AdaPCLA: Curriculum Prior Internalization For Long-Tailed Longitudinal EHR Generation

Shuai Cui[1,2] **Wenxuan Chen**[1] **Wenjie Du**[1] **Jian Lou**[3] **Dan Li**[2,3] **Wenjie Feng**[1,2,*]
[1]University of Science and Technology of China
[2]State Key Laboratory of AI Safety, Beijing 100086, China
[3]Sun Yat-sen University

Abstract

Generative modeling of longitudinal Electronic Health Records (EHR) is increasingly important for privacy-preserving research, yet standard autoregressive models tend to underrepresent the co-occurrence structure of tail events (i.e., diagnoses), reducing the fidelity and faithfulness of generated data for rare subpopulations. To this end, we propose AdaPCLA, a curriculum prior-internalization framework for long-tailed longitudinal *diagnosis-code* generation. AdaPCLA treats label-wise empirical priors as a temporary optimization scaffold: prior-aware logit adjustment is applied early in training so that rare-code gradients stay active, and the correction is then annealed away so that rare-code temporal and co-occurrence dependencies are internalized into the model parameters. The resulting generator performs inference without any external bias, and it can be adapted, without retraining, to a target cohort whose code frequencies differ from the source. Our theoretical analysis characterizes rare-code logit updates through the label-wise empirical Neural Tangent Kernel (NTK) and derives a prior-internalization bound that isolates annealing speed and NTK conditioning as the factors governing how much prior signal survives scaffold removal; we state explicitly what this bound does and does not imply empirically. Experiments on MIMIC-III and MIMIC-IV show consistent gains in tail plausibility, downstream utility, and target-frequency adaptation; in particular, AdaPCLA improves TailPairSeen over HALO by 114.2% on MIMIC-III and 65.1% on MIMIC-IV, and outperforms GPT-style generation by 3.5% F1 under MIMIC-IV→MIMIC-III adaptation without retraining.

1 Introduction

Longitudinal Electronic Health Records (EHR) provide a rich view of patient trajectories, capturing diagnoses, treatments, medications, and clinical events over time. They are central to a wide range of clinical applications, including risk prediction [1–3], medication recommendation [4, 5], and decision support systems [6, 7]. However, data sensitivity and privacy regulations hinder the sharing of real-world EHR data, limiting research and development in related intelligent technologies. High-quality synthetic EHR generation is therefore crucial for privacy-preserving research [8].

Longitudinal EHRs are challenging to model due to their high dimensionality, temporal sparsity, and heterogeneity. Moreover, a few common clinical patterns coexist alongside numerous rare but clinically significant conditions, and they usually exhibit a long-tailed pattern in the empirical marginal distribution of medical codes for symptoms, indicating that few codes occur frequently while most clinically relevant codes are rare. During the training process of generative models, such rare codes

*Corresponding author: Wenjie Feng (fengwenjie@ustc.edu.cn). Code and evaluation scripts: https://github.com/worldcs27/worldcs27.github.io.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

provide limited positive evidence and weak stable learning signals for their temporal, contextual, and co-occurrence dependencies. If a model captures only the dominant head patterns while underrepresenting rare pathological dependencies, the resulting synthetic data may appear realistic on average but become unreliable for the very subpopulations that matter most clinically. Therefore, the key challenge for synthetic EHR generation is twofold: generated records should match the overall real EHR distribution, while also preserving the temporal and co-occurrence dependencies involving rare clinical events to facilitate downstream clinical modeling and adaptation to cohorts with different code frequencies.

Existing generative models attempt to solve this problem from different aspects, but none fully resolves the long-tail difficulty. GAN and VAE-based methods [9, 10] improved privacy-preserving generation but often struggled to preserve longitudinal dependency structure. Autoregressive and hierarchical models [11] capture sequential patterns better, while diffusion-style approaches [12, 13] improve sample quality and distributional coverage. Nevertheless, across these families, a common difficulty remains: *optimization is dominated by frequent and easy-to-learn clinical patterns, whereas rare but clinically important dependencies receive weak and unstable learning signals [14, 15].* As a result, synthetic EHR generators may fit common clinical patterns yet fail to preserve rare-event plausibility, downstream utility, and controllable adaptation to target code frequencies.

To this end, we propose AdaPCLA, a curriculum prior-internalization framework for long-tailed longitudinal diagnosis-code generation. AdaPCLA applies prior-aware logit adjustment early in training to maintain effective rare-code updates, then gradually anneals this correction so that rare-code temporal and co-occurrence dependencies are internalized into the model parameters. This design supports *clean inference* without an external bias after training, and it also enables *zero-shot adaptation to target code frequencies* by applying target-prior corrections without retraining. Our theoretical analysis gives two messages: the label-wise empirical Neural Tangent Kernel (NTK) characterizes how prior-induced residuals update rare-code intrinsic logits, while the prior-internalization bound identifies annealing speed and NTK conditioning as theoretical factors for retaining prior signals after scaffold removal.

We conduct a comprehensive evaluation over large real-world datasets. Experimental results show that rather than only improving aggregate scores, AdaPCLA achieves the largest gains on tail structure: compared with HALO, it improves TailPairSeen by 114.2% on MIMIC-III and 65.1% on MIMIC-IV. For the zero-shot generation setting based on MIMIC-IV training to MIMIC-III inference, AdaPCLA reaches AUPRC of 0.922 without retraining, outperforming GPT-style generation by 3.5% F1, which indicates that the learned structure can be reused when the target cohort's code frequencies differ from the source. In summary, this paper makes the following contributions:

- We propose AdaPCLA, a prior-guided curriculum framework that strengthens rare-code learning while supporting bias-free inference and zero-shot adaptation to target code frequencies.
- We analyze AdaPCLA with label-wise empirical NTK dynamics and govern the prior-internalization bound by annealing speed and the empirical NTK.
- AdaPCLA achieves significant gains in tail plausibility, downstream utility, and zero-shot adaptation to target code frequencies on real-world datasets.

2 Related Work

Longitudinal EHR generation. Synthetic EHR generation has progressed from GAN-, VAE-, and correlation-aware models for discrete or tabular medical records [9, 10, 16] to temporal structured simulators and longitudinal mixed-type generators [15, 17, 18]. Recent work further improves high-dimensional visit modeling, privacy-preserving synthesis, multimodal diffusion, and LLM-based generation without direct patient-level access [8, 11–13]. These models improve temporal dependency modeling and sample realism. Closer to our tail-oriented goal, MTGAN [19] generates multi-label clinical time series with a conditional GAN and explicitly targets underrepresented labels, HiSGT [20] uses a hierarchy- and semantics-guided Transformer for realistic longitudinal records, and EHRDiff [21] studies diffusion-based EHR synthesis; these improve the backbone or the generative family, whereas AdaPCLA changes how the training objective treats rare codes and can be layered on top of them. However, because these generators are usually trained to fit the

overall distribution of the majority of diseases, rare disease codes contribute sparse learning signals, motivating a generation objective that gives tail codes stronger optimization support without weakening longitudinal dependency modeling.

Long-tailed learning and zero-shot control. Long-tailed learning methods such as resampling, focal loss, class-balanced loss, margin-based objectives, decoupled training, and logit adjustment reshape training under imbalanced labels [14, 22–27]. These methods are effective for discriminative classification, but direct use in longitudinal multi-label EHR generation is less straightforward: resampling can disrupt patient trajectories, fixed reweighting can destabilize sparse multi-label optimization, and static prior biases can leave the generator dependent on external correction at inference time. Meanwhile, clinical code frequencies vary across hospitals and cohorts [28], motivating generation methods that can adapt to target priors without retraining while preserving longitudinal dependencies and co-occurrence structure. Although label-shift correction and controllable tabular/EHR synthesis study related forms of occurrence frequency adaptation [10, 29–31], they do not directly address zero-shot control for long-tailed longitudinal EHR generators.

A detailed discussion about closely related work is deferred to Appendix A.

3 Problem Formulation

3.1 Preliminaries

Notions. Consider a structured EHR as a sequence of patient visits. Let $\mathcal{V}$ be a $|\mathcal{V}|$-size vocabulary of clinical events, i.e., medical codes. A patient's T-visit record forms a chronological trajectory $X = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$, where each visit $\mathbf{x}_t \in \{0,1\}^{|\mathcal{V}|}$ is a multi-hot vector indicating the presence of codes at step t; $\mathbf{x}_{<t}$ denotes the trajectory history of this patient before time step t. We assume that observed trajectories are independent and identically distributed samples drawn from an unknown underlying distribution $\mathcal{P}(X)$, which we aim to approximate.

Logit Adjustment (LA). Logit adjustment can be used to guide the model's learning under class imbalance by modifying the relative margins between a target class and its competing classes [27, 32, 33]. For a given sample (x, y) with $y \in \{1, \ldots, K\}$, consider a multiclass classifier with logits $C_k(x)$ for $k = 1, \ldots, K$. The logit-adjusted softmax loss can be written as

$$\mathcal{L}_{\mathrm{LA}}(x, y) = \log \left[1 + \sum_{y' \neq y} \delta(y, y') \exp\big(C_{y'}(x) - C_y(x)\big) \right],$$

where $C_{y'}(x)$ is the logit of a competing class y', and $\delta(y, y')$ is a prior-dependent adjustment weight. Larger $\delta(y, y')$ increases the penalty when the prediction value for the competing class y' approaches or exceeds the target class y, thereby changing the optimization pressure across imbalanced classes. In prior logit-adjustment methods for imbalanced classification, $\delta(y, y')$ is typically chosen as a function of empirical class priors, yielding a prior-calibrated decision boundary.

In the longitudinal EHR generation, the prediction target at each visit corresponds to a multi-hot vector for all medical codes. Let $y \in \{0,1\}^V$ denote the multi-label target over $\mathcal{V}$ and $y_c = 1$ when event c occurs in the target visit; let $\pi_c = \mathbb{P}(y_c = 1)$ be the empirical frequency of c at the visit level. Given the intrinsic model logit $z_c(x)$, we denote the effective logit after adjustment by $\tilde{z}_c(x)$. A natural multi-label analogue applies a prior-dependent shift element-wise, $\tilde{z}_c(x) = z_c(x) + b_c$, where b_c is a bias term determined by the empirical frequency of event c.

However, a fixed prior-dependent bias can make the generator depend on this external correction and prevent it from capturing the rare-event temporal and co-occurrence structure and internalizing them in its own parameters. AdaPCLA therefore treats the prior bias as a temporary optimization scaffold: it preserves rare-event learning signals early in training and is gradually annealed away, so that inference relies on model-encoded rare-event dependencies rather than an external correction.

3.2 Problem Formulation and Requirements

In general, longitudinal EHR generation is a sequential generative modeling task. Its goal is to learn a parametric generative model $M(\cdot; \theta)$ that approximates $\mathcal{P}(X)$. For a training set of N patient

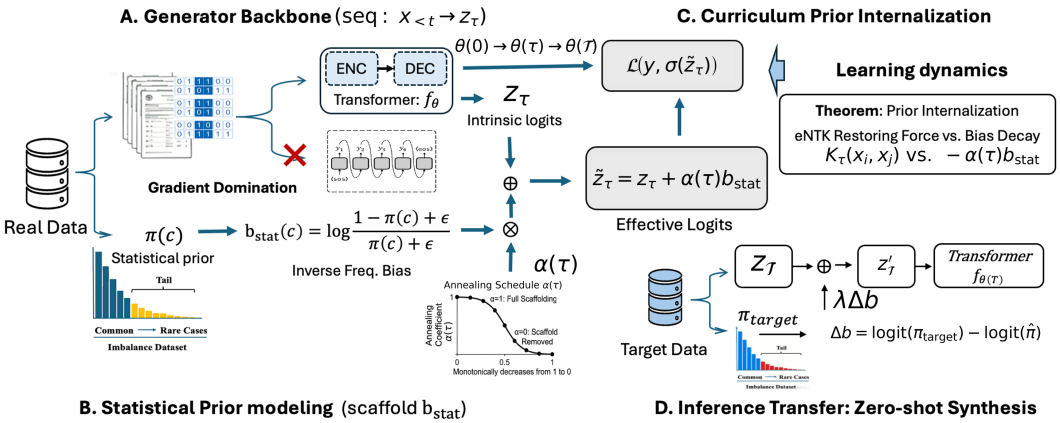


Figure 1: Schematic overview of the AdaPCLA framework.

trajectories $\mathcal{D} = \{X^i\}_{i=1}^N$, the maximum-likelihood objective can be written as

$$\theta^\star = \arg\max_\theta \frac{1}{N} \sum_{X=(\mathbf{x}_1,\ldots,\mathbf{x}_T)\in\mathcal{D}} \sum_{t=1}^{T-1} \log M(\mathbf{x}_{t+1} \mid \mathbf{x}_{1:t}; \theta). \tag{1}$$

However, real-world EHRs data are high-dimensional and sparse, and most medical codes in $\mathcal{V}$ form an extreme long-tailed distribution. This poses a substantial challenge for learning a model $M(\cdot;\theta)$ in Eq. (1) that faithfully characterizes the real data distribution $\mathcal{P}(X)$. In particular, under the binary cross-entropy loss, the gradient contributions associated with tail codes are severely overwhelmed by the large number of head samples. As a result, optimization becomes biased toward a head-dominated solution: for tail codes, gradients from the many non-occurrence labels ($y_c = 0$) can dominate the sparse gradients from occurrence labels ($y_c = 1$), pushing their logits downward and suppressing their generation probabilities. Such behavior leads the model to disregard the complex conditional dependencies of rare pathologies, ultimately fails to capture the intrinsic structure of the clinical data and to generate realistic EHR records faithfully.

Consequently, an optimal model $M(\cdot;\theta^*)$ should accurately approximate $\mathcal{P}(X)$ over the support of the data distribution, encompassing a comprehensive spectrum of medical codes and faithfully generating high-fidelity long-tailed patterns at inference time. In practice, particularly in low-resource and rare-disease settings, inadequate long-tail modeling can cause synthetic generators to miss clinically important rare-event patterns, reducing the usefulness of synthetic EHRs for downstream prediction and adapting generation to populations with different code frequencies.

Moreover, owing to the privacy constraints and the inherent difficulty of accessing and sharing large-scale real EHR datasets, the learned model is expected to be readily transferable to a new, typically small-scale dataset $\hat{\mathcal{D}}$, which may be drawn from a distinct distribution $\mathcal{P}'(X)$, thereby enabling the efficient construction of generative models that are well adapted to the target data.

4 Proposed Model

In this section, we propose AdaPCLA, a curriculum prior-internalization framework designed to counter the under-representation of rare diagnosis codes, to generate realistic longitudinal EHR records, and to adapt to a target cohort with different code frequencies without retraining. It treats empirical label priors as a temporary optimization scaffold and encourages the model to internalize this structural information by gradually removing the scaffold through annealing. Figure 1 illustrates the overall framework of AdaPCLA.

AdaPCLA consists of three core modules: *(i)* an **EHR Generator Backbone**, an autoregressive Transformer that captures longitudinal clinical dependencies; *(ii)* **Statistical Prior Modeling**, which converts long-tailed empirical prevalences into a label-wise logit scaffold $\mathbf{b}_{\text{stat}}$; and *(iii)* **Curriculum**

Prior Internalization, which anneals this scaffold so that early rare-event updates are protected while the final generator can operate without an external bias.

4.1 EHR Generator Backbone

AdaPCLA uses an autoregressive Transformer backbone to model longitudinal EHR trajectories. Given historical visits $x_{<t}$, the backbone embeds the visit events and processes the visit sequence with autoregressive masked self-attention, so that the hidden state used for predicting t-th step visit depends only on previous records and respects the temporal order:

$$H_{<t} = \mathrm{Transformer}_\theta(x_{<t}), \qquad z_t = W_o H_{t-1} + b_o \in \mathbb{R}^V,$$

where z_t denotes the intrinsic logits over the clinical-event vocabulary for the next visit. These logits are produced by the backbone parameters alone and constitute the logit space on which AdaPCLA performs prior-calibrated training during the annealing process.

Based on the shared representation H_{t-1}, the next visit is modeled with Bernoulli output heads:

$$p_\theta(x_t \mid x_{<t}) = \prod_{c=1}^{V} \mathrm{Bernoulli}(x_{t,c};\, \sigma(z_{t,c}))\,.$$

Equivalently, the visit-level training objective is the Binary Cross-Entropy loss over clinical events:

$$\mathcal{L}_{\mathrm{BCE}}(x_t, z_t) = -\sum_{c=1}^{V} \left[x_{t,c} \log \sigma(z_{t,c}) + (1 - x_{t,c}) \log(1 - \sigma(z_{t,c}))\right].$$

The factorization is imposed only at the output layer; temporal and cross-event dependencies are captured by the shared Transformer representation.

4.2 Optimization Scaffold: Statistical Prior

For the long-tailed EHR generation, a main difficulty lies in rare medical codes, which represent clinical events such as diagnoses, procedures, and medications, contribute far fewer positive updates than the common codes. Even when a positive occurrence of a rare code produces a large local BCE residual, its aggregate effect can be overwhelmed by gradients from high-frequency codes and from the many non-occurrence labels, i.e., $y_c = 0$, induced by the large code vocabulary. To make early training easier, we build a label-wise bias from empirical marginal frequencies.

Considering numerical stability, we use the smoothed marginal probability $\hat{\pi}_\epsilon(c) = \frac{\hat{\pi}(c)+\epsilon}{1+2\epsilon}$ for event c, then we define the statistical bias used in the scaffold as

$$b_{\text{stat}}(c) = -\mathrm{logit}\big(\hat{\pi}_\epsilon(c)\big) = \log\left(\frac{1 - \hat{\pi}(c) + \epsilon}{\hat{\pi}(c) + \epsilon}\right).$$

At the start of training, the model logit for c is close to its empirical log-odds, $z_{0,c}(x) \approx \mathrm{logit}\big(\hat{\pi}_\epsilon(c)\big)$. After adding $b_{\text{stat}}(c)$, the effective logit becomes $\tilde{z}_{0,c}(x) = z_{0,c}(x) + b_{\text{stat}}(c) \approx 0$. Thus, when the annealing coefficient is large early in training, the prior bias shifts the effective logit toward zero.

Since $\sigma'(u) = \sigma(u)(1 - \sigma(u))$ is maximized at $u = 0$, centering the early effective logit $\tilde{z}_{0,c}$ near zero makes a given change in the logit produce the largest change in the predicted probability, whereas a large-magnitude logit saturates the sigmoid. The same function is the curvature of the BCE loss in the logit, so rare codes also start in the region where the loss responds most strongly to parameter updates. We use this as a design principle and not as a claim about gradient magnitudes: for a positive label the residual $|\sigma(u) - y|$ actually *decreases* as u moves toward zero, so the net effect of centering on rare-code updates is an empirical question that we evaluate in Sec. 6.

We use b_{stat} only during training, rather than in the inference phase. At the beginning of training, many rare codes have low logits and receive logit-increasing gradients only from sparse occurrence labels ($y_c = 1$). Adding b_{stat} changes the residuals of these sparse rare-event observations, and those residuals then propagate through the shared backbone parameters rather than through an inference-time correction (Sec. 5.1). By reducing b_{stat} to zero gradually, the model learns to capture rare-event dependencies from the patient history rather than relying on a fixed external bias when rare events should appear. At inference time, the model will use its own logits alone for data generation.

4.3 Curriculum Prior Internalization

Retaining a fixed source-prior bias during the inference phase causes generation to remain perpetually dependent on an external correction term, which will introduce computational inefficiencies and can miscalibrate outputs when the target population has a different code distribution. Therefore, AdaPCLA utilizes the prior bias only as a training scaffold and anneals to zero gradually, encouraging rare-code dependencies to be encoded in the model parameters for bias-free inference.

During training, we use the modified logits $\tilde{z}_\tau(x) = z_\tau(x) + \alpha(\tau) b_{\text{stat}}$, where $z_\tau(x)$ is the intrinsic logit produced by the backbone, b_{stat} is the statistical prior bias, and $\alpha(\tau) \in [0, 1]$ controls the proportion of the prior bias used at step τ. The schedule starts from $\alpha(0) = 1$ and ends at $\alpha(\mathcal{T}) = 0$. Concretely, the coefficient is held at $\alpha = 1$ for the first 30% of epochs, decreases *linearly* to 0 over the next 40%, and stays at $\alpha = 0$ for the final 30%, i.e. $\alpha(\tau) = \text{clip}\big((0.7\,\mathcal{T} - \tau)/(0.4\,\mathcal{T}),\, 0,\, 1\big)$. Because the transition is linear rather than abrupt, $\sup_s |\dot{\alpha}(s)| = 1/(0.4\,\mathcal{T})$ is finite and shrinks as the number of annealing steps grows; this is exactly the quantity that enters the bound of Theorem 5.1.

