# Accelerating Access to Life-Saving Treatments to Patients

pumas AI


# Augment Workflow

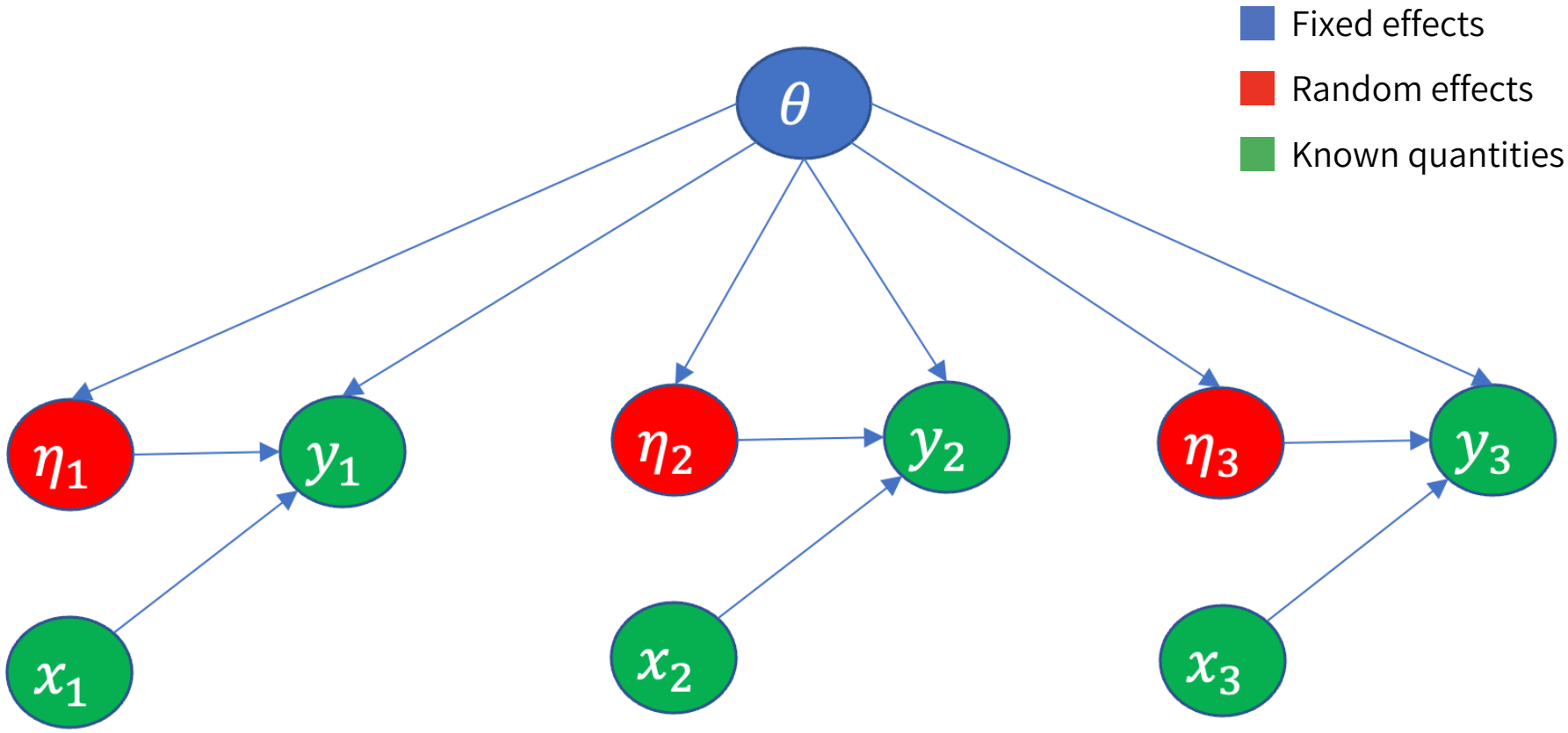
DeepPumas Workshop

**Prepared by: Mohamed Tarek**

Pumas-AI Inc

Nov 4, 2023

# Classical NLME Model



# Pumas Model

$\theta$

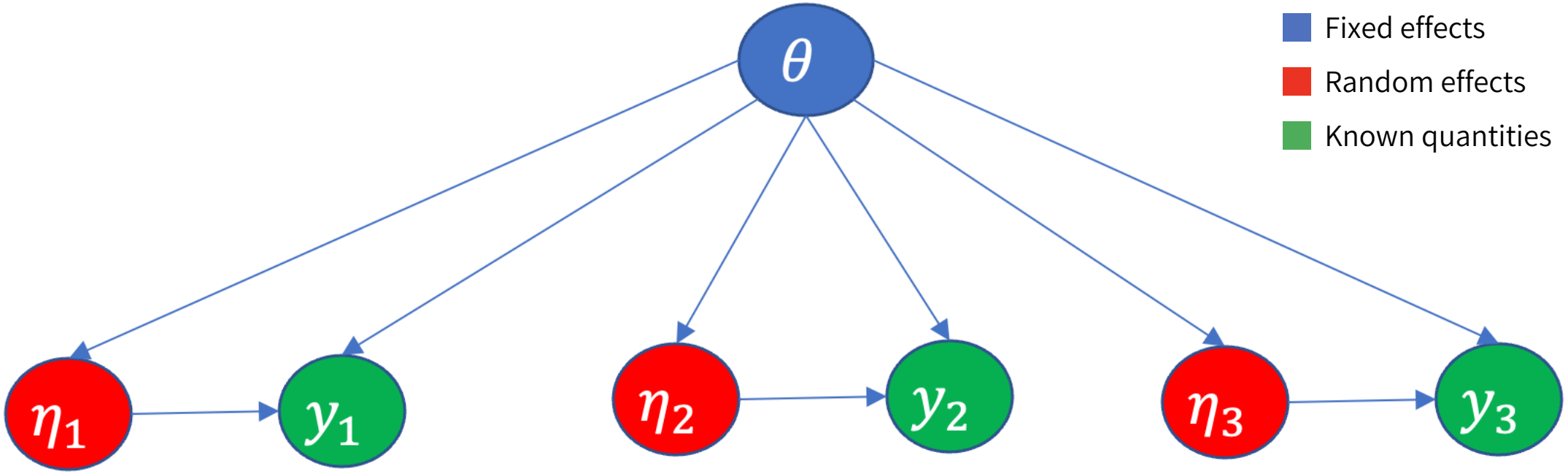
$\eta_i \mid \theta$

$x_i$

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

