Genetic Effects on Disease Progression Speed: Why It's Identifiable

Motivation: What the Current Model Misses

Key idea: Signatures are **fixed** (population archetypes). The **patient's loading** $\theta(t)$ on those signatures evolves over time. **Two mechanisms, same outcome:**

- **High baseline, slow progression:** Starts with high θ on CV signature; stays there. Early MI because already "in" the high-risk state.
- **Low baseline, fast progression:** Starts with low θ; θ grows rapidly toward CV signature. Early MI because of *rate* of accumulation.

How the current model misses this:

- Genetics enters only the **constant** part of the mean (baseline level).
- All change over time is $E_i(t)$ (GP noise), **independent of genetics**.
- Slope differences between people are treated as random variation, not predictable from genetics.
- Cannot separate "start high" vs "accumulate fast"—both get absorbed into noise.

The Question

Can we distinguish between:

1. **Heavy baseline loading**
   Person has high cardiovascular signature from age 30 onwards

2. **Fast progression**
   Person starts normal but accumulates cardiovascular risk quickly

Both lead to early-onset MI!
Can we identify which mechanism is operating?

Current Model: Only Baseline Effects

Individual signature trajectories:

$$\lambda_{ik}(t) \mid g_i = \underbrace{r_k + g_i^\top \Gamma_k}_{\text{mean (constant over time)}} + \underbrace{E_i(t)}_{\text{GP noise}}$$

where $E_i(t) \sim \mathcal{GP}(0, \Omega_\lambda)$ and E_i **is independent of** g_i **What this captures:**

- Genetics $\rightarrow$ shifts baseline level ($g_i^\top \Gamma_k$)
- High CVD PRS $\rightarrow$ higher λ_{CVD} at all ages

What this CANNOT capture:

- Genetics $\rightarrow$ affects progression speed
- All temporal variation is random GP noise ($E_i(t)$)
- Steep trajectories are treated as random individual variation

Key Principle: Mean and Variance are Independent

General Gaussian Process formulation:

$$X \mid \mu = \mu + E, \quad E \sim \mathcal{N}(0, \Sigma)$$

Critical property: E is independent of μ **For** ϕ**:**

$$\phi_{kd} \mid (\mu_d, \psi_{kd}) = \mu_d + \psi_{kd} + E, \quad E \sim \mathcal{N}(0, \Omega_\phi)$$

- Variance Ω_ϕ does NOT depend on mean $(\mu_d + \psi_{kd})$

For λ **(current):**

$$\lambda_{ik} \mid \mathrm{g}_i = r_k + \mathrm{g}_i^\top \Gamma_k + E_i(t), \quad E_i(t) \sim \mathcal{GP}(0, \Omega_\lambda)$$

- Variance Ω_λ does NOT depend on genetics (g_i)
- Therefore: No systematic genetic effects on steepness!

Extended Model: Add Genetic Slope to Mean

Individual signature trajectories:

$$\lambda_{ik}(t) \mid (\mathrm{g}_i, t) = \underbrace{r_k + \mathrm{g}_i^\top \Gamma_k^{\text{level}} + t \cdot \mathrm{g}_i^\top \Gamma_k^{\text{slope}}}_{\text{mean (NOW time-varying)}} + \underbrace{E_i(t)}_{\text{GP noise}}$$

where $E_i(t) \sim \mathcal{GP}(0, \Omega_\lambda)$ and E_i **is still independent of** g_i **and** t **What this captures:**

- Γ_k^{level}: Genetics $\rightarrow$ baseline level (as before)
- Γ_k^{slope}: Genetics $\rightarrow$ progression speed (NEW!)
- High CVD PRS $\rightarrow$ higher baseline AND steeper trajectory

Key insight: Progression speed is now in the **mean function**, not in the variance!

Why Genetic Slope IS Identifiable

Separation of systematic vs. random variation:

Systematic (Mean)	Random (Variance)
r_k (population baseline)	Ω_λ (GP covariance)
$g_i^\top \Gamma_k^{\text{level}}$ (genetic level)	Same for all individuals
$g_i^\top \Gamma_k^{\text{slope}}$ (genetic speed) NEW!	Independent of genetics

Why identifiable:

1. Mean and variance are **independent parameters**
2. If high-PRS individuals systematically progress faster: shows up in mean slope (Γ_k^{slope})
3. If progression speed is random: shows up in GP covariance (Ω_λ), same for everyone
4. These are **separate effects** $\rightarrow$ identifiable!

Concrete Example

Data pattern from 1000 individuals:

Group	Age 40	Age 60	Slope
High CVD PRS (n=500)	$\lambda = 0.4$	$\lambda = 0.8$	0.02/year
Low CVD PRS (n=500)	$\lambda = 0.3$	$\lambda = 0.5$	0.01/year

Current model: Explains level via Γ_k; treats slope difference as random GP noise **Extended model:**

Explains level via Γ_k^{level}, slope via Γ_k^{slope}

Why NOT Warping ϕ?