The training loss is the BCE loss computed with the effective logits:

$$\mathcal{L}_{\text{AdaPCLA}} = \mathcal{L}_{\text{BCE}}\big(y, \sigma(\tilde{z}_\tau(x))\big).$$

In the reported runs the positive labels are additionally up-weighted by $w = 1.5$ (Sec. 6). For clarity, the analysis in Sec. 5 is stated for the unweighted loss; Appendix L.3 gives the residual induced by the weighting and the corresponding adjustment of the constants.

At the beginning, $\alpha(\tau)$ is large, so rare-event observations exert a stronger effect on the update direction; as $\alpha(\tau)$ decays, the model must rely more on the patient history, and at the end of training $\alpha(\mathcal{T}) = 0$, so the generator uses only its own logits, $\tilde{z}_{\mathcal{T}}(x) = z_{\mathcal{T}}(x)$. Thus, AdaPCLA uses the prior bias as a temporary training aid, but inference is performed without any external bias.

4.4 Zero-Shot Adaptation to Target Code Frequencies

Data from different data versions, cohorts, or hospitals exhibits varying degrees of statistical differences and biases. Therefore, a model trained on one cohort is often applied to generate data for a target cohort with different code occurrence frequencies. AdaPCLA supports such adaptation by introducing a target-prior correction difference at inference time, without retraining. We deliberately frame the experiments in this paper as adaptation across data versions and cohorts rather than as transfer across institutions; see Appendix F for a preliminary multi-center sanity check.

Let $\hat{\pi}(c)$ be source prevalence used during training and $\pi_{\text{target}}(c)$ be target occurrence frequency for event c. We define the *logit difference* between source and target population distribution as

$$\Delta b(c) = \text{logit}\big(\pi_{\text{target}}(c)\big) - \text{logit}\big(\hat{\pi}(c)\big),$$

and modify the logits as $z'_t = z_t(\theta) + \lambda \Delta b$ for inference, where $\lambda \in [0, 1]$ controls the strength of the adjustment. $\lambda = 1$ applies the full prior correction and $\lambda = 0$ recovers the original model. Unless stated otherwise, λ is selected on a held-out *adaptation split* of the target cohort by matching generated code frequencies to target frequencies; the target test split is never used for selection. As we show in Appendix E, the result is stable over the whole grid $\lambda \in \{0, 0.25, 0.5, 0.75, 1\}$, and even $\lambda = 0$ (no correction at all) already outperforms HALO across the reported correction grid, so the reported gains are not an artifact of tuning this knob.

This correction follows the standard label-wise prior-shift rule: if the conditional distribution of patient context given event c is unchanged across the source and target populations, i.e., $\mathbb{P}_S(x \mid y_c) = \mathbb{P}_T(x \mid y_c)$, then the Bayes-optimal logit changes by

$$z^\star_{T,c}(x) = z^\star_{S,c}(x) + \text{logit}\big(\pi_T(c)\big) - \text{logit}\big(\pi_S(c)\big).$$

See Appendix K for the detailed proof. This correction is applied independently to each code at each next-visit prediction step. Under the label-wise prior-shift assumption, it steers label-wise occurrence probabilities toward the target population while reusing the longitudinal and co-occurrence structure learned by the backbone, without requiring target-domain retraining.

5 Theoretical Analysis

In this section, we analyze the training dynamics of AdaPCLA and give theoretical insights into why it can retain rare-code learning signals after the prior scaffold is digested.

First, the prior scaffold changes the BCE residuals, so the SGD update induced by these residuals is written into the intrinsic logits through the shared Transformer backbone. This happens because the scaffold is fixed and independent of the backbone parameters, i.e., $\nabla_\theta(\alpha_k b_{\text{stat}}) = 0$, and therefore $\nabla_\theta \tilde{z}_k(x) = \nabla_\theta z_k(x)$. This yields a label-wise gradient-transfer relationship: the residual of one code can influence the intrinsic logit of another code through the corresponding entry of the label-wise empirical NTK. Second, under standard lazy-training assumptions [34], we analyze how the effect of the annealed scaffold $\alpha(\tau)b_{\text{stat}}$ can be absorbed into the intrinsic logits as $\alpha(\tau)$ decays to zero. Together, these results justify that the prior bias can be injected to create an early rare-code learning signal, and the annealing schedule controls whether this signal is retained in the model parameters after the external bias is digested.

5.1 Gradient Transfer through the Shared Backbone

We identify how the prior scaffold affects rare-code intrinsic-logit updates through the label-wise empirical NTK. At training step k, the effective logits are $\tilde{z}_k(x) = z_k(x) + \alpha_k b_{\text{stat}}$, and the BCE residual is $\mathcal{G}^{(k)}_{\text{Ada}}(x) = \sigma(\tilde{z}_k(x)) - y$ which affects the SGD update direction as the scaffold. The additive term $\alpha_k b_{\text{stat}}$ is fixed with respect to the backbone parameters, so $\nabla_\theta(\alpha_k b_{\text{stat}}) = 0$ and $\nabla_\theta \tilde{z}_k(x) = \nabla_\theta z_k(x)$. The adjusted logits $\tilde{z}_k(x)$ determine the BCE residuals, and the fixed-scaffold property ensures that the corresponding parameter update changes the intrinsic logits $z_k(x)$. For a one-step SGD update with step size $\eta > 0$, the first-order intrinsic-logit update for code c is

$$\Delta z_k(c; x) = -\eta \sum_{j=1}^{V} [\mathcal{K}^{(k)}]_{c,j} \mathcal{G}^{(k)}_{\text{Ada},j}(x) + \mathcal{O}(\eta^2),$$

where $[\mathcal{K}^{(k)}]_{i,j} = \langle \nabla_\theta z_k(i; x), \nabla_\theta z_k(j; x) \rangle$. $\mathcal{K}^{(k)}$ is the label-wise empirical NTK for context x.

Therefore, the residual of code j changes the intrinsic-logit update of code c through the corresponding entry $[\mathcal{K}^{(k)}]_{c,j}$ of the label-wise empirical NTK. This relationship shows that the prior scaffold affects intrinsic logits by reshaping BCE residuals, whose effects are then transferred through backbone parameter updates. For each code c, the update is determined by BCE residuals weighted by the corresponding row of $\mathcal{K}^{(k)}$. Thus, AdaPCLA strengthens early rare-code learning signals only when those residuals can be transferred through the empirical-NTK structure.

5.2 Prior-Internalization Bound

For a fixed context x, define the expected BCE risk as

$$F_\tau(z; x) = \mathbb{E}_{Y \sim \mathbb{P}(\cdot|x)} \left[\mathcal{L}_{\text{BCE}}\big(Y, \sigma(z + \alpha(\tau) b_{\text{stat}})\big) \right].$$

Let $\tilde{z}^\star(x)$ be the effective-logit minimizer of the conditional expected BCE risk for the fixed context x. Because the effective logits are $z + \alpha(\tau)b_{\text{stat}}$, the corresponding target in intrinsic-logit coordinates is $z^\star_\tau(x) = \tilde{z}^\star(x) - \alpha(\tau)b_{\text{stat}}$. Since AdaPCLA removes the scaffold at inference time, prior internalization requires the learned intrinsic logits $z_\tau(x)$ to follow the target as $\alpha(\tau)$ decays to zero.

Theorem 5.1 (Prior-internalization bound)**.** *Under the tracking hypotheses of Appendix J, the internalization error $e_\tau(x) = z_\tau(x) - z^\star_\tau(x)$ satisfies*

$$\|e_\tau(x)\|_{\mathcal{K}^{-1}} \le e^{-\mu \lambda_{\min}(\mathcal{K})\tau} \|e_0(x)\|_{\mathcal{K}^{-1}} + \frac{B_\star(x)}{\mu \lambda_{\min}(\mathcal{K})} \sup_{s \in [0,\mathcal{T}]} |\dot{\alpha}(s)|. \tag{2}$$

Theorem 5.1 gives two design implications. First, smoother annealing reduces $\sup_s |\dot{\alpha}(s)|$, which lowers the internalization error caused by a moving scaffold. Second, a larger minimum eigenvalue $\lambda_{\min}(\mathcal{K})$ of the label-wise empirical NTK reduces the bound, meaning that the intrinsic logits can stay closer to the moving target. Because $\alpha(0) = 1$ and $\alpha(\mathcal{T}) = 0$, the moving target in intrinsic-logit coordinates shifts by $z^\star_{\mathcal{T}}(x) - z^\star_0(x) = b_{\text{stat}}$. When the internalization error $e_\tau(x)$ remains small along the annealing path, the learned intrinsic logits inherit this shift, so $z_{\mathcal{T}}(x) - z_0(x) \approx b_{\text{stat}}$. This supports the intended internalization property of AdaPCLA: the prior signal introduced by the training scaffold is absorbed into the model's own logits through the annealing trajectory, enabling inference without an external prior correction.

We stress what Theorem 5.1 does *not* give: it bounds the distance to a population-level minimizer path that is not observable from finite data, so it is a design guide rather than a testable prediction.

Table 1: Tail clinical plausibility results (mean±std over 3 independent generations).

Method	MIMIC-III				MIMIC-IV			
	PairSeen↑	TailPairSeen↑	TailCtxJSD↓	TailTopKJac↑	PairSeen↑	TailPairSeen↑	TailCtxJSD↓	TailTopKJac↑
GPT-style	0.7977 ± 0.0002	0.0384 ± 0.0005	0.6621 ± 0.0002	0.01131 ± 0.00019	0.8851 ± 0.0008	$\underline{0.0500 \pm 0.0054}$	0.6675 ± 0.0019	0.00151 ± 0.00013
LSTM	0.7272 ± 0.0001	0.0297 ± 0.0002	$\underline{0.6414 \pm 0.0003}$	**0.04905±0.00098**	0.8339 ± 0.0003	0.0355 ± 0.0006	$\underline{0.6464 \pm 0.0006}$	$\underline{0.01729 \pm 0.00046}$
EVA	0.0260 ± 0.0000	0.0023 ± 0.0000	0.6905 ± 0.0000	0.00083 ± 0.00004	$\underline{0.9042 \pm 0.0001}$	0.0482 ± 0.0015	0.6691 ± 0.0014	0.01159 ± 0.00055
SynTEG	0.0373 ± 0.0000	0.0019 ± 0.0000	0.6903 ± 0.0001	0.00002 ± 0.00000	0.2177 ± 0.0001	0.0072 ± 0.0001	0.6725 ± 0.0008	0.00006 ± 0.00000
HALO	**0.8263±0.0003**	$\underline{0.0520 \pm 0.0010}$	0.6558 ± 0.0003	0.01967 ± 0.00028	0.8768 ± 0.0010	0.0498 ± 0.0042	0.6652 ± 0.0022	0.01050 ± 0.00094
AdaPCLA	$\underline{0.7983 \pm 0.0271}$	**0.1114±0.0109**	**0.5981±0.0040**	$\underline{0.0345 \pm 0.0020}$	**0.9478±0.0032**	**0.0822±0.0022**	**0.6313±0.0033**	**0.0187±0.0013**

The closest measurable proxy (Appendix H) shows that all three schedules converge to within 10^{-2} (all-code) and 10^{-5} (tail-code) of the final logits, but the schedules are numerically indistinguishable, so we do not claim empirical verification of the speed dependence.

6 Experiments

6.1 Experimental Setting

Datasets and preprocessing. We evaluate on **MIMIC-III** [35] and **MIMIC-IV** [36]. Each patient is represented as a sequence of visits, where each visit is a deduplicated set of International Classification of Diseases (ICD) diagnosis codes. We use patient-level train/validation/test splits and apply the same preprocessing and vocabulary mapping to all methods. MIMIC-III contains 33,494/3,722/9,304 train/validation/test patient trajectories over 6,984 diagnosis codes and MIMIC-IV contains 160,884/17,877/44,691 over 9,143 codes, with 11.05 codes per real visit on average; full statistics are in Appendix C. We define the *tail set* as the lowest-frequency 20% of codes with nonzero training frequency, keeping ties together (1,471 codes on MIMIC-III, 2,125 on MIMIC-IV), computed on the training split only. The TSTR classifier predicts 25 patient-level disease groups from the MIMIC phenotyping benchmark [37], fixed before method development and defined at a different granularity from the tail ICD-code set.

Baselines. We compare AdaPCLA with representative EHR generators, including **LSTM** [38], **EVA** [15], **SynTEG** [17], **GPT-style** [13], and **HALO** (hierarchical autoregressive EHR generator) [11]. For each baseline, we follow the model architecture and training recipe described in the original paper, and use the same training cohort and evaluation protocol as AdaPCLA; any hyperparameter selection uses the validation split of the same dataset.

Metrics. We evaluate the quality of synthetic EHR data. For distributional fidelity, we examine occurrence frequency and visit-level code co-occurrence statistics. For tail clinical plausibility, we report PairSeen/TailPairSeen, TailCtxJSD, and TailTopKJac; their exact definitions are given in Appendix B. Because the paper is about *longitudinal* generation, we also report three adjacent-visit metrics, SeqBigram R^2, Transition JSD, and TailSeqPairSeen, which measure whether generated transitions between consecutive visits match the real ones. For downstream utility, we use the *Train-on-Synthetic, Test-on-Real* (TSTR) protocol and report Accuracy, F1, and AUPRC.

6.2 Experimental Results

Tail clinical plausibility. Table 1 reports tail clinical plausibility on MIMIC-III/-IV. AdaPCLA gives the strongest overall tail-focused performance across the two datasets. On MIMIC-IV it ranks first on all four metrics. In particular, it improves TailPairSeen from 0.0498 with HALO to 0.0822, a relative gain of 65.1%; and also reduces TailCtxJSD from 0.6652 to 0.6313, showing better tail-context consistency.

On MIMIC-III, AdaPCLA achieves the best TailPairSeen and TailCtxJSD. Compared with HALO, it improves TailPairSeen from 0.0520 to 0.1114, and reduces TailCtxJSD from 0.6558 to 0.5981. Although HALO has a higher overall PairSeen and LSTM has a higher TailTopKJac, AdaPCLA provides a better balance between tail support and tail-context consistency. Overall, these results show that AdaPCLA better preserves rare-event co-occurrence structure than other baselines.

Two aspects of Table 1 require comment. Several baseline standard deviations print as 0.0000 (a precision artifact; the values are below 5×10^{-5}, Appendix S), and AdaPCLA's PairSeen standard

Table 2: Complete downstream utility results (mean±std over 3 independent generations). Metrics: accuracy (Acc), F1, and AUPRC from 25-label diagnosis classification.

Method	MIMIC-III			MIMIC-IV		
	Acc↑	F1↑	AUPRC↑	Acc↑	F1↑	AUPRC↑
Baselines						
GPT-style	0.8256 ± 0.0023	0.8304 ± 0.0021	0.8902 ± 0.0020	0.8856 ± 0.0101	0.8912 ± 0.0059	0.9270 ± 0.0021
LSTM	0.5145 ± 0.0087	0.5017 ± 0.0274	0.5563 ± 0.0115	0.5441 ± 0.0143	0.5518 ± 0.0264	0.5967 ± 0.0120
EVA	0.5243 ± 0.0127	0.5167 ± 0.0108	0.5607 ± 0.0061	0.4829 ± 0.0227	0.3969 ± 0.0135	0.4896 ± 0.0080
SynTEG	0.5095 ± 0.0157	0.4016 ± 0.0449	0.5721 ± 0.0130	0.5333 ± 0.0061	0.5583 ± 0.0215	0.6602 ± 0.0097
HALO	0.8761 ± 0.0017	0.8779 ± 0.0016	0.9295 ± 0.0009	0.8592 ± 0.0030	0.8631 ± 0.0028	0.8962 ± 0.0015
Ours						
Static-PCLA (Fixed Bias)	$\underline{0.9019 \pm 0.0028}$	$\underline{0.9023 \pm 0.0027}$	$\underline{0.9455 \pm 0.0019}$	$\underline{0.8914 \pm 0.0219}$	$\underline{0.8912 \pm 0.0316}$	$\underline{0.9309 \pm 0.0135}$
AdaPCLA (Annealing)	$\mathbf{0.9040} \pm 0.0074$	$\mathbf{0.9051} \pm 0.0067$	$\mathbf{0.9514} \pm 0.0038$	$\mathbf{0.9130} \pm 0.0030$	$\mathbf{0.9104} \pm 0.0032$	$\mathbf{0.9449} \pm 0.0027$

Table 3: Compact model analysis. **(a)** Internalization ablation on MIMIC-III. **(b)** Zero-shot adaptation to target code frequencies from MIMIC-IV to MIMIC-III. In (a), Static-PCLA (Train-only) has high PairSeen but no valid tail-pair support (N/A), so it is excluded from best-value marking.

(a) Dependency → Internalization (MIMIC-III seed1)

Variant	PairSeen↑	TailPairSeen↑	TailCtxJSD↓	TailTopKJac↑
HALO	0.8261	0.0549	0.6549	0.0200
Static-PCLA (Consistent)	<u>0.8526</u>	<u>0.0755</u>	<u>0.6392</u>	<u>0.0256</u>
Static-PCLA (Train-only)	0.9992	N/A	N/A	0.0000
AdaPCLA (Ours)	0.8175	**0.1239**	**0.5940**	**0.0356**

(b) Target-frequency adaptation (IV→III)

Method	Acc↑	F1↑	AUPRC↑
LSTM	0.523	0.557	0.565
EVA	0.489	0.282	0.445
SynTEG	0.517	0.459	0.603
GPT-style	<u>0.822</u>	<u>0.836</u>	<u>0.895</u>
AdaPCLA (Ours)	**0.865**	**0.865**	**0.922**

deviation on MIMIC-III (0.0271) is larger than that of any baseline. All $\pm$ values measure variation across three cohorts from a *single* checkpoint; the cohorts differ in visit density (18.63/21.52/20.76 versus 11.05 codes per visit), which inflates PairSeen variance. After visit-size reweighting AdaPCLA remains higher, at 0.8303 ± 0.0138 (MIMIC-III) and 0.9151 ± 0.0110 (MIMIC-IV).

EVA and SynTEG show unstable tail plausibility: on MIMIC-III their PairSeen and TailPairSeen are close to zero, i.e. many generated intra-visit pairs are not supported by the training corpus. This is consistent with VAE- and GAN-based generators emitting overly dense multi-hot visits after thresholding: a dense visit contains many more intra-visit pairs, so a little noise in code selection already produces unsupported pairs.

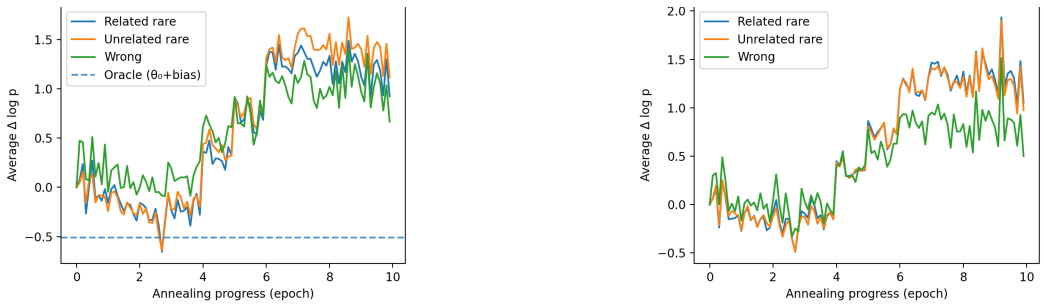
6.3 Model Analysis

We evaluate the properties and performance of AdaPCLA based on the following aspects: internalization ablation study, downstream utility, and zero-shot adaptation to target code frequencies.

Ablation study. Table 3(a) compares variants that differ in how the prior bias is used during training. *HALO* is the backbone generator without any prior bias. *Static-PCLA (Consistent)* uses the fixed prior bias both during training and inference, which tests whether a permanent bias can improve generation if it is always available. *Static-PCLA (Train-only)* uses the same fixed bias during training but removes it at inference, which tests whether the model still works after the external bias is taken away. AdaPCLA uses the bias only during training and gradually anneals to zero, and inference is performed without any external bias.

