4 算例 A：最简单的例子《多隐变量高斯模型》

为了理解坐标上升法（CAVI）中“期望背景”的含义，我们构造如下概率模型。

4.1 模型设定

假设观测变量 X_1, X_2 的生成过程受隐变量 Z_1, Z_2, Z_3 共同控制，具体逻辑如下：

- **先验分布**：假设隐变量相互独立，且服从标准高斯分布：

$$P(Z_j) = \mathcal{N}(Z_j|0,1), \quad j = 1,2,3$$

- **观测似然**：假设观测值是隐变量的线性组合加上高斯噪声：

$$\begin{aligned} P(X_1, X_2|Z) &= \textstyle\prod_{k=1}^{2} \mathcal{N}(X_k|Z_1+Z_2+Z_3, \sigma^2) \\ &= \prod_{k=1}^{2} \left[\tfrac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\tfrac{(X_k-(Z_1+Z_2+Z_3))^2}{2\sigma^2} \right) \right] \\ &= \left(\tfrac{1}{2\pi\sigma^2} \right) \exp\left(-\sum_{k=1}^{2} \tfrac{(X_k-(Z_1+Z_2+Z_3))^2}{2\sigma^2} \right) \end{aligned}$$

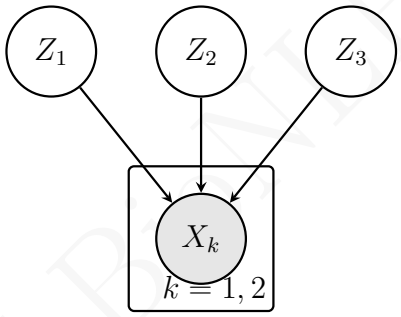

图 1: 多隐变量模型的板式记号 (Plate Notation)

4.2 联合对数概率 $\log P(X, Z)$ 的展开

忽略常数项，联合对数概率 $\log P(X, Z)$ 可以写为：

$$\begin{aligned} \log P(X, Z) &= \log P(X|Z) + \sum_{j=1}^{3} \log P(Z_j) \\ &= -\tfrac{1}{2\sigma^2} \sum_{k=1}^{2} (X_k - (Z_1+Z_2+Z_3))^2 - \tfrac{1}{2} \sum_{j=1}^{3} Z_j^2 + \text{const} \end{aligned}$$

4.3 变分后验 $q_1^*(Z_1)$ 的具体计算（配方法）

根据 CAVI 公式，我们计算 Z_1 的最优分布：

$$\log q_1^*(Z_1) = E_{q_2,q_3}[\log P(X, Z)] + \text{const}$$

我们将 $\log P(X, Z)$ 中关于 Z_1 的二项式展开，并取期望：

$$\begin{aligned} \log q_1^*(Z_1) &= E_{q_2,q_3} \left[-\tfrac{1}{\sigma^2} \sum_k (-(Z_1+Z_2+Z_3)X_k + \tfrac{1}{2}(Z_1+Z_2+Z_3)^2) - \tfrac{1}{2} Z_1^2 \right] + \dots \\ &= -\tfrac{1}{2} \left(\tfrac{2}{\sigma^2} + 1 \right) Z_1^2 + \left(\tfrac{\sum X_k - 2(E[Z_2]+E[Z_3])}{\sigma^2} \right) Z_1 + \text{const} \end{aligned}$$

由于 $\log q_1^*$ 是关于 Z_1 的二次函数，因此 $q_1^*(Z_1)$ 依然是一个高斯分布。利用配方法 (详情见附录章节A.1)，我们可以导出 $q_1^*(Z_1)$ 为高斯分布 $\mathcal{N}(\mu_1^*, (\sigma_1^2)^*)$：

$$\begin{cases} (\sigma_1^2)^* &= \frac{\sigma^2}{2+\sigma^2} \\ \mu_1^* &= \frac{\sum_{k=1}^{2} X_k - 2(\mu_2+\mu_3)}{2+\sigma^2} \end{cases}$$

Z_1 的新均值取决于 $(X_1 + X_2)$ 与其他隐变量期望值 $E[Z_2], E[Z_3]$ 的差。

逻辑注记：变分参数何时登场?

