Inference Techniques
+
Variational Autoencoders

EM Derivation: The ELBO

$$F(\phi, \theta) = \log p(x|\theta) - KL\Big(q(z|\phi)||p(z|x,\theta)\Big)$$

- **KL Divergence**
  - Measures how different two distributions are
  - Is greater than or equal to 0
    - $F(\phi, \theta)$ is a lower bound on the evidence, $\log p(x|\theta)$
    - When $q(z|\phi) = p(z|x,\theta)$, the bound is tight.

- **E-step**: Update ϕ by setting $q(z|\phi) = p(z|x,\theta)$

https://pillowlab.princeton.edu/teaching/statneuro2020/notes/notes18_LatentVariableModels.pdf

A Note on KL Divergence and “Expectations”

$$E[x] = \sum_i P(x_i) \cdot x_i$$

A Note on KL Divergence and “Expectations”

$$E[x] = \sum_i P(x_i) \cdot x_i$$

$$E_{P(x)}[x] = \sum_i P(x_i) \cdot x_i$$

A Note on KL Divergence and “Expectations”

$$E[x] = \sum_i P(x_i) \cdot x_i$$

$$E_{P(x)}[x] = \sum_i P(x_i) \cdot x_i$$

$$E_{P(x)}[f(x)] = \sum_i P(x_i) \cdot f(x_i)$$

A Note on KL Divergence and “Expectations”

$$E[x] = \sum_i P(x_i) \cdot x_i$$

$$E_{P(x)}[x] = \sum_i P(x_i) \cdot x_i$$

$$E_{P(x)}[f(x)] = \sum_i P(x_i) \cdot f(x_i)$$

- Can estimate expectations via sampling

A Note on KL Divergence and “Expectations”

$$E[x] = \sum_i P(x_i) \cdot x_i$$

$$KL(Q||P) = \sum_x Q(x) \log\left(\frac{Q(x)}{P(x)}\right)$$

$$E_{P(x)}[x] = \sum_i P(x_i) \cdot x_i$$

$$E_{P(x)}[f(x)] = \sum_i P(x_i) \cdot f(x_i)$$

- Can estimate expectations via sampling

A Note on KL Divergence and “Expectations”

$$E[x] = \sum_i P(x_i) \cdot x_i$$

$$E_{P(x)}[x] = \sum_i P(x_i) \cdot x_i$$

$$E_{P(x)}[f(x)] = \sum_i P(x_i) \cdot f(x_i)$$

$$KL(Q||P) = \sum_x Q(x) \log\left(\frac{Q(x)}{P(x)}\right)$$

$$KL(Q||P) = E_{Q(x)}\left[\log\left(\frac{Q(x)}{P(x)}\right)\right]$$

- Can estimate expectations via sampling

A Note on KL Divergence and “Expectations”

$$E[x] = \sum_i P(x_i) \cdot x_i$$

$$E_{P(x)}[x] = \sum_i P(x_i) \cdot x_i$$

$$E_{P(x)}[f(x)] = \sum_i P(x_i) \cdot f(x_i)$$

- Can estimate expectations via sampling

$$KL(Q||P) = \sum_x Q(x) \log\left(\frac{Q(x)}{P(x)}\right)$$

$$KL(Q||P) = E_{Q(x)}\left[\log\left(\frac{Q(x)}{P(x)}\right)\right]$$

- Expectation of log of difference of probability distributions Q and P (drawing values from distribution Q)

EM Derivation: The ELBO

$$F(\phi, \theta) = \log p(x|\theta) - KL\Big(q(z|\phi)||p(z|x,\theta)\Big)$$

- **KL Divergence**
  - Measures how different two distributions are
  - Is greater than or equal to 0
    - $F(\phi, \theta)$ is a lower bound on the evidence, $\log p(x|\theta)$
    - When $q(z|\phi) = p(z|x,\theta)$, the bound is tight.

- **E-step**: Update ϕ by setting $q(z|\phi) = p(z|x,\theta)$

https://pillowlab.princeton.edu/teaching/statneuro2020/notes/notes18_LatentVariableModels.pdf

Approximate Inference

- We often can't analytically determine $p(z \mid x, \theta)$

$$p(z \mid x, \theta) = \frac{p(x \mid z, \theta)p(z \mid \theta)}{\int p(x \mid z, \theta)p(z \mid \theta)dz}$$

often intractable

Laplace (Quadratic) Approximation

- Estimate $p(z \mid x, \theta)$ as a Gaussian centered around a point estimate of *z*

Laplace (Quadratic) Approximation

- Estimate $p(z|x,\theta)$ as a Gaussian centered around a point estimate of *z*

- Find the Maximum a posteriori (MAP) estimate of $\log p(z|x,\theta)$

$$\hat{z} = \underset{z}{\operatorname{argmax}} \log p(z|x,\theta)$$

Laplace (Quadratic) Approximation

- Estimate $p(z|x,\theta)$ as a Gaussian centered around a point estimate of *z*
- Find the Maximum a posteriori (MAP) estimate of $\log p(z|x,\theta)$

$$\hat{z} = \underset{z}{\operatorname{argmax}} \log p(z|x,\theta)$$

- Estimate $p(z|x,\theta)$ as $\mathcal{N}(z|\hat{z}, \mathbf{H}^{-}1)$
  - Where $\mathbf{H}$ is the Hessian of $\log p(z,x|\theta)$ evaluated at $\hat{z}$

Laplace (Quadratic) Approximation

- Estimate $p(z|x,\theta)$ as a Gaussian centered around a point estimate of *z*
- Find the Maximum a posteriori (MAP) estimate of $\log\, p(z|x,\theta)$

$$\hat{z} = \underset{z}{\text{argmax}}\ \log\, p(z|x,\theta)$$

- Estimate $p(z|x,\theta)$ as $\mathcal{N}(z|\hat{z}, \mathbf{H}^{-}1)$
  - Where $\mathbf{H}$ is the Hessian of $\log\, p(z,x|\theta)$ evaluated at $\hat{z}$
- See Murphy book 2, section 7.4.3 for a proof with the Taylor expansion of $\log\, p(z,x|\theta)$

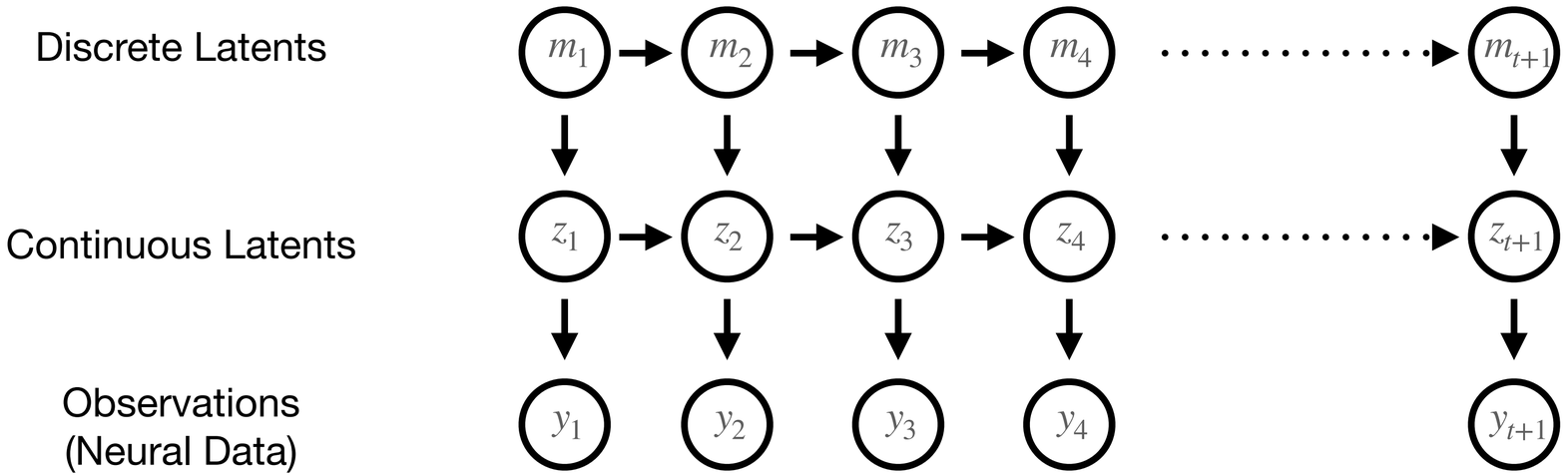
Laplace (Quadratic) Approximation

- Estimate $p(z|x,\theta)$ as a Gaussian centered around a point estimate of *z*
- Find the Maximum a posteriori (MAP) estimate of $\log\ p(z|x,\theta)$