The results show a clear difference between using a fixed bias and annealing it. Static-PCLA (Consistent) improves over HALO on several tail metrics, which means that the prior bias is useful when it is kept at inference. However, Static-PCLA (Train-only) fails after the bias is removed: TailPairSeen and TailCtxJSD are undefined, and TailTopKJac drops to 0.0000, so a fixed training bias alone does not make the model learn the rare-event structure.

AdaPCLA uses no external inference bias and achieves the best defined tail metrics in this ablation. It obtains the highest TailPairSeen 0.1239 and TailTopKJac 0.0356, and the lowest TailCtxJSD 0.5940, outperforming all the baselines. This suggests that dynamic annealing helps the backbone learn useful prior information in its own logits, instead of depending on a fixed external correction.



(a) 30 contexts

(b) 100 contexts

Figure 2: Annealing dynamics of selected probe-code probabilities under 30 and 100 probing contexts. The qualitative trends remain stable when the probing set is enlarged.

Figure 2 further visualizes this annealing process by tracking how selected probe-code probabilities change over training checkpoints. A probe code is a candidate code evaluated under a fixed patient-history context, and the probes are grouped into related rare, unrelated rare, and wrong codes. The similar trajectory patterns under 30 and 100 contexts suggest that annealing produces stable internalization dynamics as the probing set is enlarged.

Downstream predictive utility. We use the TSTR protocol to test whether synthetic data supports real downstream prediction. For each generator, we train a 25-label diagnosis classifier only on synthetic data and evaluate it on the held-out real test set. As shown in Table 2, AdaPCLA achieves the best Accuracy, F1, and AUPRC on both MIMIC-III and MIMIC-IV. These results indicate that AdaPCLA preserves predictive signals that transfer from synthetic data to real-data.

Zero-shot adaptation to target code frequencies. We also test whether AdaPCLA can adapt to a target cohort without retraining. The model is trained on the source cohort and then applied to the target cohort using only the target code occurrence rates through the delta-bias correction in Sec. 4.4. Table 3(b) reports the MIMIC-IV→MIMIC-III result: AdaPCLA reaches Acc/F1/AUPRC of $0.865/0.865/0.922$, against $0.822/0.836/0.895$ for GPT-style. A separate sensitivity sweep also applies the correction to HALO: its highest grid AUPRC is 0.835 (versus 0.822 uncorrected), whereas AdaPCLA at $\lambda = 0$ reaches 0.927; Appendix E reports both directions and distinguishes this sweep from the main protocol. The gain therefore comes from the generator rather than from a test-time correction, and AdaPCLA retains the longitudinal dependencies and co-occurrence patterns learned from the source cohort.

7 Conclusion

In this paper, we proposed AdaPCLA, a curriculum prior-internalization framework for long-tailed longitudinal diagnosis-code generation. By using empirical priors as a temporary training scaffold and gradually annealing them away, AdaPCLA strengthens rare-code learning during optimization without requiring an external bias during inference. Experiments on real-world datasets show that AdaPCLA improves tail plausibility, downstream utility, and target-frequency adaptation over strong EHR generation baselines. Since AdaPCLA is a training objective rather than a new backbone, it can be layered on top of existing longitudinal generators as a long-tail enhancement module.

Limitations. First, our zero-shot experiments adapt across MIMIC versions and cohorts, not across institutions; the eICU check (Appendix F) is a single-seed sanity study on diagnosis codes only. Second, the newly added baselines MTGAN, HiSGT and EHRDiff-Visit (Appendix D) are single-run snapshots rather than three-cohort means. Third, the privacy diagnostics (Appendix G) are attack-based measurements, not differential-privacy guarantees, and they show a higher attribute-inference recovery for AdaPCLA, a trade-off of modelling rare codes more faithfully. Finally, only ICD diagnosis codes are evaluated; other EHR modalities remain future work.

Acknowledgements

We would like to thank anonymous reviewers and area chairs for their constructive comments and efforts in improving our paper. This work was supported in part by the Open Funding Programs of State Key Laboratory of AI Safety (Project No. 2025-20). The authors declare no competing interests.

References

[1] Kim-Anh-Nhi Nguyen, Dhavalkumar Patel, Masoud Edalati, Maria Sevillano, Prem Timsina, Robert Freeman, Matthew A. Levin, David L. Reich, and Arash Kia. Electronic-medical-record-driven machine learning predictive model for hospital-acquired pressure injuries: Development and external validation. *Journal of Clinical Medicine*, 14(4), 2025. ISSN 2077-0383. doi: 10.3390/jcm14041175. URL https://www.mdpi.com/2077-0383/14/4/1175.

[2] Tianyi Liu, Andrew Krentz, Lei Lu, and Vasa Curcin. Machine learning based prediction models for cardiovascular disease risk using electronic health records data: systematic review and meta-analysis. *European Heart Journal - Digital Health*, 6(1):7–22, 10 2024. ISSN 2634-3916. doi: 10.1093/ehjdh/ztae080. URL https://doi.org/10.1093/ehjdh/ztae080.

[3] Norah Hamad Alhumaidi, Doni Dermawan, Hanin Farhana Kamaruzaman, and Nasser Alotaiq. The use of machine learning for analyzing real-world data in disease prediction and management: Systematic review. *JMIR Medical Informatics*, 13:e68898, June 2025. ISSN 2291-9694. doi: 10.2196/68898. URL https://medinform.jmir.org/2025/1/e68898.

[4] Feiyan Liu, Wenhao Wang, Jiawei Zheng, Yibo Xie, Xiaoli Wang, and Dongxiang Zhang. EDRMM: enhancing drug recommendation via multi-granularity and multi-attribute representation. *BMC Bioinform.*, 26(1):173, 2025. doi: 10.1186/S12859-025-06167-4. URL https://doi.org/10.1186/s12859-025-06167-4.

[5] Juntao Li, Haobin Yuan, Ling Luo, Tengxiao Lv, Yan Jiang, Fan Wang, Ping Zhang, Huiyi Lv, Jian Wang, Yuanyuan Sun, and Hongfei Lin. Overview of CHIP 2025 shared task 2: Discharge medication recommendation for metabolic diseases based on chinese electronic health records. *CoRR*, abs/2511.06230, 2025. doi: 10.48550/ARXIV.2511.06230. URL https://doi.org/10.48550/arXiv.2511.06230.

[6] Yu-Chen Liu, Guan-Ling Lin, Jeremiah Scholl, Yi-Chun Hung, Yu-Jing Lin, Yu-Chuan Li, and Hsuan-Chia Yang. Evaluation of diagnostic recommendations embedded in medication alerts: Prospective single-arm interventional study. *Journal of Medical Internet Research*, 27:e70731, May 2025. ISSN 1438-8871. doi: 10.2196/70731. URL https://www.jmir.org/2025/1/e70731. PMID 40424063.

[7] Huang Huang, Wei Lyu, Md Mahmud Hasan, and Shannon H Houser. Adoption of machine learning in us hospital electronic health record systems: Retrospective observational study. *Journal of Medical Internet Research*, 27:e76126, December 2025. ISSN 1438-8871. doi: 10.2196/76126. URL https://www.jmir.org/2025/1/e76126.

[8] Jinsung Yoon, Michel J. Mizrahi, Nahid Farhady Ghalaty, Thomas Jarvinen, Ashwin S. Ravi, Peter Brune, Fanyu Kong, Dave Anderson, George Lee, Arie Meir, Farhana Bandukwala, Elli Kanal, Sercan Ö. Arik, and Tomas Pfister. Ehr-safe: generating high-fidelity and privacy-preserving synthetic electronic health records. *npj Digit. Medicine*, 6, 2023. doi: 10.1038/S41746-023-00888-7. URL https://doi.org/10.1038/s41746-023-00888-7.

[9] Edward Choi, Siddharth Biswal, Bradley A. Malin, Jon Duke, Walter F. Stewart, and Jimeng Sun. Generating multi-label discrete patient records using generative adversarial networks. In Finale Doshi-Velez, Jim Fackler, David C. Kale, Rajesh Ranganath, Byron C. Wallace, and Jenna Wiens, editors, *Proceedings of the Machine Learning for Health Care Conference, MLHC 2017, Boston, Massachusetts, USA, 18-19 August 2017*, Proceedings of Machine Learning Research, pages 286–305. PMLR, 2017. URL http://proceedings.mlr.press/v68/choi17a.html.

[10] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional GAN. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7333–7343, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/254ed7d2de3b23ab10936522dd547b78-Abstract.html.

[11] Brandon Theodorou, Cao Xiao, and Jimeng Sun. Synthesize extremely high-dimensional longitudinal electronic health records via hierarchical autoregressive language model. *CoRR*, abs/2304.02169, 2023. doi: 10.48550/ARXIV.2304.02169. URL https://doi.org/10.48550/arXiv.2304.02169.

[12] Yuan Zhong, Xiaochen Wang, Jiaqi Wang, Xiaokun Zhang, Yaqing Wang, Mengdi Huai, Cao Xiao, and Fenglong Ma. Synthesizing multimodal electronic health records via predictive diffusion models. In Ricardo Baeza-Yates and Francesco Bonchi, editors, *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pages 4607–4618. ACM, 2024. doi: 10.1145/3637528.3671836. URL https://doi.org/10.1145/3637528.3671836.

[13] Yijie Hao, Huan He, and Joyce C. Ho. LLMSYN: generating synthetic electronic health records without patient-level data. In Kaivalya Deshpande, Madalina Fiterau, Shalmali Joshi, Zachary C. Lipton, Rajesh Ranganath, and Iñigo Urteaga, editors, *Machine Learning for Healthcare Conference, 16-17 August 2024, Toronto, Canada*, volume 252 of *Proceedings of Machine Learning Research*, pages 1–27. PMLR, 2024. URL https://proceedings.mlr.press/v252/hao24a.html.

[14] Jiawei Ren, Cunjun Yu, Shunan Sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Balanced meta-softmax for long-tailed visual recognition. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/2ba61cc3a8f44143e1f2f13b2b729ab3-Abstract.html.

[15] Jin Li, Benjamin J. Cairns, Jingsong Li, and Tingting Zhu. Generating synthetic mixed-type longitudinal electronic health records for artificial intelligent applications. *npj Digit. Medicine*, 6, 2023. doi: 10.1038/S41746-023-00834-7. URL https://doi.org/10.1038/s41746-023-00834-7.

[16] Amirsina Torfi and Edward A. Fox. Corgan: Correlation-capturing convolutional generative adversarial networks for generating synthetic healthcare records. In Roman Barták and Eric Bell, editors, *Proceedings of the Thirty-Third International Florida Artificial Intelligence Research Society Conference, Originally to be held in North Miami Beach, Florida, USA, May 17-20, 2020*, pages 335–340. AAAI Press, 2020. URL https://aaai.org/ocs/index.php/FLAIRS/FLAIRS20/paper/view/18458.

[17] Ziqi Zhang, Chao Yan, Thomas A. Lasko, Jimeng Sun, and Bradley A. Malin. SynTEG: a framework for temporal structured electronic health data simulation. *Journal of the American Medical Informatics Association*, 28(3):596–604, 2021. doi: 10.1093/jamia/ocaa262. URL https://doi.org/10.1093/jamia/ocaa262.

[18] Hojjat Karami, David Atienza, and Anisoara Ionescu. Synehrgy: Synthesizing mixed-type structured electronic health records using decoder-only transformers. *CoRR*, abs/2411.13428, 2024. doi: 10.48550/ARXIV.2411.13428. URL https://doi.org/10.48550/arXiv.2411.13428.

[19] Chang Lu, Chandan K. Reddy, Ping Wang, Dong Nie, and Yue Ning. Multi-label clinical time-series generation via conditional GAN. *IEEE Transactions on Knowledge and Data Engineering*, 2023. doi: 10.1109/TKDE.2023.3310909. Early access.

[20] Guanglin Zhou and Sebastiano Barbieri. Generating clinically realistic EHR data via a hierarchy- and semantics-guided transformer. In *ECAI 2025 – 28th European Conference on Artificial Intelligence*, volume 413 of *Frontiers in Artificial Intelligence and Applications*, pages 3944–3951. IOS Press, 2025.

[21] Hongyi Yuan, Songchi Zhou, and Sheng Yu. EHRDiff: Exploring realistic EHR synthesis with diffusion models. *Transactions on Machine Learning Research*, 2024.

[22] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res.*, 16:321–357, 2002. doi: 10.1613/JAIR.953. URL https://doi.org/10.1613/jair.953.

[23] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2999–3007. IEEE Computer Society, 2017. doi: 10.1109/ICCV.2017.324. URL https://doi.org/10.1109/ICCV.2017.324.

[24] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge J. Belongie. Class-balanced loss based on effective number of samples. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 9268–9277. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00949. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Cui_Class-Balanced_Loss_Based_on_Effective_Number_of_Samples_CVPR_2019_paper.html.

[25] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Aréchiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 1565–1576, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/621461af90cadfdaf0e8d4cc25129f91-Abstract.html.

[26] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=r1gRTCVFvB.

[27] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=37nvvqkCo5.

[28] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence. *Dataset Shift in Machine Learning*. MIT Press, 2009. URL https://api.semanticscholar.org/CorpusID:61294087.

[29] Zachary C. Lipton, Yu-Xiang Wang, and Alexander J. Smola. Detecting and correcting for label shift with black box predictors. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, Proceedings of Machine Learning Research, pages 3128–3136. PMLR, 2018. URL http://proceedings.mlr.press/v80/lipton18a.html.

[30] Saurabh Garg, Yifan Wu, Sivaraman Balakrishnan, and Zachary C. Lipton. A unified view of label shift estimation. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/219e052492f4008818b8adb6366c7ed6-Abstract.html.

[31] Xiangjian Jiang, Nikola Simidjievski, and Mateja Jamnik. TabStruct: Measuring structural fidelity of tabular data. *CoRR*, abs/2509.11950, 2025. doi: 10.48550/ARXIV.2509.11950. URL https://doi.org/10.48550/arXiv.2509.11950.

[32] Enhao Zhang, Chuanxing Geng, Chaohua Li, and Songcan Chen. Dynamic learnable logit adjustment for long-tailed visual recognition. *IEEE Trans. Cir. and Sys. for Video Technol.*, 34(9):7986–7997, September 2024. ISSN 1051-8215. doi: 10.1109/TCSVT.2024.3383962. URL https://doi.org/10.1109/TCSVT.2024.3383962. Issue date September 2024.

[33] Haofeng Zhang, Philip H.S. Torr, Dubing Chen, and Yuming Shen. Zero-shot logit adjustment, 2022. URL https://arxiv.org/abs/2204.11822.

[34] Lénaïc Chizat, Edouard Oyallon, and Francis R. Bach. On lazy training in differentiable programming. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 2933–2943, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/ae614c557843b1df326cb29c57225459-Abstract.html.

[35] Alistair E. W. Johnson, Tom J. Pollard, Lu Shen, Li-wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3(1):160035, 2016. doi: 10.1038/sdata.2016.35. URL https://doi.org/10.1038/sdata.2016.35.

[36] Alistair E. W. Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J. Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, Li-wei H. Lehman, Leo A. Celi, and Roger G. Mark. MIMIC-IV, a freely accessible electronic health record dataset. *Scientific Data*, 10(1):1, 2023. doi: 10.1038/s41597-022-01899-x. URL https://doi.org/10.1038/s41597-022-01899-x.

[37] Hrayr Harutyunyan, Hrant Khachatrian, David C. Kale, Greg Ver Steeg, and Aram Galstyan. Multitask learning and benchmarking with clinical time series data. *Scientific Data*, 6:96, 2019. doi: 10.1038/s41597-019-0103-9.

[38] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F. Stewart, and Jimeng Sun. Doctor AI: Predicting clinical events via recurrent neural networks. In *Proceedings of the 1st Machine Learning for Healthcare Conference*, volume 56 of *Proceedings of Machine Learning Research*, pages 301–318. PMLR, 2016. URL http://proceedings.mlr.press/v56/Choi16.html.

[39] Joseph Enguehard, Dan Busbridge, Adam Bozson, Claire Woodcock, and Nils Hammerla. Neural temporal point processes for modelling electronic health records. In Emily Alsentzer, Matthew B. A. McDermott, Fabian Falck, Suproteem K. Sarkar, Subhrajit Roy, and Stephanie L. Hyland, editors, *Machine Learning for Health Workshop, ML4H@NeurIPS 2020, Virtual Event, 11 December 2020*, Proceedings of Machine Learning Research, pages 85–113. PMLR, 2020. URL http://proceedings.mlr.press/v136/enguehard20a.html.

[40] Feng Xie, Han Yuan, Yilin Ning, Marcus Eng Hock Ong, Mengling Feng, Wynne Hsu, Bibhas Chakraborty, and Nan Liu. Deep learning for temporal data representation in electronic health records: A systematic review of challenges and methodologies. *J. Biomed. Informatics*, 126: 103980, 2022. doi: 10.1016/J.JBI.2021.103980. URL https://doi.org/10.1016/j.jbi.2021.103980.

[41] Tong Wu, Qingqiu Huang, Ziwei Liu, Yu Wang, and Dahua Lin. Distribution-balanced loss for multi-label classification in long-tailed datasets. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part IV*, Lecture Notes in Computer Science, pages 162–178. Springer, 2020. doi: 10.1007/978-3-030-58548-8_10. URL https://doi.org/10.1007/978-3-030-58548-8_10.

[42] Dekun Lin. Probability guided loss for long-tailed multi-label image classification. In Brian Williams, Yiling Chen, and Jennifer Neville, editors, *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 1577–1585. AAAI Press, 2023. doi: 10.1609/AAAI.V37I2.25244. URL https://doi.org/10.1609/aaai.v37i2.25244.

[43] Naoya Hasegawa and Issei Sato. Multiplicative logit adjustment approximates neural-collapse-aware decision boundary adjustment. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=II81zQUS1x.

[44] Yue Cheng, Jiajun Zhang, Xiaohui Gao, Weiwei Xing, and Zhanxing Zhu. Learning dynamics of logits debiasing for long-tailed semi-supervised learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=e15SYMcsTs.

[45] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 41–48, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585161. doi: 10.1145/1553374.1553380. URL https://doi.org/10.1145/1553374.1553380.

[46] Kamyar Azizzadenesheli, Anqi Liu, Fanny Yang, and Animashree Anandkumar. Regularized learning for domain adaptation under label shifts. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=rJl0r3R9KX.

[47] Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. Domain adaptation under target and conditional shift. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, JMLR Workshop and Conference Proceedings, pages 819–827. JMLR.org, 2013. URL http://proceedings.mlr.press/v28/zhang13d.html.

[48] Huan He, Owen Queen, Teddy Koker, Consuelo Cuevas, Theodoros Tsiligkaridis, and Marinka Zitnik. Domain adaptation for time series under feature and label shifts. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, Proceedings of Machine Learning Research, pages 12746–12774. PMLR, 2023. URL https://proceedings.mlr.press/v202/he23b.html.

[49] Cristóbal Esteban, Stephanie L. Hyland, and Gunnar Rätsch. Real-valued (medical) time series generation with recurrent conditional GANs. *arXiv preprint arXiv:1706.02633*, 2017.

[50] Tom J. Pollard, Alistair E. W. Johnson, Jesse D. Raffa, Leo A. Celi, Roger G. Mark, and Omar Badawi. The eICU collaborative research database, a freely available multi-center database for critical care research. *Scientific Data*, 5:180178, 2018. doi: 10.1038/sdata.2018.178.

[51] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017. doi: 10.1109/SP.2017.41.

[52] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 1322–1333. ACM, 2015. doi: 10.1145/2810103.2813677.

A Extended Related Work

A.1 Generative Modeling for Electronic Health Records

Synthetic EHR generation builds on a broader clinical sequence modeling literature, where recurrent and temporal point-process models show that diagnoses, procedures, medications, and other clinical events cannot be treated as independent observations [38–40]. This temporal view is especially important for longitudinal synthesis: a realistic generator should preserve visit-level ordering, delayed comorbidities, and clinically meaningful code co-occurrences, rather than only matching aggregate code counts.