在我们的推导中，变分参数 μ_j 和 σ_j^2 的出现经历了以下逻辑：

1. **诱导产生**：我们并未预设 $q(Z)$ 是高斯分布，但由于模型采用了高斯似然与高斯先验，数学推导的结果“诱导”出了二次项结构。
2. **参数定义**：直到我们看到二次项形式时，我们才正式定义 μ_1^* 和 $(\sigma_1^2)^*$ 作为该分布的特征统计量。
3. **迭代循环**：一旦定义完成，这些变分参数就成为了 CAVI 算法在下一轮迭代中需要的“环境背景”(即公式中的 μ_2, μ_3)。

4.4 CAVI 迭代流程

算法通过逐个更新均值 μ_j 实现：

1. **初始化**：设置 $\mu_1^{(0)}, \mu_2^{(0)}, \mu_3^{(0)}$。
2. **迭代** t：

$$\begin{aligned} \mu_1^{(t)} &\leftarrow \frac{(X_1+X_2) - 2(\mu_2^{(t-1)} + \mu_3^{(t-1)})}{2+\sigma^2} \\ \mu_2^{(t)} &\leftarrow \frac{(X_1+X_2) - 2(\mu_1^{(t)} + \mu_3^{(t-1)})}{2+\sigma^2} \\ \mu_3^{(t)} &\leftarrow \frac{(X_1+X_2) - 2(\mu_1^{(t)} + \mu_2^{(t)})}{2+\sigma^2} \end{aligned} \tag{4-1}$$

3. **收敛**：当 μ 变化量低于阈值或 ELBO 停止增长时停止。

直觉说明：这是一种“责任分担”机制。每个隐变量都在扣除其他变量的“期望贡献”后，解释剩余的观测残差。

4.5 变分后验 $q_1^*(Z_1)$ 的另一种算法 (Fixed-form VI)

如果我们不希望依赖复杂的配方法来诱导分布形状，可以采用“先预设、后优化”的策略。

4.5.1 预设变分族与参数化

我们直接定义变分分布属于高斯族，并在推导前显式地引入变分参数：

$$q_j(Z_j; \mu_j, \sigma_j^2) = \mathcal{N}(Z_j | \mu_j, \sigma_j^2), \quad j = 1, 2, 3$$

这里，μ_j 和 σ_j^2 即为待优化的变分参数。

$$q(Z) = \prod_{j=1}^{3} q_j(Z_j; \mu_j, \sigma_j^2) = \prod_{j=1}^{3} \mathcal{N}(Z_j|\mu_j, \sigma_j^2)$$

其中，待优化的变分参数集合为 $\nu = \{\mu_1, s_1^2, \mu_2, s_2^2, \mu_3, s_3^2\}$。

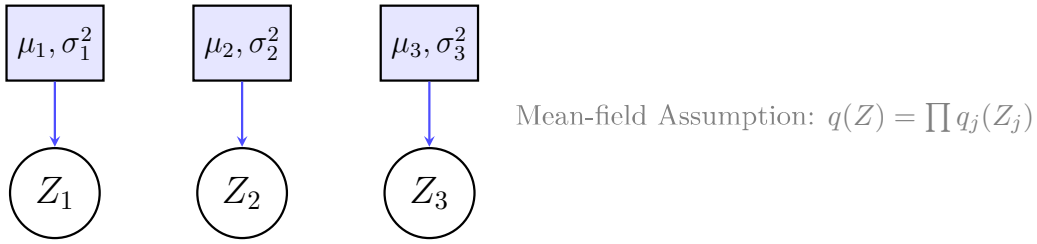

图 2: **Fixed-form VI 逻辑图**：方框代表待优化的确定性变分参数，圆圈代表由这些参数控制的高斯随机变量。注意，在变分模型中，不同 Z_j 之间没有边，体现了强行解耦的特性。

