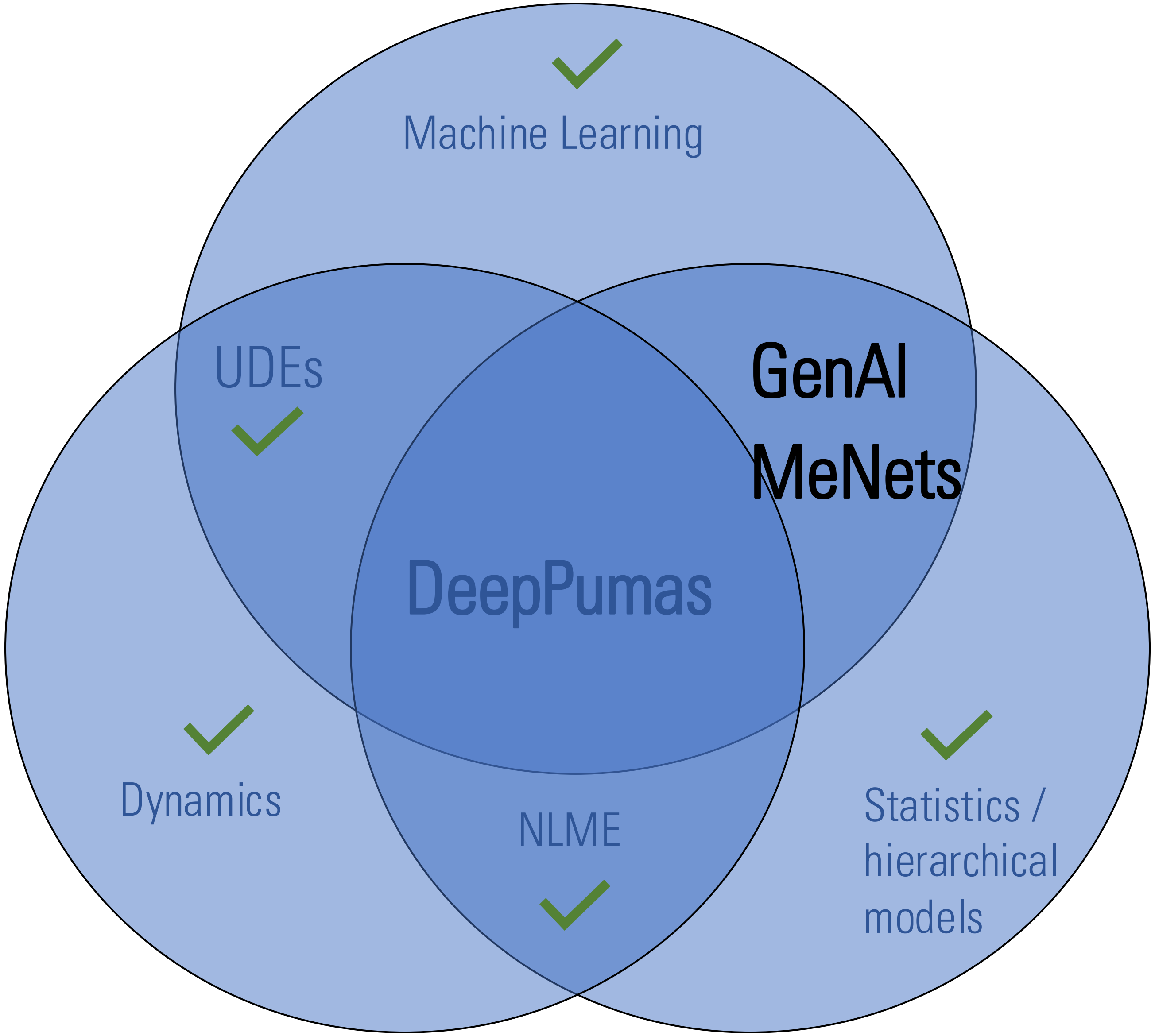
pumas AI

DeepPumas
Mixed effect neural nets

Niklas Korsbo

DeepPumas



Let's have a look at Generative AI and mixed effect neural networks!

What are Mixed Effects?