Synthetic EHR generation has progressed from GAN-, VAE-, and correlation-aware models for discrete or tabular medical records [9, 10, 16] to temporal structured simulators and longitudinal mixed-type generators [15, 17, 18]. Hierarchical autoregressive frameworks further improve high-dimensional visit modeling by directly modeling long-span patient trajectories [11]. Recent systems also study privacy-preserving synthesis, multimodal diffusion, and LLM-based generation without direct patient-level access [8, 12, 13], improving temporal dependency modeling and sample realism.

Despite this progress, most existing generators are still trained to fit the overall distribution of the majority of diseases. In long-tailed EHR data, rare disease codes contribute sparse and unstable learning signals for their temporal, contextual, and co-occurrence dependencies. As a result, a generator may improve average realism while underrepresenting rare-event plausibility and rare-code context consistency. AdaPCLA follows the motivation stated in the main text: it gives tail codes stronger optimization support through a temporary prior scaffold, while preserving longitudinal dependency modeling and enabling bias-free inference after annealing.

A.2 Long-Tailed Learning in EHR

Long-tailed learning methods reshape training under imbalanced labels, with most canonical methods developed for discriminative classification. Resampling methods rebalance the empirical distribution by constructing or selecting minority examples [22]. Loss-level methods, including focal loss [23] and class-balanced loss [24], change the gradient contribution of different labels. Margin-based objectives enlarge rare-class margins [25], decoupled training separates representation learning from classifier calibration [26], and logit adjustment adds label-frequency terms to logits to obtain a prior-calibrated decision boundary [14, 27]. Later variants further study dynamic or learnable adjustments [32].

These methods are effective for classification, but direct use in longitudinal multi-label EHR generation is less straightforward. Resampling can disrupt patient trajectories, fixed reweighting can destabilize sparse multi-label optimization, and static prior biases can leave the generator dependent on external correction at inference time. Multi-label long-tail methods [41, 42] further show that label co-occurrence and label-frequency imbalance interact, which is precisely the setting faced by longitudinal EHR synthesis.

Recent studies also analyze long-tailed learning through learning dynamics and optimization trajectories. Neural-collapse-based analyses study how imbalanced training affects decision boundaries [43], while probing-based methods use training dynamics to guide data pruning or bias mitigation [44]. However, these interventions can rely on heuristic data dropping, which may discard informative tail samples, or maintain a permanent dependency on external priors.

In contrast, AdaPCLA uses a curriculum-learning strategy [45]. By treating the statistical prior as a temporary optimization scaffold and annealing it over time, AdaPCLA strengthens rare-code learning signals while encouraging the backbone to encode prior information into its own logits. This improves tail fidelity without discarding patient trajectories or requiring a fixed inference-time bias.

A.3 Distribution Shift and Zero-Shot Adaptation

Clinical code frequencies vary across hospitals, populations, and regions due to dataset shift [28]. Standard approaches for handling such shifts typically rely on domain adaptation or fine-tuning, which require target-domain data access and additional retraining. This motivates generation meth-

ods that can adapt to target priors without retraining while preserving longitudinal dependencies and co-occurrence structure.

One relevant setting is label shift or target-prior shift, where label prevalences change while the conditional structure is assumed to be comparatively stable. Classical and modern work estimates or corrects such shifts using black-box predictors, regularized target-prior estimation, or unified label-shift estimators [29, 30, 46]. Related domain-adaptation work studies target and conditional shift [47], and recent time-series adaptation considers feature and label shifts in sequential data [48]. These studies motivate a label-wise prior-shift view for clinical synthesis: when target code occurrence frequencies change, a generator should adjust label-wise occurrence probabilities while retaining useful conditional dependencies.

Controllable tabular and EHR synthesis provides a related perspective. Conditional tabular generators [10] and synthetic EHR frameworks [8, 13] can condition generation on available attributes or external summaries. However, they are not primarily designed for zero-shot control of long-tailed longitudinal EHR generators through target-prior correction over thousands of sparse medical codes. Recent work on structural fidelity further emphasizes that preserving causal or conditional dependencies is different from matching marginal distributions [31]. This distinction is important for cross-cohort generation, where occurrence frequencies may shift while clinically meaningful co-occurrence structure should remain reusable.

AdaPCLA follows this direction by separating the learned longitudinal structure from marginal occurrence frequency. After the source prior is absorbed during training, the target-prior correction in Sec. 4.4 provides a lightweight mechanism for zero-shot adaptation to target code frequencies: the generator steers label-wise occurrence probabilities toward the target population without retraining the backbone. The intended benefit is controlled occurrence frequency adaptation while reusing the longitudinal and co-occurrence structure learned from source EHR data.

B Evaluation Metrics and Definitions

AdaPCLA is evaluated with metrics that combine standard distributional-fidelity principles [8, 9, 49] with tail-specific quantitities defined by our evaluator. We do not present the tail-specific metrics as established named metrics from prior work.

Let R be the set of real training visits, S the set of generated visits, $C(v)$ the deduplicated set of codes of visit v, and $\mathcal{T}$ the tail set computed on the training split only. Let $\mathcal{P}_D$ denote all unordered intra-visit code pairs occurring in dataset D, and $\mathrm{supp}(\mathcal{P}_R)$ the pair support of the real data. For a tail code c, let $q_R(\cdot \mid c)$ and $q_S(\cdot \mid c)$ be the empirical distributions of codes co-occurring with c in real and generated visits, and let $\mathrm{TopK}_D(c)$ be the K most frequent co-occurring codes of c in D (we use $K = 20$).

- **PairSeen** (higher is better): $\frac{1}{|\mathcal{P}_S|} \sum_{p \in \mathcal{P}_S} \mathbf{1}[p \in \mathrm{supp}(\mathcal{P}_R)]$, the fraction of generated intra-visit code pairs that are supported by the real corpus.
- **TailPairSeen** (higher): the same quantity restricted to generated pairs $p = (a, b)$ with $a \in \mathcal{T}$ or $b \in \mathcal{T}$.
- **TailCtxJSD** (lower): the support-weighted mean of $\mathrm{JSD}\big(q_R(\cdot \mid c) \,\|\, q_S(\cdot \mid c)\big)$ over tail codes c, where JSD is the Jensen–Shannon divergence.
- **TailTopKJac** (higher): $\frac{1}{|\mathcal{T}_{\mathrm{valid}}|} \sum_c \frac{|\mathrm{TopK}_R(c) \cap \mathrm{TopK}_S(c)|}{|\mathrm{TopK}_R(c) \cup \mathrm{TopK}_S(c)|}$, the Jaccard overlap of the top-K context sets, averaged over tail codes that have a valid context in both sets.
- **TailSeqPairSeen** (higher): the adjacent-visit analogue of TailPairSeen. A generated transition pair (a_t, b_{t+1}) counts as seen if it also occurs across adjacent visits in the real data and contains at least one tail code.
- **SeqBigram $\mathbf{R}^2$** (higher): the coefficient of determination between the empirical frequencies of adjacent-visit code pairs in the real and generated corpora, computed over pairs present in the real corpus; it measures how well the generator reproduces *transition* correlation rather than marginal frequencies.
- **Transition JSD** (lower): the Jensen–Shannon divergence between the real and generated adjacent-visit pair distributions.

- **Target-prior JSD** (lower): the Jensen–Shannon divergence between the generated and the target marginal code-frequency distributions, used only for the adaptation experiments.

Conventions. All tail-code sets are derived from the training split, so no test information leaks into the definition of the tail. **N/A** means that the metric is undefined because its denominator or the required tail context is absent; it is *not* a numerical zero and is never counted as a zero in averages. The two cases are kept apart explicitly: a model with no valid tail context receives TailTopKJac $= 0$ from the evaluator (a computable but empty overlap), whereas TailPairSeen and TailCtxJSD are reported as N/A for such a model, because their denominators do not exist. This is why the Static-PCLA (Train-only) row of Table 3 mixes a numeric zero with two N/A entries. The TSTR protocol follows the Train-on-Synthetic, Test-on-Real convention of Esteban et al. [49]: a classifier is trained on synthetic data only and evaluated on the held-out real test split.

C Datasets, Hyper-Parameters, and Reproducibility

Table 4 lists the dataset statistics. Table 5 lists the training configuration of AdaPCLA.

Table 4: Dataset statistics. Splits are patient-level; the vocabulary is the number of distinct ICD diagnosis codes retained after preprocessing; the tail set is the lowest non-zero frequency 20% of codes with ties kept together.

	MIMIC-III	**MIMIC-IV**
Train / Val / Test patients	33,494 / 3,722 / 9,304	160,884 / 17,877 / 44,691
Vocabulary size $\lvert\mathcal{V}\rvert$	6,984	9,143
Tail threshold (visits)	≤ 1	≤ 2
Tail codes $\lvert\mathcal{T}\rvert$	1,471	2,125
TSTR label groups	25	25

The average number of codes per real visit is 11.05 (used in the variance analysis of Sec. 6).

Table 5: Training configuration of AdaPCLA. AdaPCLA is applied as a training objective on top of a pretrained HALO checkpoint, so the backbone configuration is inherited from Theodorou et al. [11].

Component	Setting
Initialization	HALO checkpoint of the same dataset
Optimizer	Adam
Learning rate	1×10^{-4}
Batch size	48
Training epochs	10
Positive BCE weight	1.5
Annealing horizon / schedule	30% hold ($\alpha = 1$), 40% linear anneal, 30% zero ($\alpha = 0$); see Sec. 4.3
Annealing checkpoints	100 (used for the probing analyses)
Bias clip	15
Prior smoothing ϵ	1×10^{-8}

Two levels of comparison. The main comparison (Tables 1 and 2) uses one shared patient-level split, one vocabulary and one evaluator for all methods. The task-matched supplementary reproduction in Appendix D shares the evaluator, vocabulary and tail-code definition but, for MTGAN and EHRDIFF-Visit, not necessarily the same patient set, because those methods emit a different output format that must be converted before scoring. A shared evaluator does not imply a shared training cohort, and we keep the two levels visually separate.

Tail-code sets, the TSTR label groups, and the adaptation splits are fixed before any comparison and are never re-selected to favour a method. Three generated cohorts are drawn from each trained checkpoint for the main tables; the additional baselines in Appendix D are single-run snapshots, as stated there.

D Additional Baselines and Longitudinal Metrics

Additional baselines. To position AdaPCLA against recent and tail-aware generators, we also reproduced three additional methods: MTGAN [19], a conditional GAN for multi-label clinical time series; HISGT [20], a hierarchy- and semantics-guided Transformer; and a visit-level variant of EHRDIFF [21], which generates a single record vector rather than an ordered trajectory. These form a *task-matched supplementary reproduction*: they share our evaluator, vocabulary and tail-code definition, but not necessarily the same training cohort (see the protocol note in Appendix C). Table 6 reports a single saved-sample snapshot with $K = 20$.

Table 6: Supplementary comparison with recent generators on MIMIC-IV. Single saved-sample snapshot per method ($K = 20$, training-derived tail codes). HISGT is strong on overall transition correlation; AdaPCLA remains strongest on the tail metrics and on transition-distribution matching.

Method	TailPairSeen↑	TailCtxJSD↓	TailTopKJac↑	SeqBigram R^2 ↑	TSTR AUPRC↑
HALO	0.0556	0.6598	0.0206	0.3864	0.8960
HISGT	0.0500	0.6605	0.0237	**0.8756**	0.9382
MTGAN	0.0070	0.6541	0.0413	−0.2823	N/A
EHRDIFF-Visit	0.073	0.671	0.014	N/A	N/A
AdaPCLA	**0.0932**	**0.6137**	**0.0492**	0.8447	**0.9449**

EHRDIFF-Visit is reported only for visit-level metrics because the official model emits one binary record vector instead of an ordered patient trajectory; SeqBigram and the patient-level TSTR study are therefore undefined for it. MTGAN is evaluated on its converted output files, so the metrics are identical but the underlying patient sets may differ; its converted outputs do not preserve the code mapping needed to assign our 25 TSTR label groups, so its TSTR entry is N/A as well.

Longitudinal (adjacent-visit) metrics. Visit-level pairs say nothing about whether a generator respects *temporal* order, which is the central claim of a longitudinal model. Table 7 therefore reports three adjacent-visit metrics defined in Appendix B. AdaPCLA is strongest on TailSeqPairSeen and Transition JSD on both datasets, i.e. on the rare-code and transition-distribution aspects that the paper targets. HISGT is stronger on overall SeqBigram R^2 on MIMIC-IV, and we report this rather than claiming uniform dominance on every temporal statistic. The two quantities measure different things: SeqBigram R^2 is dominated by frequent transitions, whereas TailSeqPairSeen and TailCtxJSD concentrate on rare ones.

Table 7: Adjacent-visit metrics on MIMIC-III and MIMIC-IV.

Dataset	Model	SeqBigram R^2 ↑	Transition JSD↓	TailSeqPairSeen↑
MIMIC-III	HALO	0.791	0.218	0.015
	HISGT	0.062	0.416	0.005
	MTGAN	0.322	0.311	0.014
	AdaPCLA	**0.826**	**0.166**	**0.059**
MIMIC-IV	HALO	0.386	0.278	0.026
	HISGT	**0.876**	0.149	0.023
	MTGAN	−0.282	0.441	0.017
	AdaPCLA	0.845	**0.092**	**0.055**

E Target-Prior Sensitivity over λ

Section 4.4 introduces a single interpolation coefficient $\lambda \in [0, 1]$ between the uncorrected and the fully corrected logits. Table 8 reports the complete grid for both adaptation directions and both generators. λ is selected on a held-out adaptation split of the target cohort by matching the generated code frequencies to the target frequencies; the target test labels are never used. The important row is $\lambda = 0$: AdaPCLA without any correction already exceeds the highest HALO score in the reported grid in both directions, so the reported gains cannot be explained by the test-time correction alone.

Table 8: Target-prior sensitivity. Metric: held-out AUPRC. λ is selected on a target adaptation split by frequency matching. Bold marks the best value within each transfer/model row.

Transfer	Model	$\lambda = 0$	0.25	0.5	0.75	1
III→IV	HALO	0.906	**0.914**	0.909	0.907	0.907
	AdaPCLA	0.931	0.930	**0.935**	0.932	0.930
IV→III	HALO	0.822	0.834	**0.835**	0.832	0.831
	AdaPCLA	0.927	0.930	0.933	**0.934**	0.930

The curve is flat: for AdaPCLA the spread across the whole grid is at most 0.007 AUPRC, so the conclusion does not hinge on a finely tuned correction strength.

Provenance. This grid was produced by a dedicated sweep in which only λ was varied, and it is a separate evaluation from the main-protocol numbers reported in Table 3(b). The two therefore have different absolute levels — for instance AdaPCLA reaches AUPRC 0.922 in the MIV→MIII row of Table 3(b) and 0.933 at $\lambda = 0.5$ here. The two evaluations follow different protocols, and we do not attribute the gap to any single factor. We use this table only for the *relative* comparison across λ and between the two generators, both of which are computed within the same sweep; the main-text table remains the reference for absolute performance. The test-grid maxima are descriptive sensitivity results, not the criterion used to select λ.

F Multi-Center Sanity Check on eICU

The experiments in the main text adapt across MIMIC versions, which share a hospital system and overlapping collection periods. To probe behaviour beyond that setting, we ran a preliminary check with MIMIC-IV as the source and the **eICU Collaborative Research Database** [50] as the target, mapped onto the common diagnosis-code space. This is a single-seed, 10k-sample, diagnosis-code-only study. It is *not* a full TSTR evaluation, and we do not present it as evidence of proven multi-center generalisation.

Table 9: eICU sanity check (single seed, 10k samples). Δb denotes the inference-time target-prior correction. Lower is better for both JSD columns.

Target	Model	Δb	TailPairSeen↑	TailCtxJSD↓	TailTopKJac↑	Target-prior JSD↓
eICU	HALO	no	0.076	0.647	0.029	0.436
	HALO	yes	0.171	0.631	0.033	0.073
	AdaPCLA	no	0.117	0.603	0.057	0.376
	AdaPCLA	yes	**0.220**	**0.578**	**0.078**	**0.068**

Two observations follow. First, the target-prior correction helps both generators substantially on the distributional metric (Target-prior JSD drops from 0.436 to 0.073 for HALO and from 0.376 to 0.068 for AdaPCLA), which is what the label-wise construction predicts. Second, even without any correction AdaPCLA already has better tail support than HALO, and the correction stacks on top of it. The transition-level companion numbers of this run are reported together with the adjacent-visit metrics in Appendix D.

G Privacy Diagnostics

Because AdaPCLA deliberately strengthens rare-code fidelity, it is fair to ask whether it also increases memorisation or attribute disclosure. We therefore ran three standard empirical diagnostics, following the membership-inference protocol of Shokri et al. [51], the attribute-inference setting of Fredrikson et al. [52], and the duplicate/nearest-neighbour checks commonly used in synthetic-EHR audits [8]: (i) *MIA AUC*, the AUC of a membership-inference attack in the sense of Shokri et al. [51], where 0.5 corresponds to an attacker that cannot separate members from non-members; (ii) *exact* and *near* duplicate rates, where near-duplicates are patient-level code sets with Jaccard distance below 0.05 over 3,000 queries; (iii) *attribute-inference recovery*, the fraction of held-out attributes recovered by an attacker who sees the generated record.

Table 10: Privacy diagnostics on MIMIC-IV. Higher AIA recovery and higher duplicate rates mean higher risk. MIA AUC is read against the chance level 0.5: values below 0.5 indicate an attack that performs no better than chance, not a monotone "safety score".

Method	MIA AUC	Exact duplicate	Near duplicate	AIA recovery↑ (risk)
AdaPCLA	0.518	0.019	0.196	**0.113**
HISGT	0.521	0.024	0.476	0.084
MTGAN	**0.497**	**0.000**	**0.000**	0.061

The results are mixed and we report them as such. AdaPCLA's membership-inference AUC is close to chance (0.518) and it produces far fewer near-duplicates than HISGT (0.196 versus 0.476); under these two diagnostics we see no evidence of direct record memorisation. Its attribute-inference recovery is, however, the *highest* of the three methods (0.113). This is consistent with a trade-off in which modelling rare codes more faithfully also makes their conditional structure more recoverable, but our measurements do not establish that causal link. We therefore do not claim that AdaPCLA is privacy-preserving in any formal sense. These are attack-based empirical diagnostics under one choice of attack, distance and data representation; they are not differential-privacy guarantees, they cannot rule out memorisation, and a deployment would need a dedicated privacy mechanism on top of the generator.

H Annealing-Speed Proxy

Theorem 5.1 bounds the distance between the trained logits and a population-level minimizer path $z_\tau^\star$ that is not observable from finite data. The theorem error is therefore not directly measurable. We instead report the closest measurable proxy on 100 fixed MIMIC-IV contexts,

$$E_g(t) = \mathbb{E}_x \frac{\left\| \left[z_t(x) + \alpha_t b_{\text{stat}} - z_{\mathcal{T}}(x) \right]_g \right\|_2}{\sqrt{|g|}},$$

where g indexes a code group (all codes, or tail codes only) and $z_{\mathcal{T}}$ is the final checkpoint used as reference. $E_g(t)$ measures how far an intermediate checkpoint still is from the endpoint, so a schedule that removes the scaffold earlier is expected to reach a small value earlier.

The three rows come from a separate schedule sweep in which the length of the annealing phase was varied; the entries in the second column are *training-log indices* of that sweep, not epochs of the 10-epoch main runs of Table 5, so they are not comparable with the main training budget.

Table 11: Annealing-speed proxy. E_g is evaluated at the step where the bias is fully removed and at the start of annealing.

Schedule	Removal	All-code E (start→end)	Tail-code E (start→end)	Reduction
fast	50	2.0194 → 0.00202	2.5487 → 8.8e-6	99.90%
base	70	2.0194 → 0.00191	2.5487 → 7.9e-6	99.91%
slow	90	2.0194 → 0.00182	2.5487 → 5.3e-6	99.91%

What this does and does not show. It shows that, for every schedule, the intermediate logits are numerically very close to the final logits once the scaffold has been removed, both for all codes and, by three orders of magnitude, for tail codes; the reduction is above 99.9% in every case. It does *not* verify the annealing-speed dependence of Theorem 5.1. The three schedules are numerically close in this measurement (99.90% versus 99.91%), and the ordering is not even monotone in the annealing speed. Because E_g is referenced to the final checkpoint, it also cannot separate the effect of the schedule from the effect of the endpoint itself. Isolating $\sup_s |\dot{\alpha}(s)|$ empirically would require estimating the moving optimum $z_\tau^\star$, which we leave as an open problem; the bound should be read as a design guide for choosing a smooth schedule, not as a claim that our experiments confirm its speed dependence.