4.5.2 目标函数的参数化表达

此时，证据下界 ELBO 转化为关于向量 $\nu = [\mu_1, \sigma_1, \mu_2, \sigma_2, \mu_3, \sigma_3]^T$ 的标量函数：

$$\mathcal{L}(\nu) = E_q[\log P(X, Z)] + H(q)$$

其中 $H(q) = E_q[-\log q(Z)]$ 为变分分布的熵。

1. 期望能量项 (Expected Log-Likelihood) 利用高斯分布的二阶矩性质 $E_{q_\nu}[Z_j] = \mu_j$ 且 $E_q[Z_j^2] = \mu_j^2 + \sigma_j^2$，我们可以直接展开期望项：

$$\begin{aligned} E_q[\log P(X, Z)] &= E_q\left[-\tfrac{1}{2\sigma^2}\sum_{k=1}^{2}(X_k - \sum_{j=1}^{3} Z_j)^2 - \tfrac{1}{2}\sum_{j=1}^{3} Z_j^2\right] + \text{const} \\ &= -\tfrac{1}{2\sigma^2}\sum_{k=1}^{2}\left((X_k - \sum_{j=1}^{3}\mu_j)^2 + \sum_{j=1}^{3}\sigma_j^2\right) - \tfrac{1}{2}\sum_{j=1}^{3}(\mu_j^2 + \sigma_j^2) + \text{const} \end{aligned}$$

注意：此处利用了平均场假设下隐变量间的独立性，使得 $E[Z_i Z_j] = \mu_i \mu_j$ $(i \neq j)$。

2. 变分熵项 (Variational Entropy) 对于高斯变分分布，其熵项具有闭式解：

$$H(q) = E_q[-\log q(Z)] = \sum_{j=1}^{3} \frac{1}{2}\log(2\pi e \sigma_j^2)$$

经过整理，完整的证据下界 ELBO 可表示为：

$$\mathcal{L}(\nu) = \underbrace{-\frac{1}{2\sigma^2}\sum_{k=1}^{2}\left(X_k - \sum_{j=1}^{3}\mu_j\right)^2}_{\text{数据拟合项 (Data Fit)}} - \underbrace{\frac{1}{2}\sum_{j=1}^{3}\mu_j^2}_{\text{均值正则项 (Prior)}} - \underbrace{\left(\frac{1}{\sigma^2} + \frac{1}{2}\right)\sum_{j=1}^{3}\sigma_j^2}_{\text{方差惩罚项 (Uncertainty Cost)}} + \underbrace{\frac{1}{2}\sum_{j=1}^{3}\ln\sigma_j^2}_{\text{熵增项 (Spread)}} + \text{const} \quad (4\text{-}2)$$

4.5.3 优化算法的一般性策略

在得到目标函数的参数化表达后，我们不再需要配方法获得变分的参数迭代，也不需要使用坐标上升的解析更新公式（eq.3-2），我们转而通过梯度为零来直接寻找 ν^*，或者梯度下降 $\nu^{(t+1)} = \nu^{(t)} + \eta \nabla_\nu \mathcal{L}(\nu)$ 来进行数值优化。

4.5.4 梯度计算与结果一致性验证

通过对目标函数 $\mathcal{L}(\nu)$ 直接求偏导，我们可以验证 Fixed-form VI 与 CAVI 的等价性。以 μ_1 为例：

$$\nabla_{\mu_1} \mathcal{L} = \frac{1}{\sigma^2} \sum_{k=1}^{2} (X_k - \mu_1 - \mu_2 - \mu_3) - \mu_1$$