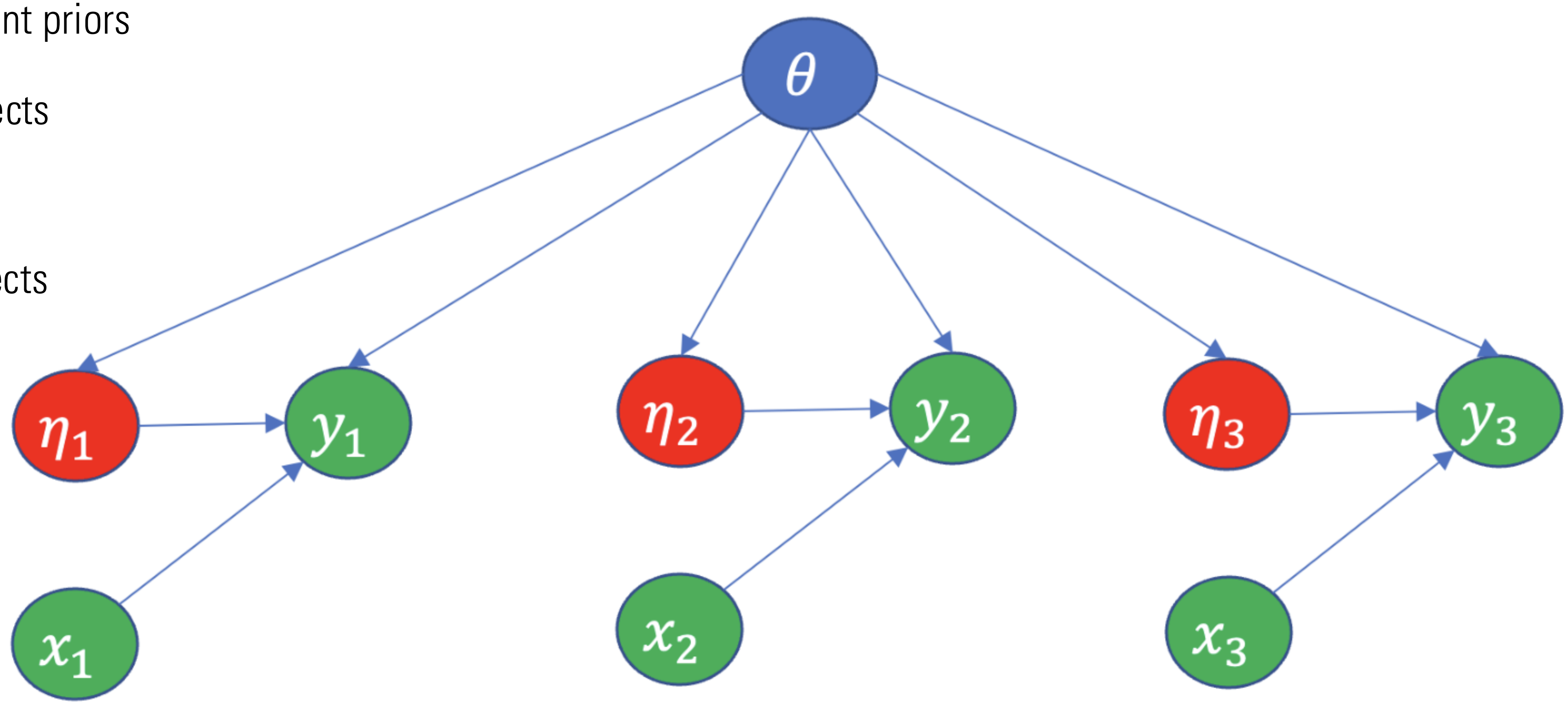
- Fixed effects, θ
  - Model parameters modelled as deterministic quantities
- Random effects, η
  - Model parameters modelled as random variables

<u>Hierarchical</u>

We typically define hierarchies where θ are shared parameters but η is subject-specific.

