

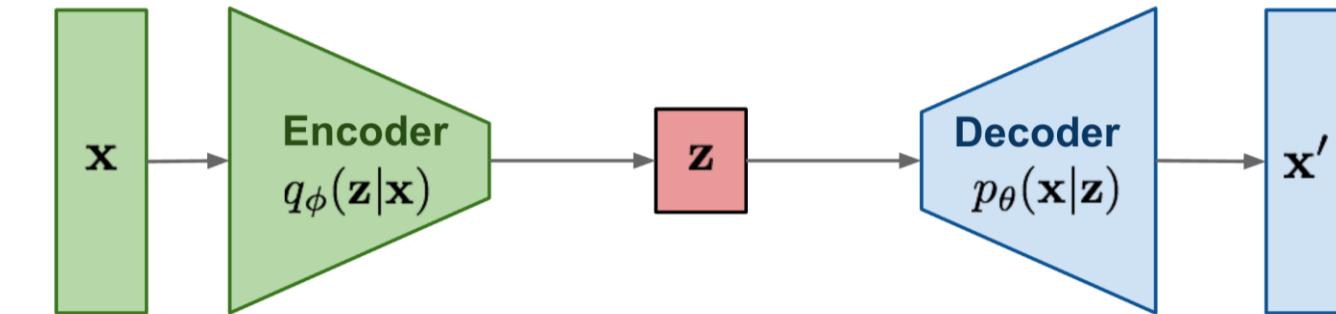
Sesión 4.2

Variational autoencoder

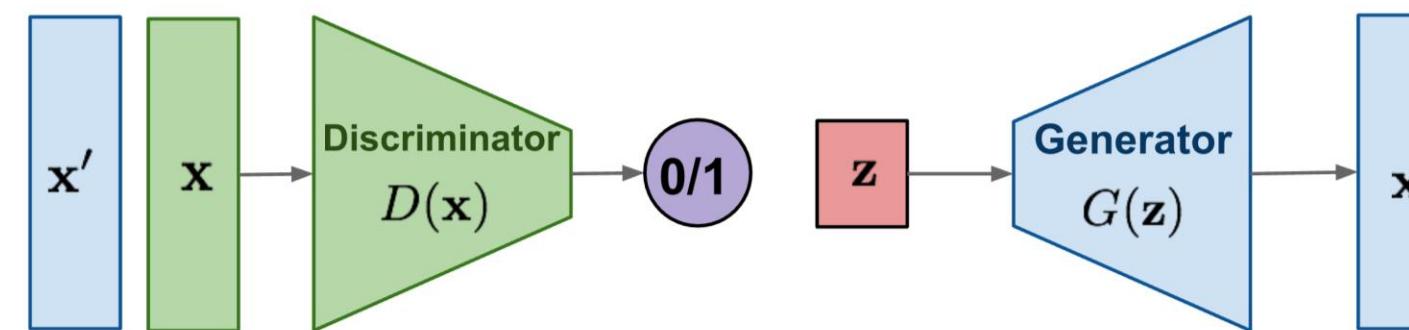
Variational inference. ELBO.

Generative model

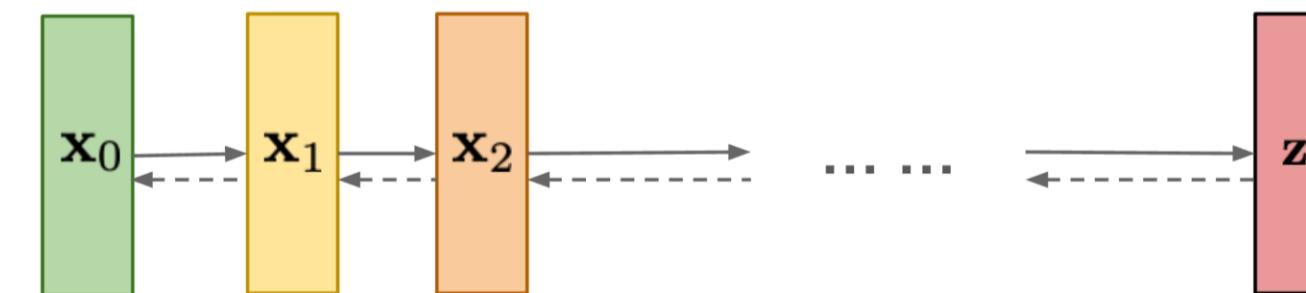
VAE: maximize variational lower bound



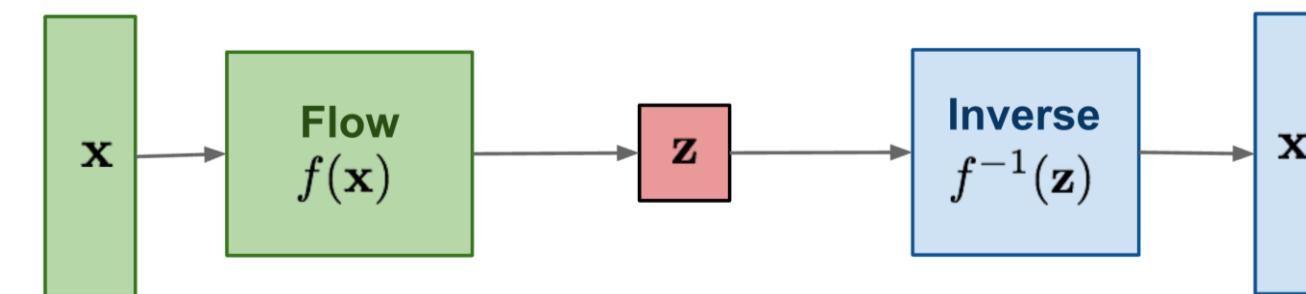
GAN: Adversarial training



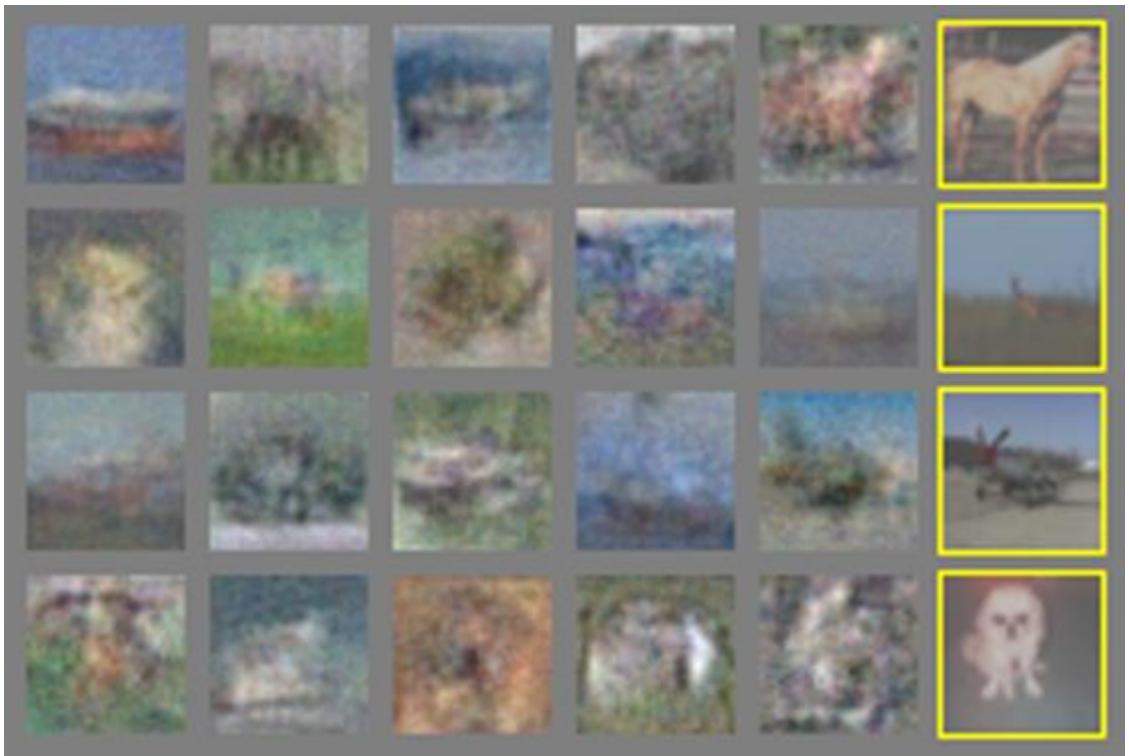
Diffusion models:
Gradually add Gaussian noise and then reverse



Flow-based models:
Invertible transform of distributions



Generative *model*



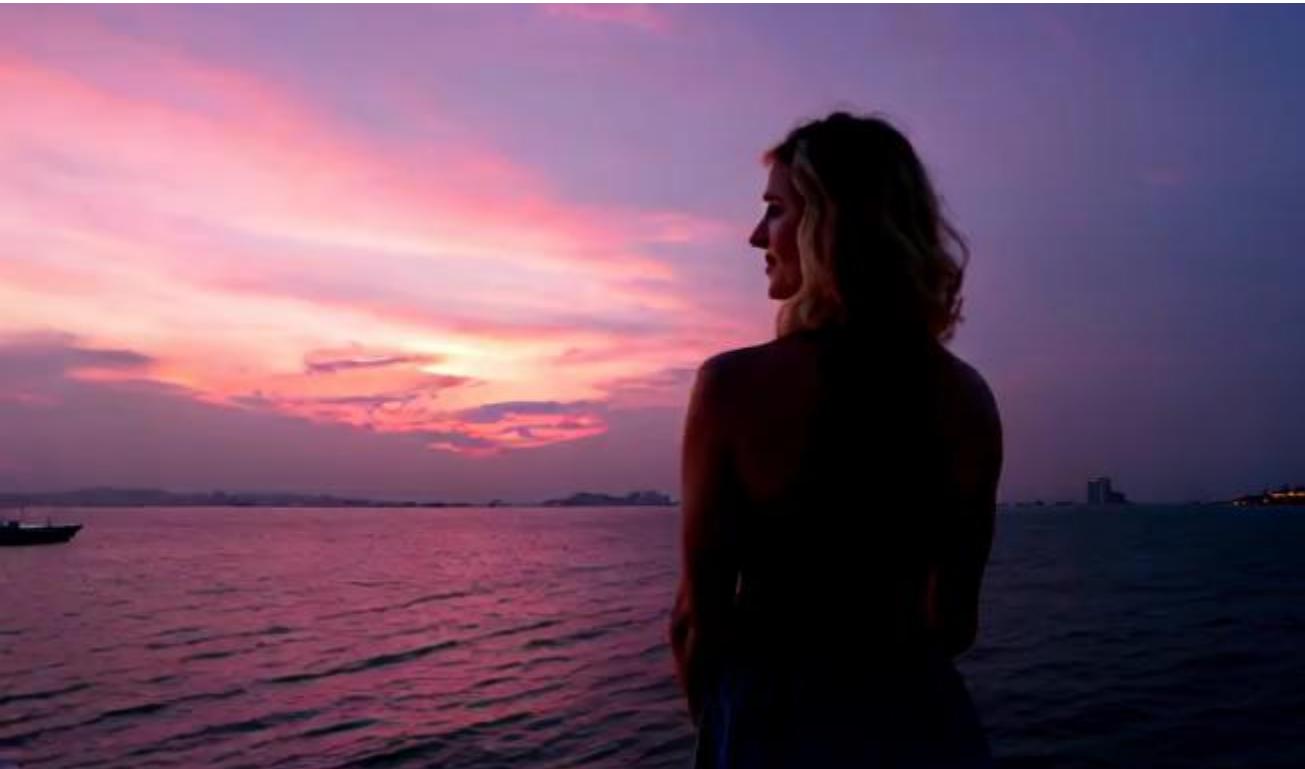
2014



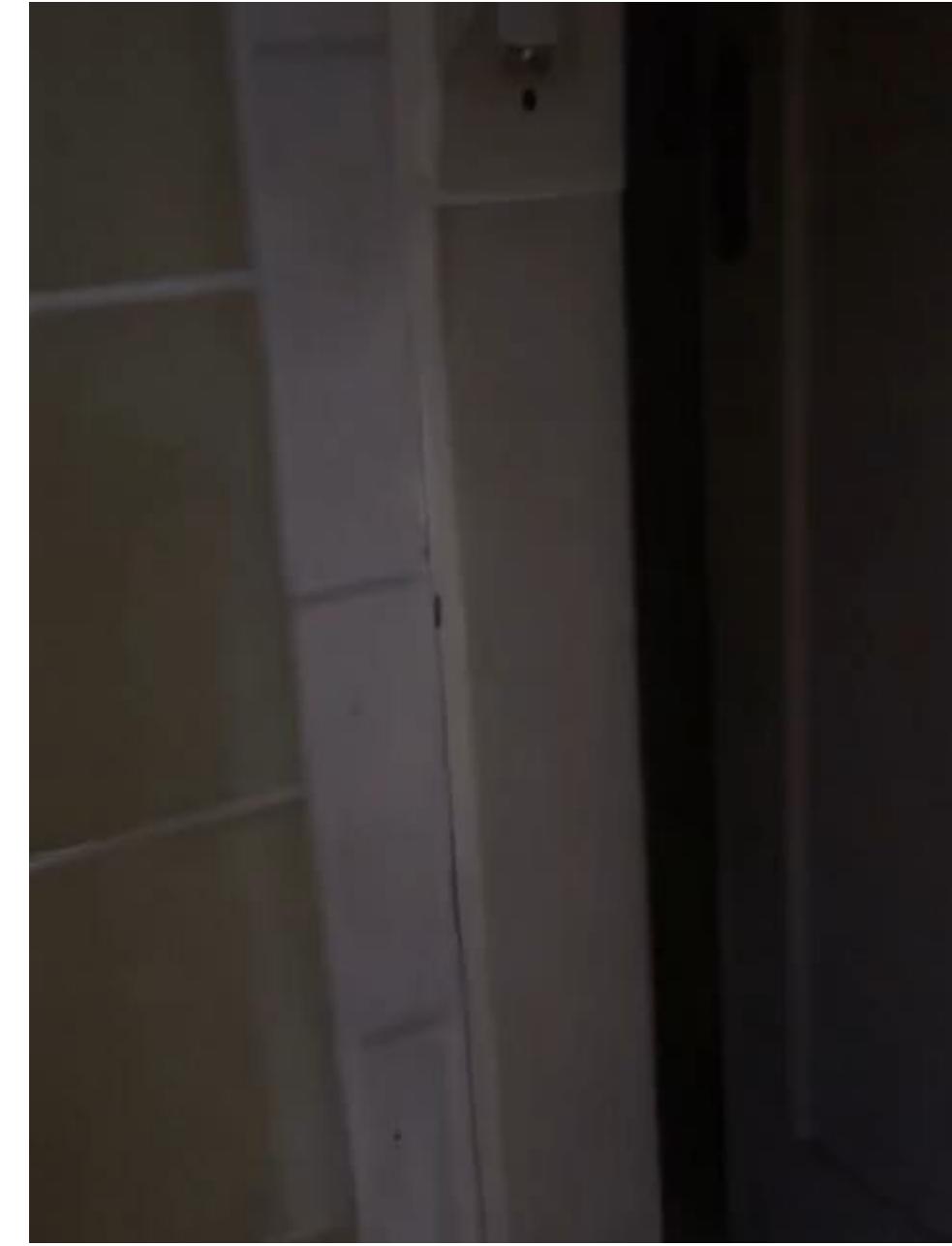
2024



Generative model



CineMaster (2025)