$$\hat{z} = \underset{z}{\text{argmax}}\ \log\ p(z|x,\theta)$$

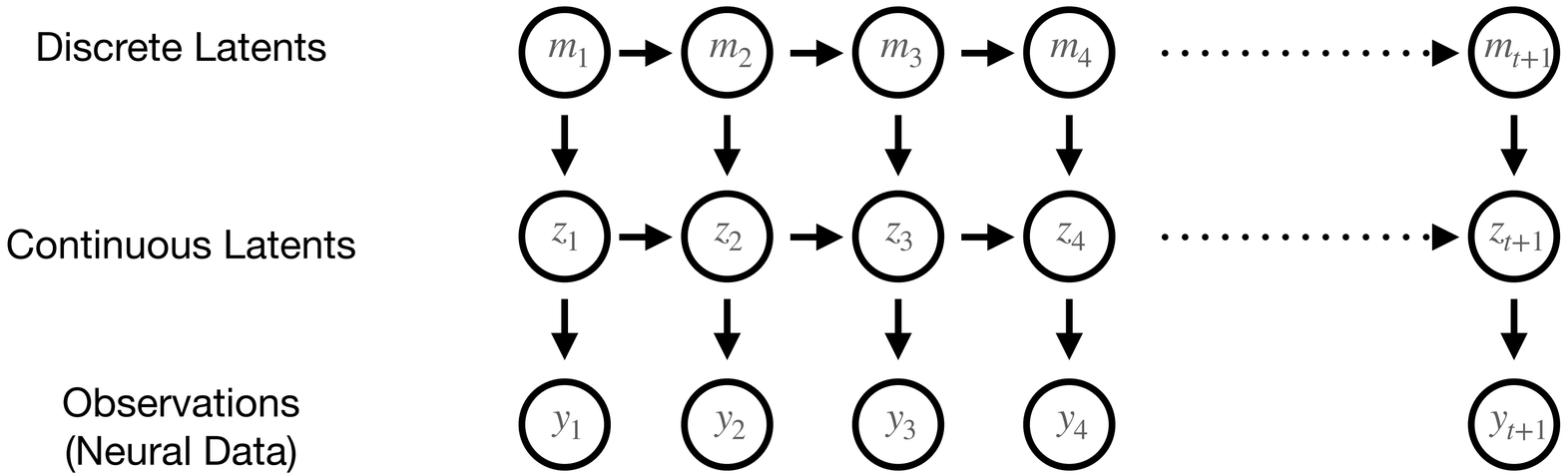
- Estimate $p(z|x,\theta)$ as $\mathcal{N}(z|\hat{z},\mathbf{H}^{-}1)$
  - Where $\mathbf{H}$ is the Hessian of $\log\ p(z,x|\theta)$ evaluated at $\hat{z}$
- Laplace Approximation is used in PLDS models!

(P)SLDS Model Fitting



- There are a number of ways to fit SLDS models - see Murphy book 2, chap 29.9

(P)SLDS Model Fitting



- There are a number of ways to fit SLDS models - see Murphy book 2, chap 29.9

- One is to assume a factorized posterior: $q(z, m) = q(z)q(m)$
  - Alternate between estimating $q(z)$, based on known discrete states, using Laplace approximation
  - And estimating $q(m)$ based on samples from $q(z)$ using forward/backward algorithm

Zoltowski et al 2020

Variational Inference

Variational Inference

$$F(\phi, \theta) = \log p(x|\theta) - KL\Big(q(z|\phi)||p(z|x,\theta)\Big)$$

- Goal is to assume a "variational distribution" of $q(z|\phi)$ (e.g. Gaussian), and directly optimize the parameters of that distribution:

$$\hat{\phi} = \underset{\phi}{\operatorname{argmin}}\ KL\left(q(z\,|\,\phi)\,||\ p(z\,|\,x,\theta)\right)$$

Variational Inference

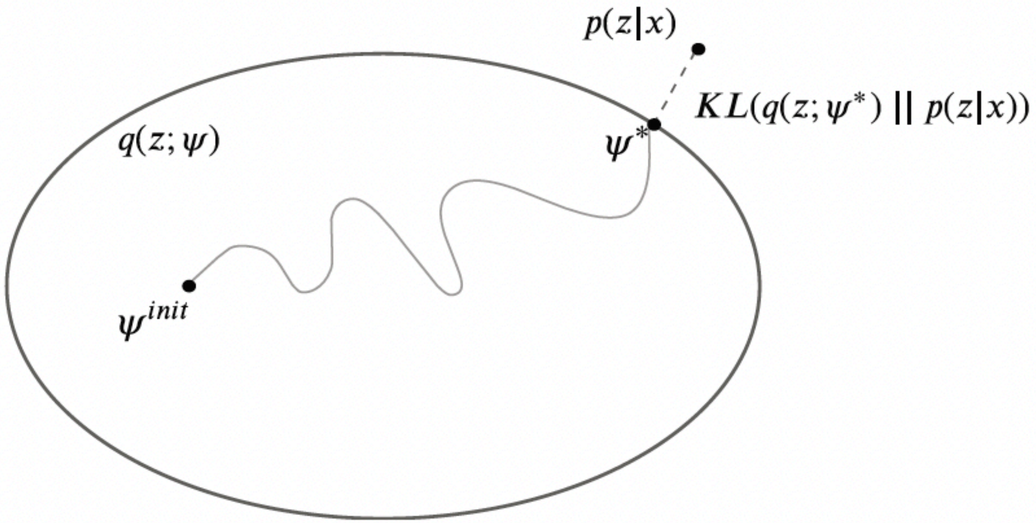


Figure 10.1: Illustration of variational inference. The large oval represents the set of variational distributions $\mathcal{Q} = \{q_{\boldsymbol{\psi}}(\boldsymbol{z}) : \boldsymbol{\psi} \in \ominus\}$, where $\ominus$ is the set of possible variational parameters. The true distribution is the point $p(\boldsymbol{z}|\boldsymbol{x})$, which we assume lies outside the set. Our goal is to find the best approximation to p within our variational family; this is the point $\boldsymbol{\psi}^*$ which is closest in KL divergence. We find this point by starting an optimization procedure from the random initial point $\boldsymbol{\psi}^{init}$. Adapted from a figure by David Blei.

Variational Inference

$$F(\phi, \theta) = \log p(x|\theta) - KL\Big(q(z|\phi)||p(z|x,\theta)\Big)$$

- Goal is to assume a "variational distribution" of $q(z|\phi)$ (e.g. Gaussian), and directly optimize the parameters of that distribution:

$$\hat{\phi} = \underset{\phi}{\text{argmin}}\ KL\left(q(z\,|\,\phi)\,||\ p(z\,|\,x,\theta)\right)$$

- Even when $q(z|\phi)$ is Gaussian, this differs from the Laplace approximation in that the parameters of the distribution get directly optimized.

Variational Inference

$$F(\phi, \theta) = \log p(x|\theta) - KL\Big(q(z|\phi)||p(z|x,\theta)\Big)$$

- Goal is to assume a "variational distribution" of $q(z|\phi)$ (e.g. Gaussian), and directly optimize the parameters of that distribution:

$$\hat{\phi} = \underset{\phi}{\operatorname{argmin}}\ KL\left(q(z\,|\,\phi)\,||\ p(z\,|\,x,\theta)\right)$$

- $q(z|\phi)$ does not need to be Gaussian

Variational Inference

$$F(\phi, \theta) = \log p(x|\theta) - KL\Big(q(z|\phi)||p(z|x,\theta)\Big)$$

- Goal is to assume a "variational distribution" of $q(z|\phi)$ (e.g. Gaussian), and directly optimize the parameters of that distribution:

$$\hat{\phi} = \underset{\phi}{\operatorname{argmin}}\ KL\left(q(z\,|\,\phi)\,||\ p(z\,|\,x,\theta)\right)$$

How can we calculate this when we can't calculate $p(z\,|\,x,\theta)$*?*

Variational Inference: Another form of the ELBO

Variational Inference: Another form of the ELBO

We can also rewrite the ELBO as

$$\text{Ł}(\boldsymbol{\psi}|\boldsymbol{\theta}, \boldsymbol{x}) = \mathbb{E}_{q_{\boldsymbol{\psi}}(\boldsymbol{z})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) + \log p_{\boldsymbol{\theta}}(\boldsymbol{z}) - \log q_{\boldsymbol{\psi}}(\boldsymbol{z})\right] \tag{10.13}$$

$$= \mathbb{E}_{q_{\boldsymbol{\psi}}(\boldsymbol{z})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\right] - D_{\mathbb{KL}}\left(q_{\boldsymbol{\psi}}(\boldsymbol{z}) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{z})\right) \tag{10.14}$$