Warping idea: Each person experiences population template at different speed

$$\pi_{idt} = \kappa \sum_k \theta_{ikt} \cdot \text{sigmoid}(\phi_{kd}(t^{\rho_i}))$$

where $\rho_i = f(g_i)$ **Problem: Not identifiable!**

- Scenario A: Person has $\rho_i = 1.5$ (fast), moderate θ_{ikt}
- Scenario B: Person has $\rho_i = 1$ (normal), steep θ_{ikt}

These produce **identical** π_{idt}: $\phi_{kd}(t)$ and θ_{ikt} are flexible → can't separate warping from steep trajectory. **Genetic slope avoids this:** $\phi_{kd}(t)$ stays fixed; genetics affects trajectories via parametric slope → clearer separation.

What We Gain: Distinguishing Two Mechanisms

Person A: High baseline, slow progression

$$\mathbb{E}[\lambda_{i,\text{CVD}}(t)] = \underbrace{0.6}_{\text{high level}} + \underbrace{0.005 \cdot t}_{\text{slow slope}}$$

- High CVD signature from age 30; moderate increase; early MI due to consistently high risk

Person B: Normal baseline, fast progression

$$\mathbb{E}[\lambda_{i,\text{CVD}}(t)] = \underbrace{0.3}_{\text{normal level}} + \underbrace{0.025 \cdot t}_{\text{fast slope}}$$

- Normal at 30; rapid increase (5× faster); early MI due to rapid accumulation

Different biology → different interventions!

Implementation: Simple Code Change

Current code:

```
# Compute lambda mean
lambda_mean = r_k + G @ Gamma_k  # (N, K)
```

Extended code:

```
# Compute lambda mean with time-varying genetic effect
t_centered = ages - 30  # (T,)
lambda_mean = (r_k +
               G @ Gamma_k_level +
               (G @ Gamma_k_slope)[:, :, None] *
                   t_centered[None, None, :])
```

That's it! 5 lines of code.

So We Can Identify Genetic Slopes—But a Catch

The extended model identifies genetic effects on progression speed **However:** $\theta = \mathrm{softmax}(\lambda)$ sums to

1.
$\Rightarrow$ Only **relative** slopes are identifiable (which signature grows faster than others). **Can we get**

***absolute* slopes?**
Yes—if we use the health signature as a calibration anchor.

Standard Model: Only RELATIVE Slopes Identifiable

- $\theta = \mathrm{softmax}(\lambda)$, so $\sum_k \theta_k = 1$.
- $\lambda_{ik}(t) = r_k + g_i^\top \gamma_{\mathrm{level},k} + t \cdot g_i^\top \gamma_{\mathrm{slope},k} + \epsilon_{ik}(t)$
- **Scale invariance:** for any constant c,

$$\theta = \mathrm{softmax}(\lambda) = \mathrm{softmax}(\lambda + c1_K).$$

- Adding the same slope c to all γ_{slope} leaves θ unchanged.
- $\Rightarrow$ Only **relative** slopes are identifiable.

Health Signature as Calibration Anchor

Idea: Use the health signature (e.g., Sig 20) with **person-specific initialization**. **Model with health:**

$$\lambda_{i,k}(t) = \begin{cases} \alpha_i + \beta_0 t + \epsilon_{i0}(t) & k = 0 \text{ (health)} \\ r_k + g_i^\top \gamma_{\text{level},k} + t \cdot g_i^\top \gamma_{\text{slope},k} + \epsilon_{ik}(t) & k = 1, \ldots, K-1 \end{cases}$$

- α_i: **person-specific** health baseline (from genetics, baseline phenotype, etc.).
- Health has its own slope β_0 (can grow or shrink).
- $\theta = \text{softmax}(\lambda)$ still sums to 1.

Why This Breaks Scale Invariance

- If we add a constant c to all **disease** λ's:

$$\lambda'_{ik} = \lambda_{ik} + c \cdot 1_{k \neq 0}$$

- Health λ_{i0} is **unchanged**.
- So θ **changes**—the health vs. disease balance shifts.
- The person-specific α_i anchor the health baseline; we can no longer freely shift all λ's.
- $\Rightarrow$ **Scale is pinned.** Absolute slopes become identifiable (relative to the health anchor).

Bottom line: Person-specific health initialization breaks the softmax scale invariance and allows identification of **absolute** progression speeds.

Extension: Genetic Slopes on Health

Health can also have genetic effects:

$$\lambda_{i0}(t) = \alpha_i + g_i^\top \gamma_{\text{level},0} + t \cdot g_i^\top \gamma_{\text{slope},0} + \epsilon_{i0}(t)$$

- α_i still person-specific (fixed or strongly informed).
- $\gamma_{\text{slope},0}$: genetic effect on *health* progression speed.
- Disease slopes $\gamma_{\text{slope},k}$ for $k \geq 1$ are identifiable on an absolute scale because α_i breaks the invariance.

Identifiable $\neq$ Recoverable

Identifiability proof (notebook): Initialize near truth $\Rightarrow r = 0.99$.