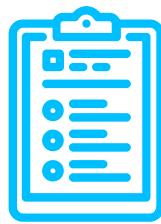


Pika (2025)



TRANSFORMATEC

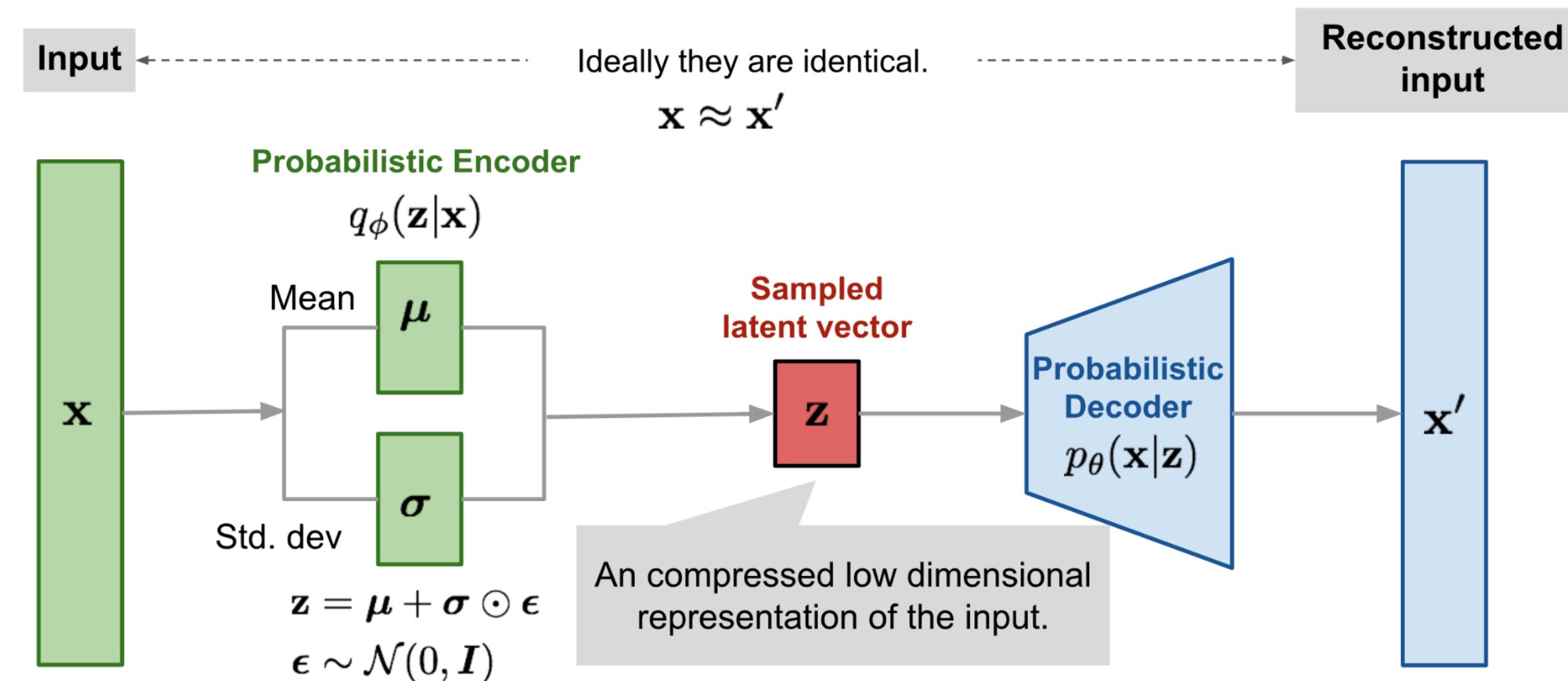
1.



Variational *Autoencoder*

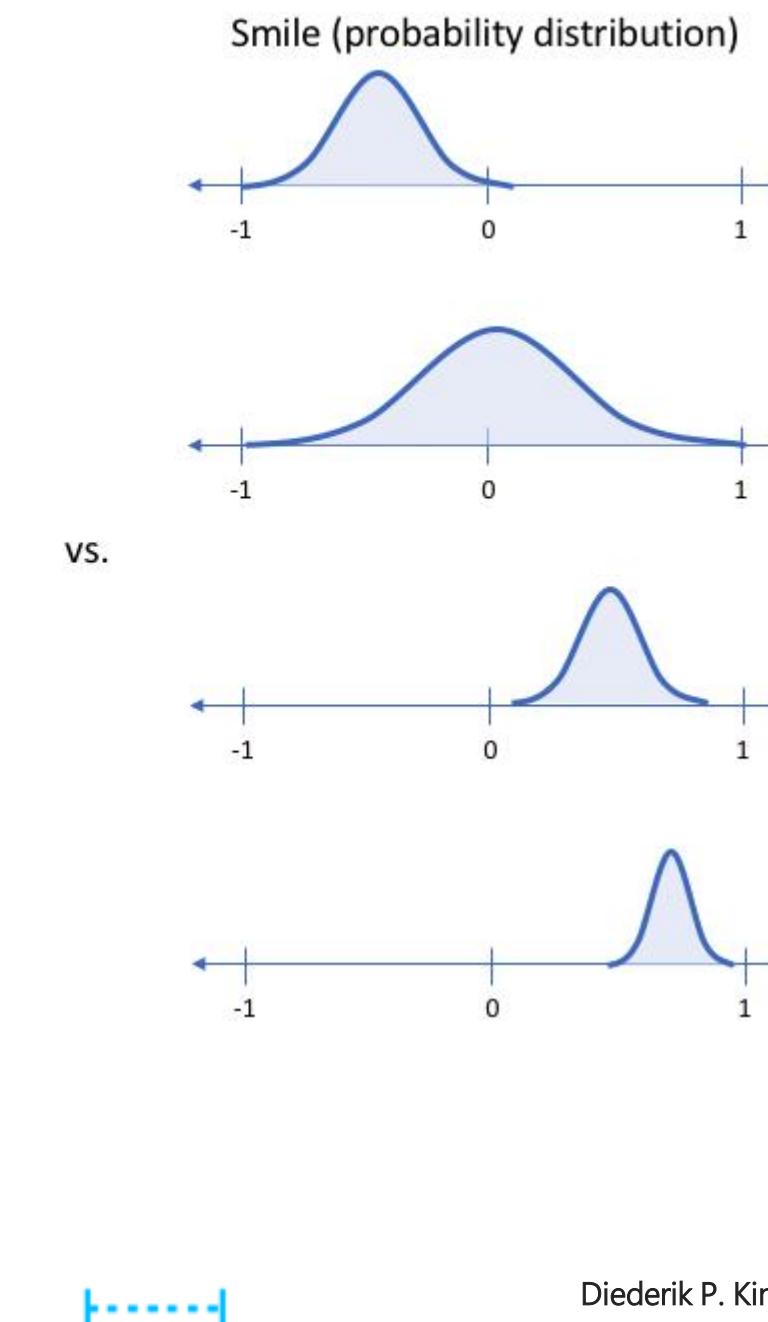
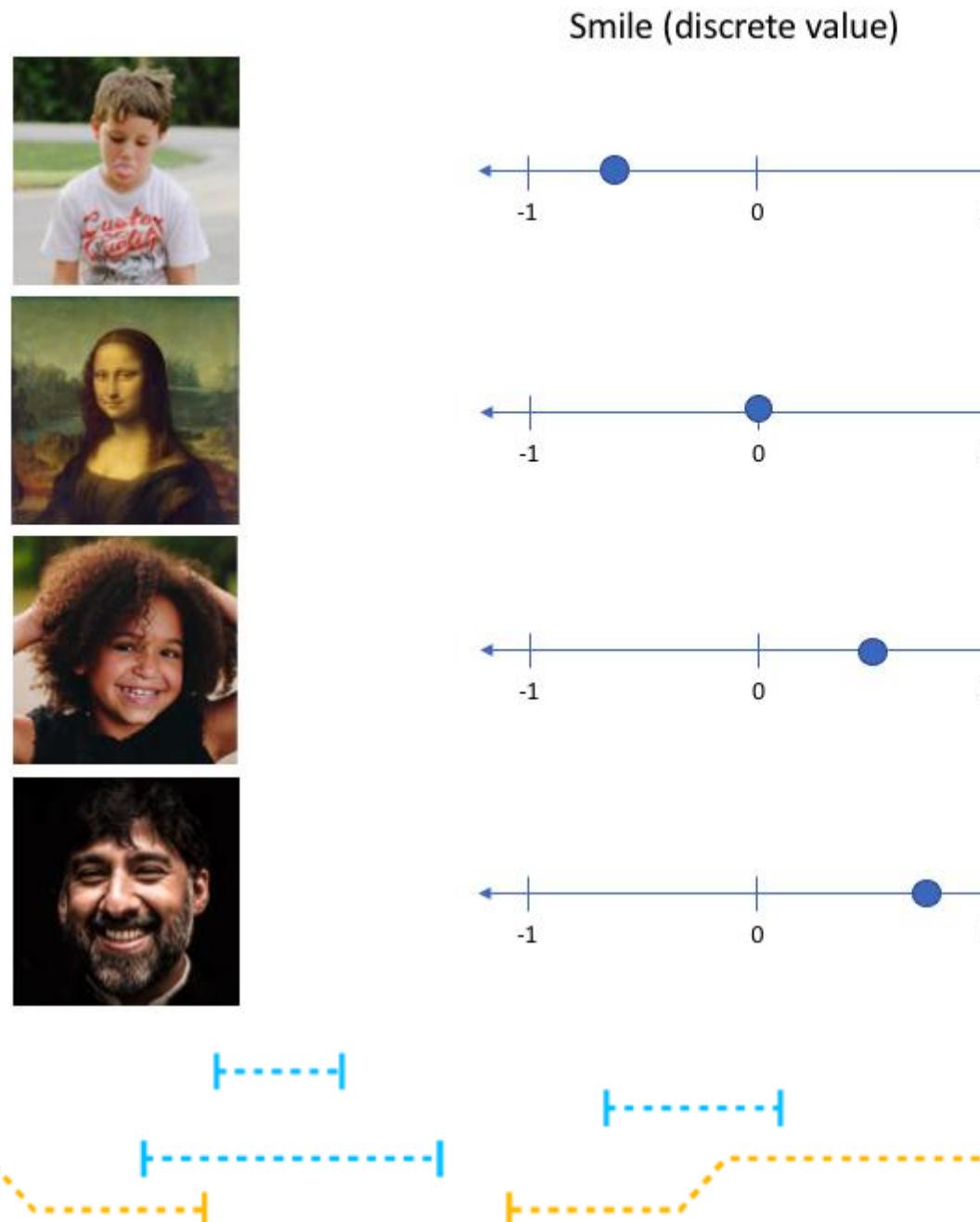


VAE



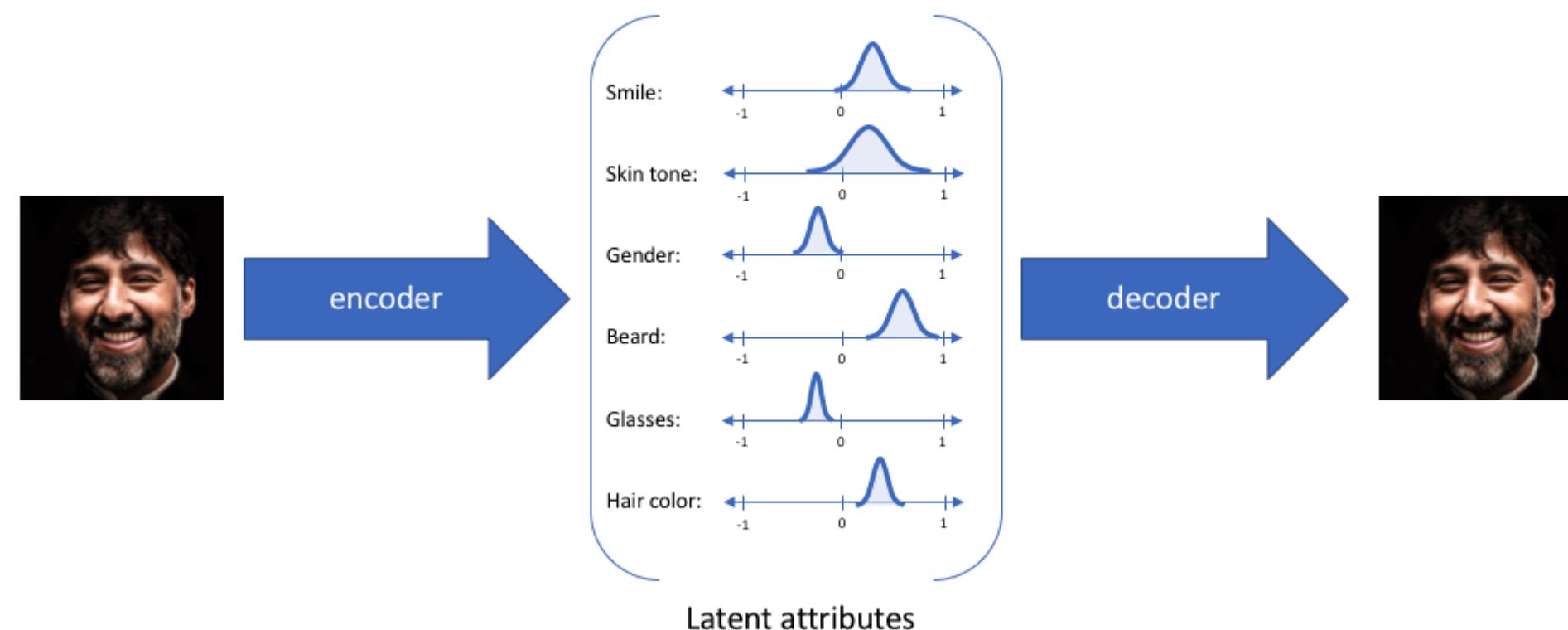
VAE

A diferencia de los autoencoders tradicionales que aprenden una función de codificación determinística, los VAEs introducen una distribución probabilística en el espacio latente



