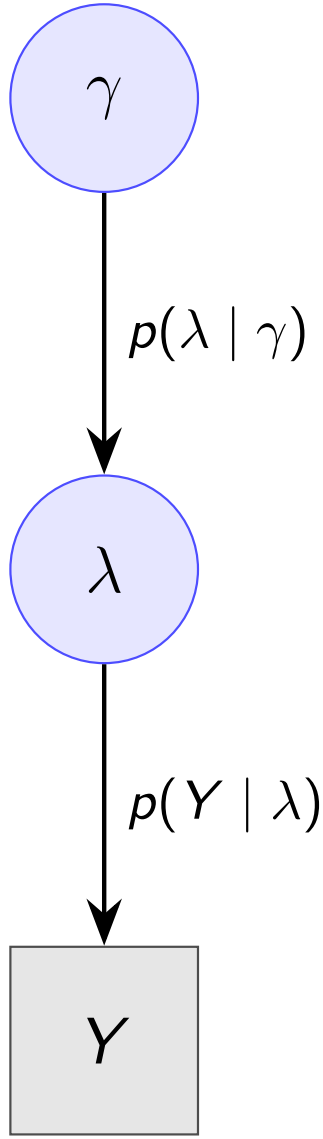
Gradient Flow in Joint MAP of Hierarchical Models

The Hierarchical Model



ALADYN:

Joint MAP Objective

Optimize **all parameters simultaneously**:

$$\underset{\lambda, \gamma}{\text{argmin}} \quad \underbrace{-\log p(Y \mid \lambda)}_{\text{NLL}} + \underbrace{-\log p(\lambda \mid \gamma)}_{\text{GP prior}}$$

Joint MAP Objective

Optimize **all parameters simultaneously**:

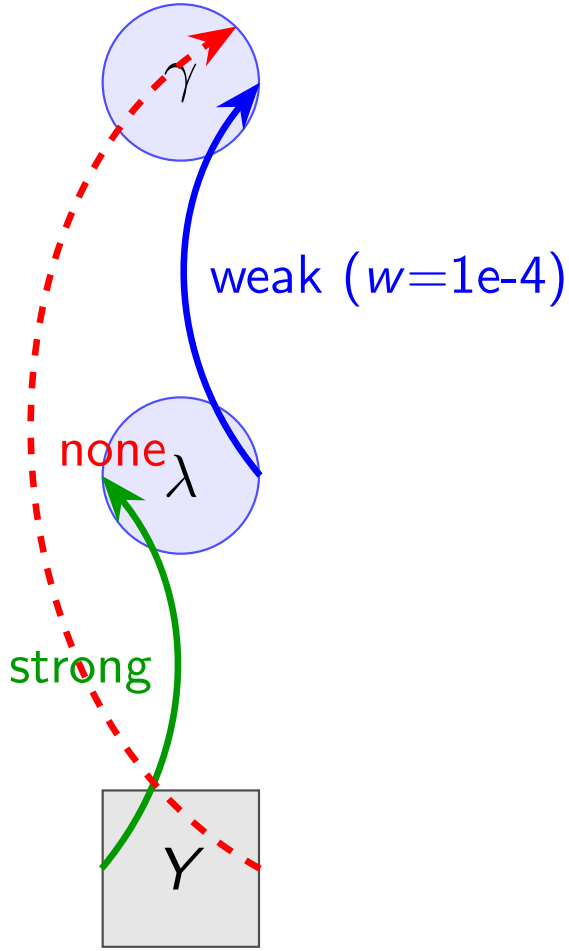
$$\underset{\lambda,\gamma}{\operatorname{argmin}} \underbrace{-\log p(Y \mid \lambda)}_{\text{NLL}} + \underbrace{-\log p(\lambda \mid \gamma)}_{\text{GP prior}}$$

Take gradients:

$$\frac{\partial \mathcal{L}}{\partial \lambda} = \underbrace{\frac{\partial}{\partial \lambda}\big[-\log p(Y \mid \lambda)\big]}_{\text{data signal}} + \underbrace{\frac{\partial}{\partial \lambda}\big[-\log p(\lambda \mid \gamma)\big]}_{\text{prior signal}}$$

$$\frac{\partial \mathcal{L}}{\partial \gamma} = \underbrace{\frac{\partial}{\partial \gamma}\big[-\log p(Y \mid \lambda)\big]}_{\text{zero — } \gamma \text{ not in likelihood}} + \underbrace{\frac{\partial}{\partial \gamma}\big[-\log p(\lambda \mid \gamma)\big]}_{\text{prior signal only}}$$

λ Screens γ from the Data



- λ gets **strong** gradient from data
- γ gets **weak** gradient from prior only
- γ gets **no** direct gradient from data

Why It Usually Doesn't Matter

λ compensates.

For prediction:

- Fix $\phi, \gamma, \psi, \kappa$ from training
- Re-estimate λ_{new} for each new individual
- λ_{new} has $K \times T$ free parameters
- Adapts to new data regardless of γ quality

Other examples:

- `lme4`: variance components weakly estimated, BLUPs still good
- Stan `optimizing`: hyperparameters only get prior gradient
- GAMs (`mgcv`): smoothing penalties co-estimated

Why It Usually Doesn't Matter

λ compensates.

For prediction:

- Fix $\phi, \gamma, \psi, \kappa$ from training
- Re-estimate λ_{new} for each new individual
- λ_{new} has $K \times T$ free parameters
- Adapts to new data regardless of γ quality

Other examples:

- `lme4`: variance components weakly estimated, BLUPs still good
- Stan `optimizing`: hyperparameters only get prior gradient
- GAMs (`mgcv`): smoothing penalties co-estimated

It only matters when γ is a primary scientific output.