Variational Inference: Another form of the ELBO

We can also rewrite the ELBO as

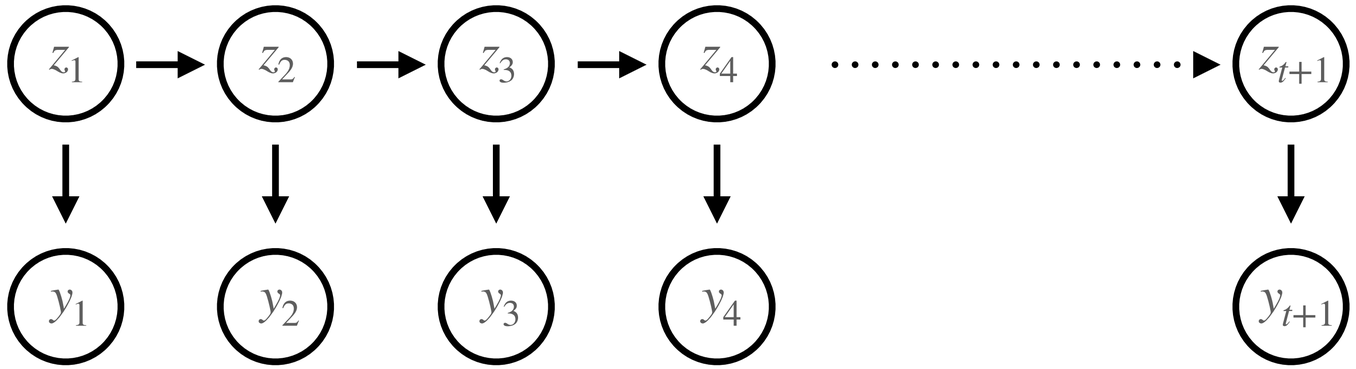
$$Ł(\boldsymbol{\psi}|\boldsymbol{\theta}, \boldsymbol{x}) = \mathbb{E}_{q_{\boldsymbol{\psi}}(\boldsymbol{z})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) + \log p_{\boldsymbol{\theta}}(\boldsymbol{z}) - \log q_{\boldsymbol{\psi}}(\boldsymbol{z})\right] \tag{10.13}$$

$$= \mathbb{E}_{q_{\boldsymbol{\psi}}(\boldsymbol{z})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\right] - D_{\mathbb{KL}} \left(q_{\boldsymbol{\psi}}(\boldsymbol{z}) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{z})\right) \tag{10.14}$$

$$\text{ELBO} = \text{expected log likelihood} - \text{KL from posterior to prior} \tag{10.15}$$

The KL term acts like a regularizer, preventing the posterior from diverging too much from the prior.

Example: Latent variable models with nonlinear emissions models (fLDS)

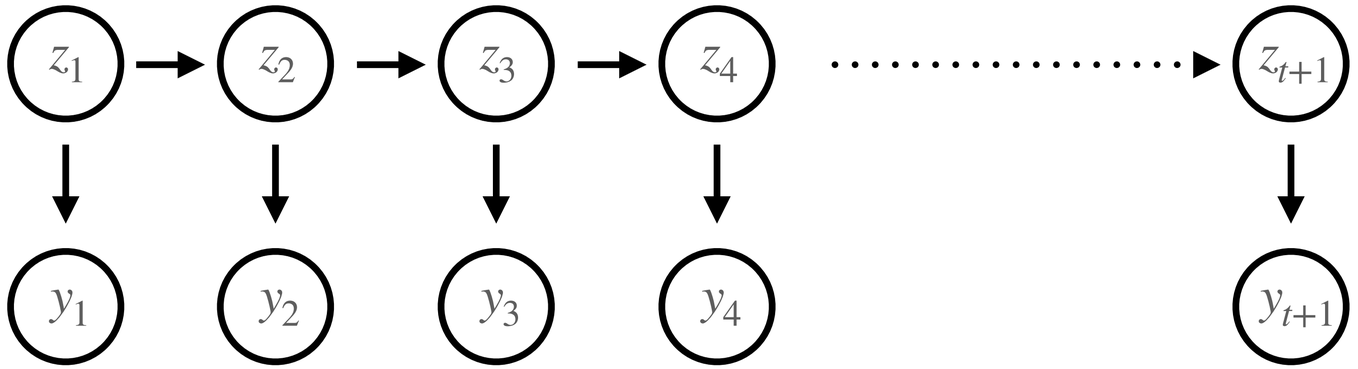


Linear Dynamics

$y_t = f(z_t) + \text{noise}$

Gao et al 2016

Example: Latent variable models with nonlinear emissions models (fLDS)



Linear Dynamics

$y_t = f(z_t) + \text{noise}$

- Note: Laplace Approximation does not work well because the Hessian is not easily tractable

Gao et al 2016

Gradient-based Variational EM: Solving for the parameters

- Need to estimate:
  - Parameters of the latent variable model (e.g. emissions), θ
  - Parameters of inference (e.g. mean/stdev of the latents), ϕ

Gradient-based Variational EM: Solving for the parameters

- Need to estimate:
  - Parameters of the latent variable model (e.g. emissions), θ
    - Sample from the latents and then get gradients of θ
  - Parameters of inference (e.g. mean/stdev of the latents), ϕ

Gradient-based Variational EM: Solving for the parameters

The gradient wrt the generative parameters $\boldsymbol{\theta}$ is easy to compute, since we can push gradients inside the expectation, and use a single Monte Carlo sample:

$$\nabla_{\boldsymbol{\theta}} \text{Ł}(\boldsymbol{\theta}, \boldsymbol{\phi}|\boldsymbol{x}) = \nabla_{\boldsymbol{\theta}} \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})\right] \tag{10.31}$$

$$= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \left[\nabla_{\boldsymbol{\theta}} \left\{\log p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})\right\}\right] \tag{10.32}$$

$$\approx \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}^s) \tag{10.33}$$

where $\boldsymbol{z}^s \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$. This is an unbiased estimate of the gradient, so can be used with SGD.

Gradient-based Variational EM: Solving for the parameters

- Need to estimate:
  - Parameters of the latent variable model (e.g. emissions), θ
    - Sample from the latents and then get gradients of θ
  - Parameters of inference (e.g. mean/stdev of the latents), ϕ

Gradient-based Variational EM: Solving for the parameters

- Need to estimate:
  - Parameters of the latent variable model (e.g. emissions), θ
    - Sample from the latents and then get gradients of θ
  - Parameters of inference (e.g. mean/stdev of the latents), ϕ
    - More complicated :)

Gradient-based Variational EM: Solving for the parameters

The gradient wrt the inference parameters $\boldsymbol{\phi}$ is harder to compute since

$$\nabla_{\boldsymbol{\phi}} \text{Ł}(\boldsymbol{\theta}, \boldsymbol{\phi}|\boldsymbol{x}) = \nabla_{\boldsymbol{\phi}} \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})\right] \tag{10.34}$$

$$\neq \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \left[\nabla_{\boldsymbol{\phi}} \left\{\log p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})\right\}\right] \tag{10.35}$$

Gradient-based Variational EM: Solving for the parameters

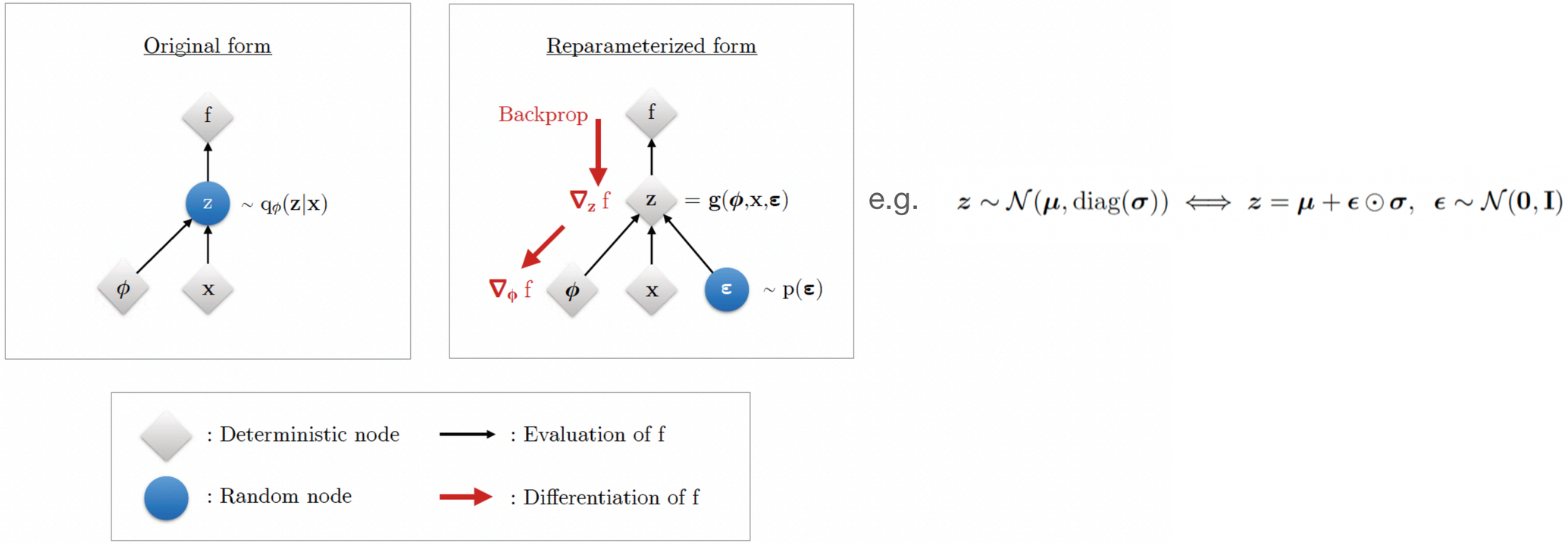
The gradient wrt the inference parameters $\boldsymbol{\phi}$ is harder to compute since