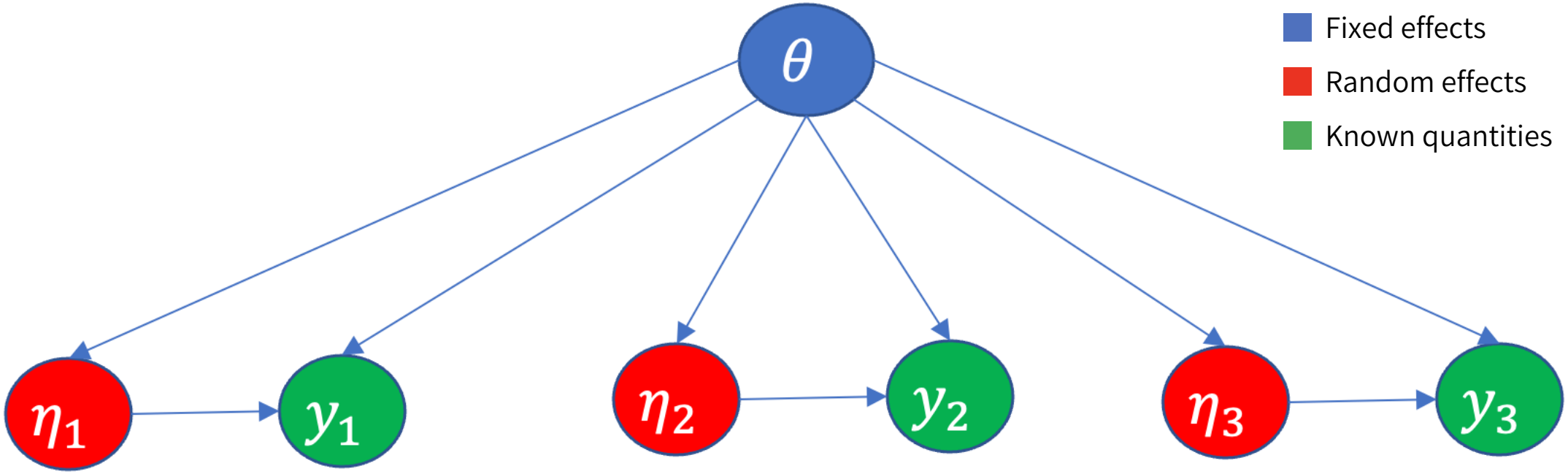
```
@model begin
  @param begin
    θ ∈ VectorDomain(4, lower = zeros(4))
    Ω ∈ PSDDomain(2)
    Σ ∈ RealDomain(lower = 0.0)
    a ∈ RealDomain(lower = 0.0, upper = 1.0)
  end
  @random begin
    η ~ MvNormal(Ω)
  end
  @covariates sex wt etn
  @pre begin
    θ1 := θ[1]
    Ka = θ1
    CL = θ[2] * ((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
```