Reparameterization: $\lambda = \mu(\gamma) + \delta$

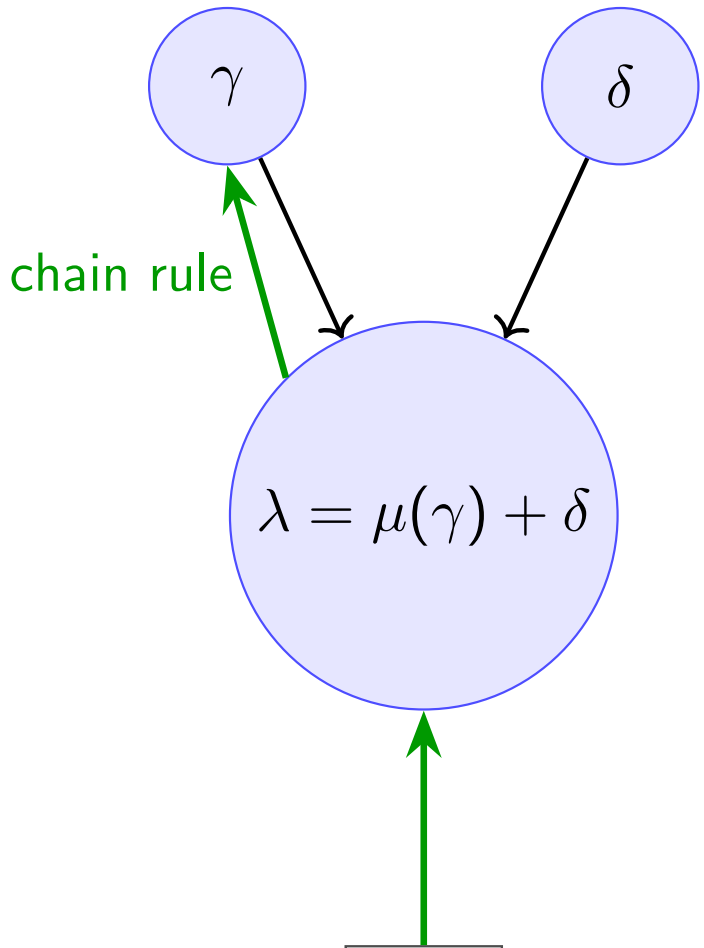
Change of variables (same MAP, different computation graph):

$$\underset{\delta,\,\gamma}{\operatorname{argmin}} \quad \underbrace{-\log p\big(Y \mid \mu(\gamma) + \delta\big)}_{\text{NLL — } \gamma \text{ is inside}} + \quad \underbrace{-\log p(\delta)}_{\text{GP prior on residual}}$$

Reparameterization: $\lambda = \mu(\gamma) + \delta$

Change of variables (same MAP, different computation graph):

$$\underset{\delta,\gamma}{\text{argmin}} \quad \underbrace{-\log p\big(Y \mid \mu(\gamma) + \delta\big)}_{\text{NLL — } \gamma \text{ is inside}} + \underbrace{-\log p(\delta)}_{\text{GP prior on residual}}$$



Same Loss, Different Gradients

Algebraically identical (substitute $\delta = \lambda - \mu(\gamma)$):

	Original	**Reparam**
Free params	λ, γ	δ, γ
NLL	$-\log p(Y \mid \lambda)$	$-\log p(Y \mid \mu(\gamma)$ -
Prior	$(\lambda - \mu(\gamma))^\top \Omega^{-1} (\lambda - \mu(\gamma))$	$\delta^\top \Omega^{-1} \delta$
γ gets NLL gradient?	No	Yes

Same Loss, Different Gradients

Algebraically identical (substitute $\delta = \lambda - \mu(\gamma)$):

	Original	**Reparam**
Free params	λ, γ	δ, γ
NLL	$-\log p(Y \mid \lambda)$	$-\log p(Y \mid \mu(\gamma)$ -
Prior	$(\lambda - \mu(\gamma))^\top \Omega^{-1} (\lambda - \mu(\gamma))$	$\delta^\top \Omega^{-1} \delta$
γ gets NLL gradient?	No	Yes

Same stationary points. Different optimization paths.

Analogy: minimize $f(x)$ with penalty $(x-5)^2$

- Original: optimize x directly
- Reparam: let $x = 5 + z$, optimize z with penalty z^2
- Same answer — but 5 gets gradient from f in the second form

Empirical Evidence: ALADYN

	Original	**Reparam**
Mean $\|\gamma\|$	0.006	0.081
γ correlation	0.37	
ψ correlation	0.76	
ϕ correlation	0.94	

- ϕ (disease trajectories): nearly identical — both fit the data well
- ψ (disease–signature assignments): moderately different — reparam less stable
- γ (genetic effects): very different — original $\gamma \approx 0$

- **Original γ is effectively zero** — genetic effects not estimated
- **Reparam γ is 14$\times$ larger** — but is it signal or overfitting?

Summary

1. In **all** joint MAP hierarchical models, hyperparameters (γ) get no gradient from the likelihood — only from the prior

2. This is **fine for prediction**: λ compensates, and we re-estimate λ for new individuals

3. It's a problem **only when γ is a scientific quantity of interest** (e.g., genetic slope recovery)

4. Reparameterization gives γ data signal via chain rule, at the cost of less stable ψ assignments

5. **Both are standard.** Centered (original) is the default in most software. Non-centered (reparam) is the standard alternative when hyperparameter estimation matters.