$$\nabla_{\boldsymbol{\phi}} \mathrm{L}(\boldsymbol{\theta}, \boldsymbol{\phi}|\boldsymbol{x}) = \nabla_{\boldsymbol{\phi}} \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})\right] \tag{10.34}$$

$$\neq \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \left[\nabla_{\boldsymbol{\phi}} \left\{\log p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})\right\}\right] \tag{10.35}$$

- Two standard approaches:
  - Blackbox Variational Inference
    - Uses fun math trick (score function estimator) to rewrite the above in a way that takes the expectation of a gradient
  - Reparameterization trick

Reparameterization Trick



e.g. $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma})) \iff \boldsymbol{z} = \boldsymbol{\mu} + \boldsymbol{\epsilon} \odot \boldsymbol{\sigma}, \;\; \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

Figure 10.4: Illustration of the reparameterization trick. The objective f depends on the variational parameters $\boldsymbol{\phi}$, the observed data $\boldsymbol{x}$, and the latent random variable $\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$. On the left, we show the standard form of the computation graph. On the right, we show a reparameterized form, in which we move the stochasticity into the noise source $\boldsymbol{\epsilon}$, and compute $\boldsymbol{z}$ deterministically, $\boldsymbol{z} = g(\boldsymbol{\phi}, \boldsymbol{x}, \boldsymbol{\epsilon})$. The rest of the graph is deterministic, so we can backpropagate the gradient of the scalar f wrt $\boldsymbol{\phi}$ through $\boldsymbol{z}$ and into $\boldsymbol{\phi}$. From Figure 2.3 of [KW19a]. Used with kind permission of Durk Kingma.

Reparameterization Trick

The key trick is to rewrite the random variable $\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ as some differentiable (and invertible) transformation g of another random variable $\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$, which does not depend on $\boldsymbol{\phi}$, i.e., we assume we can write

$$\boldsymbol{z} = g(\boldsymbol{\phi}, \boldsymbol{x}, \boldsymbol{\epsilon}) \tag{10.36}$$

For example,

$$\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma})) \iff \boldsymbol{z} = \boldsymbol{\mu} + \boldsymbol{\epsilon} \odot \boldsymbol{\sigma}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{10.37}$$

Using this, we have

$$\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[f(\boldsymbol{z})\right] = \mathbb{E}_{p(\boldsymbol{\epsilon})}\left[f(\boldsymbol{z})\right] \quad \text{s.t.} \quad \boldsymbol{z} = g(\boldsymbol{\phi}, \boldsymbol{x}, \boldsymbol{\epsilon}) \tag{10.38}$$

where we define

$$f_{\boldsymbol{\theta},\boldsymbol{\phi}}(\boldsymbol{z}) = \log p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \tag{10.39}$$

Hence

$$\nabla_{\boldsymbol{\phi}} \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[f(\boldsymbol{z})\right] = \nabla_{\boldsymbol{\phi}} \mathbb{E}_{p(\boldsymbol{\epsilon})}\left[f(\boldsymbol{z})\right] = \mathbb{E}_{p(\boldsymbol{\epsilon})}\left[\nabla_{\boldsymbol{\phi}} f(\boldsymbol{z})\right] \tag{10.40}$$

Black-box variational inference

10.2.3.1 Estimating the gradient using REINFORCE

To estimate the gradient of the ELBO, we will use the **score function estimator**, also called the **REINFORCE** estimator (Section 6.3.4). In particular, suppose we write the ELBO as

$$Ł(\boldsymbol{\psi}) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{\psi})}\left[\tilde{\mathcal{L}}(\boldsymbol{\psi}, \boldsymbol{z})\right] = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{\psi})}\left[\log p(\boldsymbol{x}, \boldsymbol{z}) - \log q(\boldsymbol{z}|\boldsymbol{\psi})\right] \tag{10.65}$$

Then from Equation (6.58) we have

$$\nabla_{\boldsymbol{\psi}} Ł(\boldsymbol{\psi}) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{\psi})}\left[\tilde{\mathcal{L}}(\boldsymbol{\psi}, \boldsymbol{z}) \nabla_{\boldsymbol{\psi}} \log q(\boldsymbol{z}|\boldsymbol{\psi})\right] \tag{10.66}$$

We can then compute a Monte Carlo approximation to this:

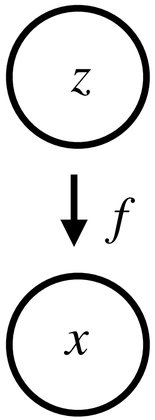
$$\widehat{\nabla_{\boldsymbol{\psi}} Ł(\boldsymbol{\psi}_t)} = \frac{1}{S}\sum_{s=1}^{S} \tilde{\mathcal{L}}(\boldsymbol{\psi}, \boldsymbol{z}_s) \nabla_{\boldsymbol{\psi}} \log q_{\boldsymbol{\psi}}(\boldsymbol{z}_s)|_{\boldsymbol{\psi}=\boldsymbol{\psi}_t} \tag{10.67}$$

We can pass this to any kind of gradient optimizer, such as SGD or Adam.

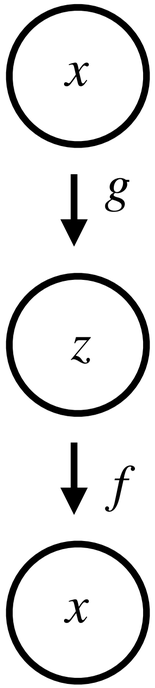
Variational Autoencoder (VAE)

Variational Autoencoder (VAE)

- Classically in latent variable models, we directly optimize ***z***
  - E.g., let's say ***z*** is a 1d Gaussian and ***x*** has *T* time points, then we would optimize a mean and stdev. for all *T* time points.



Variational Autoencoder (VAE)



- Classically in latent variable models, we directly optimize ***z***
  - E.g., let's say ***z*** is a 1d Gaussian and ***x*** has *T* time points, then we would optimize a mean and stdev. for all *T* time points.

- VAEs use "amortized inference", where ***z*** is learned as a function of ***x***, so only the parameters of *g* (neural network parameters) are learned
  - In the above example, there would be separate neural networks trained to predict ***z***'s means and stdevs: $\mu_t = g_\mu(x_t)$ and $\sigma_t = g_\sigma(x_t)$

VAEs vs. Standard Autoencoders

- Often perform similarly

VAEs vs. Standard Autoencoders

- Often perform similarly
- VAEs are better generative models

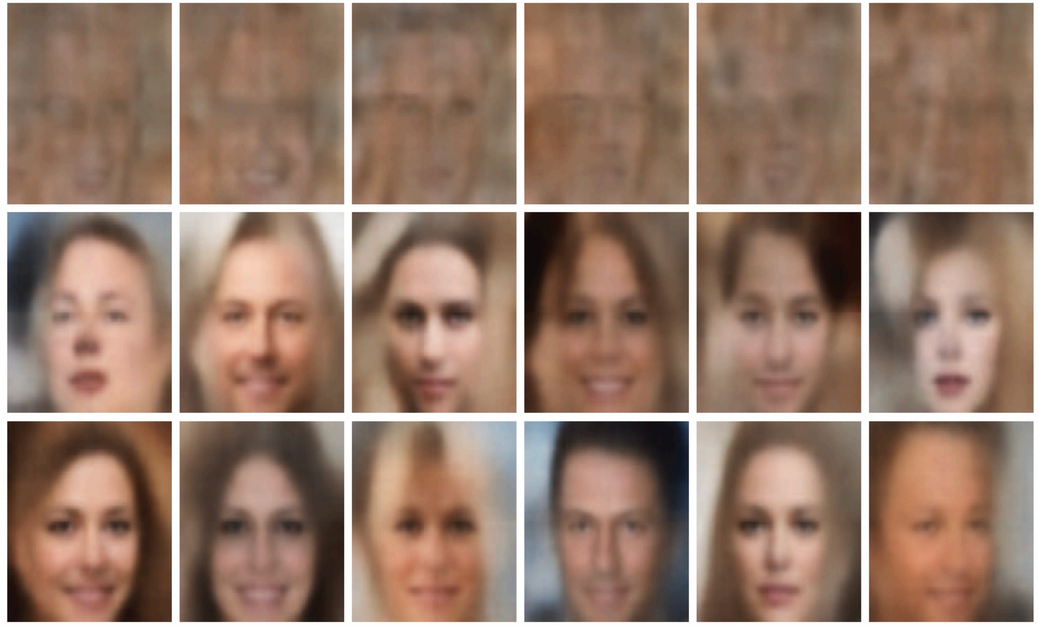


Figure 21.2: Illustration of unconditional image generation using (V)AEs trained on CelebA. Row 1: deterministic autoencoder. Row 2: β-VAE with $\beta = 0.5$. Row 3: VAE (with $\beta = 1$). Generated by celeba_vae_ae_comparison.ipynb.