令梯度为零，解得如下结果，与 eq.(4-1) 结果一致。

$$\mu_1^* = \frac{\sum X_k - 2(\mu_2 + \mu_3)}{2 + \sigma^2} \tag{4-3}$$

类似的，从 $\mathcal{L}(\nu)$ 提取包含 σ_1^2 的项，只保留与 σ_1^2 有关的部分：

$$\mathcal{L}_{(\sigma_1^2)} = -\left(\frac{1}{\sigma^2} + \frac{1}{2}\right) \sigma_1^2 + \frac{1}{2} \ln \sigma_1^2 + \text{const}$$

对 σ_1^2 求偏导

$$\nabla_{\sigma_1^2} \mathcal{L} = -\left(\frac{1}{\sigma^2} + \frac{1}{2}\right) + \frac{1}{2\sigma_1^2}$$

令梯度为零，设 $\nabla_{\sigma_1^2} \mathcal{L} = 0$，我们有 $\frac{1}{2\sigma_1^2} = \frac{2+\sigma^2}{2\sigma^2}$。最终解得：

$$(\sigma_1^2)^* = \frac{\sigma^2}{2 + \sigma^2} \tag{4-4}$$

逻辑闭环：

- **在解析路径中**，这种形式是通过配方“凑”出来的。
- **在数值路径中**，这种形式是通过寻找 Loss 函数的极值点“走”出来的。

这证明了对于高斯模型，**预设高斯分布族**这一行为并没有损失精确度，反而简化了推断的思考量。[4]

[4] 方法论对比：

- **配方法路径**：追求解析解，计算效率极高，但仅适用于共轭模型。
- **预设路径**：追求通用性，可处理非共轭模型（如 $P(X|Z)$ 是神经网络），是现代深度学习中 VAE 等模型的基础。

5 算例 B：«在算例 A 中引入深度学习模型 ϕ 后，多隐变量高斯模型的混合推断方案»

5.1 算例的定义，及训练策略的调整

5.1.1 模型设定的一个变化

此节中，我们基本考虑与算例 A 相同的设定，仅仅将隐变量集合 Z 中的一个变分因子 Z_1 进行调整，其变分后验分布不再由独立的 μ_1 控制，而是由一个深度学习模型 ϕ 输出，称其为摊销因子。在变分推断的语境下，“摊销”这一术语反映了从“局部参数优化”向“全局映射函数”的范式转移。[5] 这意味着 Z_1 的推断过程从“逐点参数优化”转向“全局函数映射”。

- **摊销因子** Z_1：$q_\phi(Z_1|X)$，其中 ϕ 是神经网络（编码器）的参数。它学习的是从观测 X 到隐空间 Z_1 的通用映射。
- **经典因子** Z_2, Z_3：保持 $q_2(Z_2)$ 和 $q_3(Z_3)$，由坐标参数 μ_2, μ_3 控制。

同时，对隐变量的变分的计算通过 Fixed form VI 进行，与章节4.5相同。此时，变分族定义为：$q(Z) = q_\phi(Z_1|X)q_2(Z_2)q_3(Z_3)$。这里有一个最主要的变化便是变分后验：

$$q_\phi(Z_1|X) = \mathcal{N}(\mu_\phi(X), \sigma^2_\phi(X)) \tag{5-1}$$

5.1.2 概率图模型 (PGM)

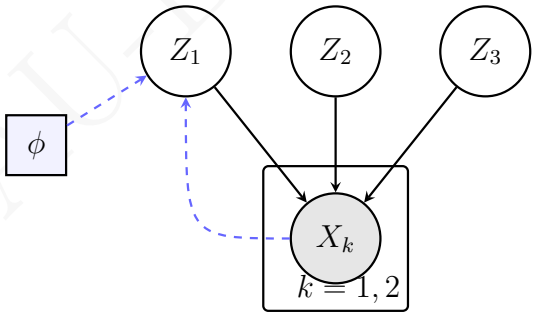

图 3: 摊销变分推断下的 PGM。实线代表生成模型 $P(X|Z)$，蓝色虚线代表由 ϕ 参数化的摊销推断路径 $q_\phi(Z_1|X)$。

此时的概率图模型反映了生成过程（实线）与推断路径（虚线）的结合。特别是 Z_1 的推断不再是孤立的，而是由 X 通过参数 ϕ 引导的。ϕ 是神经网络的权重，决定了从 X 到 Z_1 的映射逻辑。