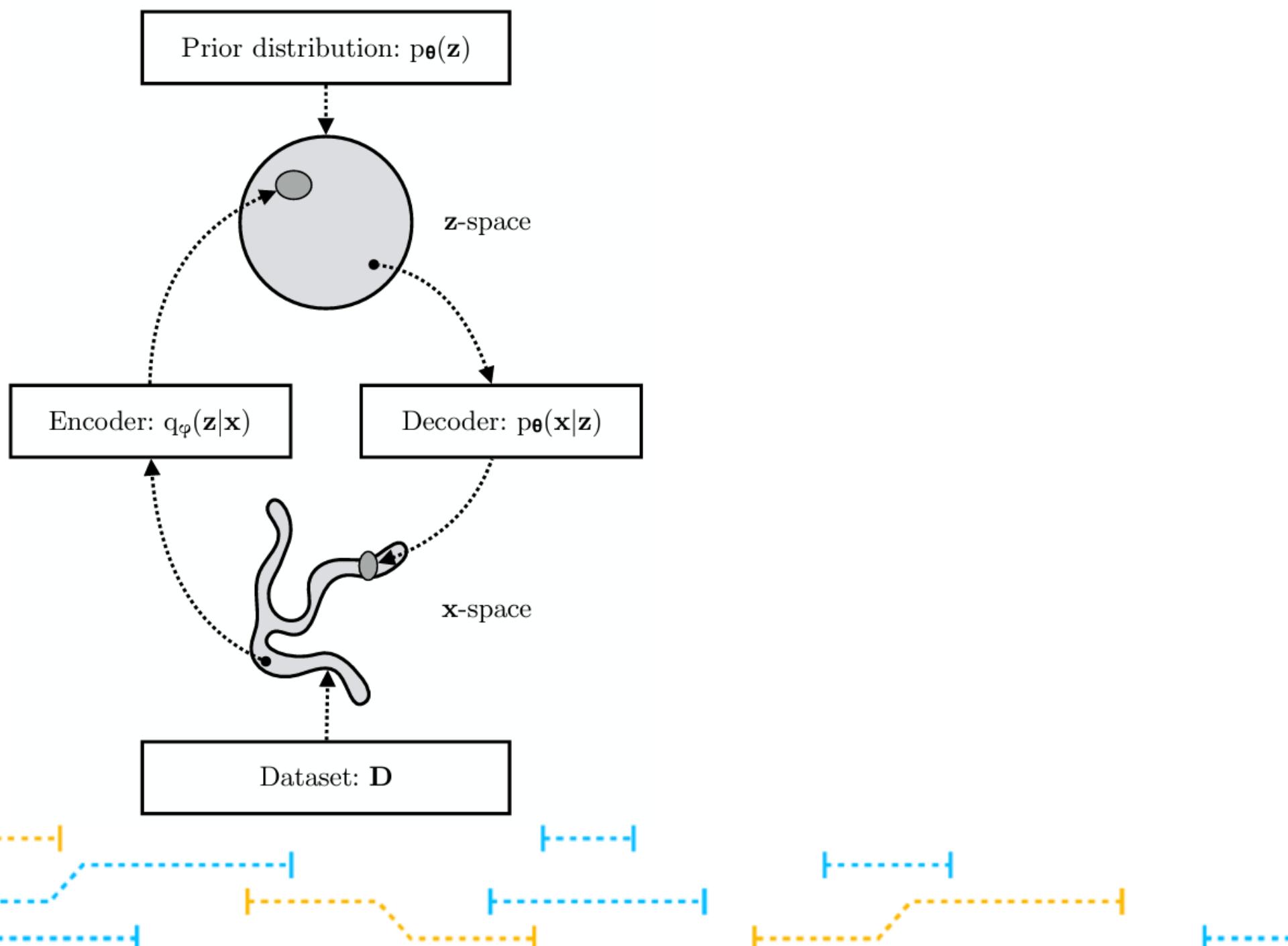
VAE

A diferencia de los autoencoders tradicionales que aprenden una función de codificación determinística, los VAEs introducen una distribución probabilística en el espacio latente



VAE

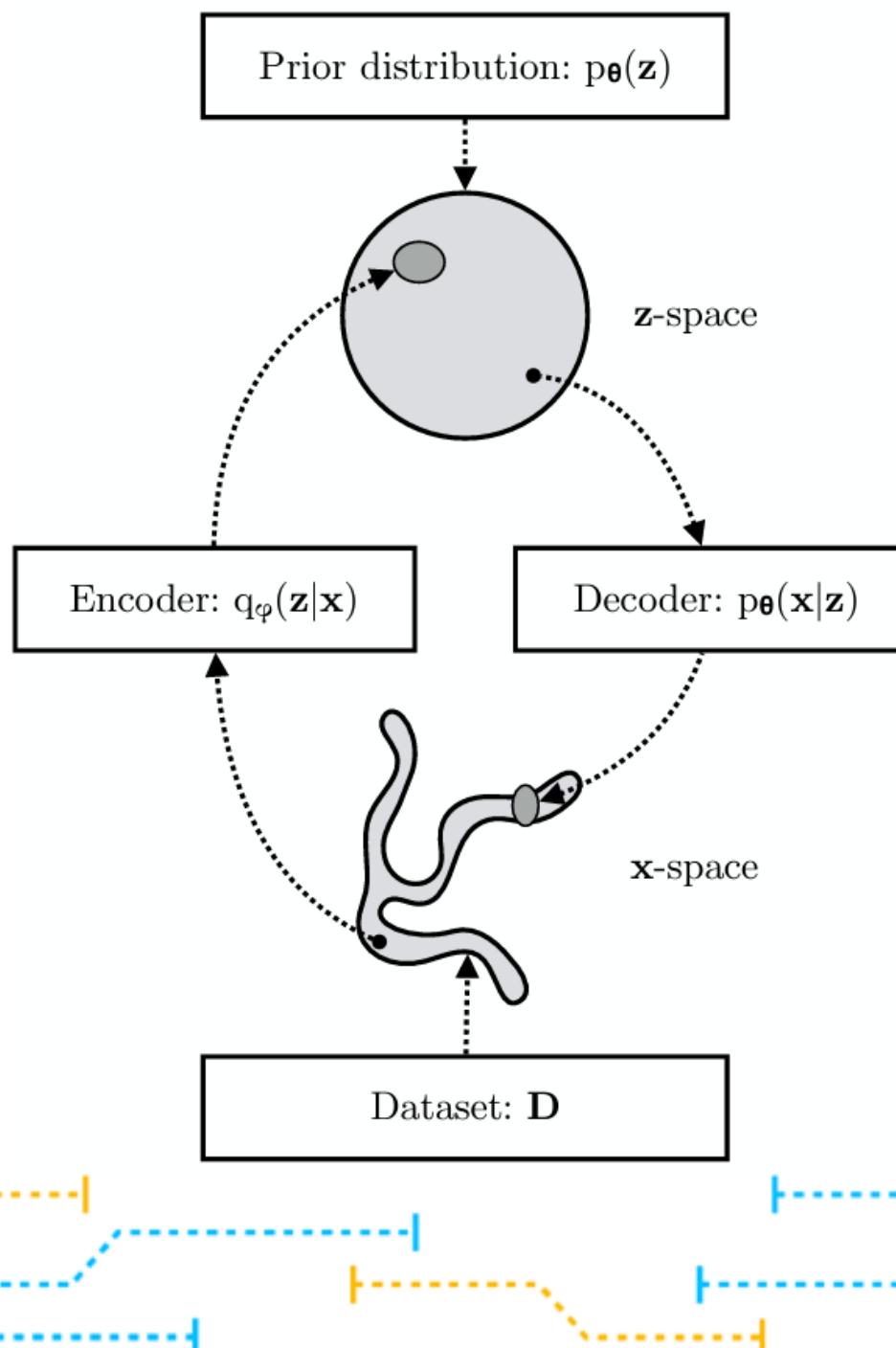
El **encoder**, parametrizado por ϕ , transforma los datos de entrada en una distribución probabilística sobre el espacio latente.
 El **decoder**, parametrizado por θ , utiliza estas representaciones latentes para reconstruir los datos de entrada.



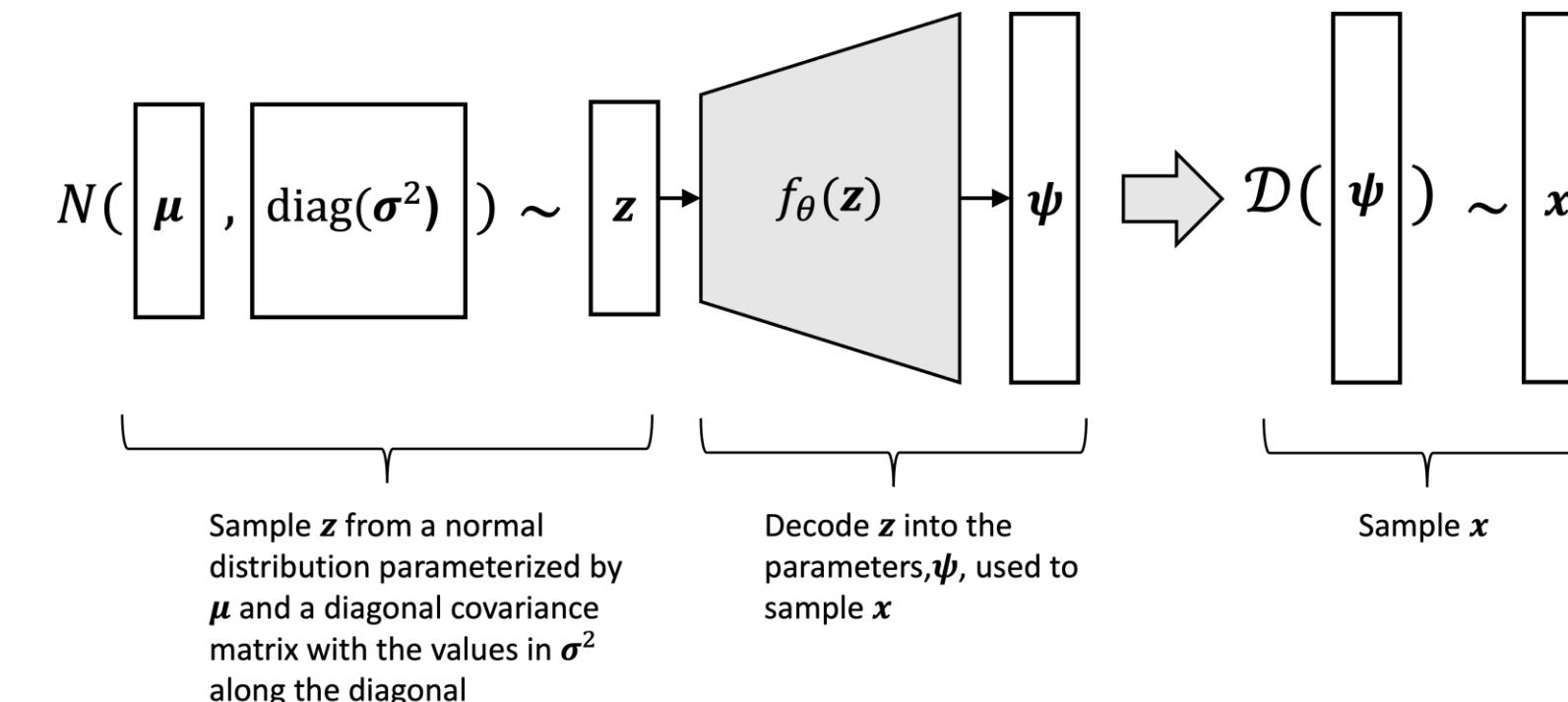
VAE

El **encoder**, parametrizado por ϕ , transforma los datos de entrada en una distribución probabilística sobre el espacio latente.

El **decoder**, parametrizado por θ , utiliza estas representaciones latentes para reconstruir los datos de entrada.