I Complete Model Analysis and Additional Main-Text Analyses

I.1 Model Analysis

In this section, we conduct a deep-dive analysis to examine the core theoretical claims of AdaPCLA: (1) the internalization of prior knowledge, (2) the translation of tail fidelity into downstream utility, and (3) the disentanglement of structure and occurrence frequency for zero-shot control.

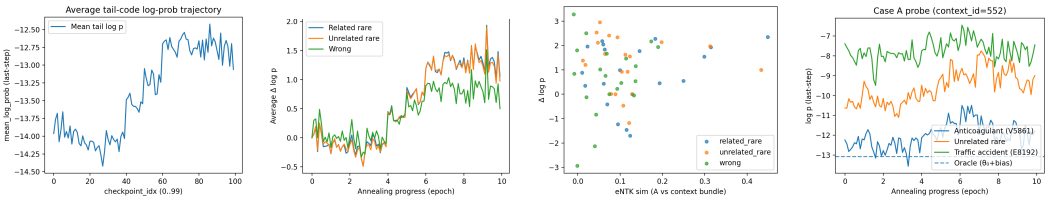
I.1.1 Prior-Internalization Ablation

To examine whether the prior bias is absorbed into the model or remains an external correction, we compare dynamic annealing against static-bias variants in Table 3. This analysis follows the curriculum construction in Sec. 4.3 and focuses on whether the backbone can preserve rare-event structure during bias-free inference.

Effect of Removing a Fixed Bias The ablation shows a clear difference between using a fixed bias and annealing it. The *Static-PCLA (Train-only)* variant applies the fixed bias during training but removes it at inference. After this bias is removed, TailPairSeen and TailCtxJSD cannot be computed, and TailTopKJac drops to 0.0000. This negative control indicates that a fixed training bias alone does not make the backbone encode rare-event structure into its own logits. Instead, the model remains dependent on the fixed correction and fails to preserve tail plausibility once the correction is removed at inference time.

Evidence of Prior Internalization AdaPCLA does not use an external bias at inference, but still achieves the best tail metrics in this ablation. It obtains the highest TailPairSeen **0.1239** and TailTopKJac **0.0356**, and the lowest TailCtxJSD **0.5940**. This result is consistent with the gradient dynamics analyzed in Sec. 4.3: the prior scaffold changes BCE residuals during training, and the resulting updates are written into the intrinsic logits through the shared Transformer backbone. It suggests that dynamic annealing helps the backbone learn useful prior information in its own logits, instead of depending on a fixed external correction during sampling.

Probability Dynamics during Annealing Beyond the ablation table, we further examine how selected code probabilities change as the annealing coefficient $\alpha(\tau)$ decays. Figure 3 reports four complementary views, from global tail statistics to a single clinical context. As Fig. 3(a) shows, the average last-step log-probability of a broad set of tail codes steadily increases across checkpoints, indicating that the curriculum increases tail-code probabilities as a whole. On the micro-probing set, Fig. 3(b) shows systematic differences across probe types (related_rare / unrelated_rare / wrong), while Fig. 3(c) is consistent with the label-wise empirical NTK analysis in Sec. 4.3: probability gains $\Delta \log p$ grow with the initial gradient similarity between a disease and its context bundle. Finally, Fig. 3(d) evaluates the context of Case A (cardiovascular patient, Sec. I.2). *Anticoagulant* (V5861), the clinically correct tail medication for Atrial Fibrillation, rises and converges to the Oracle line, whereas *Traffic accident* (E8192), the hallucination produced by HALO, remains at the bottom. This is consistent with the Case Study: AdaPCLA later generates Anticoagulant because its probability is selectively elevated during annealing. Additional robustness checks for micro-probing are provided in Appendix Q.



(a) Global tail trend: average last-step log-probability of a broad set of tail codes over 100 checkpoints.

(b) Micro-probe dynamics: average $\Delta \log p$ over 100 probing contexts for related, unrelated, and wrong probes.

(c) eNTK Gradient Bridge: initial eNTK similarity vs. $\Delta \log p$; higher similarity implies larger probability gains.

(d) Case A probe: Anticoagulant (V5861) vs. Traffic accident (E8192) in the cardiovascular context; see Sec. I.2.

Figure 3: Dynamics of curriculum prior internalization. (a) Global average tail log-probability increases across checkpoints. (b) Micro-probe probability trajectories highlight type-dependent gains across related, unrelated, and wrong probes. (c) eNTK similarity correlates with per-item $\Delta \log p$, consistent with the gradient-bridge mechanism. (d) Single-context probe for Case A: Anticoagulant rises to the Oracle target, while Traffic accident remains low.

I.1.2 Annealing Schedule Effects on Related-Rare Probes

We compare different annealing schedules by analyzing probability trajectories on tail-relevant probe codes. We do not use a global logit-difference norm such as $\|z_{\mathcal{T}} - z_0 - b_{\text{stat}}\|$ as the main empirical measure. First, the fixed scaffold b_{stat} has a much larger norm than the logit changes induced by training, making schedule-dependent changes difficult to isolate. Second, the norm aggregates over the entire vocabulary and does not focus on rare-code dimensions that are relevant to a fixed patient-history context.

To measure these schedule effects, we use related_rare gain and AUC summaries based on micro-probe trajectories. Following the micro-probe setting in the main text, we focus on pairs (x, c) labeled as related_rare, where c is a tail code clinically related to the patient-history context x. For each pair, we record the last-step log-probability trajectory $\{\log p_t\}_{t=0}^{T}$ over 100 micro-probe checkpoints and define three summaries:

- **Final gain**: $\Delta \log p = \log p_T - \log p_0$, measuring the net log-probability change at the end of annealing for each related_rare code.
- **AUC**:
$$\text{AUC} = \sum_{t=1}^{T} \max\big(0,\, \log p_t - \log p_0\big),$$
which sums the positive log-probability changes along the annealing trajectory.
- **Normalized AUC**:
$$\text{AUC}_{\text{norm}} = \frac{\text{AUC}}{(T-1)\, \max_t \max\big(0,\, \log p_t - \log p_0\big)},$$
which rescales AUC to approximately $[0, 1]$ and summarizes the distribution of log-probability gains across checkpoints relative to the final checkpoint.

These summaries depend only on increments of $\log p_t$ and do not explicitly involve the global bias b_{stat}. They measure the magnitude and consistency of log-probability increases for tail-relevant related_rare codes along the annealing trajectory.

On the related_rare micro-probes, we compute final gain, AUC, and normalized AUC for four annealing schedules: **slow**, **base**, **cosine**, and **fast**. For final gain $\Delta \log p$, the mean values follow
$$\text{base} \;<\; \text{cosine} \;<\; \text{fast} \;<\; \text{slow},$$
but the differences are small compared to the standard deviations (gain_std ≈ 1.0). We therefore do *not* interpret final gain alone as providing a robust ordering across schedules.

AUC and normalized AUC show a clearer ordering. For both raw AUC and normalized AUC, we observe
$$\text{slow} \;\ll\; \text{base} \approx \text{cosine} \;<\; \text{fast},$$

where the **fast** schedule has the largest related_rare AUC, while the **slow** schedule has the smallest AUC. The normalized AUC values follow the same ordering, indicating that fast annealing increases related_rare log-probabilities earlier and more steadily than slow annealing. We therefore use AUC and normalized AUC as the main schedule-dependent summaries in our study of annealing schedules, and use final gain as a supporting quantity.

Table 12: Schedule-dependent probability changes on related_rare micro-probes. We report the mean (and standard deviation) of final gain $\Delta \log p$, AUC, and normalized AUC over all related_rare pairs. Larger AUC / $\mathrm{AUC}_{\mathrm{norm}}$ indicate larger and more sustained log-probability gains for tail-relevant codes.

Schedule	$\Delta \log p$	AUC	$\mathrm{AUC}_{\mathrm{norm}}$
slow	1.04 ($\pm$1.09)	44.5 ($\pm$32.2)	0.195 ($\pm$0.101)
base	0.99 ($\pm$1.06)	68.3 ($\pm$44.0)	0.290 ($\pm$0.141)
cosine	1.02 ($\pm$1.05)	71.4 ($\pm$46.5)	0.289 ($\pm$0.149)
fast	1.04 ($\pm$1.05)	101.4 ($\pm$59.6)	0.401 ($\pm$0.178)

These related_rare gain and AUC summaries should not be interpreted as direct numerical estimates of the theoretical prior-internalization error $\|e_\tau(x)\|_{\mathcal{K}^{-1}}$. They summarize log-probability changes for selected related_rare codes, rather than reconstructing the label-wise empirical NTK flow or the moving target $z^\star_\tau(x)$. Accordingly, this experiment provides an empirical analysis of schedule-dependent probability trajectories on tail-relevant codes.

Taken together, the results show that annealing speed affects the timing and magnitude of related_rare probability increases during training. Fast annealing produces earlier and more sustained log-probability gains, whereas slow annealing produces later and weaker accumulated gains. We use these findings as the schedule-dependent mechanism analysis, not as a strict numerical validation of the prior-internalization bound.

I.2 Qualitative Analysis: Clinical Case Study

To move beyond aggregate statistics, we examine the **Clinical Coherence** of synthetic patient trajectories. We present two complementary case studies on MIMIC-IV to illustrate how AdaPCLA captures complex dependencies that baselines miss.

Case A: Longitudinal Trajectory Consistency. We select a real patient with a complex history of *Cardiovascular and Metabolic disorders*. As detailed in Table 13, given the context of Visits 1–2 (containing Hyperlipidemia, Atrial Fibrillation, and Hypertension), we compare the generated third visit across models.

- **Ground Truth Logic:** The patient's condition progresses logically: chronic cardiovascular issues persist, requiring continued medication (Anticoagulants) and monitoring.
- **Baseline Failure (Hallucination):** The HALO baseline captures generic codes (Hypertension) but suffers from severe **semantic hallucination**. It predicts *"Traffic accident (E8192)"*, a completely irrelevant external cause disconnected from the patient's medical history. This typifies the "Gradient Domination" failure: without the scaffold, the model's attention is scattered by noise. LSTM and GPT similarly fail to recall the specific comorbidity structure (e.g., predicting unrelated HIV or Pneumonia codes).
- **AdaPCLA Success (Coherence):** In contrast, AdaPCLA recovers the treatment context of this patient. It predicts *"Long-term use of anticoagulants (V5861)"*, a code tightly tied to the history of *Atrial Fibrillation (42731)*, and *"Chest pain"*, a plausible symptom given the cardiac history. This is consistent with AdaPCLA having encoded the co-occurrence between atrial fibrillation and long-term anticoagulant use, a dependency that marginal frequencies alone would not favour. The example documents code-level agreement on one selected patient; it is not evidence of causal or treatment-level understanding.

Case B: Tail Co-occurrence Topology In Table 14 (Appendix), we investigate the local neighborhood of a specific tail code: *Squamous cell carcinoma of skin (17322)*. We examine the top-10 co-occurring codes generated by each model.

- **Manifold Reconstruction:** AdaPCLA successfully reconstructs the correct disease manifold. It identifies 5 out of the top 10 true comorbidities, including specific procedures like *Aortocoronary bypass (V4581)* and chronic conditions like *Hyperlipidemia*.
- **Baseline Scattering:** HALO's predictions are scattered and generic (e.g., Dizziness, Chronic pain), lacking disease-specific correlation. LSTM and GPT fail to generate any meaningful co-occurrences for this rare code, indicating **Mode Collapse** in the tail region.

These two cases show, on two selected patients, that the quantitative gaps of Table 1 are visible at the level of individual codes and co-occurrence neighborhoods. They are illustrative examples rather than a systematic clinical validation, and we draw no causal or treatment-level conclusion from them.

Table 13: **Qualitative Comparison on MIMIC-IV (Case A) — Full version.** Generated Visit 3 codes conditioned on the same patient history (Real Visit 1–2 above). **Teal** = correct; **Orange** = rare/tail; **Red** = hallucination.

Model / Source	**Medical Codes (Visit)**
Real Visit 1 *(Context)*	Hyperlipidemia (2724), AV block (42613), Anemia (2851), Thoracic aortic aneurysm (4412), Atrial fib (42731), Fluid overload (27669), Chr pulmon heart (4168), Dvrtclo colon (56210), Valvular dis (3963/3970), Hypertension (4019).
Real Visit 2 *(Context)*	Post-proc (V4589), Hyperlipidemia (2724), Atrial fib (42731), Dvrtclo colon (56210), Hyposmolality (2761), Hypertension (4019).
Ground Truth *(Real Visit 3)*	**Core:** Hypertension (4019), Hyperlipidemia (2724). **Target tail structure:** Atrial fib (42731), Long-term anticoagul (V5861), Acute cholecystitis (5750), Dvrtclo colon (56210).
AdaPCLA *(Ours)*	✓ Hypertension (4019), ✓ Hyperlipidemia (2724) *Captured logic:* ✓ **Atrial fib (42731)**, ✓ **Long-term anticoagul (V5861)** *Others:* ✗ Chest pain (78659); *(Missed: 5750, 56210)*
HALO *(Baseline)*	✓ Hypertension (4019), ✓ Hyperlipidemia (2724), ✓ Dvrtclo colon (56210) *Hallucinations:* ✗ Traffic acc motcycl (E8192), ✗ Esophageal reflux (53081) *(Missed: Atrial fib, Anticoagulants, Cholecystitis)*
LSTM *(Baseline)*	✗ Lumbar disc (72210), ✗ HIV status (V08), ✗ Pneumonia (486), ✗ Sleep apnea (78057) *(Missed: Failed to capture any meaningful patient-specific codes.)*
GPT-style *(Baseline)*	✓ Hypertension (4019); ✗ CHF (4280), ✗ Juv osteochondrosis (7324), ✗ Hx brain injury (V1552) *Analysis:* Generic prediction mixed with unrelated codes.

Table 14: **Case B: Top-10 co-occurring codes — Full version.** Tail code: Squamous cell carcinoma of skin (17322). LSTM and GPT omitted (no tail co-occurrence). **Teal** = overlap Real & AdaPCLA; **Red** = HALO scattered; **Gray** = none.

Rank	Real	HALO	AdaPCLA
1	Atrial fibrillation (42731)	✗ Dizziness and giddiness (7804)	✓ Cor ath unsp vsl ntv/gft (41400)
2	✓ Aortocoronary bypass (V4581)	✗ Chronic pain NEC (33829)	Syncope and collapse (7802)
3	Gout NOS (2749)	✗ Somatization disorder (30081)	Trnspl status-pancreas (V4283)
4	Joint replaced knee (V4365)	–	✓ Aortocoronary bypass (V4581)
5	✓ Hyperlipidemia NEC/NOS (2724)	–	✓ Hypertension NOS (4019)
6	✓ Long-term use anticoagul (V5861)	–	DMI wo cmp nt st uncntrl (25001)
7	✓ Cor ath unsp vsl ntv/gft (41400)	–	Kidney transplant status (V420)
8	✓ Hypertension NOS (4019)	–	✓ Hyperlipidemia NEC/NOS (2724)
9	Polycystic kidney NOS (75312)	–	✓ Long-term use anticoagul (V5861)
10	Crnry athrscl natve vssl (41401)	–	–

I.3 Prediction Reliability and Internalization Consistency

Beyond transferability and controllability, we investigate two complementary aspects: (1) the reliability of predicted probabilities, especially in the long tail; and (2) the consistency between observed curriculum dynamics and the theory in Sec. 4.3.

Prediction Confidence Reliability We report retained **ECE**, **Brier Score**, and **NLL** measurements for overall and tail-only labels. The stored evaluations contain different numbers of record–label pairs, so these are auxiliary descriptive diagnostics rather than a controlled same-record comparison; the values and this limitation are reported in Appendix O.

Expanded Probing Analysis We further enlarge the probing set from 30 to 100 contexts and recompute related_rare / unrelated_rare / wrong probe trajectories. Table 16 and Figure 2 show that the trajectory scale and type separation remain qualitatively consistent under the larger probe set.

Summary of Outputs The confidence metrics provide auxiliary descriptive diagnostics, while expanded probing examines curriculum dynamics. Exploratory $\lambda_{\min}(\mathcal{K})$ correlation analysis is reported in Appendix N and treated as non-conclusive evidence.

J Prior-Internalization Error Bound Under Lazy Training

We analyze the curriculum-induced logit dynamics for a fixed input context x. Recall the effective logits $\tilde{z}_\tau(x) = z_\tau(x) + \alpha(\tau) b_{\text{stat}}$, where $\alpha : [0, \mathcal{T}] \to [0, 1]$ is absolutely continuous with an essentially bounded derivative, $\alpha(0) = 1$, and $\alpha(\mathcal{T}) = 0$.

Why we use an expected-risk surrogate A pointwise BCE loss with a single hard multi-label target $y \in \{0, 1\}^V$ generally does not admit a finite minimizer in logit space, because the minimizer is attained only in the limit $z_c \to \pm\infty$. To obtain a well-posed moving optimum, we work with the *conditional expected* BCE risk at a fixed context x:

$$F_\tau(z; x) = \mathbb{E}_{Y \sim \mathbb{P}(\cdot|x)} \Big[\mathcal{L}_{\text{BCE}}\big(Y, \sigma(z + \alpha(\tau) b_{\text{stat}})\big) \Big].$$

We suppress the explicit dependence on x in what follows.

Under the lazy-training approximation, we assume that the label-wise empirical NTK remains approximately constant along the trajectory. Because we analyze the dynamics of different labels for the *same* context x, the kernel here is the label-wise coupling matrix

$$\mathcal{K}_{i,j} = \langle \nabla_\theta z(i;x), \nabla_\theta z(j;x) \rangle ,$$

rather than the usual sample–sample NTK. On the active label subspace we assume $\mathcal{K}$ is positive definite, so that $\mathcal{K}^{-1}$ is well-defined there. We write $\|v\|_{\mathcal{K}^{-1}} := \sqrt{v^\top \mathcal{K}^{-1} v}$.

The kernel-induced gradient flow in intrinsic-logit coordinates is $\dot{z}_\tau = -\mathcal{K} \nabla F_\tau(z_\tau)$.

Assumptions

- **A1 (Local strong monotonicity / local strong convexity type condition).** There exists a convex neighborhood $\mathcal{D} \subset \mathbb{R}^V$ and a constant $\mu > 0$ such that, for all $\tau \in [0, \mathcal{T}]$ and all $u, v \in \mathcal{D}$,
$$\langle u - v, \nabla F_\tau(u) - \nabla F_\tau(v) \rangle \geq \mu \|u - v\|^2.$$
This is the standard first-order formulation of strong monotonicity of ∇F_τ on $\mathcal{D}$; when F_τ is twice differentiable, it is implied by the local Hessian lower bound $\nabla^2 F_\tau(z) \succeq \mu I$ on $\mathcal{D}$.
- **A2 (Shifted-risk smoothness and schedule regularity).** Write $F_\tau(z) = H(z + \alpha(\tau) b_{\text{stat}})$, where the effective-logit risk H is twice continuously differentiable on a neighborhood containing the translated trajectories. The schedule $\alpha : [0, \mathcal{T}] \to [0, 1]$ is absolutely continuous with an essentially bounded derivative and endpoints $\alpha(0) = 1$, $\alpha(\mathcal{T}) = 0$. For the schedule of Sec. 4.3, $\dot{\alpha}(\tau) = -1/(0.4\,\mathcal{T})$ on $(0.3\mathcal{T}, 0.7\mathcal{T})$ and is zero on the two constant segments, almost everywhere. Thus $\operatorname{ess\,sup}_s |\dot{\alpha}(s)| = 1/(0.4\,\mathcal{T})$. All derivative statements below hold almost everywhere; $\sup_s |\dot{\alpha}(s)|$ denotes the essential supremum.
- **A3 (Constant, nondegenerate kernel on the active subspace).** The label-wise kernel $\mathcal{K}$ is constant and symmetric on the active label subspace, and is positive definite there with $\lambda_{\min}(\mathcal{K}) > 0$. In particular, the $\mathcal{K}^{-1}$–inner product and the induced norm $\| \cdot \|_{\mathcal{K}^{-1}}$ used below are well-defined on that subspace.