[5] **计算成本的分摊 (Computational Cost)**。**传统 VI** (Z_2, Z_3)：对于每一个新的观测点 $X^{(i)}$，都必须重新运行迭代优化算法（如 CAVI）来寻找最优的变分参数 $\mu^{(i)}$。**摊销 VI** (Z_1)：虽然训练神经网络 ϕ 的初始成本很高，但一旦训练完成，对于任何新样本，只需一次前向计算即可瞬间获得后验分布。高昂的训练成本被分摊到了海量的预测任务中。

参数规模的压缩 (Parameter Scalability)。**非摊销模式**：变分参数的数量与样本量 N 成正比（即 $O(N)$），在大规模数据集下会遭遇“维度灾难”。**摊销模式**：参数量仅由神经网络的权重 ϕ 决定，与样本量 N 无关（即 $O(1)$）。这使得模型能够处理数以亿计的数据点，而不增加推断参数的负担。

概率图中的全局性 (Global Perspective)。在概率图模型（PGM）中，ϕ 位于 Plate（矩形框）之外。这在数学上准确地表达了 ϕ 是所有观测数据 $X_1, \ldots, X_N$ **共同分摊**并学习出来的全局逻辑，而非属于某个特定样本的局部变量。

5.1.3 混合迭代策略：解析与梯度的协同

由于隐变量 Z_1 嵌套在由参数 ϕ 定义的神经网络中，其后验分布无法获得解析解。我们采用一种“解析更新 + 梯度更新”的混合方案（Hybrid Optimization）：

1. **局部因子 Z_2, Z_3 (CAVI)**：保持解析更新。在计算期望背景时，直接使用神经网络当前输出的期望值 $E_{q_\phi}[Z_1] = \mu_\phi(X)$。
2. **全局参数 ϕ (SGD)**：利用**重参数化技巧 (Reparameterization Trick)** 将期望算子内部的梯度传回 ϕ，解决随机节点不可导的问题。

5.2 ELBO 的推导和计算

5.2.1 ELBO 定义

在混合推断框架下，全变分分布假设为 $q(Z) = q_\phi(Z_1|X)q(Z_2)q(Z_3)$。证据下界 (ELBO) 的定义为：

$$\text{ELBO} = E_{q_\phi q_2 q_3}\left[\log P(X, Z_1, Z_2, Z_3) - \log q_\phi(Z_1|X)q(Z_2)q(Z_3)\right]$$

为了便于推导迭代公式，我们将 ELBO 展开并按照变量的依赖关系进行分解：展开联合概率分布 $P(X, Z)$，包含观测模型的似然项和各隐变量的先验项：

$E_{q_\phi(Z_1|X)q_2q_3}[\log P(X|Z_1, Z_2, Z_3) + \log P(Z_1) + \log P(Z_2) + \log P(Z_3)]$.

将变分分布分解并代入后，ELBO 可以写为针对不同参数的贡献项之和：

$$\begin{aligned}\text{ELBO} = & \underbrace{E_{q_\phi}\left[E_{q_2q_3}[\log P(X|Z_1, Z_2, Z_3)] + \log P(Z_1) - \log q_\phi(Z_1|X)\right]}_{\text{与}\phi\text{ 相关的摊销项}} \\ & + \sum_{j=2}^{3} \underbrace{E_{q_j}[\log P(Z_j) - \log q(Z_j)]}_{\text{与}\mu_j\text{ 相关的坐标项}} + \text{const}\end{aligned}$$