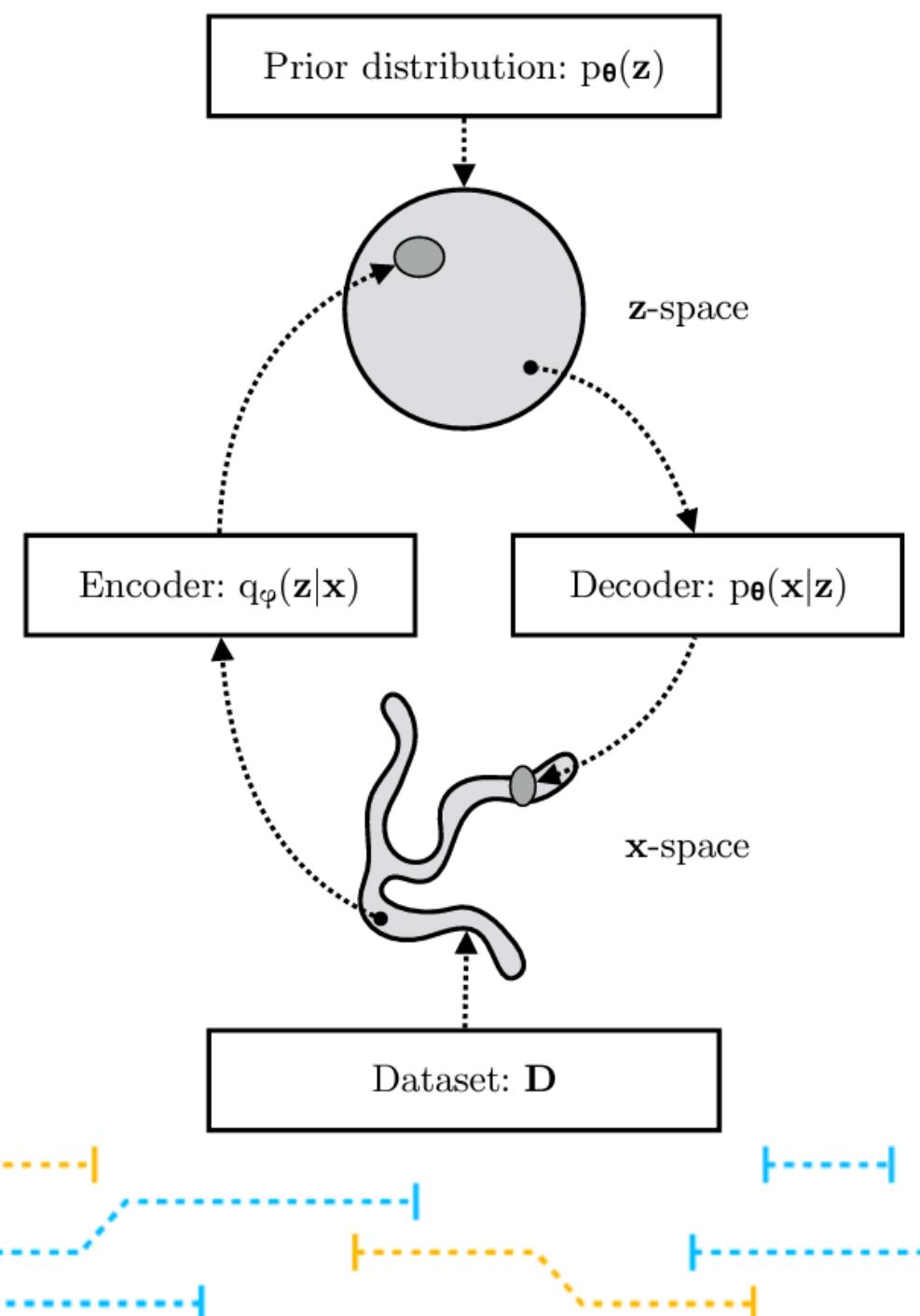


Supongamos que se nos da un conjunto de datos $x_1, \dots, x_n \in \mathbb{R}^D$ generados por una VAE.

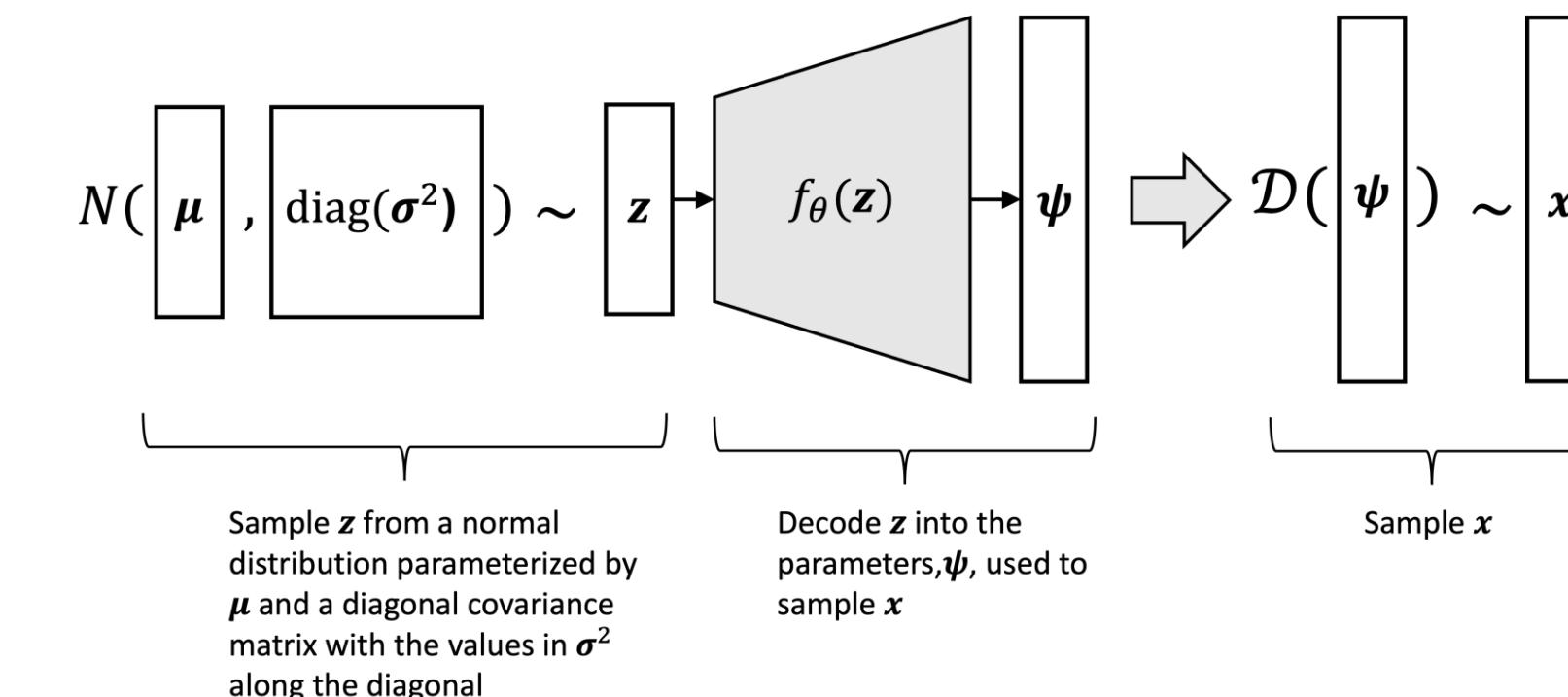


VAE

El **encoder**, parametrizado por ϕ , transforma los datos de entrada en una distribución probabilística sobre el espacio latente.
 El **decoder**, parametrizado por θ , utiliza estas representaciones latentes para reconstruir los datos de entrada.



Supongamos que se nos da un conjunto de datos $x_1, \dots, x_n \in \mathbb{R}^D$ generados por una VAE.



Posterior distribution representa cómo los datos observados actualizan la distribución de las variables latentes.

$$p_\theta(z_i | x_i) = \frac{p_\theta(x_i | z_i)p(z_i)}{\int p_\theta(x_i | z_i)p(z_i) dz_i}$$

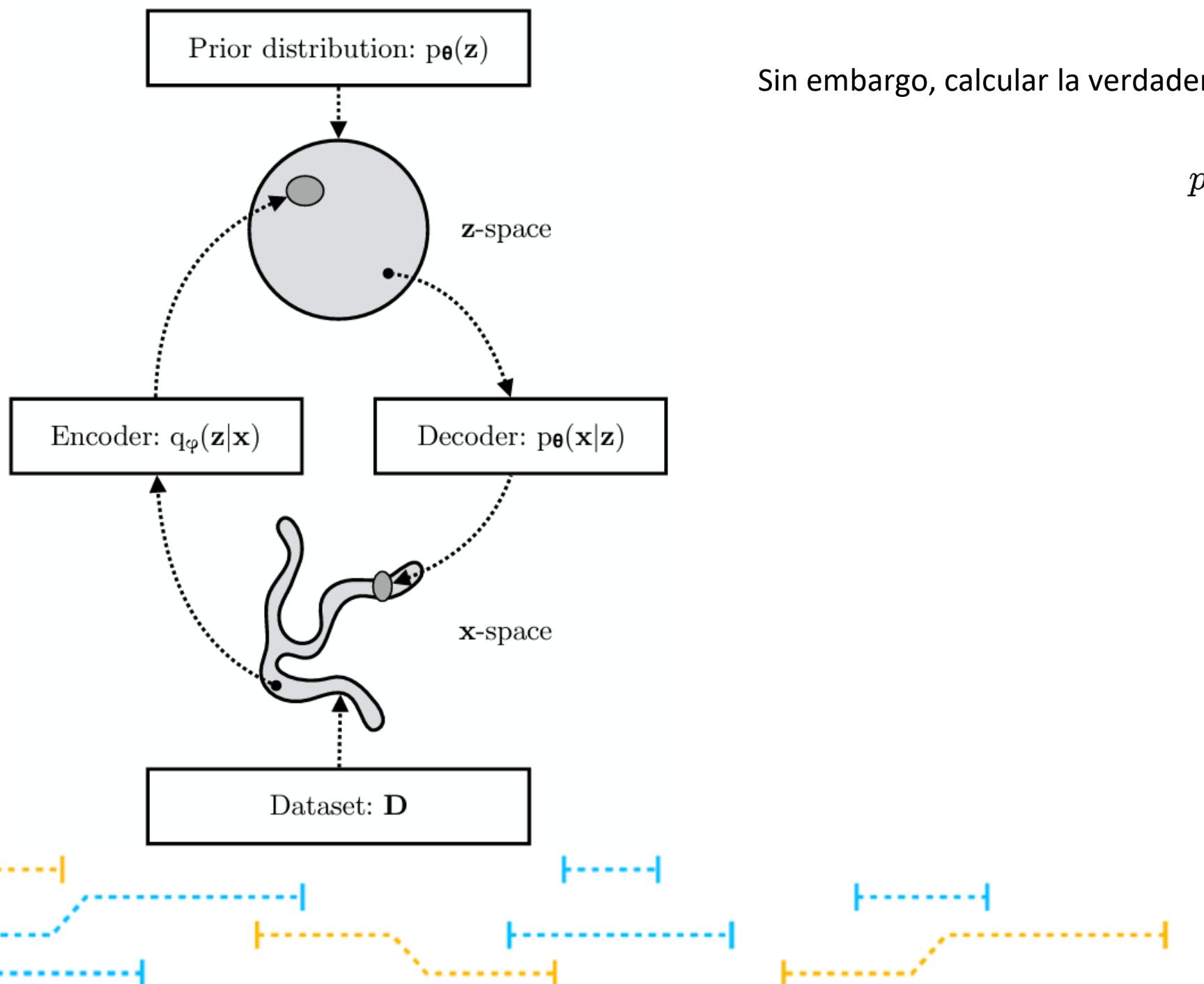
Bayes Theorem

TRANSFORMATEC

VAE

El **encoder**, parametrizado por ϕ , transforma los datos de entrada en una distribución probabilística sobre el espacio latente.

El **decoder**, parametrizado por θ , utiliza estas representaciones latentes para reconstruir los datos de entrada.



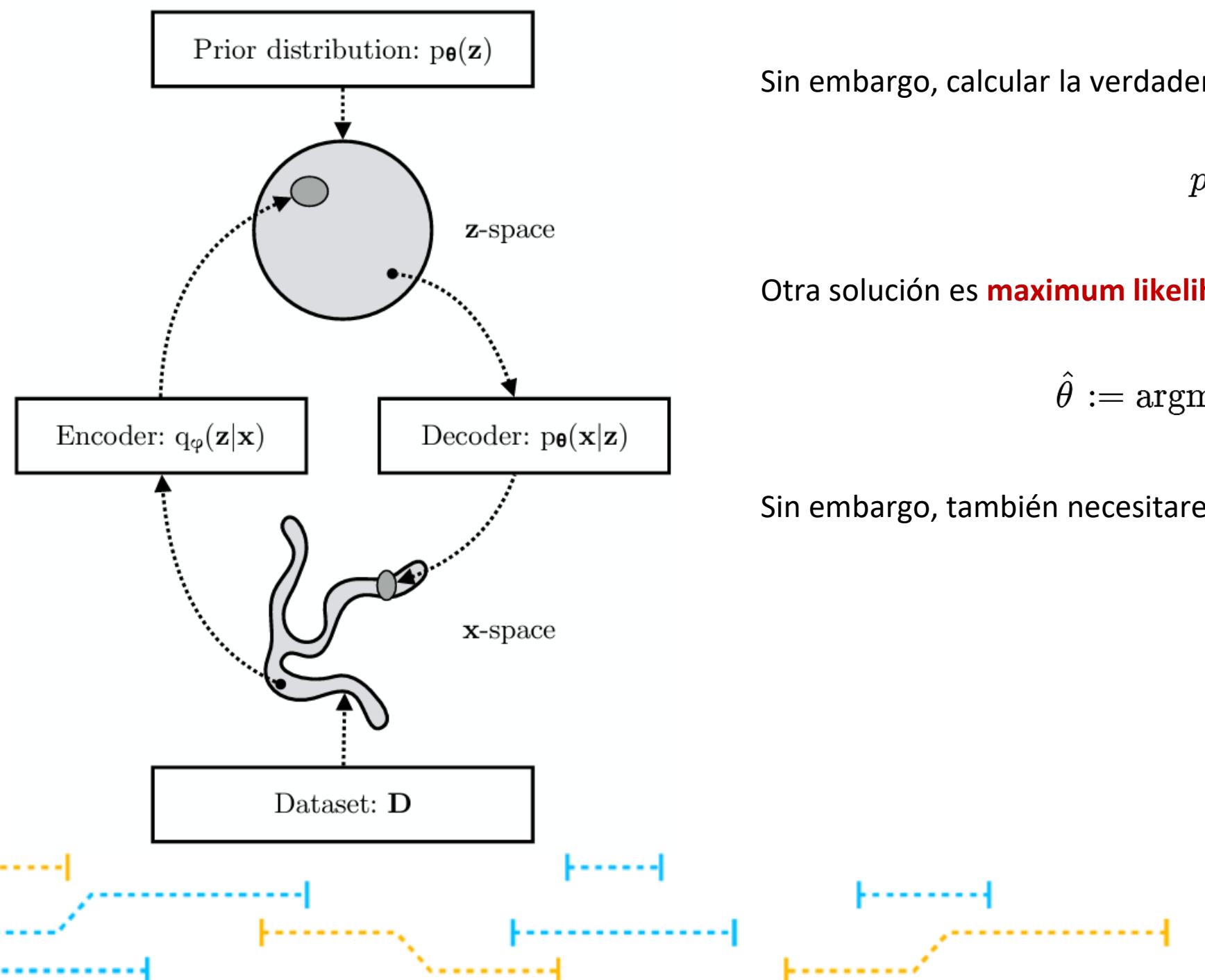
Sin embargo, calcular la verdadera posterior distribution es intratable debido a la **integral en el denominador**.

$$p_\theta(z_i | \mathbf{x}_i) = \frac{p_\theta(\mathbf{x}_i | z_i)p(z_i)}{\int p_\theta(\mathbf{x}_i | z_i)p(z_i) dz_i}$$

VAE

El **encoder**, parametrizado por ϕ , transforma los datos de entrada en una distribución probabilística sobre el espacio latente.