But in practice: We don't know the true slopes!

- Initialize $\gamma_{\text{slope}} = 0$ (no prior knowledge)
- Initialize γ_{level} from regression of average disease burden on genetics
- Original formulation: $r = 0.77$ (relative), $r = -0.85$ (absolute) $\leftarrow$ wrong sign!

Why? In the original model, λ is a free parameter:

$$\mathcal{L} = \underbrace{-\mathbb{E}[\log p(Y \mid \pi)]}_{\text{NLL (no } \gamma_{\text{slope}}!)} + w \underbrace{\|\lambda - \lambda_{\text{mean}}(\gamma)\|^2}_{\text{GP penalty (only gradient source)}}$$

γ_{slope} only appears in the penalty—invisible to the data likelihood.

Fix 1: Reparameterize λ

Old: λ is a free $N \times K \times T$ parameter (76,500 d.o.f.)

$$\lambda \text{ (free)} \qquad \gamma_{\text{slope}} \text{ only in penalty}$$

New: Decompose λ into parametric mean + residual:

$$\lambda_{ik}(t) = \underbrace{r_k + \mathbf{g}_i^\top \gamma_{\text{level},k} + t \cdot \mathbf{g}_i^\top \gamma_{\text{slope},k}}_{\lambda_{\text{mean}}(\gamma)} + \underbrace{\delta_{ik}(t)}_{\text{residual}}$$

Now γ_{slope} flows through the forward pass:

$$\gamma_{\text{slope}} \to \lambda_{\text{mean}} \to \lambda \to \text{softmax} \to \theta \to \pi \to \text{NLL}$$

γ_{slope} gets gradient from the data, not just from the penalty.

```
def get_lambda(self):
    return self.get_lambda_mean() + self.delta
```

Fix 2: Two-Phase Training

Problem: δ has $N \times K \times T = 76{,}500$ parameters vs. γ_{slope}'s $P \times K = 15$.
δ is more expressive and absorbs temporal structure before slopes can learn.

Solution: Freeze δ first.

Phase 1: δ frozen

- $\gamma_{\text{slope}}, \gamma_{\text{level}}, \psi, \kappa$ learn
- Only way to change λ is through slopes
- Slopes **must** learn temporal structure
- 1000 epochs, LR = 0.008

Phase 2: δ unfrozen

- All parameters fine-tune jointly
- δ captures individual residuals
- AUC improves (0.60 $\rightarrow$ 0.77)
- Slopes continue to **improve**
- Early stopping on slope correlation

Fix 3: Proper GP Kernel on δ

δ should be smooth (individual noise), not systematic trends.

Old penalty (white noise): $\frac{1}{NKT}\sum_{i,k,t}\delta_{ik}(t)^2$
$\rightarrow$ Penalizes magnitude only; δ can still have linear trends.

New penalty (SE kernel): $\frac{1}{2}\delta^\top\Omega_\lambda^{-1}\delta$ via Cholesky solve
$\rightarrow$ Penalizes temporal structure; linear trends in δ are **expensive**.

Effect: Systematic temporal trends (slopes) are pushed into γ_{slope} where they belong. δ captures only smooth, individual-level residuals.

Matches the GP kernel in the production model (`clust_huge_amp_vectorized.py`):
SE kernel with $\ell = T/4$, weight $W = 10^{-4}$.

Recovery Results: From Zero Initialization

All runs start with $\gamma_{\text{slope}} = 0$ (no cheating).

Method	Standard (relative)	Health anchor (absolute)
Original (free λ)	$r = 0.77$	$r = -0.85$
+ Reparameterize	$r = 0.83$	$r = -0.92$
+ Two-phase training	$r = 0.82$	$r = 0.90$
+ GP kernel (full fix)	$r = 0.86$	$r = 0.91$
True init (identifiability)	$r = 0.99$	$r = 0.97$

Three ingredients:

1. Reparameterize: $\lambda = \lambda_{\text{mean}}(\gamma) + \delta$ (slopes get NLL gradient)
2. Two-phase: freeze δ first (slopes must learn)
3. GP kernel on δ: penalize temporal structure in residuals

Summary

Part 1—Genetic slopes are identifiable:

- Add Γ_k^{slope} to the mean of λ (separate from GP noise)
- Distinguishes baseline vs. progression-speed mechanisms

Part 2—From relative to absolute slopes:

- Softmax $\Rightarrow$ only relative slopes identifiable
- Health signature with person-specific α_i $\Rightarrow$ breaks scale invariance
- $\Rightarrow$ Absolute progression speeds identifiable

Part 3—Recovery from realistic initialization:

- Reparameterize: $\lambda = \lambda_{\text{mean}}(\gamma) + \delta$ (slopes get NLL gradient)
- Two-phase training: freeze δ first, then fine-tune jointly
- GP kernel on δ: penalizes temporal structure in residuals
- $r = 0.86$ (relative), $r = 0.91$ (absolute) from $\gamma_{\text{slope}} = 0$