针对 Z_1 的重参数化处理 为了使 ϕ 的梯度可计算，我们利用重参数化技巧 $Z_1 = \mu_\phi(X) + \sigma_\phi(X) \odot \epsilon$，将摊销项改写为关于噪声分布 $\epsilon \sim \mathcal{N}(0, I)$ 的期望形式。此时，全系统的目标函数 $\text{ELBO}(\phi, \mu_2, \mu_3)$ 为：[6]

$$\begin{aligned}\text{ELBO}(\phi, \mu_2, \mu_3) = & \underbrace{E_{\epsilon\sim\mathcal{N}(0,I)}\left[E_{q_2q_3}[\log P(X|Z_1(\epsilon, \phi), Z_2, Z_3)] + \log P(Z_1(\epsilon, \phi)) - \log q_\phi(Z_1(\epsilon, \phi)|X)\right]}_{\text{针对}\phi\text{ 的重参数化摊销项}} \\ & + \sum_{j=2}^{3} \underbrace{E_{q_j}[\log P(Z_j) - \log q(Z_j)]}_{\text{针对}\mu_j\text{ 的坐标解析项}} + \text{const}\end{aligned}$$

5.2.2 ELBO 中期望的计算，参数 ϕ, μ_2, μ_3 的交替优化

在得到重参数化的 ELBO 表达式后，推断的核心在于如何处理其中的期望算子 E。在混合推断框架下，算法采取了“解析路径”与“随机采样路径”并行的策略。[7]

在第 t 次迭代中，算法按如下步骤交替进行：

[6] $\text{ELBO}(\phi, \mu_2, \mu_3)$ 这个记号的选取同时提醒读者，后期优化的参数分别是 ϕ, μ_2, μ_3.

[7] Z_2, Z_3 负责捕捉数据中符合经典统计分布的部分，利用解析解确保在已知背景下的绝对最优。ϕ 驱动的 Z_1 负责学习数据中复杂的、非线性的隐特征，通过大数据的“摊销”效应分摊推断成本。

1. 解析更新坐标因子 $\mu_j(j=2,3)$ ：对于经典隐变量 Z_2, Z_3，其变分分布 q_j 具有简单的闭式解（参考 eq.4-1）。在更新 μ_j 时，摊销因子 Z_1 的期望直接由神经网络当前的输出均值 $E_{q_\phi}[Z_1] = \mu_\phi(X;\phi^{(t-1)})$ 替代（以 Z_2 为例）：

$$\mu_2^{(t)} \leftarrow \frac{(X_1+X_2) - 2\left(\mu_\phi(X;\phi^{(t-1)}) + \mu_3^{(t-1)}\right)}{2+\sigma^2} \tag{5-2}$$

2. 针对摊销因子 ϕ 的蒙特卡洛采样 (结合重参化) ：由于 ϕ 嵌套在随机采样层中，其期望无法解析。我们将 Z_1 表达为噪声变量 $\epsilon \sim \mathcal{N}(0,I)$，进行重参数，从噪声分布中抽取 L 个样本 $\{\epsilon^{(l)}\}_{l=1}^L$，构造近似梯度。

$$Z_1(\epsilon;\phi) = \mu_\phi(X) + \sigma_\phi(X) \odot \epsilon$$

此时，针对 ϕ 的优化目标梯度 $\nabla_\phi \text{ELBO}(\phi, \mu_2, \mu_3)$ 可通过蒙特卡洛采样近似估计（它代替了期望计算）：

$$\nabla_\phi \text{ELBO} \approx \frac{1}{L}\sum_{l=1}^{L} \nabla_\phi \Big[\log P(X, Z_1^{(l)}, \mu_2^{(t-1)}, \mu_3^{(t-1)}) - \log q_\phi(Z_1^{(l)}|X)\Big] \tag{5-3}$$
$$\text{where} \quad Z_1^{(l)} = \mu_\phi(X) + \sigma_\phi(X) \odot \epsilon^{(l)}, \quad \epsilon^{(l)} \sim \mathcal{N}(0,I)$$