El **decoder**, parametrizado por θ , utiliza estas representaciones latentes para reconstruir los datos de entrada.



Sin embargo, calcular la verdadera posterior distribution es intratable debido a la **integral en el denominador**.

$$p_\theta(\mathbf{z}_i | \mathbf{x}_i) = \frac{p_\theta(\mathbf{x}_i | \mathbf{z}_i)p(\mathbf{z}_i)}{\int p_\theta(\mathbf{x}_i | \mathbf{z}_i)p(\mathbf{z}_i) d\mathbf{z}_i}$$

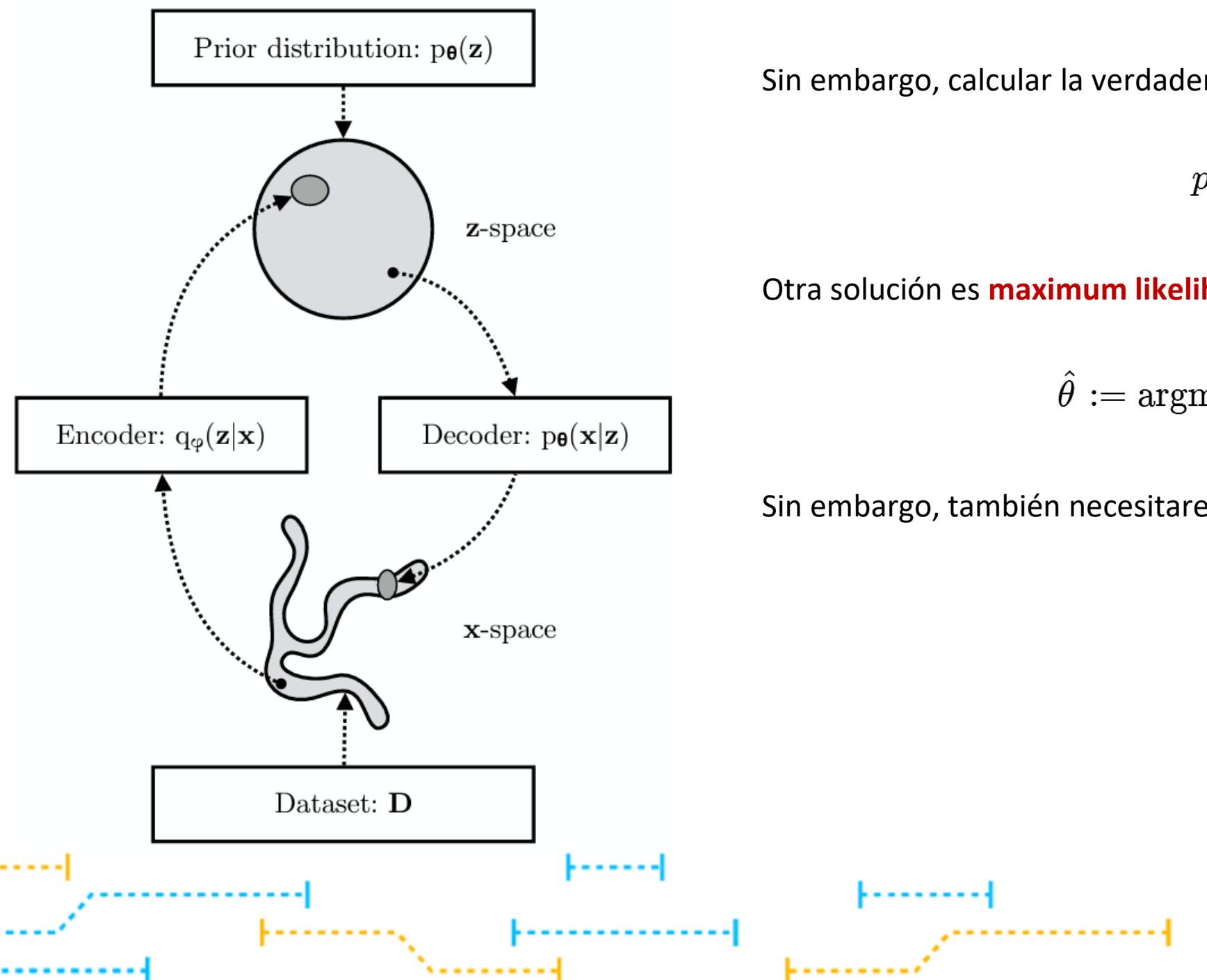
Otra solución es **maximum likelihood estimation**:

$$\hat{\theta} := \operatorname{argmax}_{\theta} \prod_{i=1}^n p_\theta(\mathbf{x}_i) = \operatorname{argmax}_{\theta} \prod_{i=1}^n \int p_\theta(\mathbf{x}_i | \mathbf{z}_i)p(\mathbf{z}_i) d\mathbf{z}_i$$

Sin embargo, también necesitaremos resolver dicha integral. :c

VAE

El **encoder**, parametrizado por ϕ , transforma los datos de entrada en una distribución probabilística sobre el espacio latente.
 El **decoder**, parametrizado por θ , utiliza estas representaciones latentes para reconstruir los datos de entrada.



Sin embargo, calcular la verdadera posterior distribution es intratable debido a la **integral en el denominador**.

$$p_\theta(z_i | x_i) = \frac{p_\theta(x_i | z_i)p(z_i)}{\int p_\theta(x_i | z_i)p(z_i) dz_i}$$

Otra solución es **maximum likelihood estimation**:

$$\hat{\theta} := \operatorname{argmax}_\theta \prod_{i=1}^n p_\theta(x_i) = \operatorname{argmax}_\theta \prod_{i=1}^n \int p_\theta(x_i | z_i)p(z_i) dz_i$$

Sin embargo, también necesitaremos resolver dicha integral. :c

Aquí entra **Variational Inference** para solucionar el problema :D

ELBO

Definimos la **evidencia** como $\log p(x; \theta)$

Es la probabilidad marginal de los datos observados bajo el modelo actual.

Cuantifica cuán bien el modelo con parámetros θ explica los datos observados, es decir es la probabilidad de que los datos hayan sido generados por el modelo tal como está parametrizado

Nosotros podemos probar que la evidencia está acotada por el **ELBO** (Evidence Lower BOund):

$$\log p(x; \theta) \geq E_{Z \sim q} \left[\log \frac{p(x, Z; \theta)}{q(Z)} \right]$$

$$ELBO := E_{Z \sim q} \left[\log \frac{p(x, Z; \theta)}{q(Z)} \right]$$



ELBO

Definimos la **evidencia** como $\log p(x; \theta)$

Es la probabilidad marginal de los datos observados bajo el modelo actual.

Cuantifica cuán bien el modelo con parámetros θ explica los datos observados, es decir es la probabilidad de que los datos hayan sido generados por el modelo tal como está parametrizado

Nosotros podemos probar que la evidencia está acotada por el **ELBO** (Evidence Lower BOund):

$$\log p(x; \theta) \geq E_{Z \sim q} \left[\log \frac{p(x, Z; \theta)}{q(Z)} \right] \quad ELBO := E_{Z \sim q} \left[\log \frac{p(x, Z; \theta)}{q(Z)} \right]$$

La demostración es directa:

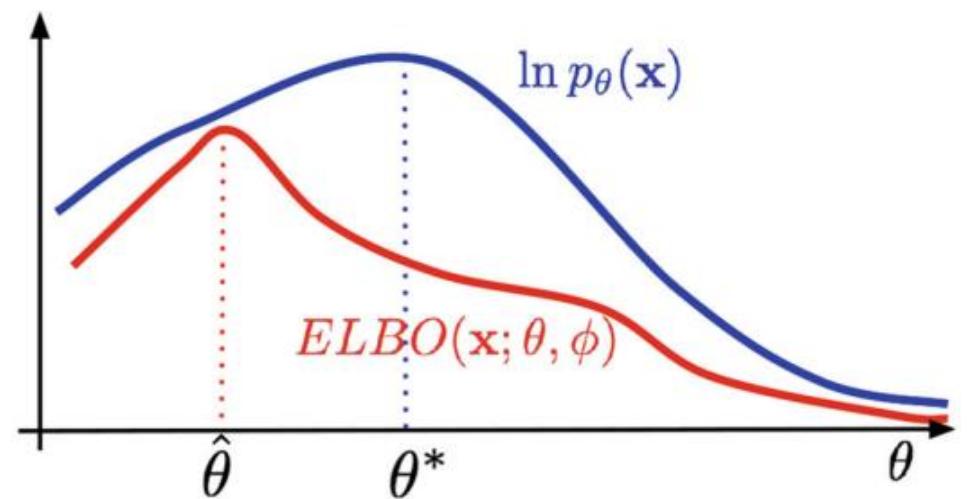
$$\begin{aligned} \log p(x; \theta) &= \log \int p(x, z; \theta) dz \\ &= \log \int p(x, z; \theta) \frac{q(z)}{q(z)} dz \\ &= \log E_{Z \sim q} \left[\frac{p(x, Z)}{q(z)} \right] \geq E_{Z \sim q} \left[\log \frac{p(x, Z)}{q(z)} \right] \end{aligned}$$

Jensen's Inequality
:D

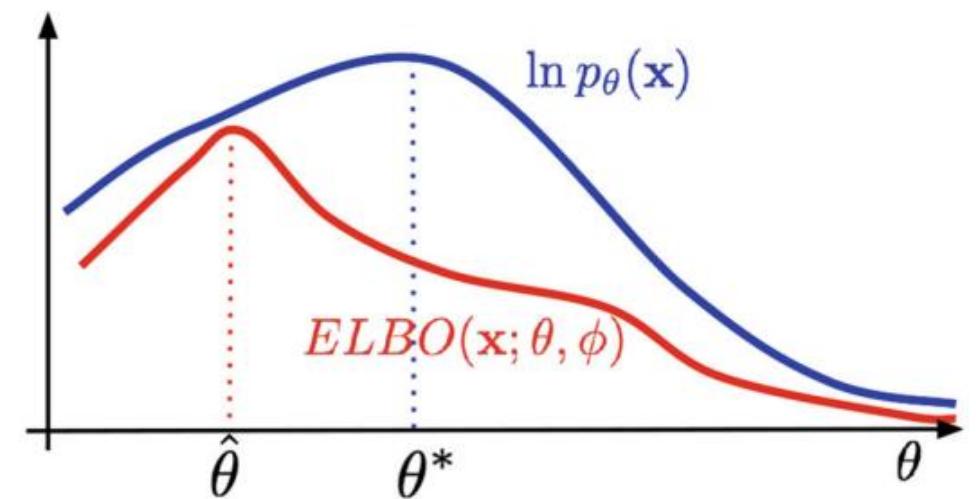


ELBO

$$\text{ELBO}(\phi, \theta) = \sum_{i=1}^n E_{z_i \sim q} [\log p_\theta(\mathbf{x}_i, \mathbf{z}_i) - \log q(\mathbf{z}_i \mid \mathbf{x}_i)]$$



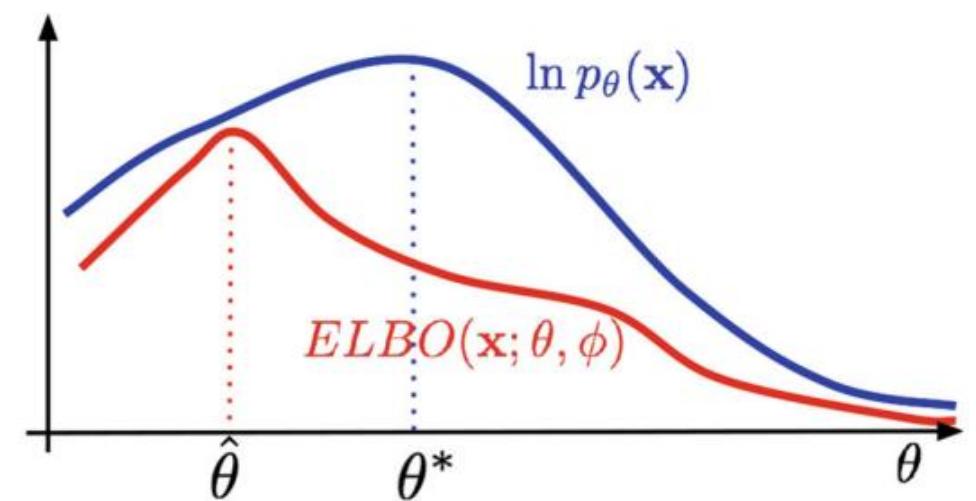
ELBO



$$\begin{aligned}
 \text{ELBO}(\phi, \theta) &= \sum_{i=1}^n E_{z_i \sim q} [\log p_\theta(\mathbf{x}_i, \mathbf{z}_i) - \log q(\mathbf{z}_i \mid \mathbf{x}_i)] \\
 &= \sum_{i=1}^n \int q(\mathbf{z}_i \mid \mathbf{x}_i) [\log p_\theta(\mathbf{x}_i, \mathbf{z}_i) - \log q(\mathbf{z}_i \mid \mathbf{x}_i)] d\mathbf{z}_i
 \end{aligned}$$