Lemma J.1 (Regularity of the moving minimizer)**.** *Let $\tilde{z}^\star$ be the fixed finite minimizer of H, and assume $z_\tau^\star = \tilde{z}^\star - \alpha(\tau) b_{\text{stat}} \in \mathcal{D}$ throughout the interval. Under A2 the path $z_\tau^\star$ is absolutely continuous, with*

$$\|\dot{z}_\tau^\star\|_{\mathcal{K}^{-1}} \leq B_\star |\dot{\alpha}(\tau)| \quad \text{for almost every } \tau, \qquad B_\star = \|b_{\text{stat}}\|_{\mathcal{K}^{-1}}. \tag{3}$$

Proof. The affine identity $z_\tau^\star = \tilde{z}^\star - \alpha(\tau) b_{\text{stat}}$ follows from the shifted risk $F_\tau(z) = H(z + \alpha(\tau) b_{\text{stat}})$. Absolute continuity is inherited from α, and $\dot{z}_\tau^\star = -\dot{\alpha}(\tau) b_{\text{stat}}$ almost everywhere. Taking the norm gives the claim. ■

Theorem J.1 (Prior-internalization error in the $\mathcal{K}^{-1}$-norm)**.** *Define the internalization error $e_\tau = z_\tau - z_\tau^\star$ and $r_\tau = \|e_\tau\|_{\mathcal{K}^{-1}}$. Assume additionally that $z_\tau, z_\tau^\star \in \mathcal{D}$ for all $\tau \in [0, \mathcal{T}]$. Under Assumptions A1–A3 together with this invariant-neighborhood condition, for all $\tau \in [0, \mathcal{T}]$,*

$$r_\tau \leq e^{-\mu \lambda_{\min}(\mathcal{K}) \tau} r_0 + \frac{B_\star}{\mu \lambda_{\min}(\mathcal{K})} \sup_{s \in [0, \mathcal{T}]} |\dot{\alpha}(s)|. \tag{4}$$

Proof We follow three steps: (i) define the internalization error and a Lyapunov function; (ii) derive a scalar differential inequality for the radius r_τ; (iii) apply Grönwall to close the bound.

Step 1: internalization error and Lyapunov function. Because $\dot{z}_\tau = -\mathcal{K} \nabla F_\tau(z_\tau)$ and $\nabla F_\tau(z_\tau^\star) = 0$, we have

$$\dot{e}_\tau = -\mathcal{K}\big(\nabla F_\tau(z_\tau) - \nabla F_\tau(z_\tau^\star)\big) - \dot{z}_\tau^\star.$$

Define the Lyapunov function $V_\tau = \frac{1}{2}\|e_\tau\|_{\mathcal{K}^{-1}}^2 = \frac{1}{2} e_\tau^\top \mathcal{K}^{-1} e_\tau$. Since $\mathcal{K}$ is constant and symmetric,

$$\begin{aligned} \dot{V}_\tau &= e_\tau^\top \mathcal{K}^{-1} \dot{e}_\tau \\ &= -e_\tau^\top \big(\nabla F_\tau(z_\tau) - \nabla F_\tau(z_\tau^\star)\big) - e_\tau^\top \mathcal{K}^{-1} \dot{z}_\tau^\star. \end{aligned}$$

Step 2: Lyapunov bound and the radius reduction (Lyapunov-to-r bridge) Since $z_\tau, z_\tau^\star \in \mathcal{D}$ for all $\tau \in [0, \mathcal{T}]$, Assumption A1 with $u = z_\tau$ and $v = z_\tau^\star$ gives

$$e_\tau^\top \big(\nabla F_\tau(z_\tau) - \nabla F_\tau(z_\tau^\star)\big) \geq \mu \|e_\tau\|^2.$$

Moreover, since $\mathcal{K} \succeq \lambda_{\min}(\mathcal{K}) I$ on the active subspace,

$$\|e_\tau\|^2 \geq \lambda_{\min}(\mathcal{K}) \|e_\tau\|_{\mathcal{K}^{-1}}^2 = \lambda_{\min}(\mathcal{K}) r_\tau^2.$$

Using Cauchy–Schwarz in the $\mathcal{K}^{-1}$ inner product together with Lemma J.1,

$$|e_\tau^\top \mathcal{K}^{-1} \dot{z}_\tau^\star| \leq \|e_\tau\|_{\mathcal{K}^{-1}} \|\dot{z}_\tau^\star\|_{\mathcal{K}^{-1}} \leq B_\star |\dot{\alpha}(\tau)|\, r_\tau.$$

Combining the above inequalities yields the *Lyapunov inequality*

$$\dot{V}_\tau \leq -\mu \lambda_{\min}(\mathcal{K}) r_\tau^2 + B_\star |\dot{\alpha}(\tau)|\, r_\tau. \tag{5}$$

We now isolate the key reduction from $\dot{V}_\tau$ to a scalar inequality for r_τ. Because $V_\tau = \frac{1}{2} r_\tau^2$ and $r_\tau \geq 0$, the chain rule gives $\dot{V}_\tau = r_\tau \dot{r}_\tau$ at points where $r_\tau > 0$; substituting into (5) and dividing by r_τ yields

$$\dot{r}_\tau \leq -\mu \lambda_{\min}(\mathcal{K}) r_\tau + B_\star |\dot{\alpha}(\tau)|. \tag{6}$$

(The norm of an absolutely continuous path is absolutely continuous. On the set where $r_\tau = 0$, its derivative is zero almost everywhere; hence the same inequality holds there almost everywhere.) We refer to the passage from (5) to (6) as the *radius reduction step*.

Step 3: Grönwall closure Applying Grönwall's inequality to Eq. (6) gives

$$r_\tau \leq e^{-\mu \lambda_{\min}(\mathcal{K}) \tau} r_0 + B_\star \int_0^\tau e^{-\mu \lambda_{\min}(\mathcal{K})(\tau - s)} |\dot{\alpha}(s)|\, ds.$$

Finally,

$$\int_0^\tau e^{-\mu \lambda_{\min}(\mathcal{K})(\tau - s)} ds \leq \frac{1}{\mu \lambda_{\min}(\mathcal{K})},$$

which yields Eq. (4). ■

Corollary (Prior internalization: structural identity vs. realized endpoint deviation) *Structural assumption (ideal moving optimum)* Assume additionally that the effective-risk minimizer $\tilde{z}^\star(x)$ depends only on the conditional distribution $\mathbb{P}(Y \mid x)$ and is therefore independent of $\alpha(\tau)$. Then the quasi-static intrinsic minimizer admits the affine representation $z_\tau^\star = \tilde{z}^\star - \alpha(\tau) b_{\text{stat}}$, and, together with the schedule endpoints $\alpha(0) = 1$ and $\alpha(\mathcal{T}) = 0$, the *ideal endpoint shift identity* $z_{\mathcal{T}}^\star - z_0^\star = b_{\text{stat}}$ holds. Without this structural assumption, one should *not* treat $z_{\mathcal{T}}^\star - z_0^\star = b_{\text{stat}}$ as automatic.

Realized logits vs. internalization error. Under the ideal endpoint identity $z_{\mathcal{T}}^\star - z_0^\star = b_{\text{stat}}$, the exact algebraic decomposition $z_{\mathcal{T}} - z_0 - b_{\text{stat}} = e_{\mathcal{T}} - e_0$ rewrites the realized endpoint deviation from b_{stat} into a difference of internalization errors at $\tau = \mathcal{T}$ and $\tau = 0$.

Using Theorem J.1 to control $\|e_\tau\|_{\mathcal{K}^{-1}}$ and the triangle inequality $\|e_{\mathcal{T}} - e_0\|_{\mathcal{K}^{-1}} \leq \|e_{\mathcal{T}}\|_{\mathcal{K}^{-1}} + \|e_0\|_{\mathcal{K}^{-1}}$, with $a = \mu \lambda_{\min}(\mathcal{K})$,

$$\|z_{\mathcal{T}} - z_0 - b_{\text{stat}}\|_{\mathcal{K}^{-1}} \leq \big(1 + e^{-a\mathcal{T}}\big) \|e_0\|_{\mathcal{K}^{-1}} + \frac{B_\star}{a} \sup_{s \in [0, \mathcal{T}]} |\dot{\alpha}(s)|.$$

Thus the endpoint identity is exact up to a transient initialization term and an annealing-induced internalization term. In particular, if $e_0 = 0$ or $\|e_0\|_{\mathcal{K}^{-1}} = \mathcal{O}(\sup_s |\dot{\alpha}(s)|)$, then

$$\|z_{\mathcal{T}} - z_0 - b_{\text{stat}}\|_{\mathcal{K}^{-1}} = \mathcal{O}\left(\sup_{s \in [0, \mathcal{T}]} |\dot{\alpha}(s)| \right).$$

K Proof of Zero-Shot Transfer

Proof. Fix a prediction step and a code c. Let $\mathcal{D}_S$ and $\mathcal{D}_T$ denote the source and target domains. Under the label-wise prior-shift assumption, $\mathbb{P}_S(x \mid y_c) = \mathbb{P}_T(x \mid y_c)$, while only the class prior changes from $\pi_S(c)$ to $\pi_T(c)$.

The Bayes-optimal per-code logit in domain $\mathcal{D} \in \{\mathcal{D}_S, \mathcal{D}_T\}$ is

$$z^\star_{\mathcal{D},c}(x) = \log \frac{\mathbb{P}_{\mathcal{D}}(y_c = 1 \mid x)}{\mathbb{P}_{\mathcal{D}}(y_c = 0 \mid x)}.$$

Applying Bayes' rule gives

$$z^\star_{\mathcal{D},c}(x) = \log \frac{\mathbb{P}_{\mathcal{D}}(x \mid y_c = 1)}{\mathbb{P}_{\mathcal{D}}(x \mid y_c = 0)} + \log \frac{\pi_{\mathcal{D}}(c)}{1 - \pi_{\mathcal{D}}(c)}.$$

The first term is invariant across domains by assumption, so subtracting the source and target expressions yields

$$z^\star_{T,c}(x) - z^\star_{S,c}(x) = \mathrm{logit}\big(\pi_T(c)\big) - \mathrm{logit}\big(\pi_S(c)\big).$$

Hence the exact label-wise Bayes correction is obtained by adding the delta bias $\Delta b(c) = \mathrm{logit}\big(\pi_T(c)\big) - \mathrm{logit}\big(\pi_S(c)\big)$. This proves the claim. The main-text interpolation parameter λ is exact when $\lambda = 1$; values $\lambda \in [0, 1)$ correspond to a practical interpolation heuristic rather than to the exact Bayes identity. ■ □

Remark. This observation is deliberately label-wise and step-wise. It justifies marginal prior correction for each code at a fixed prediction step, but it does not by itself imply exact matching of the full autoregressive joint distribution after rollout.

L Theoretical Analysis and Proofs

In this section, we provide supporting proofs for the claims used in the main text.

L.1 Proof of the Static Centering Property

We prove a local centering statement rather than a global optimality claim.

Bayesian decomposition For a fixed medical code c, the Bayes-optimal effective logit under BCE is the posterior log-odds $\tilde{z}^\star_c(x) = \log \frac{\mathbb{P}(y_c=1|x)}{\mathbb{P}(y_c=0|x)}$. Applying Bayes' rule gives the standard decomposition

$$\tilde{z}^\star_c(x) = \underbrace{\log \frac{\mathbb{P}(x \mid y_c = 1)}{\mathbb{P}(x \mid y_c = 0)}}_{\mathcal{S}_c(x)} + \underbrace{\log \frac{\pi(c)}{1 - \pi(c)}}_{\mathrm{logit}(\pi(c))}. \tag{7}$$

Here $\mathcal{S}_c(x)$ is the structural term, while $\mathrm{logit}(\pi(c))$ is the prevalence term.

Prior-centered initialization Let $\hat{\pi}_\epsilon(c) = \frac{\hat{\pi}(c)+\epsilon}{1+2\epsilon}$ and $b_{\text{stat}}(c) = -\mathrm{logit}\big(\hat{\pi}_\epsilon(c)\big)$. Assume that, near initialization, the intrinsic logits satisfy $z_{0,c}(x) = \mathrm{logit}\big(\hat{\pi}_\epsilon(c)\big) + r_c(x)$, with $|r_c(x)| \ll 1$. Then for any static additive shift $b(c)$ the initial effective logit is $\tilde{z}_{0,c}(x) = z_{0,c}(x) + b(c) = \mathrm{logit}\big(\hat{\pi}_\epsilon(c)\big) + r_c(x) + b(c)$. The unique shift that centers the nominal operating point (the case $r_c(x) = 0$) at zero is $b(c) = -\mathrm{logit}\big(\hat{\pi}_\epsilon(c)\big) = b_{\text{stat}}(c)$, for which $\tilde{z}_{0,c}(x) = r_c(x) \approx 0$.

Local sensitivity maximization For BCE with sigmoid link, the local logit-to-probability sensitivity and the pointwise curvature are both $\sigma'(u) = \sigma(u)(1 - \sigma(u))$. This quantity is uniquely maximized at $u = 0$. Therefore, among additive shifts depending only on $\hat{\pi}_\epsilon(c)$, the choice $b_{\text{stat}}(c)$ is the unique shift that centers the local operating point at the maximal-sensitivity region of the sigmoid. This establishes the centering claim used in the main text.

Finally, because the scaffold is an x-independent additive term, it translates the logit coordinate without changing the structural component $\mathcal{S}_c(x)$ in Eq. (7). Thus it acts as a local optimization scaffold rather than as a modification of the underlying structure. ■

L.2 Lemma: NTK Gradient Flow under BCE Loss

Before analyzing the curriculum dynamics, we formalize the continuous-time gradient flow of neural network outputs under Binary Cross-Entropy (BCE) loss.

Lemma L.1 (BCE-NTK Gradient Flow)**.** *Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ be the training dataset. Let $\mathbf{f}(t) = [f_{\theta(t)}(x_1), \ldots, f_{\theta(t)}(x_n)]^\top \in \mathbb{R}^n$ denote the intrinsic logits of the neural network at time t, and $\mathbf{y} = [y_1, \ldots, y_n]^\top \in \{0,1\}^n$ be the corresponding labels. Under continuous-time gradient descent on the BCE loss, the evolution of the network outputs follows the non-linear Ordinary Differential Equation (ODE)*

$$\frac{d\mathbf{f}(t)}{dt} = -H(t) \cdot (\sigma(\mathbf{f}(t)) - \mathbf{y}),$$

where $H(t) \in \mathbb{R}^{n \times n}$ is the empirical NTK matrix with entries

$$H_{i,j}(t) = \left\langle \nabla_\theta f_{\theta(t)}(x_i), \nabla_\theta f_{\theta(t)}(x_j) \right\rangle.$$

Proof. The parameter trajectory under gradient flow satisfies

$$\frac{d\theta(t)}{dt} = -\nabla_\theta L(\theta(t)) = -\nabla_\theta \left(\sum_{i=1}^n \ell_{\mathrm{BCE}}(y_i, \sigma(f_{\theta(t)}(x_i))) \right).$$

Applying the chain rule gives

$$\frac{d\theta(t)}{dt} = -\sum_{i=1}^n \big(\sigma(f_{\theta(t)}(x_i)) - y_i\big) \nabla_\theta f_{\theta(t)}(x_i).$$

For any x_k,

$$\begin{aligned} \frac{df_{\theta(t)}(x_k)}{dt} &= \left\langle \nabla_\theta f_{\theta(t)}(x_k), \frac{d\theta(t)}{dt} \right\rangle \\ &= -\sum_{i=1}^n \big(\sigma(f_{\theta(t)}(x_i)) - y_i\big) \left\langle \nabla_\theta f_{\theta(t)}(x_k), \nabla_\theta f_{\theta(t)}(x_i) \right\rangle. \end{aligned}$$

Stacking these equations over $k = 1, \ldots, n$ yields the claimed matrix form. ■ □

Remark. Under AdaPCLA, the *parameter gradient* of the effective logits coincides with that of the intrinsic logits, because the scaffold is independent of θ: $\nabla_\theta \tilde{f}_{\theta(t)}(x) = \nabla_\theta f_{\theta(t)}(x)$. This holds at a fixed training step. Along the continuous-time flow the two quantities differ by an explicit annealing term,

$$\frac{d}{dt}\tilde{z}(t) = \frac{d}{dt} z(t) + \dot{\alpha}(t) b_{\text{stat}},$$

so the effective logits follow the same gradient-flow identity only up to $\dot{\alpha}(t) b_{\text{stat}}$. That term is precisely what the tracking bound of Appendix J accounts for, and it is why a smoother schedule yields a smaller error; a single SGD step with α held fixed may neglect it.

L.3 Positive-Class Weighting of the BCE Residual

The reported AdaPCLA training configuration up-weights positive labels by $w = 1.5$ (Table 5). For a single label with target $y \in \{0,1\}$ and effective logit $u = \tilde{z}$, the weighted loss is

$$\ell_w(y,u) = -w\, y \log \sigma(u) - (1-y) \log\big(1 - \sigma(u)\big), \qquad \frac{\partial \ell_w}{\partial u} = \big[1 + (w-1)y\big]\sigma(u) - w\, y.$$

For $y = 0$ this equals $\sigma(u)$, and for $y = 1$ it equals $w\,(\sigma(u) - 1)$. The weighting therefore multiplies the residual of positive labels by w and leaves negative labels unchanged, so the one-step update of the main text holds with the residual replaced by $W\mathcal{G}$, where $W = \mathrm{diag}(w \text{ on positives}, 1 \text{ otherwise})$. The Jacobian and the label-wise NTK are unchanged, but the residual is reweighted separately for each label. This is not generally equivalent to a single global learning-rate rescaling.

The weighting does move the optimum: the minimizer of the weighted conditional risk is $\sigma^\star = w\pi/(w\pi + 1 - \pi)$ rather than π, where π is the conditional occurrence probability. Consequently the moving optimum in the tracking bound of Appendix J is displaced, and its constants should be read as those of the weighted problem. We state the bound for the unweighted loss because the argument is cleaner there; we do not claim that the unweighted formula is the exact training objective.

M Proof of Learning Dynamics under Multi-Label Conditional Factorization

We now write the discrete first-order update under the actual AdaPCLA training objective. Fix one SGD step and condition on the current annealing value α_t as constant during that step. Define the effective logits $\tilde{z}^t(x) = z^t(x) + \alpha_t b_{\text{stat}}$, where $z^t(x) = f_{\theta^t}(x) \in \mathbb{R}^V$ denotes the intrinsic logits.