VAEs vs. Standard Autoencoders

- Often perform similarly
- VAEs are better generative models

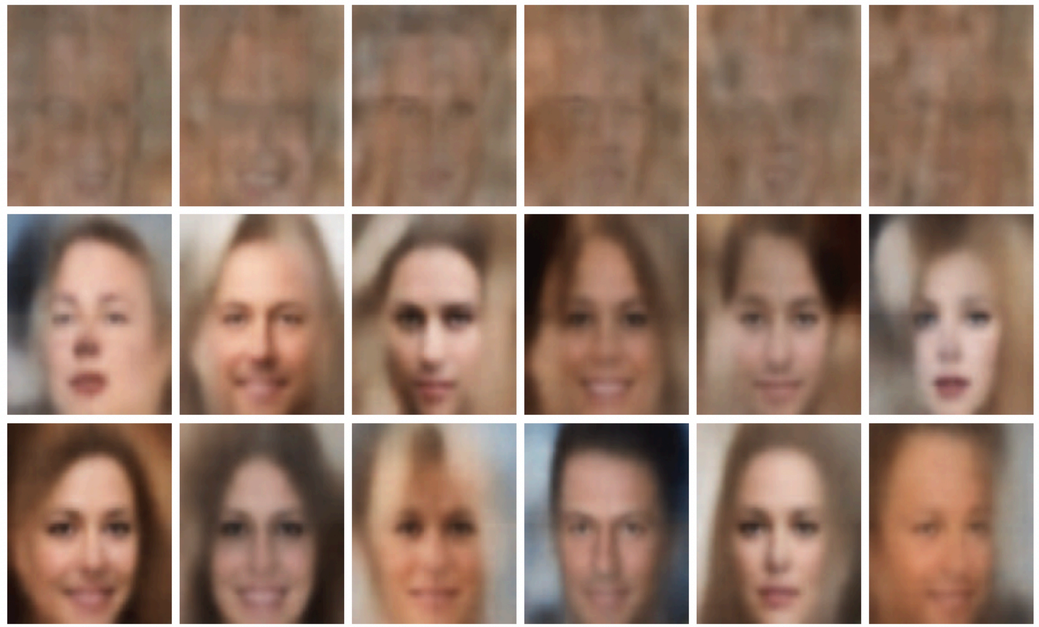
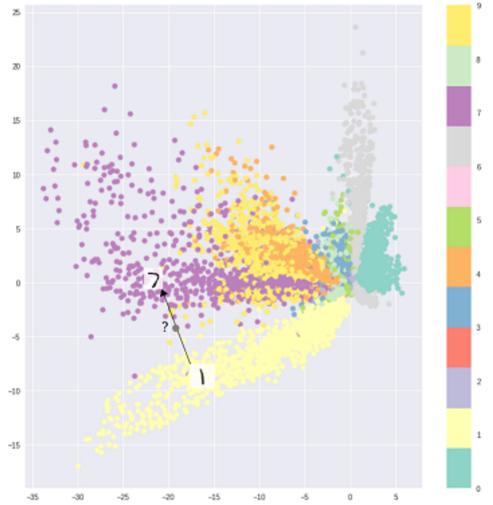


Figure 21.2: Illustration of unconditional image generation using (V)AEs trained on CelebA. Row 1: deterministic autoencoder. Row 2: β-VAE with $\beta = 0.5$. Row 3: VAE (with $\beta = 1$). Generated by celeba_vae_ae_comparison.ipynb.

AE Latents on MNIST



VAE Latents on MNIST

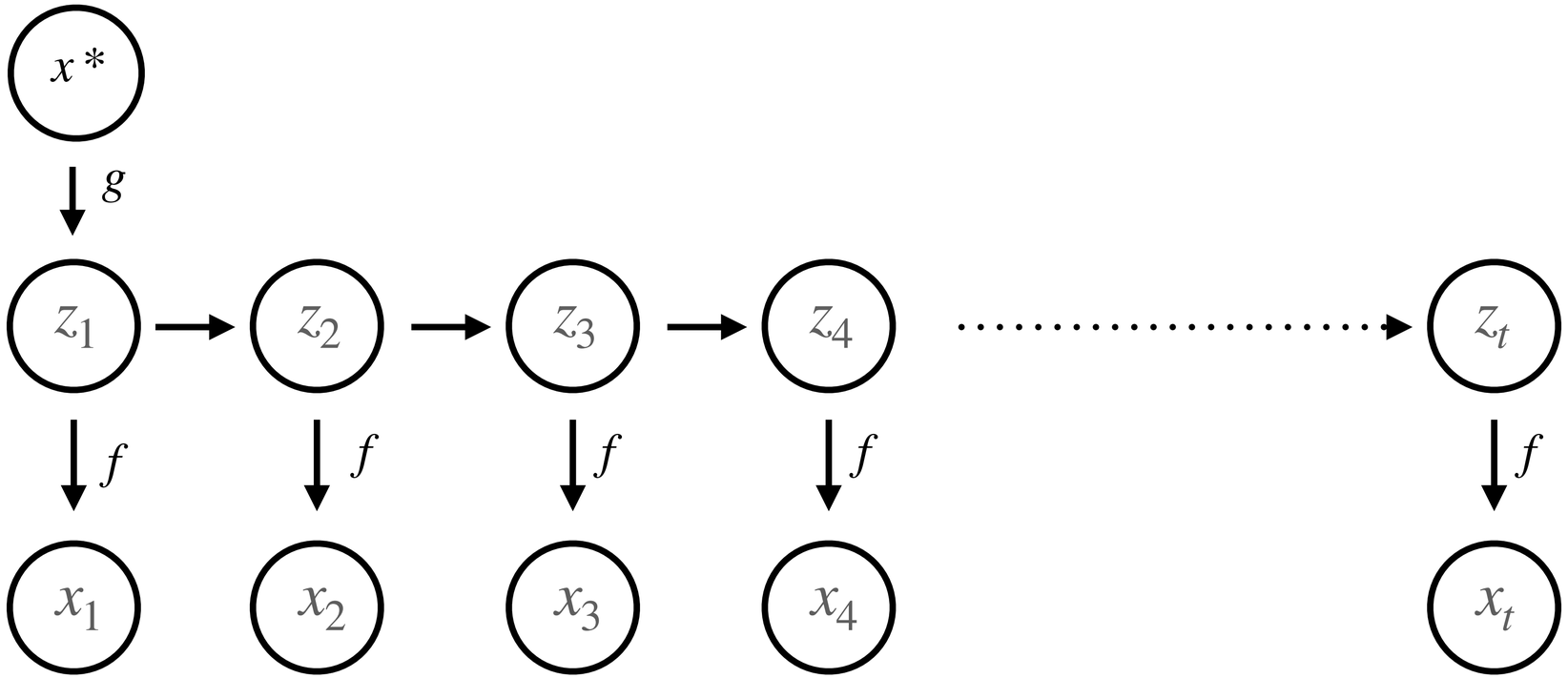
https://stats.stackexchange.com/questions/324340/when-should-i-use-a-variational-autoencoder-as-opposed-to-an-autoencoder

VAEs vs. Standard Autoencoders

- Often perform similarly
- VAEs are better generative models
- VAEs are flexible probabilistic models (can change prior, noise distributions, etc).

