output contract. It evaluates the orchestration layer itself and does not constitute a head-to-head generation-quality benchmark against external specialist frameworks; comparisons against standalone TimeGAN, CTGAN/SDV, and external diffusion systems (e.g. RoentGen-class models) were not run and remain future work.

9. Only limited aspects of clinical expert knowledge have been incorporated, through heuristic scoring; no clinician review was performed.

10. The public datasets used differ in distribution from local hospital data, so the observations may not transfer to other medical centers.

11. Downstream results are five-seed for the major TSTR analyses and single-seed for distributional snapshots; the observed downstream gains (≈ 0.02 AUROC or less) are modest and, for CXR and for EHR (61-record hold-out), have confidence intervals that include zero (Section 11.1).

12. The near-perfect ECG condition-fidelity AUROC (≈ 0.9985) co-occurs with a negative MI beat-template correlation (−0.27); this raises the possibility, not a demonstrated finding, that the fidelity classifier partly exploits synthetic-specific shortcuts rather than clinically meaningful morphology (Section 10.3). Combined with ≈ 42.7–43.4% Mode-B source recovery (Section 10.5), this independent ECG generator's outputs should not be read as validated, independently generated MI morphology, and Mode B itself remains bounded real-waveform perturbation rather than an independently learned generative model.

12.3. Future Work

Possible future work includes adding spatial conditioning on CXR, retraining the CXR generator to remove the latent-level colour contamination and to support additional pathologies, generalizing ECG pathologies with a learned class-conditioned generator, better arbitrary constraints on tabular data, optional formal differential privacy at train time, learning clinically interpretable prompt parsers, adding a genuine router reject path, and a broader matched-generator ablation spanning a full grid of training-set sizes and multiple

EHR datasets (extending the preliminary n = 100 comparison in Section 10.4.1). MIMIC-IV integration is specified (an acquisition/preprocessing specification exists) but is blocked pending credentialed access. 3D and time-resolved cardiac data such as MRI/CT can be made router-compatible but will require significantly different backbones and compute hardware. The five-seed re-evaluation of the downstream tasks (seeds 42–46, mean ± SD) is reported in Table 4. The experiment scripts, split definitions and configuration/provenance files recording LoRA rank, sampler steps, guidance scale and seeds are included in the public repository; model checkpoints and generated synthetic samples are not currently released (see the Data availability).

13. Conclusion

This paper presented SynForge-Cardio, a multi-modality orchestration framework for synthetic CXR, 12-lead ECG, and EHR data generation aimed at privacy-conscious cardiovascular machine-learning workflows. The system combines a common natural-language clinical prompt interface, rule-based deterministic routing, established modality-specific generators run as independent pipelines, and a shared validation / privacy-screening / reporting layer in a workstation-scale full-stack implementation. We provided generative and evaluation formulae, an experimental protocol on public datasets, and empirical measurements of framework behaviour (94.6% routing accuracy, 100% direct/routed equivalence on evaluated cases, 59 unnecessary calls avoided, 100% validation/reporting completeness, 1108 tests collected with 1105 passed / 3 skipped / 0 failed), distributional fidelity, empirical privacy-risk screening, and downstream utility. Downstream gains are small and mixed — small but positive for ECG MI-vs-NORM (Mode-B bounded real-waveform augmentation), small for EHR (positive, but on a 61-record hold-out only) and not distinguishable from zero for the CXR tasks — and no regime substitutes for real data. The results support SynForge-Cardio as a practical research tool for multi-modality synthetic-data coordination, while leaving clinical validation, formal privacy accounting, patient-level joint synthesis, and single-modality performance competition as separate goals.

Sustainable Development Goals

This work relates to United Nations Sustainable Development Goal 3 (Good Health and Well-being), by supporting privacy-conscious development and evaluation of cardiovascular machine-learning methods on public data, and to Goal 9 (Industry, Innovation and Infrastructure), by providing a reproducible,

workstation-scale research framework for synthetic-data generation and validation. SynForge-Cardio is a research tool; no clinical deployment or patient-outcome benefit is claimed.

Funding
This research received no external funding.