下面我们对该公式进行计算。

联合分布项 $\log P(X, Z_1^{(l)}, \mu_2, \mu_3)$：这一项由“似然”和“先验”组成：

$$\log P(X, Z_1^{(l)}, \mu_2, \mu_3) = \underbrace{\log P(X|Z_1^{(l)}, \mu_2, \mu_3)}_{\text{似然项}} + \underbrace{\log P(Z_1^{(l)})}_{\text{先验项}} + \text{const}$$

其中似然项计算是均方误差（MSE）的负数：$-\frac{1}{2\sigma^2}\|X - (Z_1^{(l)} + \mu_2 + \mu_3)\|^2$. 先验项反映了 $Z_1^{(l)}$ 距离先验中心的距离：$-\frac{1}{2}\|Z_1^{(l)}\|^2$.

变分密度项 $\log q_\phi(Z_1^{(l)}|X)$：这是 $Z_1^{(l)}$ 在它自己所属分布下的对数概率密度：

$$\log q_\phi(Z_1^{(l)}|X) = -\frac{1}{2}\left[\log(2\pi\sigma_\phi^2) + \frac{(Z_1^{(l)} - \mu_\phi)^2}{\sigma_\phi^2}\right] = -\frac{1}{2}\left[\log(2\pi\sigma_\phi^2) + \frac{1}{2}(\epsilon^{(l)})^2\right]$$

在已知坐标背景 μ_2, μ_3 的情况下，针对神经网络参数 ϕ 的梯度估算式为：

$$\nabla_\phi \text{ELBO} \approx \frac{1}{L}\sum_{l=1}^{L} \nabla_\phi \left[\underbrace{-\frac{1}{2\sigma^2}\sum_{k=1}^{2}(X_k - Z_1^{(l)} - \mu_2 - \mu_3)^2}_{\text{A. 重构路径}} \underbrace{-\frac{1}{2}(Z_1^{(l)})^2}_{\text{B. 先验路径}} + \underbrace{\log \sigma_\phi(X)}_{\text{C. 熵路径}}\right] \tag{5-4}$$

$$\text{其中,} \begin{cases} Z_1^{(l)} = \mu_\phi(X) + \sigma_\phi(X)\cdot\epsilon^{(l)} \ \text{（重参数化采样节点）} \\ \epsilon^{(l)} \sim \mathcal{N}(0,1) \ \text{（独立标准高斯噪声样本）} \\ \mu_2, \mu_3 \ \text{（由上一轮 CAVI 更新得到的确定性常量）} \end{cases}$$

3. 参数步进 使用随机梯度上升 $\phi^{(t)} \leftarrow \phi^{(t-1)} + \eta \cdot \nabla_\phi \text{ELBO}$ 更新网络权重：

— 计算 ELBO 各项 —

```
# A. 重构路径 (Reconstruction Path)
recon_term = - (0.5 / (sigma_obs**2)) * torch.sum((X - (z1_reparam + mu2 + mu3))**2, dim=1)

# B. 先验路径 (Prior Path)
prior_term = - 0.5 * (z1_reparam**2)

# C. 熵路径 (Entropy Path)
entropy_term = 0.5 * log_var_phi

# 汇总 ELBO (取 Batch 的平均值)
elbo = torch.mean(recon_term + prior_term + entropy_term)

# 4. 反向传播与优化
loss = -elbo
loss.backward()
optimizer.step()
```

5.3 总结：混合推断

在这种设定下，模型展现出了极强的灵活性：Z_2, Z_3 负责捕捉数据中符合经典统计分布的部分（严谨、解析），而 ϕ 驱动的 Z_1 则负责学习数据中复杂的、非线性的隐特征（强大、灵活）。