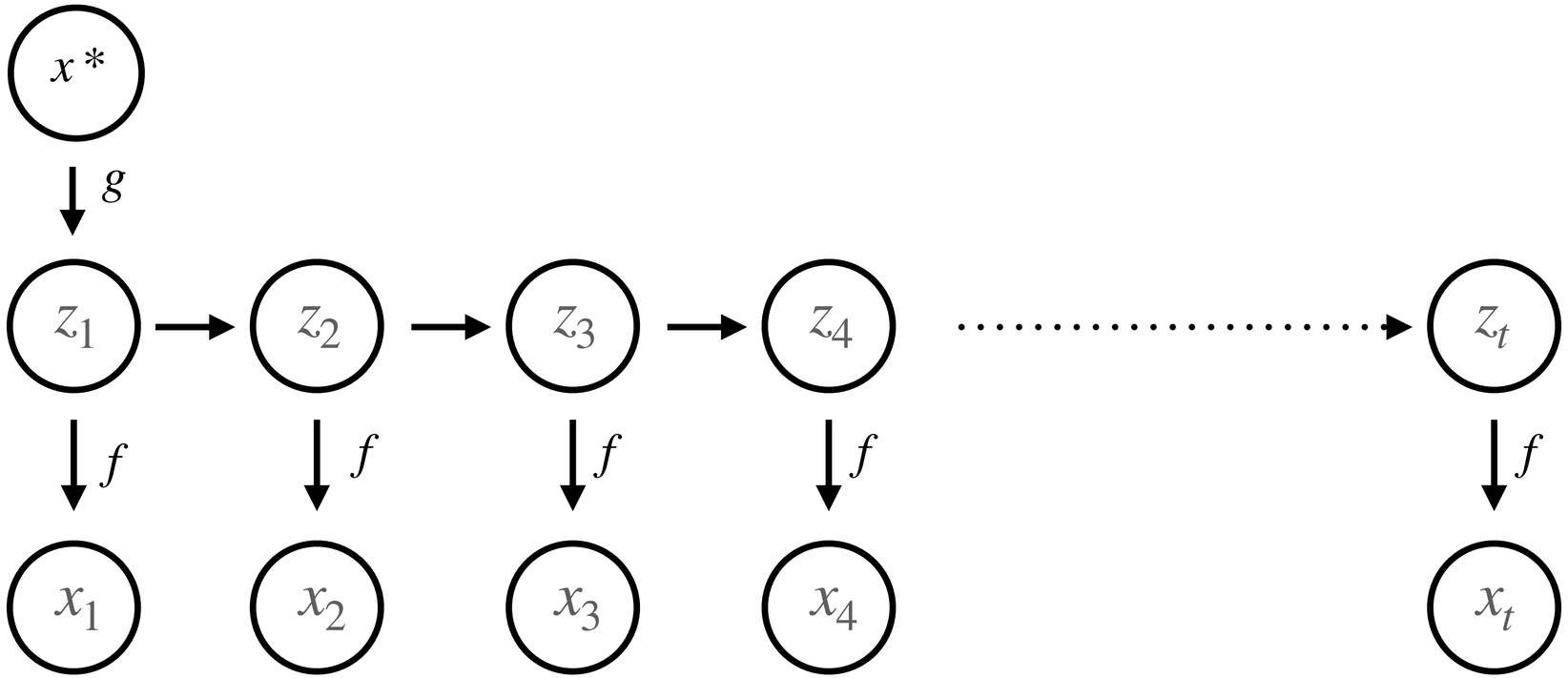
Sequential Variational Autoencoder (sVAE)

- VAE where the latents have a recurrent structure (e.g. are an RNN)



Sequential Variational Autoencoder (sVAE)

- VAE where the latents have a recurrent structure (e.g. are an RNN)



- LFADS (JC paper) is based on an SVAE model

Many more inference methods!

- Whole class of sampling (Monte Carlo) methods
- Markov Chain Monte Carlo (MCMC)
- Particle filtering
- Many more!