Data availability
This study analyzed three third-party datasets and generated synthetic data from them; no new patient data were collected. NIH ChestX-ray14 (Wang et al., CVPR 2017 [10]) is provided by the NIH Clinical Center, which we acknowledge as the data provider, and is available from the original download page (https://nihcc.app.box.com/v/ChestXray-NIHCC). This study used its first image archive (images_001; 4,999 radiographs from 1,335 patients), obtained through the public Hugging Face repository https://huggingface.co/datasets/alkzar90/NIH-Chest-X-ray-dataset, together with the NIH metadata file Data_Entry_2017_v2020.csv. PTB-XL [11] (version 1.0.3 [25]; CC BY 4.0) is available from PhysioNet at https://physionet.org/content/ptb-xl/1.0.3/ (DOI: https://doi.org/10.13026/kfzx-aw45). The UCI Cleveland Heart Disease data (file processed.cleveland.data; 303 records, all retained; Section 9.2) are part of the UCI Heart Disease dataset [13] (CC BY 4.0; https://archive.ics.uci.edu/dataset/45/heart+disease; DOI: https://doi.org/10.24432/C52P4X). Each dataset must be obtained from its provider and remains subject to the terms and attribution requirements stated by that provider; this study does not redistribute any of these datasets. The datasets were accessed and used in accordance with the applicable terms and attribution requirements of their respective providers. MIMIC-IV [12] requires credentialed access, was not used in any reported experiment, and no MIMIC-IV data are included. The source code, tests, experiment scripts, split definitions, preprocessing manifests, configuration and provenance files, and the result and privacy-screening artifacts contained in the repository are publicly available at https://github.com/sreyas15/synforge-cardio (MIT licence). Trained model checkpoints and generated synthetic samples are not currently in that repository, and their public release has not been confirmed; the synthetic outputs have not been privacy-cleared for distribution (see the privacy-screening findings in Section 10.5).

Code availability
The source code, tests, experiment scripts, split definitions, configuration and provenance files, and result artifacts described in this paper are available at https://github.com/sreyas15/synforge-cardio under the MIT licence (inspected at commit c1d62857c402a42f53abe52f38fc6eb31ea3425e, 12 October 2026). Trained model checkpoints and generated synthetic samples are not in the repository, so results that depend on them cannot be reproduced from the public repository. The matched-size EHR ablation (Table 3d) script (backend/app/experiments/ehr_matched_size_ablation.py), configuration, per-run results and summary are in the repository (the jitter and CTGAN arms were forced by narrowing the routing thresholds); the code revision used when the ablation was run is unrecoverable. The original log of the reported test totals is not in the repository.

Author contributions
S.S.K. contributed to conceptualization, methodology, software development, investigation, data curation, formal analysis, visualization, and writing of the original draft. K.N. contributed to methodology, validation, investigation, and review and editing of the manuscript. A.T.M. contributed to supervision, project administration, methodology, and critical review and editing of the manuscript. R.S.G. contributed to validation, investigation, and review and editing of the manuscript. All authors reviewed and approved the final manuscript.

Additional information
Competing interests: The authors declare no competing interests.

References
[1] I. Goodfellow et al., “Generative adversarial nets,” in Proc. NeurIPS, 2014.

[2] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” in Proc. NeurIPS, 2020.

[3] R. Rombach et al., “High-resolution image synthesis with latent diffusion models,” in Proc. IEEE/CVF CVPR, pp. 10684–10695, 2022.

[4] L. Zhang, A. Rao, and M. Agrawala, “Adding conditional control to text-to-image diffusion models,” in Proc. IEEE/CVF ICCV, pp. 3836–3847, 2023.

[5] E. J. Hu et al., “LoRA: Low-rank adaptation of large language models,” in Proc. ICLR, 2022.

[6] N. Patki, R. Wedge, and K. Veeramachaneni, “The Synthetic Data Vault,” in Proc. IEEE DSAA, pp. 399–410, 2016.

[7] L. Xu et al., “Modeling tabular data using conditional GAN,” in Proc. NeurIPS, 2019.

[8] J. Yoon, D. Jarrett, and M. van der Schaar, “Time-series generative adversarial networks,” in Proc. NeurIPS, 2019.

[9] Q. Wen et al., “Time series data augmentation for deep learning: A survey,” in Proc. IJCAI, pp. 4653–4660, 2021.

[10] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, “ChestX-ray8: Hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases,” in Proc. IEEE CVPR, pp. 2097–2106, 2017, doi:10.1109/CVPR.2017.369.

[11] P. Wagner et al., “PTB-XL, a large publicly available electrocardiography dataset,” Scientific Data, vol. 7, art. 154, 2020, doi:10.1038/s41597-020-0495-6.

[12] A. E. Johnson et al., “MIMIC-IV, a freely accessible electronic health record dataset,” Scientific Data, vol. 10, art. 1, 2023.