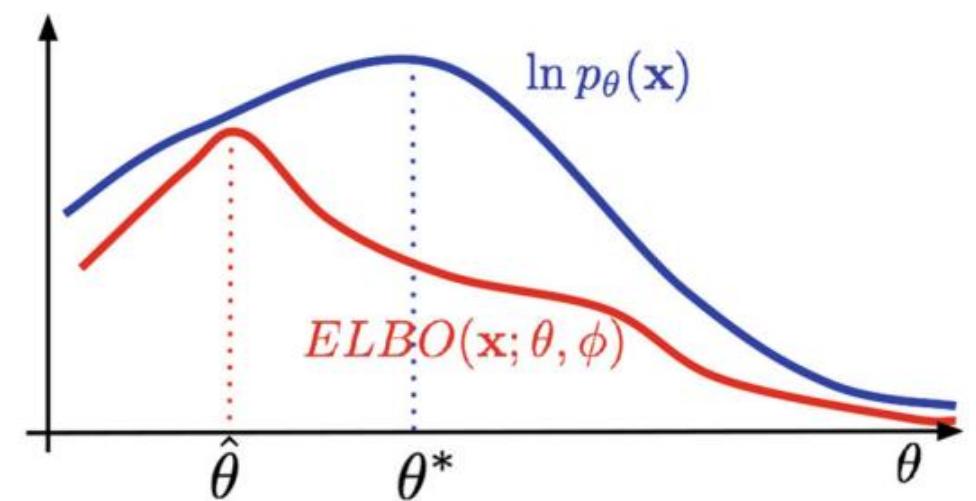
ELBO



$$\begin{aligned}
 \text{ELBO}(\phi, \theta) &= \sum_{i=1}^n E_{z_i \sim q} [\log p_\theta(\mathbf{x}_i, \mathbf{z}_i) - \log q(\mathbf{z}_i \mid \mathbf{x}_i)] \\
 &= \sum_{i=1}^n \int q(\mathbf{z}_i \mid \mathbf{x}_i) [\log p_\theta(\mathbf{x}_i, \mathbf{z}_i) - \log q(\mathbf{z}_i \mid \mathbf{x}_i)] d\mathbf{z}_i \\
 &= \sum_{i=1}^n \int q(\mathbf{z}_i \mid \mathbf{x}_i) [\log p_\theta(\mathbf{x}_i \mid \mathbf{z}_i) + \log p(\mathbf{z}_i) - \log q(\mathbf{z}_i \mid \mathbf{x}_i)] d\mathbf{z}_i
 \end{aligned}$$



ELBO



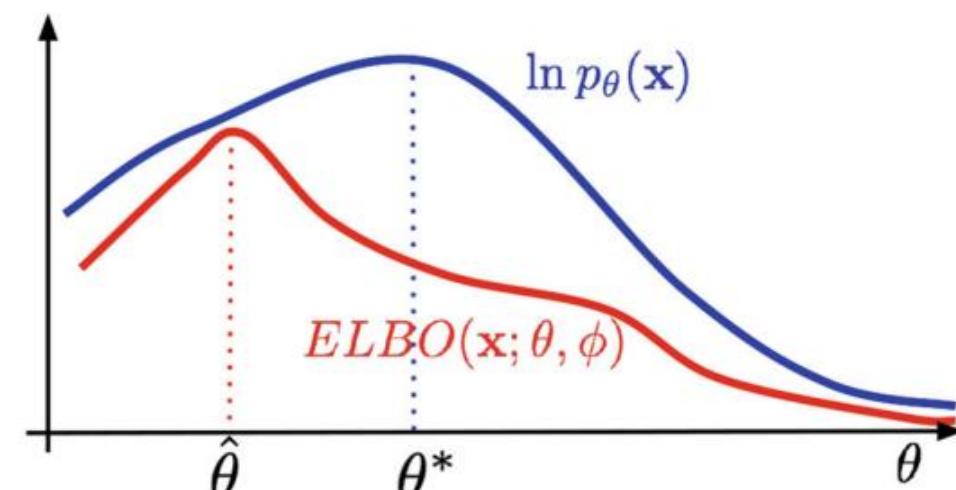
$$\begin{aligned}
 \text{ELBO}(\phi, \theta) &= \sum_{i=1}^n E_{z_i \sim q} [\log p_\theta(\mathbf{x}_i, \mathbf{z}_i) - \log q(\mathbf{z}_i \mid \mathbf{x}_i)] \\
 &= \sum_{i=1}^n \int q(\mathbf{z}_i \mid \mathbf{x}_i) [\log p_\theta(\mathbf{x}_i, \mathbf{z}_i) - \log q(\mathbf{z}_i \mid \mathbf{x}_i)] d\mathbf{z}_i \\
 &= \sum_{i=1}^n \int q(\mathbf{z}_i \mid \mathbf{x}_i) [\log p_\theta(\mathbf{x}_i \mid \mathbf{z}_i) + \log p(\mathbf{z}_i) - \log q(\mathbf{z}_i \mid \mathbf{x}_i)] d\mathbf{z}_i \\
 &= \sum_{i=1}^n E_{z_i \sim q} [\log p_\theta(\mathbf{x}_i \mid \mathbf{z}_i)] + \boxed{\sum_{i=1}^n \int q(\mathbf{z}_i \mid \mathbf{x}_i) [\log p(\mathbf{z}_i) - \log q(\mathbf{z}_i \mid \mathbf{x}_i)] d\mathbf{z}_i}
 \end{aligned}$$



TRANSFORMATEC

Diederik P. Kingma and Max Welling (2013) "Auto-Encoding Variational Bayes".
arXiv preprint arXiv:1312.6114

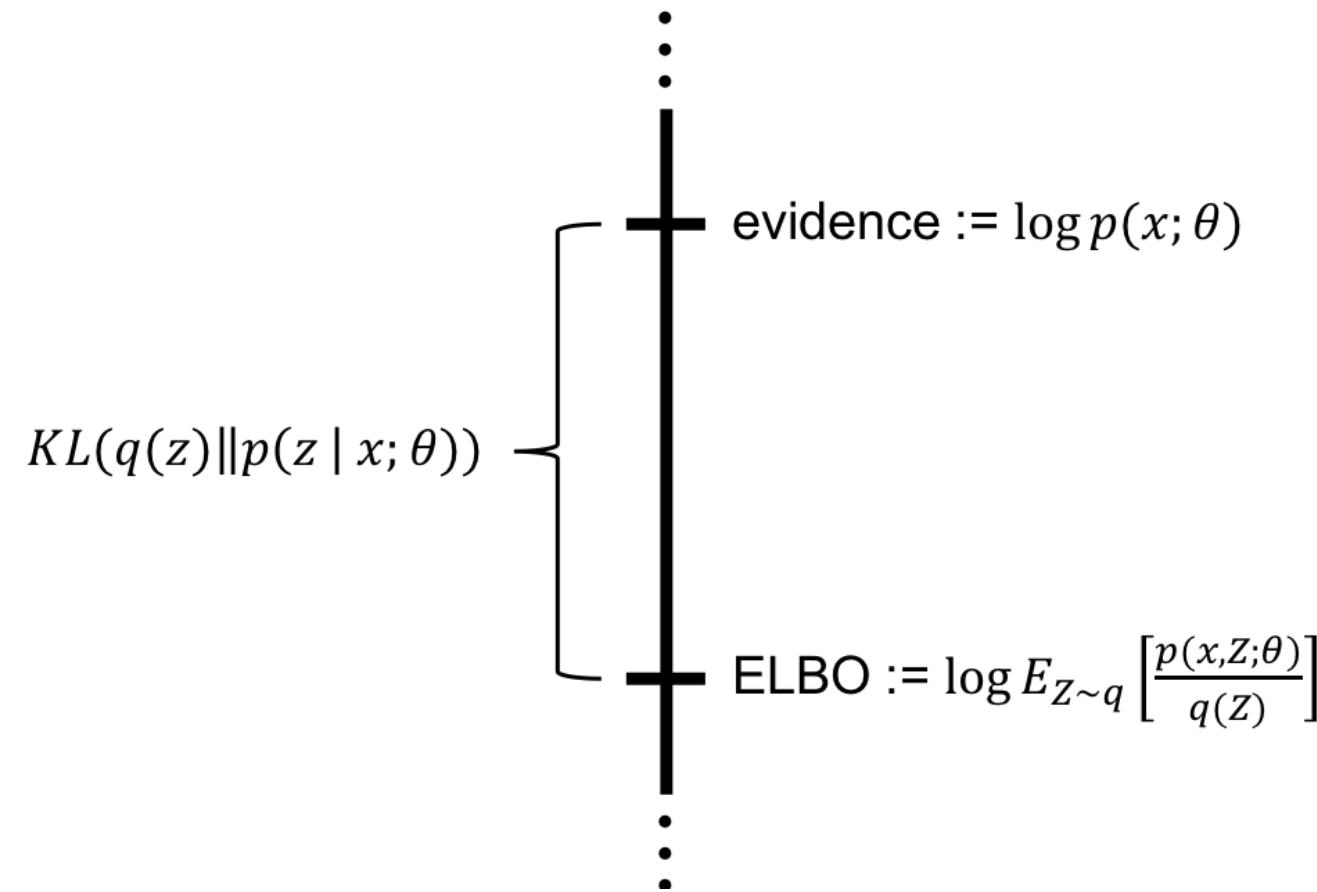
ELBO



$$\begin{aligned}
 \text{ELBO}(\phi, \theta) &= \sum_{i=1}^n E_{z_i \sim q} [\log p_\theta(\mathbf{x}_i, \mathbf{z}_i) - \log q(\mathbf{z}_i \mid \mathbf{x}_i)] \\
 &= \sum_{i=1}^n \int q(\mathbf{z}_i \mid \mathbf{x}_i) [\log p_\theta(\mathbf{x}_i, \mathbf{z}_i) - \log q(\mathbf{z}_i \mid \mathbf{x}_i)] d\mathbf{z}_i \\
 &= \sum_{i=1}^n \int q(\mathbf{z}_i \mid \mathbf{x}_i) [\log p_\theta(\mathbf{x}_i \mid \mathbf{z}_i) + \log p(\mathbf{z}_i) - \log q(\mathbf{z}_i \mid \mathbf{x}_i)] d\mathbf{z}_i \\
 &= \sum_{i=1}^n E_{z_i \sim q} [\log p_\theta(\mathbf{x}_i \mid \mathbf{z}_i)] + \sum_{i=1}^n \int q(\mathbf{z}_i \mid \mathbf{x}_i) [\log p(\mathbf{z}_i) - \log q(\mathbf{z}_i \mid \mathbf{x}_i)] d\mathbf{z}_i \\
 &= \sum_{i=1}^n E_{z_i \sim q} [\log p_\theta(\mathbf{x}_i \mid \mathbf{z}_i)] + \boxed{\sum_{i=1}^n E_{z_i \sim q} \left[\log \frac{p(\mathbf{z}_i)}{q(\mathbf{z}_i \mid \mathbf{x}_i)} \right]}
 \end{aligned}$$



ELBO

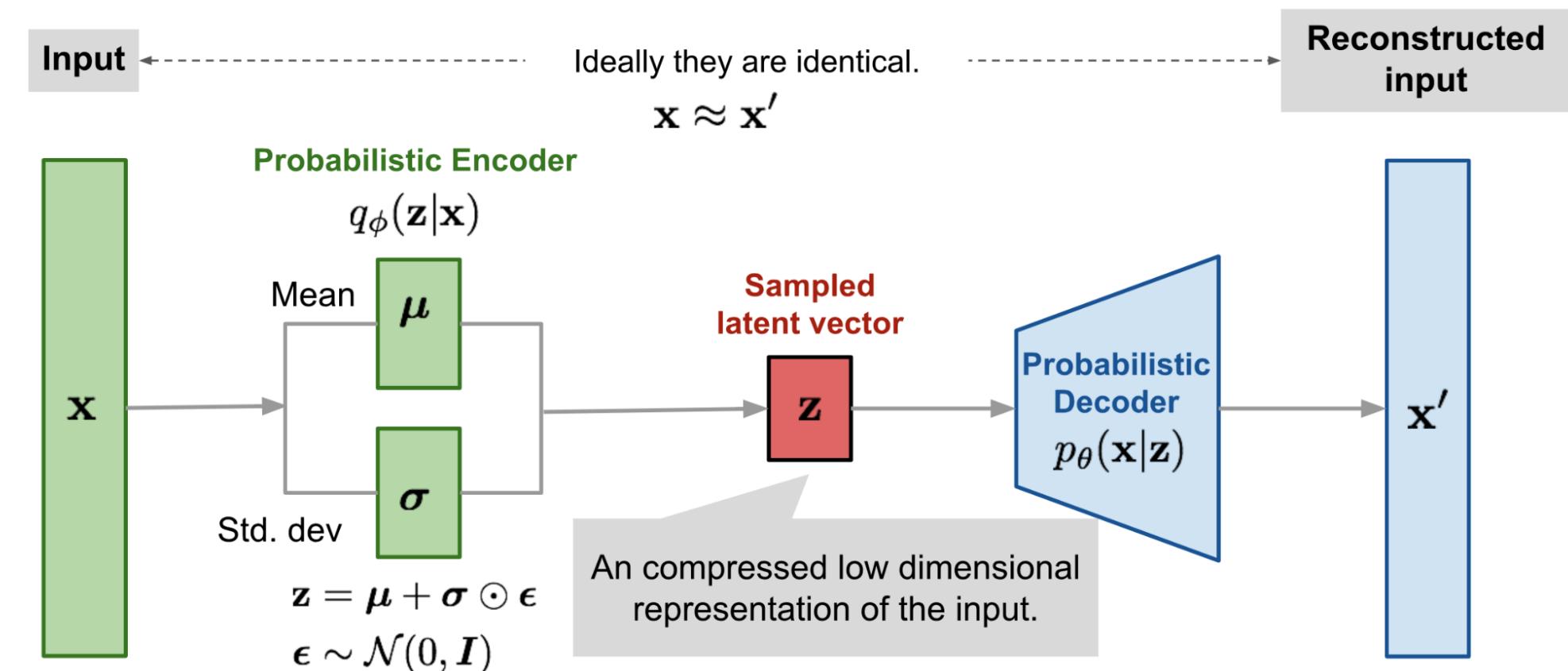


$$\begin{aligned}
 \text{ELBO}(\phi, \theta) &= \sum_{i=1}^n E_{\mathbf{z}_i \sim q} [\log p_\theta(\mathbf{x}_i, \mathbf{z}_i) - \log q(\mathbf{z}_i \mid \mathbf{x}_i)] \\
 &= \sum_{i=1}^n \int q(\mathbf{z}_i \mid \mathbf{x}_i) [\log p_\theta(\mathbf{x}_i, \mathbf{z}_i) - \log q(\mathbf{z}_i \mid \mathbf{x}_i)] d\mathbf{z}_i \\
 &= \sum_{i=1}^n \int q(\mathbf{z}_i \mid \mathbf{x}_i) [\log p_\theta(\mathbf{x}_i \mid \mathbf{z}_i) + \log p(\mathbf{z}_i) - \log q(\mathbf{z}_i \mid \mathbf{x}_i)] d\mathbf{z}_i \\
 &= \sum_{i=1}^n E_{\mathbf{z}_i \sim q} [\log p_\theta(\mathbf{x}_i \mid \mathbf{z}_i)] + \sum_{i=1}^n \int q(\mathbf{z}_i \mid \mathbf{x}_i) [\log p(\mathbf{z}_i) - \log q(\mathbf{z}_i \mid \mathbf{x}_i)] d\mathbf{z}_i \\
 &= \sum_{i=1}^n E_{\mathbf{z}_i \sim q} [\log p_\theta(\mathbf{x}_i \mid \mathbf{z}_i)] + \sum_{i=1}^n E_{\mathbf{z}_i \sim q} \left[\log \frac{p(\mathbf{z}_i)}{q(\mathbf{z}_i \mid \mathbf{x}_i)} \right] \\
 &= \boxed{\sum_{i=1}^n E_{\mathbf{z}_i \sim q} [\log p_\theta(\mathbf{x}_i \mid \mathbf{z}_i)] - KL(q(\mathbf{z}_i \mid \mathbf{x}_i) \parallel p(\mathbf{z}_i))}
 \end{aligned}$$