# Covariate Free Model



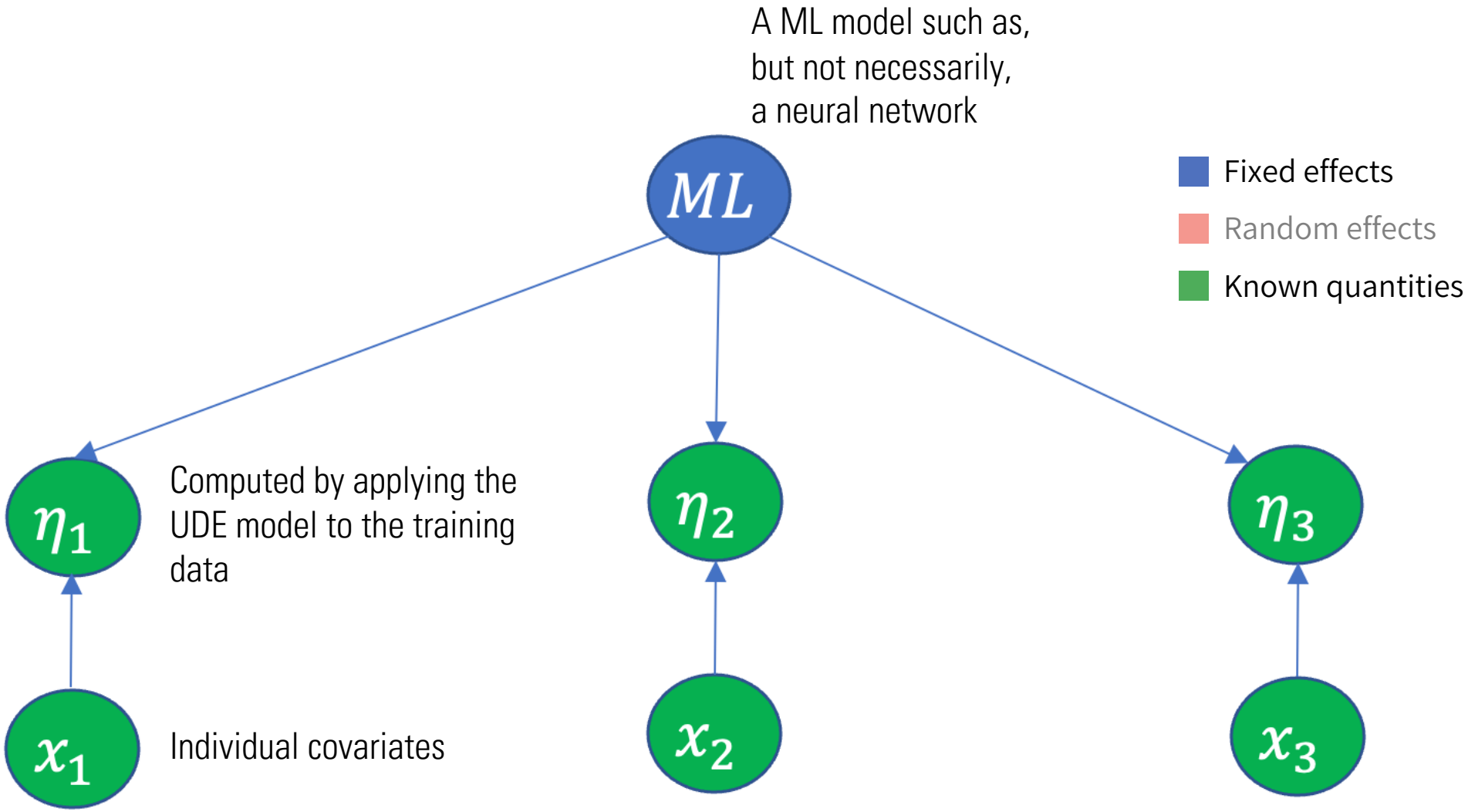

The NLME UDE models we have seen are covariate free!
They cannot exploit known heterogeneity in the patients to give better predictions!
In absence of observed data, their best prediction is independent of the individual under consideration.
This is far from ideal.