Let x_o be an observing example and (x_u, y_u) an updating example. We study the change in the log-probability vector $\log \sigma(\tilde{z}^t(x_o))$. A first-order Taylor expansion gives

$$\log \sigma(\tilde{z}^{t+1}(x_o)) = \log \sigma(\tilde{z}^t(x_o)) + \left\langle \nabla_\theta \log \sigma(\tilde{z}^t(x_o))\big|_{\theta^t},\, \theta^{t+1} - \theta^t \right\rangle + \mathcal{O}(\|\theta^{t+1} - \theta^t\|^2).$$

Therefore,

$$\Delta \log \sigma(\tilde{z}^t(x_o)) = \nabla_\theta \log \sigma(\tilde{z}^t(x_o))\big|_{\theta^t} (\theta^{t+1} - \theta^t) + \mathcal{O}(\|\theta^{t+1} - \theta^t\|^2). \tag{8}$$

Under SGD on the AdaPCLA BCE loss,

$$\theta^{t+1} - \theta^t = -\eta \left(\nabla_\theta \mathcal{L}_{\text{BCE}}\big(y_u, \sigma(\tilde{z}^t(x_u))\big)\right)^\top.$$

Because $\alpha_t b_{\text{stat}}$ is independent of θ within this step, $\nabla_\theta \tilde{z}^t(x) = \nabla_\theta z^t(x)$. Applying the chain rule on both factors in Eq. (8) yields the following compact decomposition. For readability, write

$$\begin{aligned}
D_o^t &:= \nabla_{\tilde{z}} \log \sigma(\tilde{z}^t(x_o))\big|_{\tilde{z}^t}, \\
J_o^t &:= \nabla_\theta z^t(x_o)\big|_{\theta^t}, \\
J_u^t &:= \nabla_\theta z^t(x_u)\big|_{\theta^t}, \\
R_u^t &:= \nabla_{\tilde{z}} \mathcal{L}_{\text{BCE}}(y_u, \sigma(\tilde{z}^t(x_u)))\big|_{\tilde{z}^t}.
\end{aligned}$$

Then

$$\begin{aligned}
&\nabla_\theta \log \sigma(\tilde{z}^t(x_o))\big|_{\theta^t}(\theta^{t+1} - \theta^t) \\
&\quad = -\eta\, D_o^t J_o^t (J_u^t)^\top (R_u^t)^\top \\
&\quad = -\eta\, \mathcal{A}_{\text{Ada}}^t(x_o) \mathcal{K}^t(x_o, x_u) \mathcal{G}_{\text{Ada}}^t(x_u, y_u),
\end{aligned} \tag{9}$$

where

$$\mathcal{A}_{\text{Ada}}^t(x_o) = \operatorname{diag}\big(1 - \sigma(\tilde{z}^t(x_o))\big),$$

$$\mathcal{K}^t(x_o, x_u) = \nabla_\theta z^t(x_o)\big|_{\theta^t} \left(\nabla_\theta z^t(x_u)\big|_{\theta^t}\right)^\top,$$

and

$$\mathcal{G}_{\text{Ada}}^t(x_u, y_u) = \left(\nabla_{\tilde{z}} \mathcal{L}_{\text{BCE}}(y_u, \sigma(\tilde{z}^t(x_u)))\big|_{\tilde{z}^t}\right)^\top = \sigma(\tilde{z}^t(x_u)) - y_u.$$

Substituting Eq. (9) into Eq. (8) gives

$$\Delta \log \sigma(\tilde{z}^t(x_o)) = -\eta\, \mathcal{A}_{\text{Ada}}^t(x_o) \mathcal{K}^t(x_o, x_u) \mathcal{G}_{\text{Ada}}^t(x_u, y_u) + \mathcal{O}(\eta^2).$$

The matrix $\mathcal{A}_{\text{Ada}}^t(x_o)$ is diagonal because sigmoid factorizes across labels:

$$\frac{\partial \log \sigma(\tilde{z}_c)}{\partial \tilde{z}_k} = \begin{cases} 1 - \sigma(\tilde{z}_c), & c = k, \\ 0, & c \neq k. \end{cases}$$

Therefore, as in independent BCE, the output nonlinearity does not introduce explicit cross-label terms. Any cross-label effect is determined by the shared-backbone kernel $\mathcal{K}^t(x_o, x_u)$.

Finally, because $\tilde{z}^t(x) = z^t(x) + \alpha_t b_{\text{stat}}$ differs from $z^t(x)$ by a θ-independent constant, the intrinsic-logit increment obeys

$$\Delta z^t(x_o) = -\eta\, \mathcal{K}^t(x_o, x_u) \mathcal{G}_{\text{Ada}}^t(x_u, y_u) + \mathcal{O}(\eta^2),$$

which is the form used in the main text. The deterministic effect induced by changes in α_t across optimization steps is analyzed separately in Appendix J. ■

N Exploratory Analysis of $\lambda_{\min}(\mathcal{K})$ and Related-Rare Gain/AUC

The prior-internalization bound in Appendix J and Sec. 4.3 identifies the minimum eigenvalue $\lambda_{\min}(\mathcal{K})$ of the label-wise empirical NTK as a factor that reduces the internalization-error bound. As an **exploratory** analysis, we compute $\lambda_{\min}(\mathcal{K}_x)$ at the initial checkpoint for each probing context and compare it with the empirical related-rare AUC and final gain. The label-wise empirical NTK is restricted to the related-rare codes of the same context.

30-context results: On the 30-context micro-probe set, the Spearman correlations are positive but weak: Spearman($\lambda_{\min}(\mathcal{K}_{\text{init}})$, related-rare AUC) ≈ 0.30 ($p \approx 0.11$) and Spearman($\lambda_{\min}(\mathcal{K}_{\text{init}})$, related-rare gain) ≈ 0.16 ($p \approx 0.40$). The signs are consistent with the bound, but the correlations do not reach conventional significance.

100-context robustness check: We extend the analysis to 100 probing contexts. The correlations become close to zero: Spearman($\lambda_{\min}(\mathcal{K}_{\text{init}})$, AUC) ≈ 0.02 ($p \approx 0.86$) and Spearman($\lambda_{\min}(\mathcal{K}_{\text{init}})$, gain) ≈ -0.06 ($p \approx 0.53$). The final-gain correlation is negative under this larger probing set.

Conclusion: We do *not* treat the $\lambda_{\min}(\mathcal{K})$ and related-rare AUC/gain analysis as empirical validation of the bound. The 30-context result is statistically modest, and the 100-context extension shows that the correlation is not robust. We report this analysis for completeness as non-conclusive evidence.

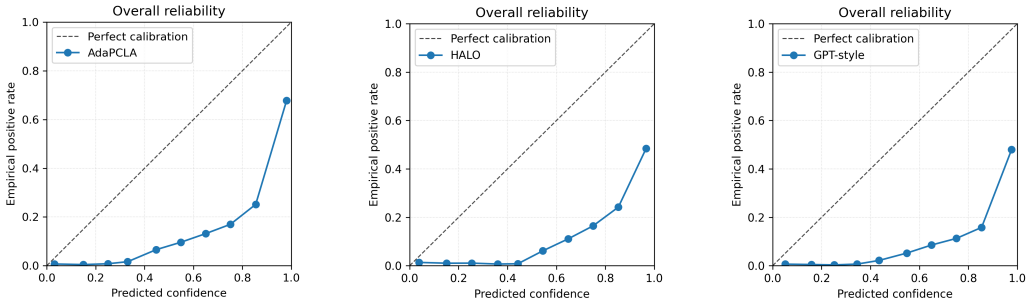
O Prediction Reliability Details and Additional Results

This section provides detailed prediction-reliability statistics for the downstream 25-label DiagnosisModel evaluated on the real MIMIC-IV test split. We report ECE, Brier Score, and Negative Log-Likelihood (NLL) for AdaPCLA, HALO, and GPT-style under two scopes: (i) overall (all labels), and (ii) tail-only (bottom-1/3 labels by training-set frequency).

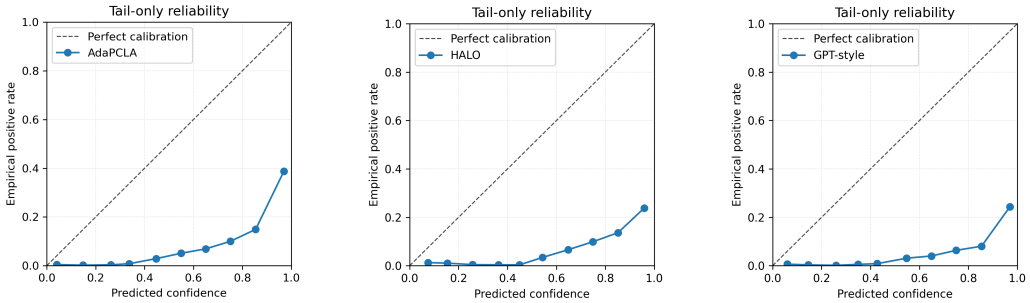
Table 15: **Prediction reliability on held-out real MIMIC-IV records.** Lower is better for ECE, Brier and NLL. Tail-only results restrict to the 8 rare label groups among the 25. The three models were scored on held-out record sets of different sizes (AdaPCLA: 17,877 records; GPT-style and HALO: 44,691), as the #Pairs column shows; the comparison is therefore indicative rather than exact, and we do not use it to rank models.

Model	Scope	ECE	Brier	NLL	#Pairs
AdaPCLA	overall	0.13	0.06	0.27	446,925
GPT-style	overall	0.27	0.14	0.55	1,117,275
HALO	overall	0.37	0.20	0.65	1,117,275
AdaPCLA	tail	0.19	0.08	0.34	143,016
GPT-style	tail	0.31	0.16	0.60	357,528
HALO	tail	0.44	0.24	0.73	357,528

Figure 4 shows the corresponding reliability diagrams for both the overall and tail-only scopes. Across confidence bins, AdaPCLA's curve stays closer to the identity line than the other two, especially in the high-confidence region, while GPT-style and HALO show larger confidence errors, particularly on rare labels. Given the difference in record-set sizes noted in the table caption, we read this as a consistent tendency rather than as a precise ranking.



(a) AdaPCLA (overall) (b) HALO (overall) (c) GPT-style (overall)



(d) AdaPCLA (tail-only) (e) HALO (tail-only) (f) GPT-style (tail-only)

Figure 4: **Reliability diagrams on held-out real MIMIC-IV records.** Top row: overall results across all 25 labels; bottom row: tail-only results restricted to the 8 rare label groups. AdaPCLA has the smallest deviation from the identity line, whereas GPT-style and HALO have larger confidence errors, particularly on tail labels. See the table caption for the difference in record-set sizes.

P Bias-Free Inference Algorithm

Algorithm 1 Bias-free inference process of AdaPCLA

Input: trained generator f_{θ^*} after annealing; maximum number of visits $T_{\max}$; Bernoulli sampling rule $\mathcal{S}$.
Output: synthetic visit sequence $\tilde{X} = (\tilde{x}_1, \ldots, \tilde{x}_{\tilde{T}})$.
Initialize $\tilde{X} \leftarrow \emptyset$.
for $t = 1$ **to** $T_{\max}$ **do**
  Compute intrinsic logits without an external bias: $z_t \leftarrow f_{\theta^*}(\tilde{x}_{<t}) \in \mathbb{R}^V$.
  Map to probabilities: $p_t \leftarrow \sigma(z_t)$.
  Sample the next visit: $\tilde{x}_t \leftarrow \mathcal{S}(p_t)$ (independent Bernoulli over the clinical-event vocabulary).
  Apply the stopping rule for an empty visit or termination criterion.
  if stopping rule is triggered **then**
    break
  end if
  Append $\tilde{x}_t$ to $\tilde{X}$.
end for
return $\tilde{X}$.

Q Micro-Probing Results under 30 and 100 Contexts

This section reports quantitative summaries for the micro-probing experiments with 30 and 100 patient-history contexts. For each context and probe type (related_rare / unrelated_rare / wrong),

we compute the AUC of $\Delta \log p(\tau)$ over annealing checkpoints and the final $\Delta \log p$ at the end of annealing. We then report the mean and standard deviation across contexts.

Table 16: **Micro-probing summaries under 30 and 100 contexts.** For each probe type, we report the number of contexts, the mean and standard deviation of the AUC of $\Delta \log p(\tau)$, and the mean and standard deviation of the final $\Delta \log p$ at the end of annealing. The 100-context setting keeps the trajectory scale similar to the 30-context setting and provides a larger probing set for the robustness check.

Setting	**Type**	**#Ctx**	**AUC mean**	**AUC std**	**Final $\Delta \log p$ mean**	**Final $\Delta \log p$ std**
30-context	related_rare	30	5.28	5.84	0.92	1.10
30-context	unrelated_rare	30	6.06	5.28	1.12	0.98
30-context	wrong	30	5.85	8.12	0.66	1.25
100-context	related_rare	100	5.80	6.02	1.05	1.11
100-context	unrelated_rare	100	5.66	4.96	0.98	0.93
100-context	wrong	100	4.32	7.30	0.50	1.19

Figure 2 in the main text tracks selected probe-code probabilities under 30 and 100 probing contexts. Table 16 reports the corresponding AUC of $\Delta \log p(\tau)$ and final $\Delta \log p$ for each probe type.

R Dependency Versions (Experimental Environment)

All experiments reported in this paper were run on a server with 8×NVIDIA RTX 4090 GPUs. Table 17 lists the main software and dependency versions used for training and evaluation. We recommend using compatible versions for reproduction.

Table 17: Main dependency versions used in our experiments for reproducibility.

Component	Version
Python	3.10.x
PyTorch (with CUDA)	2.0.x / 2.1.x
CUDA (driver / toolkit)	11.8 or 12.x
NumPy	$\geq$1.24
SciPy	$\geq$1.10
matplotlib	$\geq$3.7

S Complete Results with Standard Deviation

This section restates Table 1 and Table 2 for convenience and clarifies the reporting precision of the smallest standard deviations. In a four-decimal column, 0.0000 means that the raw standard deviation is below 5×10^{-5} (below 5×10^{-6} in the five-decimal TailTopKJac column); the affected cells are EVA and SynTEG on MIMIC-III. These values are small because both generators are extremely stable across cohorts on the MIMIC-III vocabulary; they are not exact zeros.

T Detailed Annealing Analysis (Supplementary)

This section provides extended analyses referenced in Sec. I.1. We report the micro-probing setup (30 clinical contexts, each with three disease types: related_rare, unrelated_rare, wrong), the eNTK trajectory over 100 annealing checkpoints, and the Balanced Accuracy / Geometric Mean (bACC/GM) curves on the probing set. The bACC/GM panel is reported for completeness, and it is a **negative** result: at the decision threshold used for the probes, bACC stays at 0.5 and GM at 0 across all checkpoints, i.e. this thresholded diagnostic does not separate related from unrelated tail codes. We therefore do not use it as evidence that AdaPCLA selectively elevates structurally relevant codes; the probability-trajectory panels of Fig. 3 are the relevant evidence for the internalization dynamics, and they show type-dependent but overlapping trajectories.

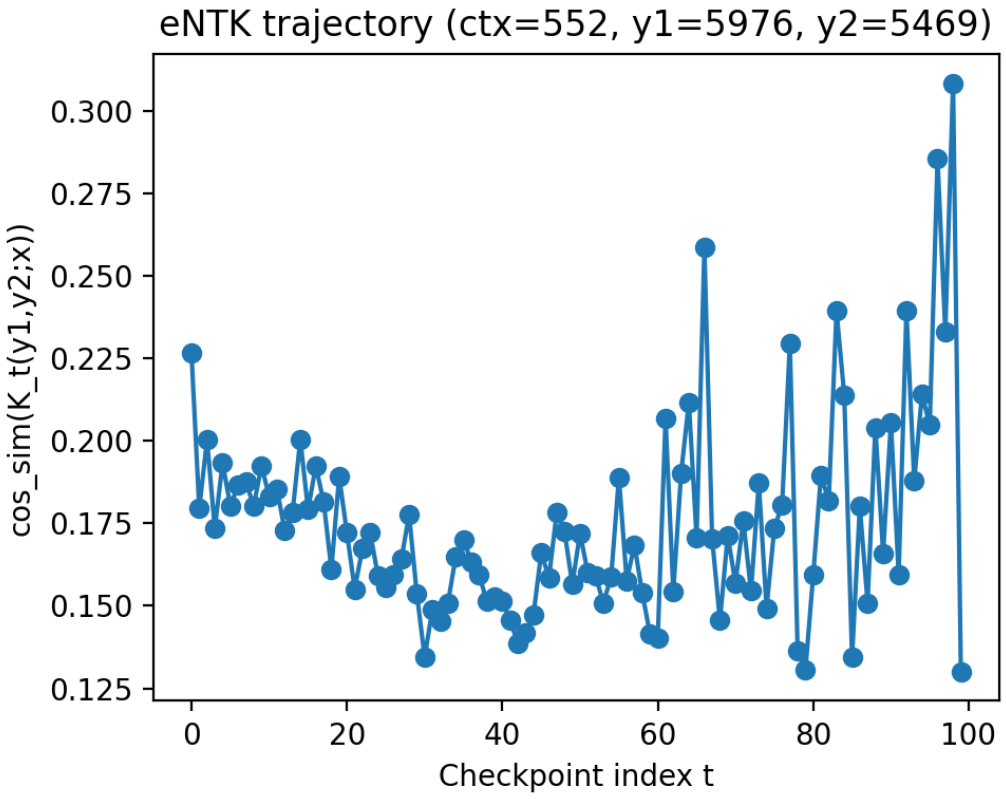
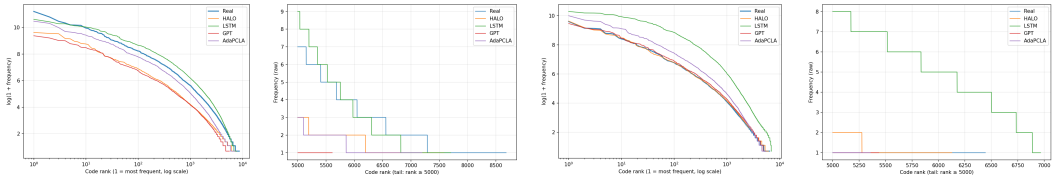


Figure 5: eNTK cosine similarity trajectory over 100 annealing checkpoints for a representative (context, disease pair). The gradient bridge evolves as training proceeds.

U Frequency–Rank Plots

This section reports frequency–rank plots for MIMIC-III and MIMIC-IV, comparing Real training data with synthetic data generated by HALO, LSTM, GPT-style, and AdaPCLA. These plots summarize the ranked marginal occurrence-frequency distribution of medical codes. They do not evaluate code-set support, tail clinical plausibility, rare-code context consistency, or downstream predictive utility. Therefore, we use them only as supplementary distributional-fidelity views. Model comparison should rely on Table 1 for PairSeen, TailPairSeen, TailCtxJSD, and TailTopKJac, Table 2 for downstream utility, and Sec. I.2 for qualitative clinical examples.



(a) Frequency–rank distribution (full range, log-scaled rank) on MIMIC-IV: Real, HALO, LSTM, GPT-style, and AdaPCLA.

(b) Frequency–rank distribution (tail range, rank ≥ 5000, raw frequency) on MIMIC-IV: Real, HALO, LSTM, GPT-style, and AdaPCLA.

(c) Frequency–rank distribution (full range, log-scaled rank) on MIMIC-III: Real, HALO, LSTM, GPT-style, and AdaPCLA.

(d) Frequency–rank distribution (tail range, rank ≥ 5000, raw frequency) on MIMIC-III: Real, HALO, LSTM, GPT-style, and AdaPCLA.

Figure 7: Frequency–rank plots for the full range and tail range on MIMIC-IV and MIMIC-III. These plots summarize marginal occurrence-frequency distributions only.

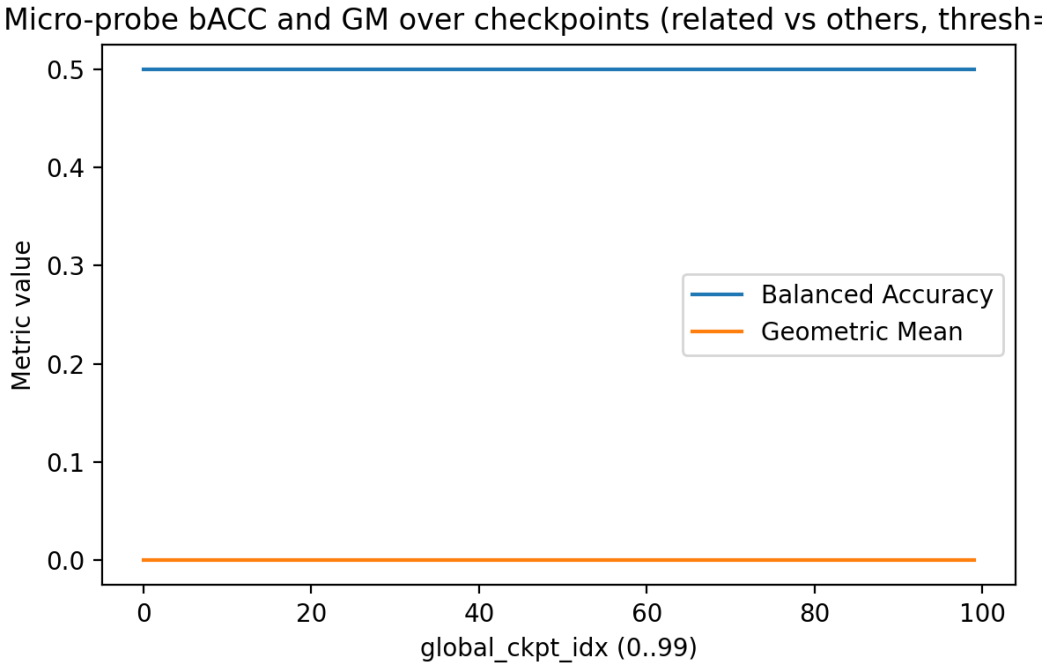


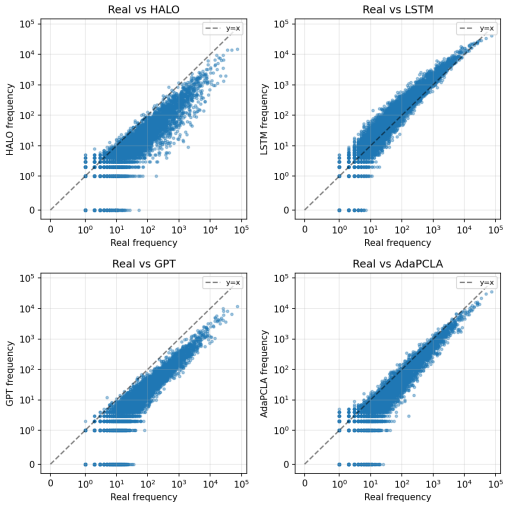
Figure 6: Balanced Accuracy and Geometric Mean on the micro-probing set (related_rare vs. unrelated_rare + wrong) over annealing checkpoints.