Dimensionality in Linear and Nonlinear Models

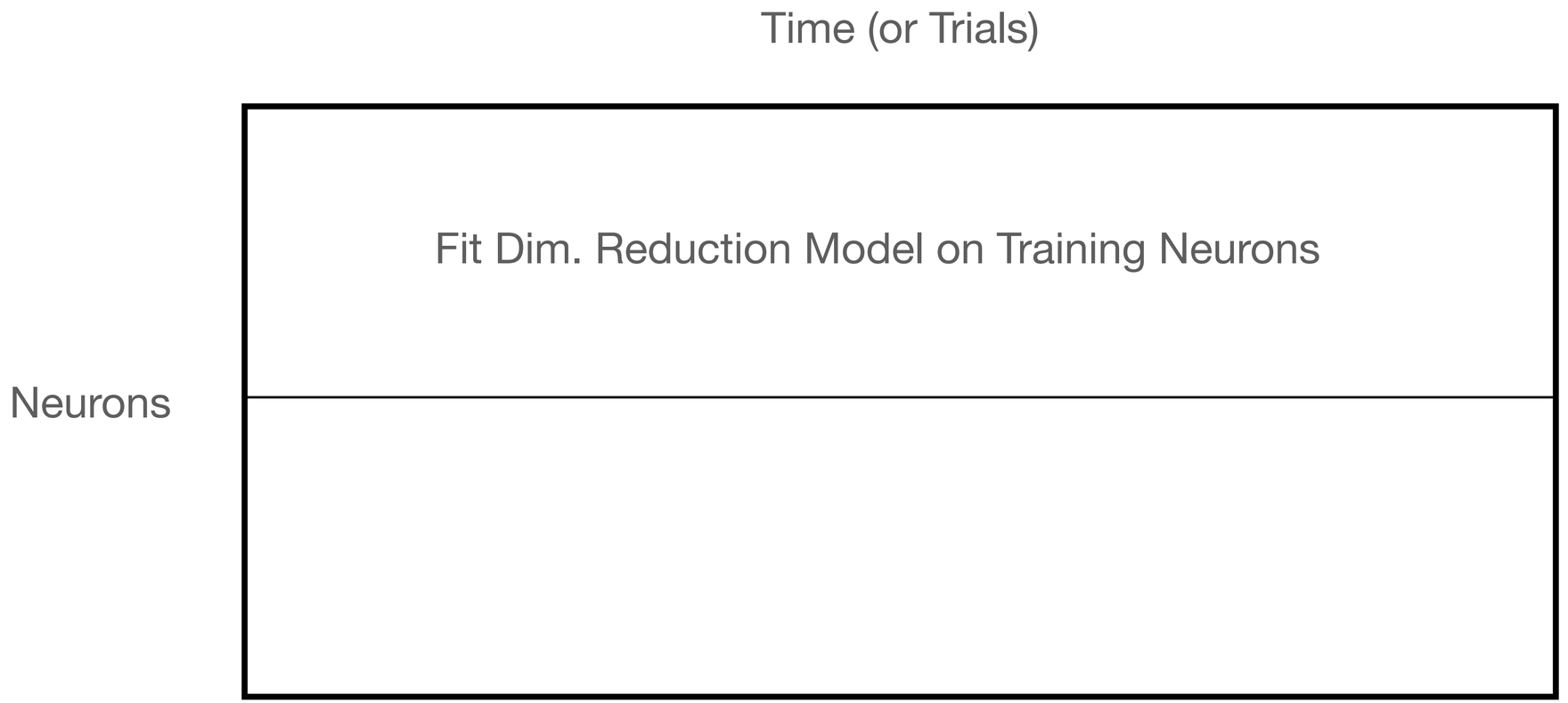
- *See HW*

Bi-cross-validation with Latent Variable Models + Dim. Reduction

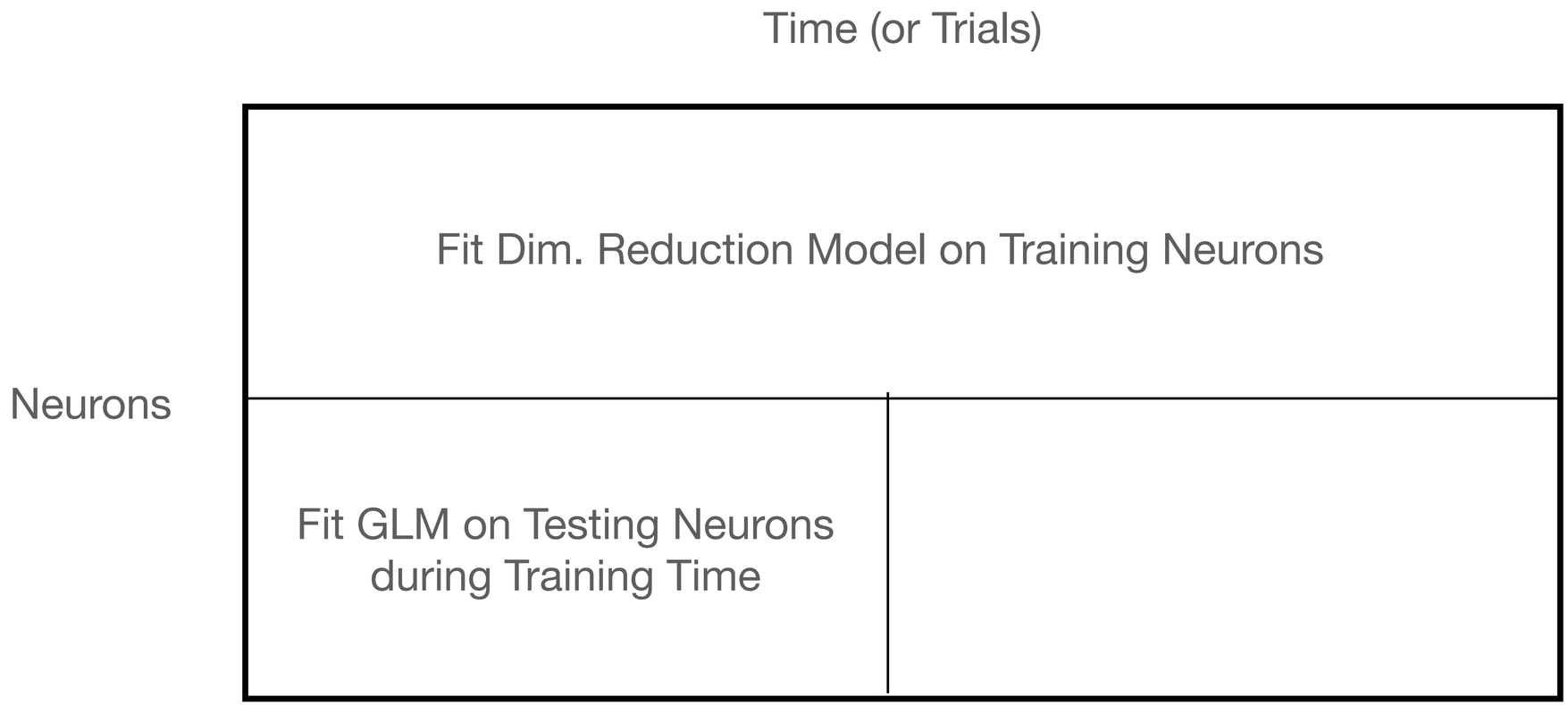
Time (or Trials)

Neurons

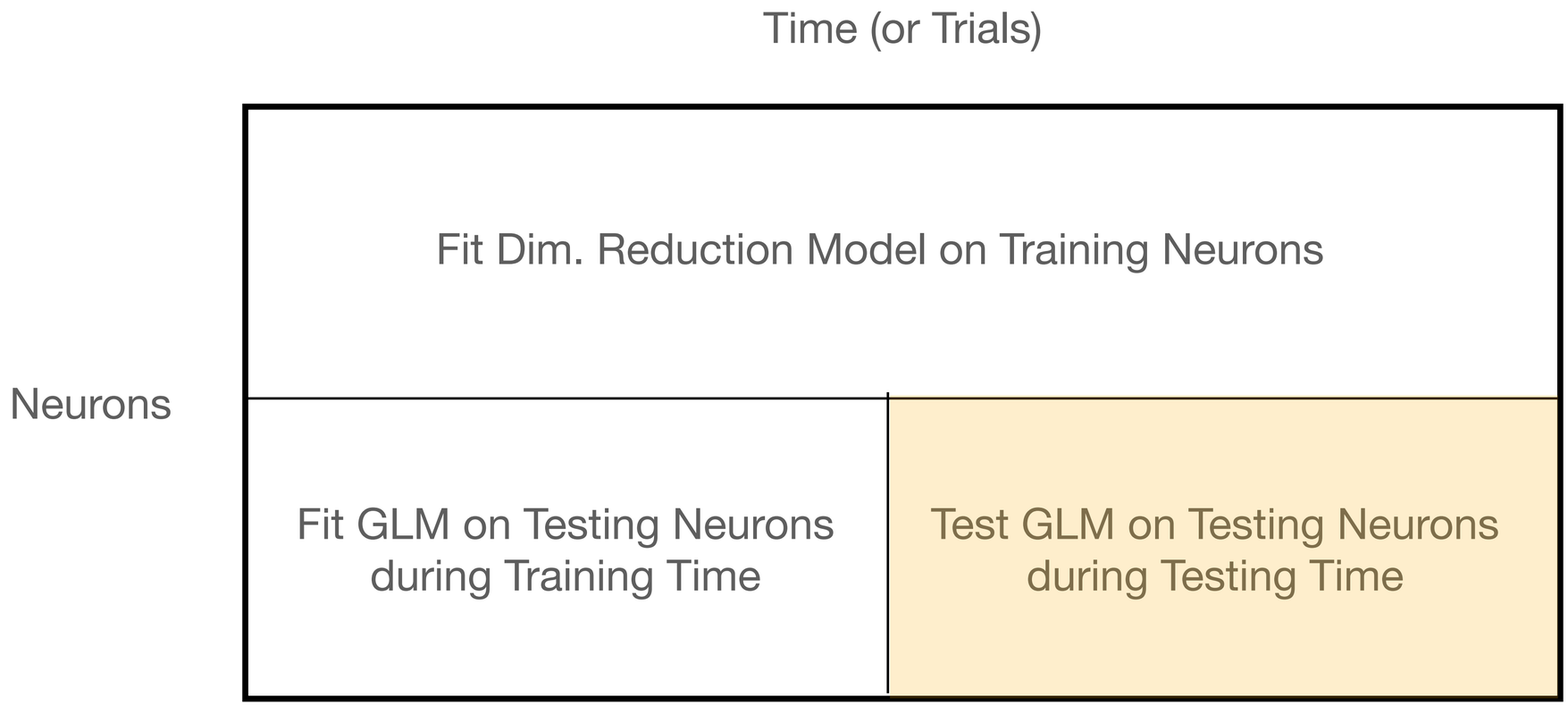
Bi-cross-validation with Latent Variable Models + Dim. Reduction



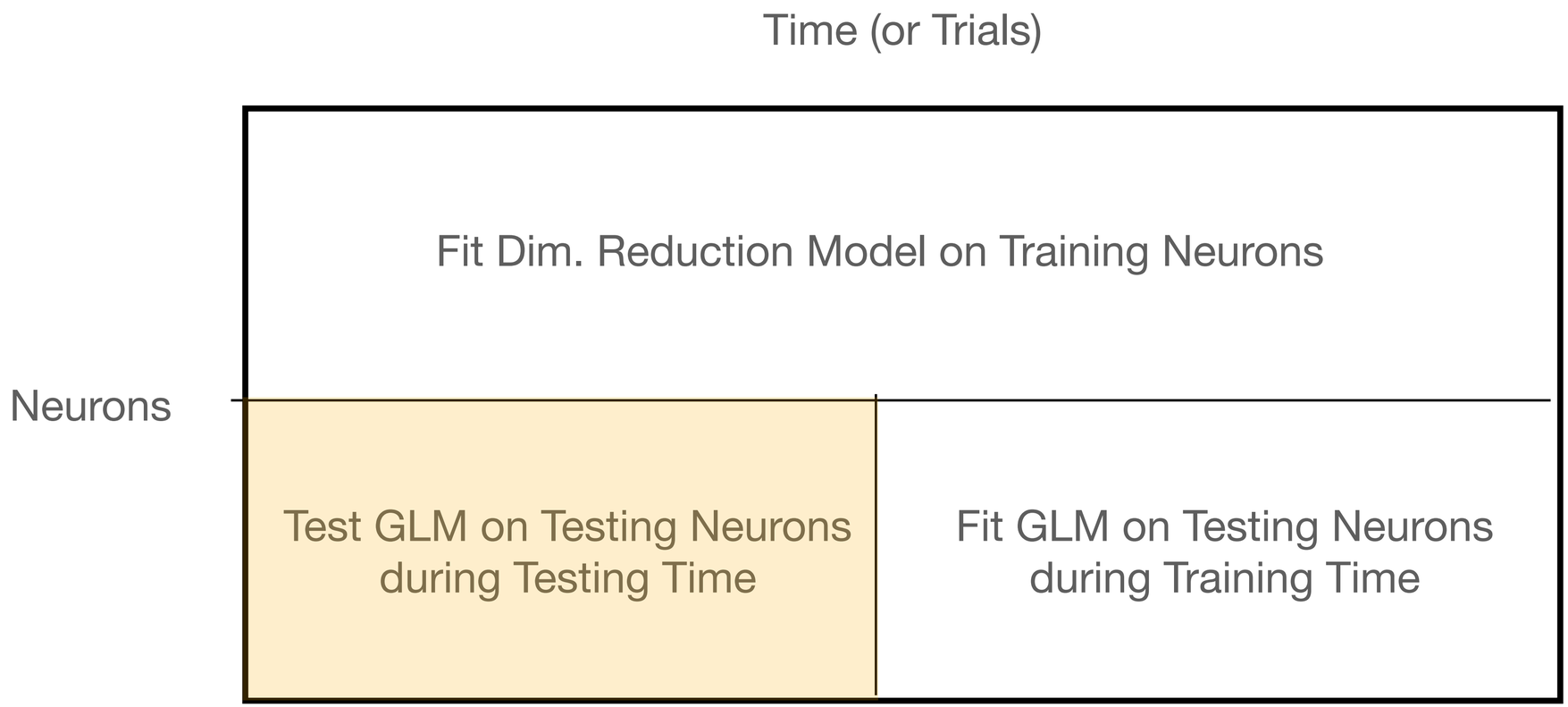
Bi-cross-validation with Latent Variable Models + Dim. Reduction