[13] A. Janosi, W. Steinbrunn, M. Pfisterer, and R. Detrano, “Heart Disease,” UCI Machine Learning Repository, 1989, doi:10.24432/C52P4X. [Online]. Available: https://archive.ics.uci.edu/dataset/45/heart+disease

[14] A. Yale et al., “Generation and evaluation of privacy preserving synthetic health data,” Neurocomputing, vol. 416, pp. 244–255, 2020.

[15] A. Gonzales et al., “Synthetic data in health care: A narrative review,” PLOS Digital Health, vol. 2, no. 1, e0000082, 2023.

[16] G. Huang et al., “Densely connected convolutional networks,” in Proc. IEEE CVPR, 2017.

[17] K. He et al., “Deep residual learning for image recognition,” in Proc. IEEE CVPR, 2016.

[18] L. Breiman, “Random forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[19] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” Journal of Big Data, vol. 6, art. 60, 2019.

[20] J. Fonseca and F. Bacao, “Tabular and latent space synthetic data generation: A literature review,” Journal of Big Data, vol. 10, art. 115, 2023.

[21] World Health Organization, “Cardiovascular diseases (CVDs),” Fact sheet, 2021.

[22] M. Heusel et al., “GANs trained by a two time-scale update rule converge to a local Nash equilibrium,” in Proc. NeurIPS, 2017.

[23] M. Bińkowski et al., “Demystifying MMD GANs,” in Proc. ICLR, 2018.

[24] A. Hirsch, F. W. Ten, K. S. Krüger et al., “The Multimodal Anonymizer: a fully local multi-agent AI system for medical data deidentification,” medRxiv, 2026, doi:10.64898/2026.05.28.26353952.

[25] P. Wagner, N. Strodthoff, R.-D. Bousseljot, W. Samek, and T. Schaeffter, “PTB-XL, a large publicly available electrocardiography dataset (version 1.0.3),” PhysioNet, 2022, doi:10.13026/kfzx-aw45.

[26] J. Irvin et al., “CheXpert: A large chest radiograph dataset with uncertainty labels and expert comparison,” in Proc. AAAI Conf. Artif. Intell., vol. 33, no. 1, pp. 590–597, 2019, doi:10.1609/aaai.v33i01.3301590.

[27] A. E. W. Johnson et al., “MIMIC-CXR, a de-identified publicly available database of chest radiographs with free-text reports,” Scientific Data, vol. 6, art. 317, 2019, doi:10.1038/s41597-019-0322-0.

Appendix A. Symbol Glossary

Table 5: Principal symbols used in the paper.

Symbol	Meaning
D_real^M, D_syn^M	Per-modality real and synthetic datasets (not patient-linked)
x^M, y^M	Modality-specific sample and label (CXR image / ECG matrix / EHR vector)
p, c = τ(p)	Text prompt and text embedding
z_t, β_t, ᾱ_t	Diffusion latent, noise schedule, cumulative product

ε_θ	Conditional denoising network
$x(t)$, a_w, b_w, θ_w	Parametric ECG wave parameters
L_R, L_S, L_U	TimeGAN loss terms
$M(n)$	Size-aware tabular model selector
FID, KID	Image metrics (SSIM only for paired image-to-image diagnostics)
DTW, PSD_sim	ECG metrics
$D_{n,m}$	KS statistic
ΔCorr	Frobenius norm of the correlation-matrix difference
DCR($\hat{t}$), DCR_0.05	Nearest-real distance and its 5th percentile

Appendix B. Router benchmark prompts and failure cases

The router benchmark contains 52 fixed prompts across ten categories (clear CXR / ECG / EHR, rule-augmentation, synonym variants, explicit multi-modal, ambiguous-no-keyword, unsupported-modality, malformed, and an "arrhythmia" quirk category). 37 are well-defined. Representative entries:

Prompt (verbatim)	**Expected**	**Router output**
Generate a chest X-ray showing cardiomegaly	CXR	CXR ✓
Produce a 12-lead ECG for a healthy patient	ECG	ECG ✓
Create tabular clinical data for a diabetic patient	EHR	EHR ✓
Generate an EHR record for a patient with heart failure	EHR + CXR	EHR + CXR ✓
Create tabular data for a patient with atrial fibrillation	EHR + ECG	EHR + ECG ✓
Generate EHR, ECG, and CXR data as three independent outputs for the same clinical scenario	EHR + ECG + CXR	EHR + ECG + CXR ✓
Generate a patient record for someone with arrhythmia	EHR	EHR + ECG ✗ (substring "rhythm")
Create tabular data for a patient with cardiac arrhythmia	EHR	EHR + ECG ✗ (substring "rhythm")
(ambiguous / unsupported / malformed, 15 prompts)	—	falls back to all three modalities

The two "arrhythmia" prompts are the only two routing errors on the well-defined subset (both ECG false positives). All 15 special-category prompts default to generating all three supported modalities; none crash. A genuine reject path is future work.