Function Loss



VAE



$$\max \mathcal{L}(\phi, \beta) = \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) - D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z}))$$



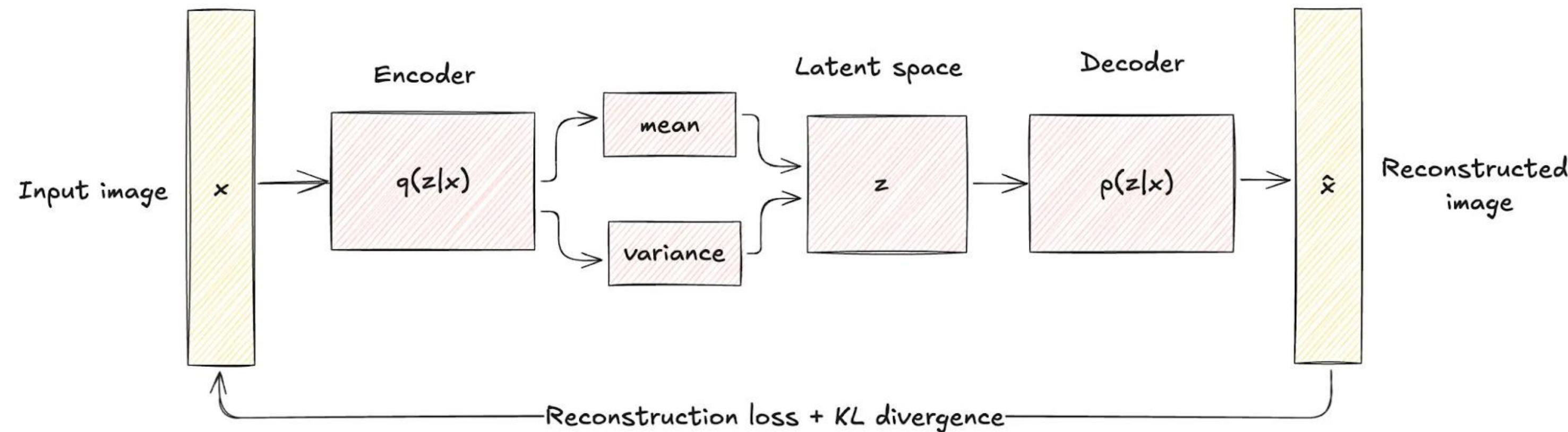
VAE

$$\max \mathcal{L}_i(\phi, \beta) = \frac{1}{L} \sum_{l=1}^L \log p_{\theta}(x_i|z_l) - D_{\text{KL}}(q_{\phi}(z|x_i)||p_{\theta}(z))$$

Decoder **Encoder** **Fixed**



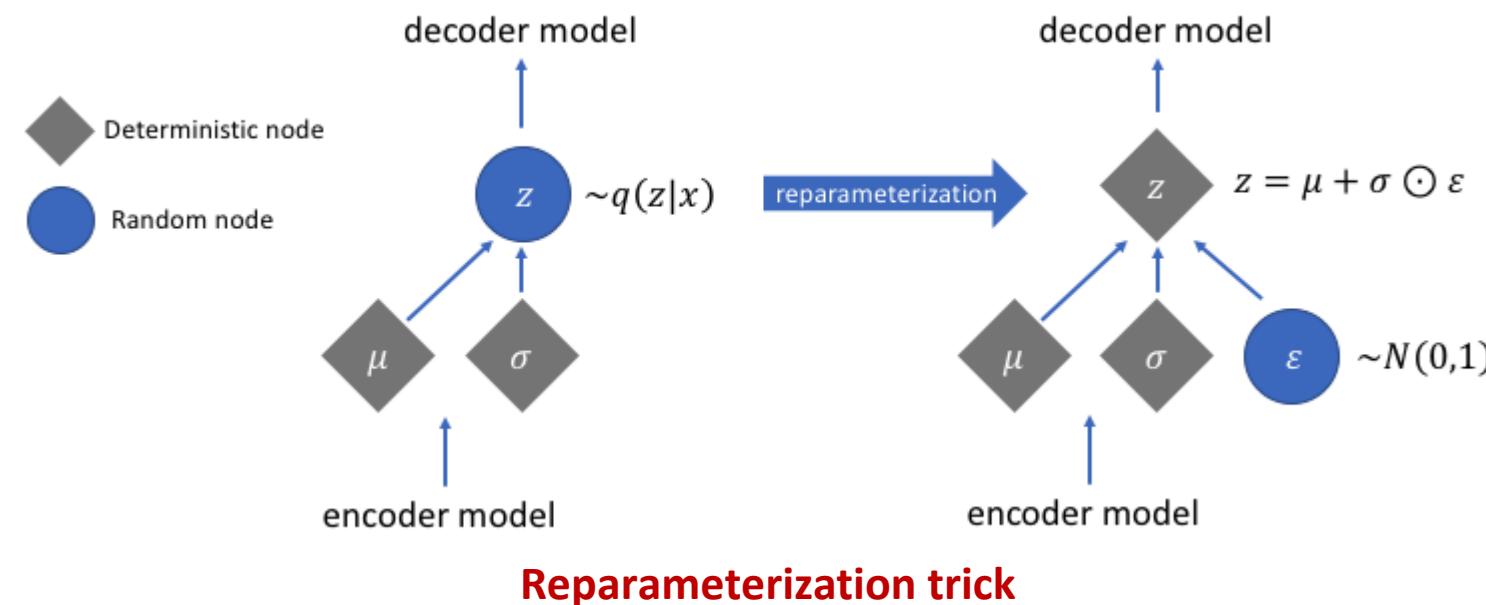
La operación de muestreo es **no diferenciable**



VAE

$$\max \mathcal{L}_i(\phi, \beta) = \frac{1}{L} \sum_{l=1}^L \log p_{\theta}(x_i|z_l) - D_{\text{KL}} \left(q_{\phi}(z|x_i) || p_{\theta}(z) \right)$$

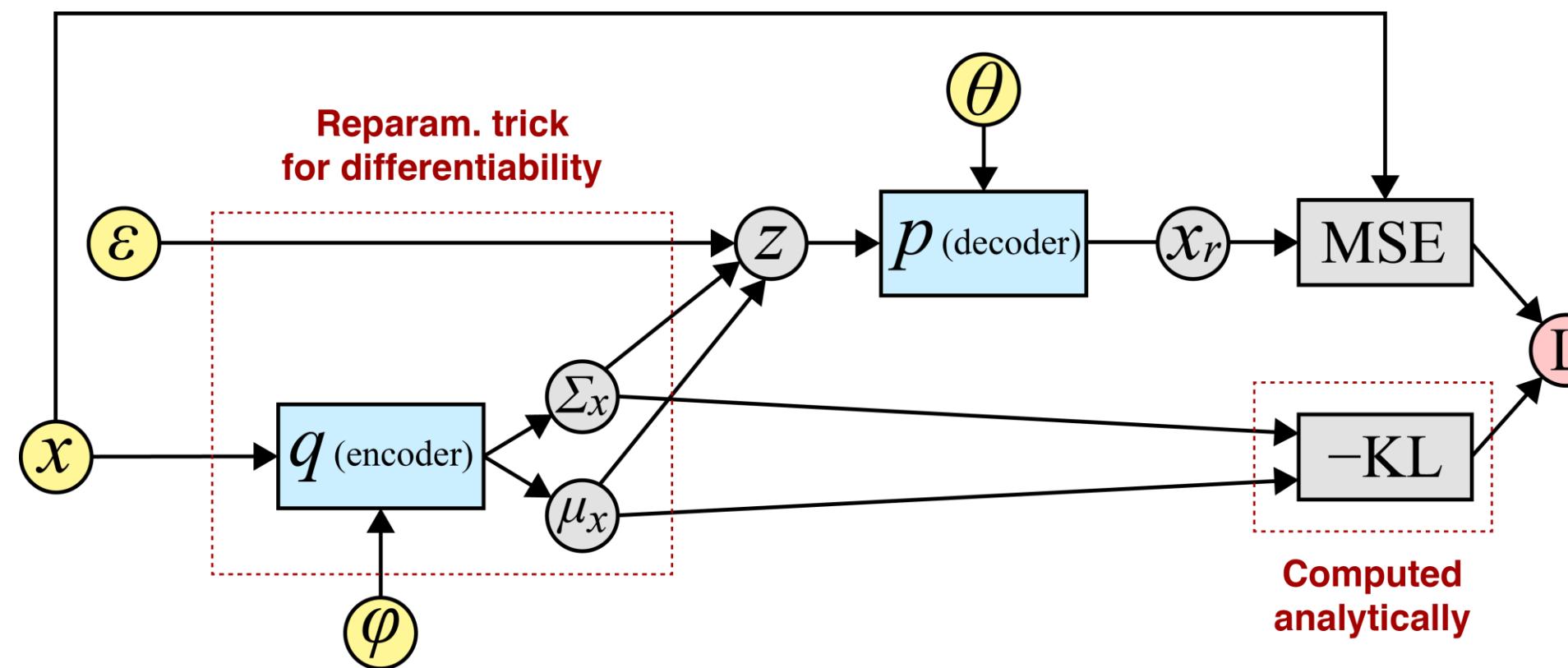
Decoder Encoder Fixed



VAE

$$\max \mathcal{L}_i(\phi, \beta) = \frac{1}{L} \sum_{l=1}^L \log p_{\theta}(x_i|z_l) - D_{\text{KL}} \left(q_{\phi}(z|x_i) || p_{\theta}(z) \right)$$

Decoder Encoder Fixed



VAE

$$\max \mathcal{L}_i(\phi, \beta) = \frac{1}{L} \sum_{l=1}^L \log p_{\theta}(x_i | z_l) - D_{\text{KL}} \left(q_{\phi}(z|x_i) || p_{\theta}(z) \right)$$

Decoder **Encoder** **Fixed**

La divergencia KL entre $q(z|x)$ y $p(z)$ es: $D_{KL}(p(z|x)||p(z)) = \int q(z|x) \log \frac{q(z|x)}{p(z)} dz$



VAE

$$\max \mathcal{L}_i(\phi, \beta) = \frac{1}{L} \sum_{l=1}^L \log p_{\theta}(x_i | z_l) - D_{\text{KL}} \left(q_{\phi}(z|x_i) || p_{\theta}(z) \right)$$

Decoder Encoder Fixed

La divergencia KL entre $q(z|x)$ y $p(z)$ es: $D_{KL}(p(z|x)||p(z)) = \int q(z|x) \log \frac{q(z|x)}{p(z)} dz$

Asumimos:

- $q(z|x) \sim \mathcal{N}(z; \mu, \sigma^2)$. Entonces:
- $p(z) = \mathcal{N}(z; 0, I)$, cuya densidad es:

$$q(z|x) = \frac{1}{(2\pi)^{d/2} \prod_{j=1}^d \sigma_j} \exp \left(-\frac{1}{2} \sum_{j=1}^d \frac{(z_j - \mu_j)^2}{\sigma_j^2} \right)$$

$$p(z) = \frac{1}{(2\pi)^{d/2}} \exp \left(-\frac{1}{2} \sum_{j=1}^d z_j^2 \right)$$



VAE

$$\max \mathcal{L}_i(\phi, \beta) = \frac{1}{L} \sum_{l=1}^L \log p_{\theta}(x_i | z_l) - D_{\text{KL}} \left(q_{\phi}(z|x_i) || p_{\theta}(z) \right)$$

Decoder **Encoder** **Fixed**

La divergencia KL entre $q(z|x)$ y $p(z)$ es: $D_{KL}(p(z|x)||p(z)) = \int q(z|x) \log \frac{q(z|x)}{p(z)} dz$

Asumimos:

- $q(z|x) \sim \mathcal{N}(z; \mu, \sigma^2)$. Entonces:
- $p(z) = \mathcal{N}(z; 0, I)$, cuya densidad es:

$$q(z|x) = \frac{1}{(2\pi)^{d/2} \prod_{j=1}^d \sigma_j} \exp \left(-\frac{1}{2} \sum_{j=1}^d \frac{(z_j - \mu_j)^2}{\sigma_j^2} \right)$$

$$p(z) = \frac{1}{(2\pi)^{d/2}} \exp \left(-\frac{1}{2} \sum_{j=1}^d z_j^2 \right)$$