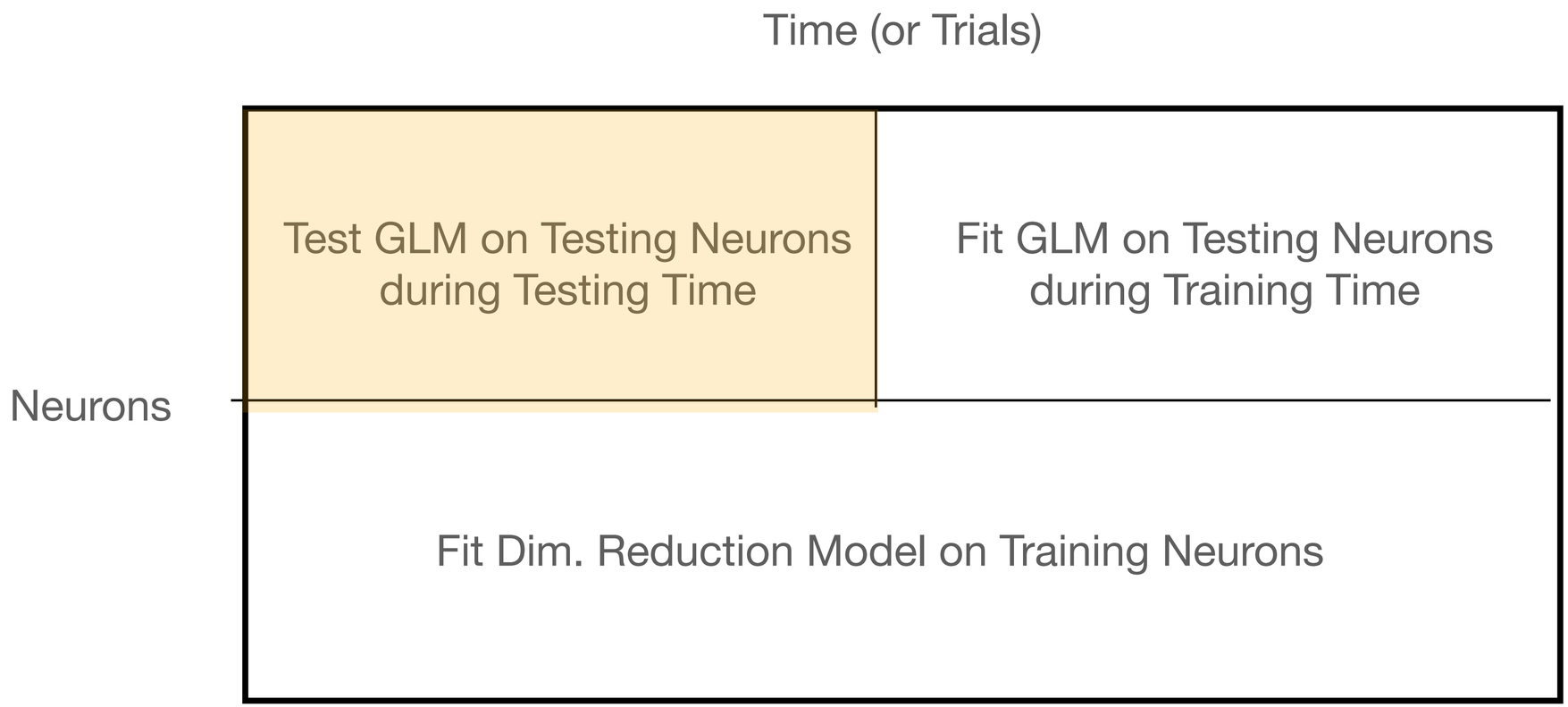
Bi-cross-validation with Latent Variable Models + Dim. Reduction



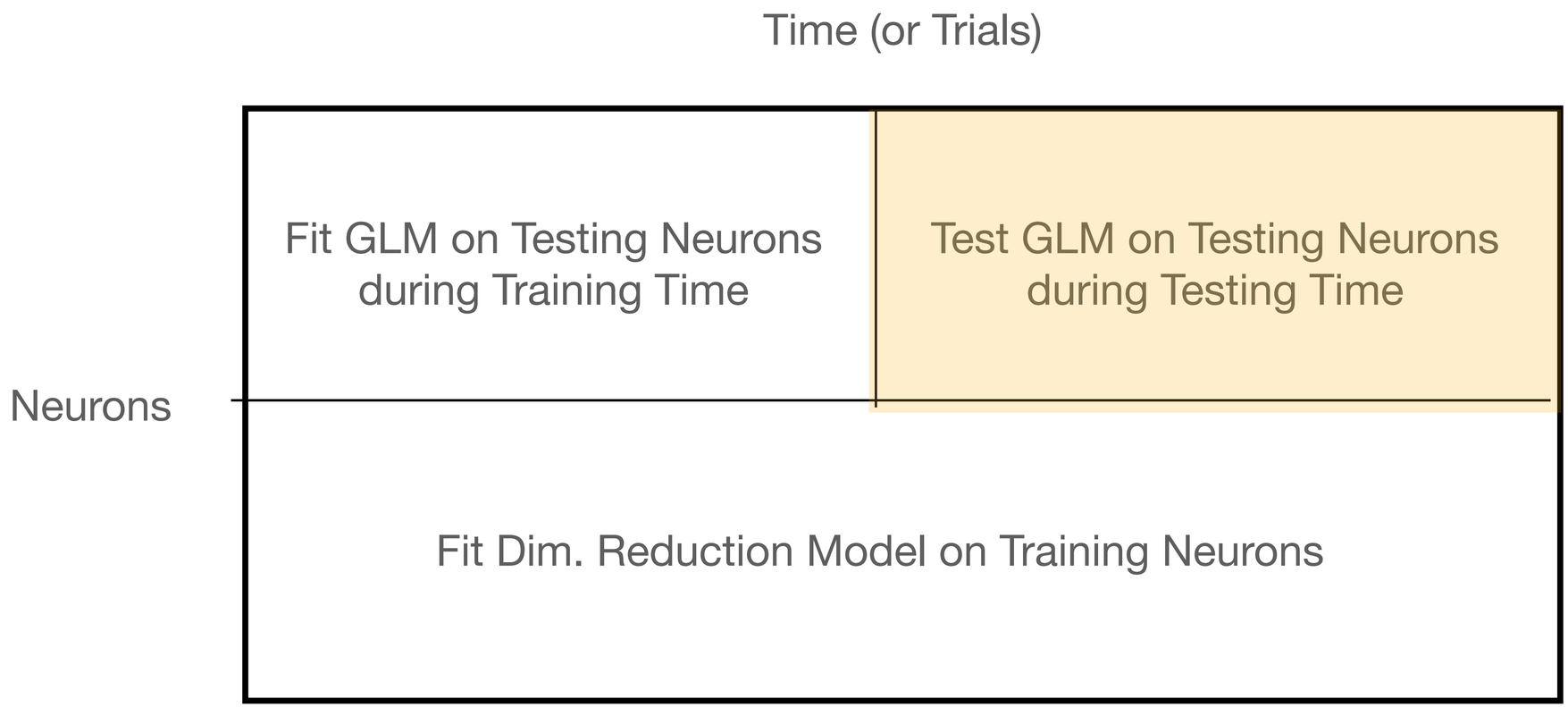
Bi-cross-validation with Latent Variable Models + Dim. Reduction



Bi-cross-validation with Latent Variable Models + Dim. Reduction



Bi-cross-validation with Latent Variable Models + Dim. Reduction



Resources

- Probabilistic Machine Learning Book 2, Chaps. 7,10, 29
  - https://probml.github.io/pml-book/book2.html
- Good blog post on cross-validation:
  - https://alexhwilliams.info/itsneuronalblog/2018/02/26/crossval/