V Frequency–Frequency and Rank–Rank Scatters

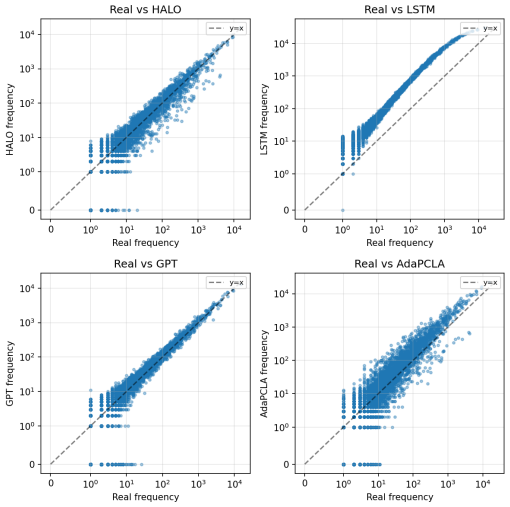
This section reports per-code marginal occurrence-frequency views that complement the frequency–rank plots in Sec. U. Each point corresponds to one medical code present in the Real training set.

Frequency–frequency scatter: For each code c, the horizontal axis is its visit-level frequency in Real and the vertical axis is its visit-level frequency in a given model (one subplot per model: HALO, LSTM, GPT-style, AdaPCLA). Both axes use a symmetric log scale. Points near the diagonal $y = x$ indicate that the model's marginal frequency for that code is close to Real; points above or below the diagonal indicate higher or lower generated frequency than Real. These plots evaluate code-level marginal occurrence frequencies, but they do not measure tail clinical plausibility or rare-event co-occurrence structure.

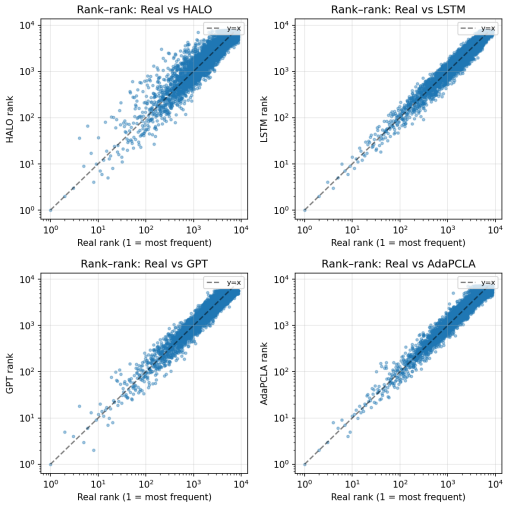
Rank–rank scatter: For each code c, the horizontal axis is its rank in Real (1 = most frequent) and the vertical axis is its rank in the model's synthetic data; both axes are log-scaled. Points near the diagonal indicate agreement between the Real code rank and the synthetic code rank. Codes never generated by a model are assigned a rank beyond the maximum. Together, the frequency–frequency and rank–rank figures report per-code marginal alignment with the training distribution.



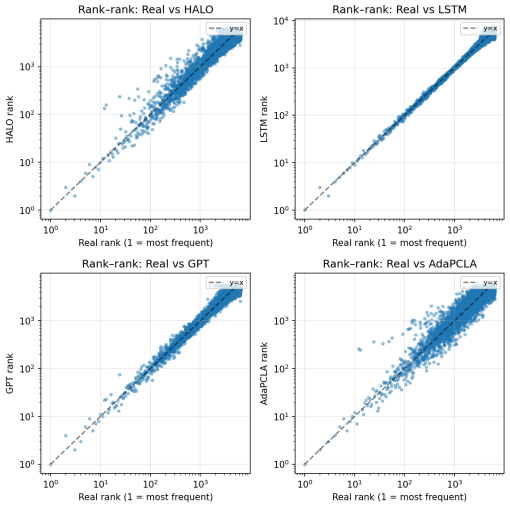
(a) Frequency–frequency scatter (full range) on MIMIC-IV: each subplot compares Real with one model (HALO, LSTM, GPT-style, AdaPCLA). Axes: visit-level frequency (symlog). Points near the diagonal indicate code-level marginal-frequency alignment.



(b) Frequency–frequency scatter (full range) on MIMIC-III: Real compared with HALO, LSTM, GPT-style, and AdaPCLA. Axes: visit-level frequency (symlog).



(c) Rank–rank scatter (full range) on MIMIC-IV: each subplot shows Real rank and model rank per code (log scale). Points near the diagonal indicate agreement in code ordering.



(d) Rank–rank scatter (full range) on MIMIC-III: Real rank and model rank per code.

Figure 8: Per-code marginal occurrence-frequency alignment between Real and synthetic generators, shown with frequency–frequency and rank–rank scatter plots.

W Code Co-occurrence Block Structure

This section reports code–code co-occurrence matrices for the top-200 most frequent medical codes on MIMIC-IV. We compute visit-level co-occurrence counts for Real, HALO, LSTM, and AdaPCLA, and then apply a PPMI (Positive Pointwise Mutual Information) transform. To keep the panels comparable, we perform hierarchical clustering on the Real PPMI matrix using average linkage and apply the resulting code order to all models. The visualization compares how well each generator preserves frequent-code co-occurrence patterns learned from the Real training data.

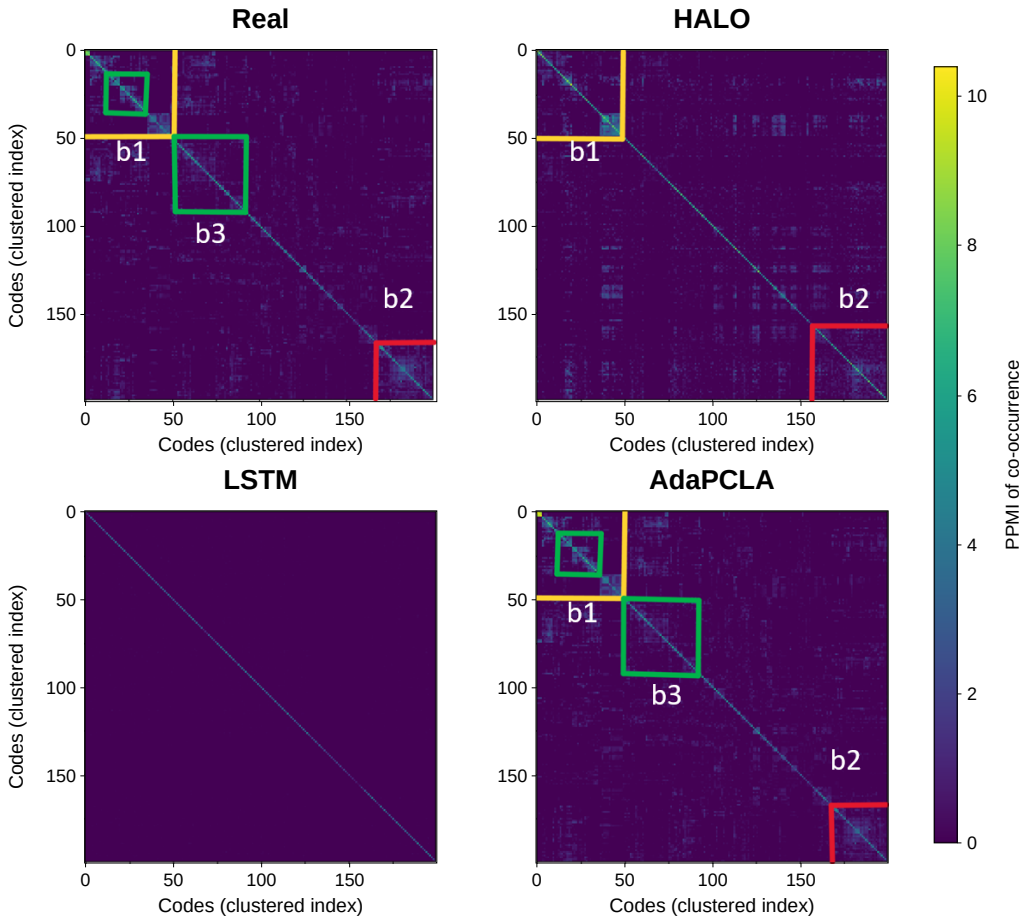


Figure 9: PPMI-transformed code–code co-occurrence heatmaps for the top-200 most frequent codes on MIMIC-IV. All panels use the same Real-based hierarchical clustering order along both axes. AdaPCLA better preserves the frequent-code co-occurrence structure of Real than HALO and LSTM. All four panels are drawn with the single colour scale shown at the right. Note that this figure and Fig. 10 use different colour scales for the same quantity, so comparisons should be read within a figure rather than across figures.

X Additional Experimental Figures

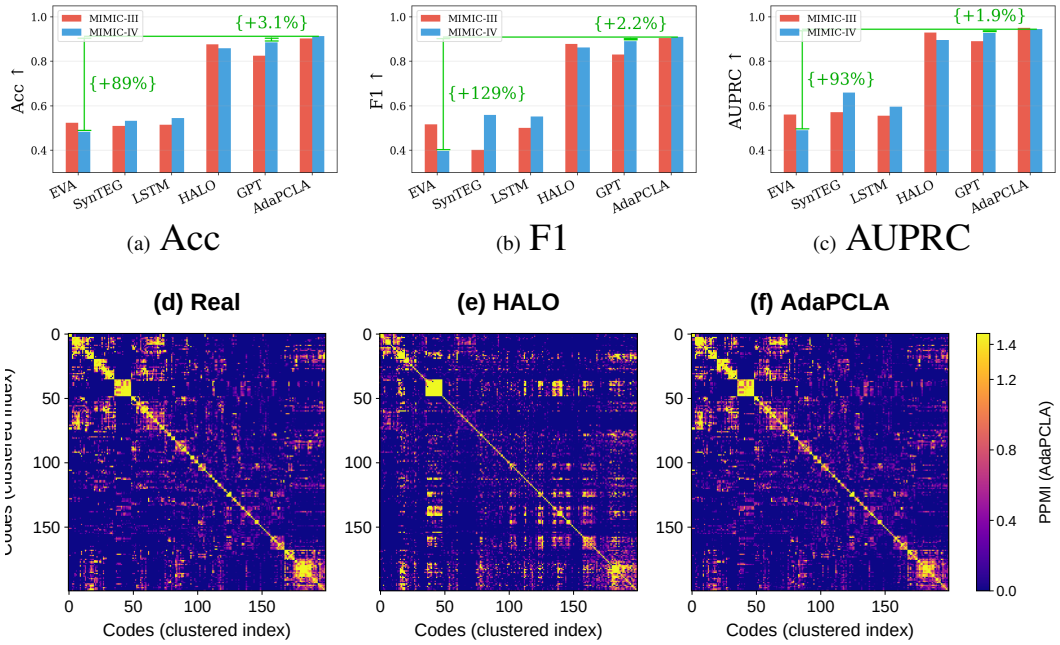


Figure 10: Downstream utility and code co-occurrence summaries on MIMIC-IV. **Row 1:** Accuracy, F1, and AUPRC under the TSTR protocol. **Row 2:** code–code co-occurrence heatmaps for Real, HALO, and AdaPCLA; the original rendered matrices and their ordering are retained. The colour bar is reproduced from the AdaPCLA source panel; these panels provide qualitative block-pattern comparisons.

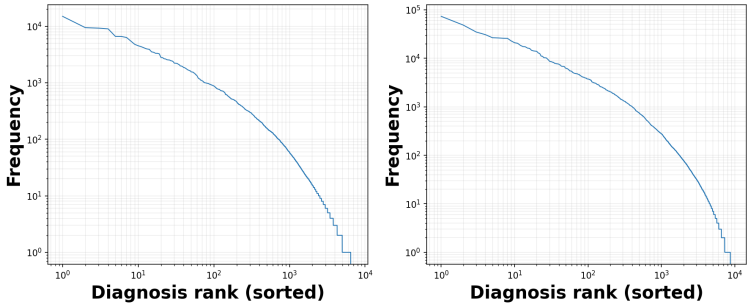


Figure 11: Medical code frequency versus rank for Real training data on MIMIC-III and MIMIC-IV.

Y Code-Bucket Conditional Distributions

This section reports the full $P(\text{code} \mid \text{label})$ heatmaps referenced in Sec. 6. We partition medical codes into three frequency buckets: head (most frequent), mid, and tail (long tail). For each bucket, we compute the conditional distribution of codes given the bucket label. The heatmaps use a 3×4 layout: each row corresponds to one bucket, and each column corresponds to one data source (Real training data, HALO, LSTM, or AdaPCLA) on MIMIC-IV. These plots provide supplementary distributional-fidelity views for code-level probability structure within each frequency regime. AdaPCLA shows closer alignment with Real across the three buckets than HALO and LSTM, which is consistent with the tail plausibility and co-occurrence results in Table 1.

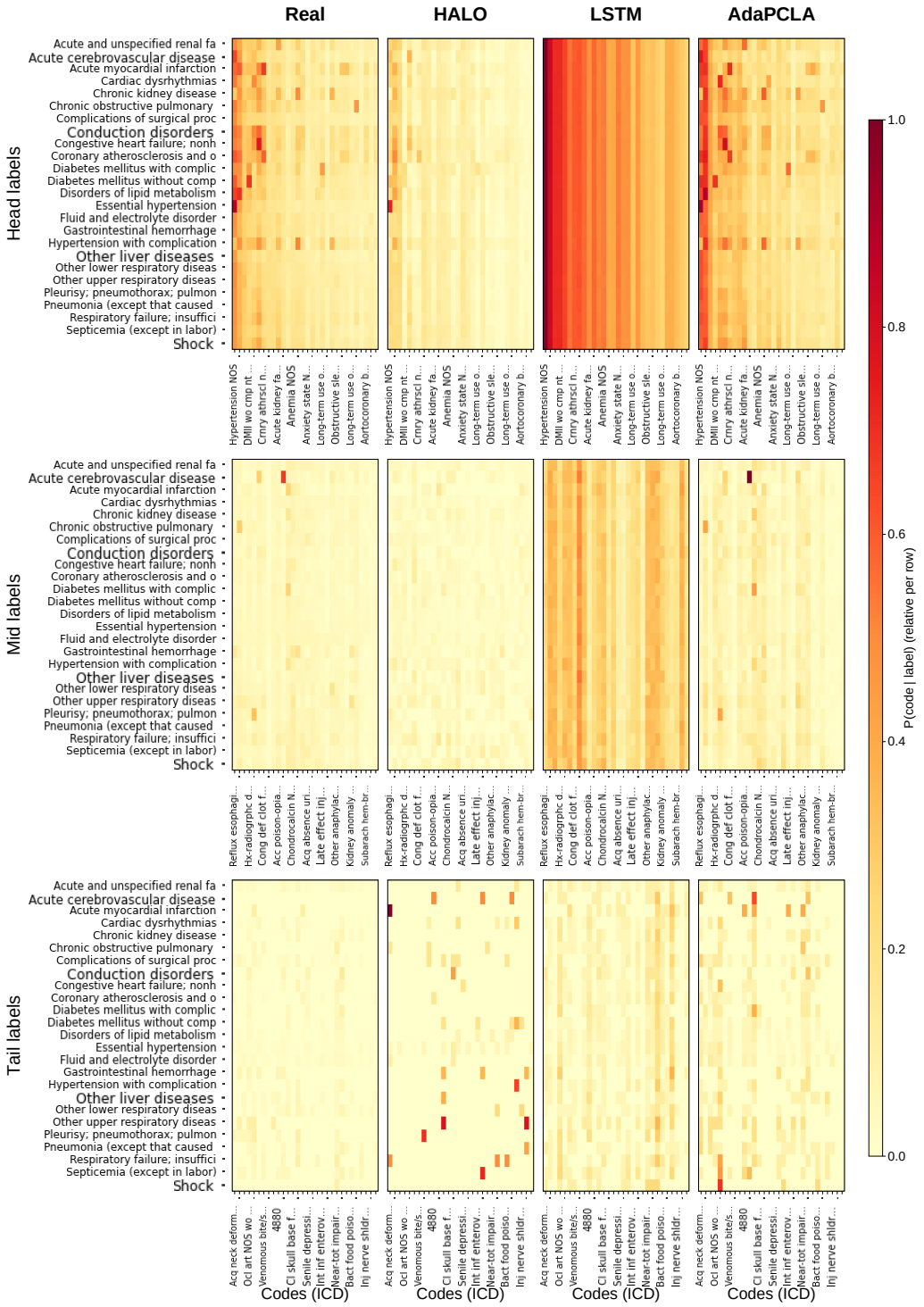


Figure 12: $P(\text{code} \mid \text{label})$ heatmaps for head, mid, and tail code buckets on MIMIC-IV. Each row corresponds to one frequency bucket, and each column corresponds to Real, HALO, LSTM, or AdaPCLA. All heatmap cells and the original label wording are retained; every third code label is displayed to reduce crowding.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the scope explicitly, including the diagnosis-code restriction, the target-frequency (not multi-center) adaptation setting, and the fact that the internalization bound is a design guide rather than an empirically verified prediction; the Limitations paragraph records the remaining gaps.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The final section ends with a four-part Limitations paragraph covering (i) adaptation across MIMIC versions rather than across institutions, (ii) the single-run status of the added baselines, (iii) the privacy trade-off documented in Appendix G, and (iv) the diagnosis-code-only scope.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Theorem 5.1 in the main text is stated with explicit reference to Assumptions A1–A3, which are listed and used in Appendix J; Lemma J.1 and Theorem J.1 are numbered and come with complete proofs, and the zero-shot correction is proved in Appendix K.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Appendix C reports full dataset statistics and the complete training configuration, Appendix B gives exact definitions of every metric, and Sec. 6 describes the splits, baselines, and protocol.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code and evaluation scripts are released at the URL given in the first-page footnote. MIMIC-III, MIMIC-IV and eICU are distributed by PhysioNet under credentialed data-use agreements and are therefore not redistributed with the paper.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Sec. 6 lists datasets, splits, baselines and metrics; Appendix C adds dataset statistics and hyper-parameters, and Appendix D describes how the additional baselines were reproduced.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: Every entry of the two main tables reports mean±std, and Sec. 6 explains what the standard deviation measures (three generated cohorts from a single trained checkpoint, not independent training runs) and reports the visit-size-reweighted check that confirms the ordering. The baselines added after review are explicitly marked as single-run snapshots, and we do not claim significance for them.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All experiments run on a server with 8 NVIDIA RTX 4090 GPUs; software versions are listed in Appendix R, and training time per model is on the order of hours.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Synthetic EHR generation carries a dual-use risk: more faithful tail modelling can also increase memorisation or attribute leakage. Appendix G measures this directly and reports a higher attribute-inference recovery for our method, which we discuss as a trade-off and as a limitation.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: What we release is code and a training objective, not a trained generator or a data release; no real patient records leave the credentialed-access environment, and all evaluation uses de-identified public benchmarks. The residual privacy risks that this class of model inherits are measured in Appendix G and acknowledged in the Limitations.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The creators or original owners of assets (e.g., code, data, models), used in the paper, are properly credited. MIMIC-III, MIMIC-IV and eICU are used under the PhysioNet credentialed data-use agreement; the baseline implementations follow their official releases, which are cited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release the code implementing the training objective and the evaluation scripts at the URL in the first-page footnote, with documentation of the configuration in Appendix C. No new dataset is introduced.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development does not involve LLMs as an important, original, or non-standard component; the method uses an autoregressive Transformer trained on diagnosis codes and initialized from the corresponding HALO checkpoint.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.