Appendix C. CXR Prompt-Conditioning Template (Reproducibility)

This appendix documents, at the level of literal strings and file locations, how a clinical request becomes the text and conditioning signals actually passed to the CXR diffusion pipeline for the final product configuration reported in Section 10.2 (the frozen Cardiomegaly / No-Finding scope; Figure 9; the 500-image Cardiomegaly product run).

Call chain (file:line, as of the frozen configuration). Intent parsing extracts a canonical condition keyword from the caller's free text via a fixed alias table (app/generation/cxr/prompt_parser.py:parse_cxr_prompt).
The model router dispatches type="cxr_conditioned" requests to app/generation/cxr/final_conditioning_adapter.py:generate (line 126); for the supported conditions (Cardiomegaly, No Finding) with no explicit checkpoint override, this calls app/generation/cxr/candidate_a_adapter.py:generate, using the checkpoint recorded in

data/processed/cxr/final_cardiomegaly/selected_checkpoint.json (currently candidate_a_norm5/checkpoints/candidate_a_nc_step1500.pt). Because that checkpoint's mechanism is "candidate_a_appended_pathology_token_norm_constrained", generation is performed by app/generation/cxr/diffusion_lora_candidate_a_norm_constrained.py:generate_candidate_a_image_norm _ constrained (lines 308–334).

User-provided (varies per request): only the matched condition keyword extracted from the caller's free text by the alias table in prompt_parser.py (e.g., "cardiomegaly"), plus the requested image count and generation seed. The caller's literal free text is not itself passed to the diffusion model; only the canonical condition label survives past parse_cxr_prompt.

Automatically templated (fixed for a given condition, varies only with the condition label): the base text prompt is built by build_finding_prompt(condition) in app/generation/cxr/pathology_conditioning.py (lines 148–160). For Cardiomegaly this produces the literal string "Chest radiograph showing cardiomegaly". For No Finding, a separate hard-coded literal string bypasses build_finding_prompt: "Chest radiograph showing no acute abnormality".

Fixed across every generation regardless of condition: the negative prompt, defined once in app/generation/cxr/diffusion_lora.py (line 67) as the literal string "color, painting, 3d render, illustration, cartoon, text, label, watermark, low quality, artifacts".

Non-textual conditioning channel, in addition to (not a substitute for) the text prompt above: a one-hot pathology vector (condition_name_to_vector, pathology_conditioning.py line 98) is passed through a trained PathologyTokenProjector MLP, hard-norm-clamped (apply_hard_norm_guard, max_norm = 10.0, target_norm = 5.0 per the checkpoint's norm_constraint metadata), and appended as a 78th token to the

CLIP text encoder's 77-token output (append_pathology_token) before cross-attention. This is a separate learned embedding appended after text encoding, not additional prompt text, and it is what carries the pathology-conditioning signal beyond the coarse text template above.

Generation settings relevant to the conditioning mechanism (DiffusionLoRAConfig, diffusion_lora.py lines 71–86; consistent with Table 2): LoRA rank r = 16, α = 16, applied to the U-Net's to_q / to_k / to_v / to_out.0 projection layers; DPM-Solver multistep sampler, 15 steps; classifier-free guidance scale 7.5; 512×512 output resolution; fp16 base precision.

Other prompt-template code paths exist in the repository but do not correspond to the reported results and should not be conflated with the above: (i) diffusion_lora.py's FIXED_EVAL_PROMPTS / DISEASE_PROMPTS plain-text lookup, used only for the earlier F3/F4/Stage-A FID gates (the within-pipeline FID comparison in Section 10.2), carries no appended pathology token; (ii) an earlier "v2" / "Stage B" pathology-vector mechanism (diffusion_lora_v2_conditioning.py) is superseded and is not reached by a default request for a supported condition. The Candidate-A norm-constrained path documented above is the one actually reached by the live product path for Cardiomegaly / No-Finding requests, and is what produced Figure 9 and the Section 10.2 500-image product-run numbers.