DeepPumas

- θ – all the population parameters.
    - Shared across subjects
- η – random effects of all subjects.
    - η_1 specific to subject 1
    - Typically has θ-dependent priors
- x – covariates of all the subjects
    - x_1 specific to subject 1
- y – responses of all the subjects
    - y_1 specific to subject 1



DeepPumas

In a Pumas model

θ

$\eta_i \mid \theta$

x_i

$y_i|\theta, \eta_i, x_i$

```
@model begin
  @param begin
    θ ∈ VectorDomain(4, lower = zeros(4))
    Ω ∈ PSDDomain(2)
    Σ ∈ RealDomain(lower = 0.0)
    a ∈ RealDomain(lower = 0.0, upper = 1.0)
  end
  @random begin
    η ~ MvNormal(Ω)
  end
  @covariates sex wt etn
  @pre begin
    θ1 := θ[1]
    Ka = θ1
    CL = θ[2] * ((wt / 70)^0.75) * (θ[4]^sex) *
      exp(η[1])
    Vc = θ[3] * exp(η[2])
  end
  @dynamics begin
    Depot' = -Ka * Depot
    Central' = Ka * Depot - (CL / Vc) * Central
    Res' = Depot - Central
  end
  @derived begin
    conc = @. Central / Vc
    dv ~ @. Normal(conc, conc * Σ)
    T_max = maximum(t)
  end
  @observed begin
    obs_cmax = maximum(dv)
  end
end
```

Don’t assign too much meaning to the random effects

- Indicates unknown parameters that vary between subjects (or whatever hierarchy we use)
- Usually tied very closely to a fixed effect in pharmacometrics. $CL = tvCL \cdot \exp(\eta_{cl})$.
- Enables degrees of freedom along which the model can account for outcome heterogeneity.

DeepPumas

Random effects during simulation?

Simple

Just sample and use

θ

$\eta_i \mid \theta$

x_i

$y_i \mid \theta, \eta_i, x_i$

```
@model begin
  @param begin
    θ ∈ VectorDomain(4, lower = zeros(4))
    Ω ∈ PSDDomain(2)
    Σ ∈ RealDomain(lower = 0.0)
    a ∈ RealDomain(lower = 0.0, upper = 1.0)
  end
  @random begin
    η ~ MvNormal(Ω)
  end
  @covariates sex wt etn
  @pre begin
    θ1 := θ[1]
    Ka = θ1
    CL = θ[2] * ((wt / 70)^0.75) * (θ[4]^sex) *
      exp(η[1])
    Vc = θ[3] * exp(η[2])
  end
  @dynamics begin
    Depot' = -Ka * Depot
    Central' = Ka * Depot - (CL / Vc) * Central
    Res' = Depot - Central
  end
  @derived begin
    conc = @. Central / Vc
    dv ~ @. Normal(conc, conc * Σ)
    T_max = maximum(t)
  end
  @observed begin
    obs_cmax = maximum(dv)
  end
end
```

Fitting with random effects

Their effect is largely determined by how they contribute to the loss function of a model fit

Conditional probability

Probability of the response $\boldsymbol{y}$ according to the model given specific values of $\boldsymbol{\theta}$, $\boldsymbol{\eta}$ and $\boldsymbol{x}$

$$p_c(\boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{x})$$

Fit model by simply finding the values of $\boldsymbol{\theta}$ and $\boldsymbol{\eta}$ that maximizes the conditional probability?

Fitting with random effects

Their effect is largely determined by how they contribute to the loss function of a model fit

<u>Marginal probability (!)</u>

Integrates out the effect of the random effects

$$p_m(\boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{x}) = \int p_c(\boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{x}) \cdot p_{prior}(\boldsymbol{\eta} \mid \boldsymbol{\theta}) \, d\boldsymbol{\eta}$$

Average conditional probability weighted by a prior.