# Covariate Free Model

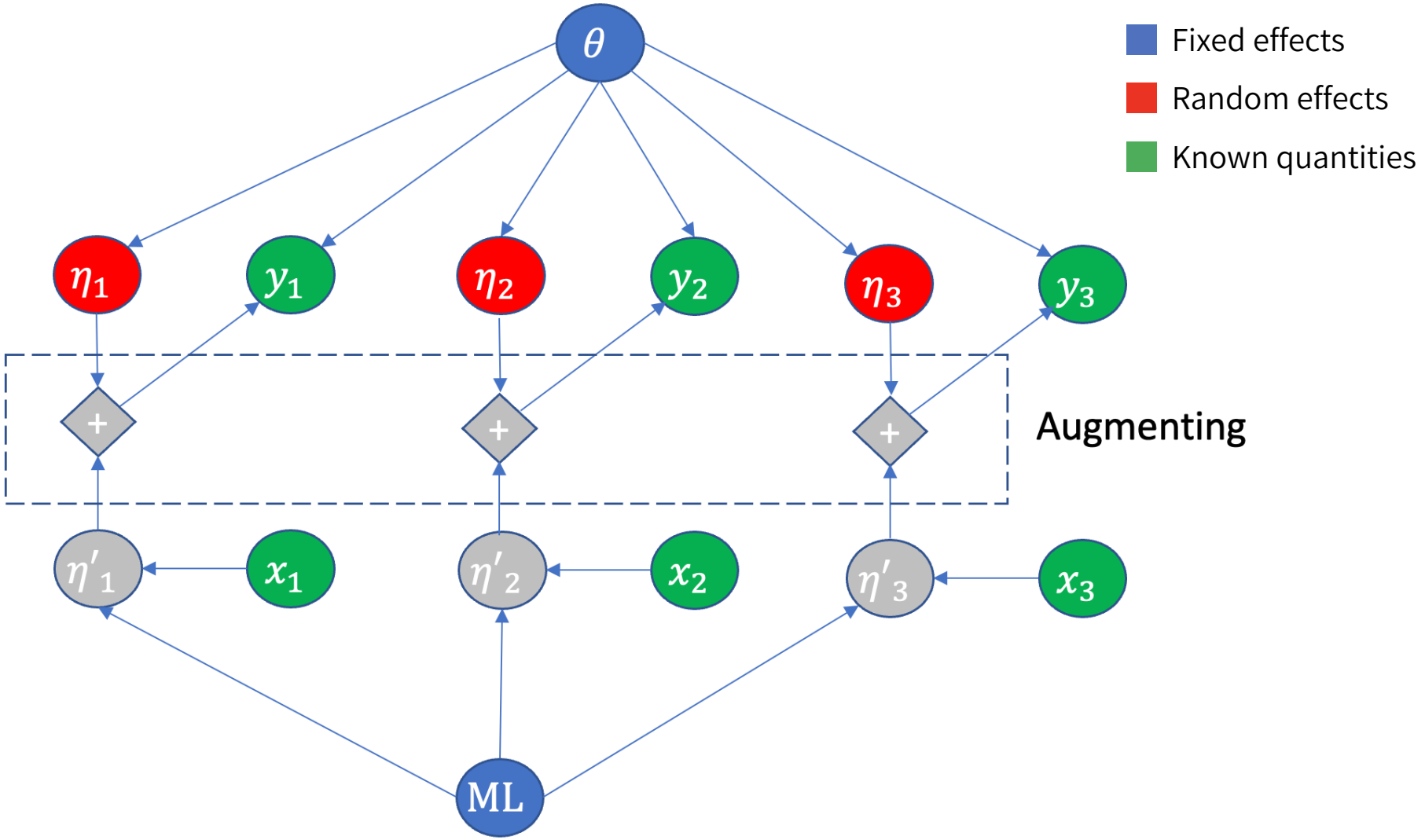



However, if we could find a function mapping the covariates to the random effects of the UDE model, we could give better predictions even in absence of observed data.

# Supervised Learning



# Augmented Model



# In short

1. Fit an NLME model to describe individual time courses using random effects
2. Extract an approximation of the posterior distribution of the random effects for each subject in the training data.
3. Fit a machine learning model to predict these posterior distributions from covariates
4. Augment the original NLME model with the machine learning prediction of the random effect value.