Different methods/approximations: Laplace, FOCE and EM

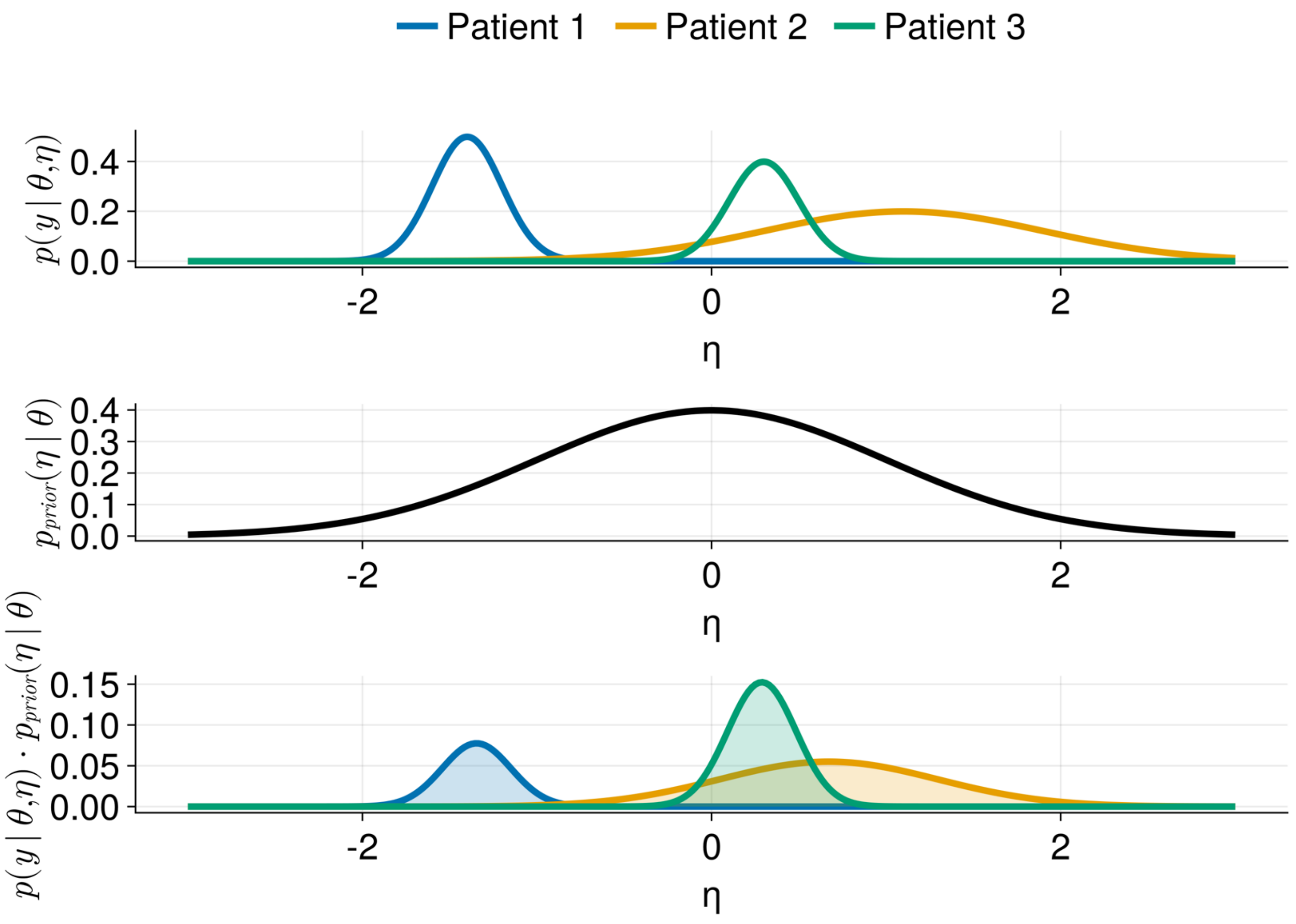
DeepPumas

Fitting with random effects

Their effect is largely determined by how they contribute to the loss function of a model fit

Conditional likelihood

Marginal likelihood



DeepPumas

Fitting algorithms in DeepPumas

	Marginal likelihood	Conditional likelihood
Prior / regularization on θ	MAP(FO()) MAP(FOCE()) MAP(LaplaceI())	JointMAP()
No prior / regularization on θ	FO() FOCE() LaplaceI()	N/A

Mixed effect neural networks (MeNets)

- What happens when you use random effects as part of your NN input?

- Let’s see in an exercise!