Desarrollamos:

$$\log \frac{q(z|x)}{p(z)} = \log \left(\frac{1}{(2\pi)^{d/2} \prod_{j=1}^d \sigma_j} \exp \left(-\frac{1}{2} \sum_{j=1}^d \frac{(z_j - \mu_j)^2}{\sigma_j^2} \right) \cdot (2\pi)^{d/2} \exp \left(\frac{1}{2} \sum_{j=1}^d z_j^2 \right) \right) = -\sum_{j=1}^d \log \sigma_j - \frac{1}{2} \sum_{j=1}^d \frac{(z_j - \mu_j)^2}{\sigma_j^2} + \frac{1}{2} \sum_{j=1}^d z_j^2$$



VAE

$$\max \mathcal{L}_i(\phi, \beta) = \frac{1}{L} \sum_{l=1}^L \log p_{\theta}(x_i | z_l) - D_{\text{KL}} \left(q_{\phi}(z|x_i) || p_{\theta}(z) \right)$$

Decoder **Encoder** **Fixed**

La divergencia KL entre $q(z|x)$ y $p(z)$ es: $D_{KL}(p(z|x)||p(z)) = \int q(z|x) \log \frac{q(z|x)}{p(z)} dz$

Asumimos:

- $q(z|x) \sim \mathcal{N}(z; \mu, \sigma^2)$. Entonces:
- $p(z) = \mathcal{N}(z; 0, I)$, cuya densidad es:

$$q(z|x) = \frac{1}{(2\pi)^{d/2} \prod_{j=1}^d \sigma_j} \exp \left(-\frac{1}{2} \sum_{j=1}^d \frac{(z_j - \mu_j)^2}{\sigma_j^2} \right)$$

$$p(z) = \frac{1}{(2\pi)^{d/2}} \exp \left(-\frac{1}{2} \sum_{j=1}^d z_j^2 \right)$$

Desarrollamos:

$$\log \frac{q(z|x)}{p(z)} = \log \left(\frac{1}{(2\pi)^{d/2} \prod_{j=1}^d \sigma_j} \exp \left(-\frac{1}{2} \sum_{j=1}^d \frac{(z_j - \mu_j)^2}{\sigma_j^2} \right) \cdot (2\pi)^{d/2} \exp \left(\frac{1}{2} \sum_{j=1}^d z_j^2 \right) \right) = -\sum_{j=1}^d \log \sigma_j - \frac{1}{2} \sum_{j=1}^d \frac{(z_j - \mu_j)^2}{\sigma_j^2} + \frac{1}{2} \sum_{j=1}^d z_j^2$$

Reemplazando: $D_{KL}(p(z|x)||p(z)) = \mathbb{E}_{q(z|x)} \left[\log \frac{q(z|x)}{p(z)} \right] = -\sum_{j=1}^d \log \sigma_j - \frac{1}{2} \sum_{j=1}^d \mathbb{E}_{q(z|x)} \left[\frac{(z_j - \mu_j)^2}{\sigma_j^2} \right] + \frac{1}{2} \sum_{j=1}^d \mathbb{E}_{q(z|x)} [z_j^2]$



VAE

$$\max \mathcal{L}_i(\phi, \beta) = \frac{1}{L} \sum_{l=1}^L \log p_{\theta}(x_i | z_l) - D_{\text{KL}} \left(q_{\phi}(z|x_i) || p_{\theta}(z) \right)$$

Decoder **Encoder** **Fixed**

La divergencia KL entre $q(z|x)$ y $p(z)$ es:

$$D_{KL}(p(z|x)||p(z)) = -\sum_{j=1}^d \log \sigma_j - \frac{1}{2} \sum_{j=1}^d \mathbb{E}_{q(z|x)} \left[\frac{(z_j - \mu_j)^2}{\sigma_j^2} \right] + \frac{1}{2} \sum_{j=1}^d \mathbb{E}_{q(z|x)} [z_j^2]$$


VAE

$$\max \mathcal{L}_i(\phi, \beta) = \frac{1}{L} \sum_{l=1}^L \log p_\theta(x_i|z_l) - D_{\text{KL}}(q_\phi(z|x_i) || p_\theta(z))$$

Decoder **Encoder** **Fixed**

La divergencia KL entre $q(z|x)$ y $p(z)$ es: $D_{KL}(p(z|x)||p(z)) = -\sum_{j=1}^d \log \sigma_j - \frac{1}{2} \sum_{j=1}^d \mathbb{E}_{q(z|x)} \left[\frac{(z_j - \mu_j)^2}{\sigma_j^2} \right] + \frac{1}{2} \sum_{j=1}^d \mathbb{E}_{q(z|x)} [z_j^2]$

Termino 1 $-\sum_{j=1}^d \log \sigma_j$ Este término es independiente de z , por lo que queda como está.



VAE

$$\max \mathcal{L}_i(\phi, \beta) = \frac{1}{L} \sum_{l=1}^L \log p_{\theta}(x_i | z_l) - D_{\text{KL}} \left(q_{\phi}(z|x_i) || p_{\theta}(z) \right)$$

Decoder **Encoder** **Fixed**

La divergencia KL entre $q(z|x)$ y $p(z)$ es: $D_{KL}(p(z|x)||p(z)) = -\sum_{j=1}^d \log \sigma_j - \frac{1}{2} \sum_{j=1}^d \mathbb{E}_{q(Z|X)} \left[\frac{(z_j - \mu_j)^2}{\sigma_j^2} \right] + \frac{1}{2} \sum_{j=1}^d \mathbb{E}_{q(Z|X)} [z_j^2]$

Termino 1 $-\sum_{j=1}^d \log \sigma_j$ Este término es independiente de z , por lo que queda como está.

Termino 2 $\mathbb{E}_{q(Z|X)} \left[\frac{(z_j - \mu_j)^2}{\sigma_j^2} \right]$

Dado que $q(z|x) \sim \mathcal{N}(\mu_j, \sigma_j^2)$, la variable $z_j - \mu_j$ tiene distribución $\mathcal{N}(0, \sigma_j^2)$.

Entonces:

$$\mathbb{E}_{q(Z|X)} \left[\frac{(z_j - \mu_j)^2}{\sigma_j^2} \right] = \frac{1}{\sigma_j^2} \mathbb{E} [(z_j - \mu_j)^2] = \frac{\sigma_j^2}{\sigma_j^2} = 1$$

Por lo tanto:

$$-\frac{1}{2} \sum_{j=1}^d \mathbb{E}_{q(Z|X)} \left[\frac{(z_j - \mu_j)^2}{\sigma_j^2} \right] = -\frac{1}{2} \sum_{j=1}^d 1 = -\frac{d}{2}$$



VAE

$$\max \mathcal{L}_i(\phi, \beta) = \frac{1}{L} \sum_{l=1}^L \log p_{\theta}(x_i | z_l) - D_{\text{KL}} \left(q_{\phi}(z|x_i) || p_{\theta}(z) \right)$$

Decoder **Encoder** **Fixed**

La divergencia KL entre $q(z|x)$ y $p(z)$ es: $D_{KL}(p(z|x)||p(z)) = -\sum_{j=1}^d \log \sigma_j - \frac{1}{2} \sum_{j=1}^d \mathbb{E}_{q(Z|x)} \left[\frac{(z_j - \mu_j)^2}{\sigma_j^2} \right] + \frac{1}{2} \sum_{j=1}^d \mathbb{E}_{q(Z|x)} [z_j^2]$

Termino 1 $-\sum_{j=1}^d \log \sigma_j$ Este término es independiente de z , por lo que queda como está.

Termino 2 $\mathbb{E}_{q(Z|x)} \left[\frac{(z_j - \mu_j)^2}{\sigma_j^2} \right]$

Dado que $q(z|x) \sim \mathcal{N}(\mu_j, \sigma_j^2)$, la variable $z_j - \mu_j$ tiene distribución $\mathcal{N}(0, \sigma_j^2)$. Entonces:

$$\mathbb{E}_{q(Z|x)} \left[\frac{(z_j - \mu_j)^2}{\sigma_j^2} \right] = \frac{1}{\sigma_j^2} \mathbb{E} [(z_j - \mu_j)^2] = \frac{\sigma_j^2}{\sigma_j^2} = 1$$

Por lo tanto:

$$-\frac{1}{2} \sum_{j=1}^d \mathbb{E}_{q(Z|x)} \left[\frac{(z_j - \mu_j)^2}{\sigma_j^2} \right] = -\frac{1}{2} \sum_{j=1}^d 1 = -\frac{d}{2}$$



Termino 3 $\mathbb{E}_{q(Z|x)} [z_j^2]$

Habíamos definido z_j con media μ_j y varianza σ_j^2 . Entonces:

$$\mathbb{E}[z_j^2] = \text{var}(z_j) + (\mathbb{E}[z_j])^2 = \sigma_j^2 + \mu_j^2$$

Por lo tanto:

$$\frac{1}{2} \sum_{j=1}^d \mathbb{E}_{q(Z|x)} [z_j^2] = \frac{1}{2} \sum_{j=1}^d (\sigma_j^2 + \mu_j^2)$$

VAE

$$\max \mathcal{L}_i(\phi, \beta) = \frac{1}{L} \sum_{l=1}^L \log p_{\theta}(x_i | z_l) - D_{\text{KL}} \left(q_{\phi}(z|x_i) || p_{\theta}(z) \right)$$

Decoder **Encoder** **Fixed**

La divergencia KL entre $q(z|x)$ y $p(z)$ es: $D_{KL}(p(z|x)||p(z)) = -\sum_{j=1}^d \log \sigma_j - \frac{1}{2} \sum_{j=1}^d 1 + \frac{1}{2} \sum_{j=1}^d (\sigma_j^2 + \mu_j^2)$



VAE

$$\max \mathcal{L}_i(\phi, \beta) = \frac{1}{L} \sum_{l=1}^L \log p_{\theta}(x_i | z_l) - D_{\text{KL}} \left(q_{\phi}(z|x_i) || p_{\theta}(z) \right)$$

Decoder **Encoder** **Fixed**

La divergencia KL entre $q(z|x)$ y $p(z)$ es:

$$D_{KL}(p(z|x)||p(z)) = - \sum_{j=1}^d \log \sigma_j - \frac{1}{2} \sum_{j=1}^d 1 + \frac{1}{2} \sum_{j=1}^d (\sigma_j^2 + \mu_j^2)$$

$$= \frac{1}{2} \sum_{j=1}^d [-2 \log \sigma_j - 1 + \sigma_j^2 + \mu_j^2] = \frac{1}{2} \sum_{j=1}^d [-\log \sigma_j^2 - 1 + \sigma_j^2 + \mu_j^2]$$



VAE

$$\max \mathcal{L}_i(\phi, \beta) = \frac{1}{L} \sum_{l=1}^L \log p_{\theta}(x_i | z_l) - D_{\text{KL}} \left(q_{\phi}(z|x_i) || p_{\theta}(z) \right)$$

Decoder **Encoder** **Fixed**

La divergencia KL entre $q(z|x)$ y $p(z)$ es:

$$D_{KL}(p(z|x)||p(z)) = - \sum_{j=1}^d \log \sigma_j - \frac{1}{2} \sum_{j=1}^d 1 + \frac{1}{2} \sum_{j=1}^d (\sigma_j^2 + \mu_j^2)$$

$$= \frac{1}{2} \sum_{j=1}^d [-2 \log \sigma_j - 1 + \sigma_j^2 + \mu_j^2] = \frac{1}{2} \sum_{j=1}^d [-\log \sigma_j^2 - 1 + \sigma_j^2 + \mu_j^2]$$

$$= \frac{1}{2} \sum_{j=1}^d [-\log \text{var} - 1 + \exp(\log \text{var}) + \mu_j^2]$$



VAE

$$\max \mathcal{L}_i(\phi, \beta) = \frac{1}{L} \sum_{l=1}^L \log p_{\theta}(x_i|z_l) - D_{\text{KL}} \left(q_{\phi}(z|x_i) || p_{\theta}(z) \right)$$

Decoder **Encoder** **Fixed**

La divergencia KL entre $q(z|x)$ y $p(z)$ es:

$$D_{KL}(p(z|x)||p(z)) = -\sum_{j=1}^d \log \sigma_j - \frac{1}{2} \sum_{j=1}^d 1 + \frac{1}{2} \sum_{j=1}^d (\sigma_j^2 + \mu_j^2)$$

$$= \frac{1}{2} \sum_{j=1}^d [-2 \log \sigma_j - 1 + \sigma_j^2 + \mu_j^2] = \frac{1}{2} \sum_{j=1}^d [-\log \sigma_j^2 - 1 + \sigma_j^2 + \mu_j^2]$$

$$= \frac{1}{2} \sum_{j=1}^d [-\log \text{var} - 1 + \exp(\log \text{var}) + \mu_j^2]$$

Preferimos usar **log var** porque:

- **Garantizar la positividad:** La varianza debe ser estrictamente positiva. Al parametrizar la varianza como $\sigma^2 = \exp(\log \text{var})$ nos aseguramos de que, sin importar el valor de **log var** (que puede ser cualquier número real), σ^2 siempre será positivo.
- **Estabilidad numérica y optimización:** Trabajar en el espacio logarítmico puede mejorar la convergencia durante el entrenamiento, ya que los cambios relativos en la varianza se representan de forma más suave.



VAE

$$\max \mathcal{L}_i(\phi, \beta) = \frac{1}{L} \sum_{l=1}^L \log p_{\theta}(x_i | z_l) - D_{\text{KL}} \left(q_{\phi}(z|x_i) || p_{\theta}(z) \right)$$

Decoder **Encoder** **Fixed**

La divergencia KL entre $q(z|x)$ y $p(z)$ es:

$$D_{KL}(p(z|x)||p(z)) = - \sum_{j=1}^d \log \sigma_j - \frac{1}{2} \sum_{j=1}^d 1 + \frac{1}{2} \sum_{j=1}^d (\sigma_j^2 + \mu_j^2)$$

$$= \frac{1}{2} \sum_{j=1}^d [-2 \log \sigma_j - 1 + \sigma_j^2 + \mu_j^2] = \frac{1}{2} \sum_{j=1}^d [-\log \sigma_j^2 - 1 + \sigma_j^2 + \mu_j^2]$$

$$= \frac{1}{2} \sum_{j=1}^d [-\log \text{var} - 1 + \exp(\log \text{var}) + \mu_j^2]$$

CODIGO:

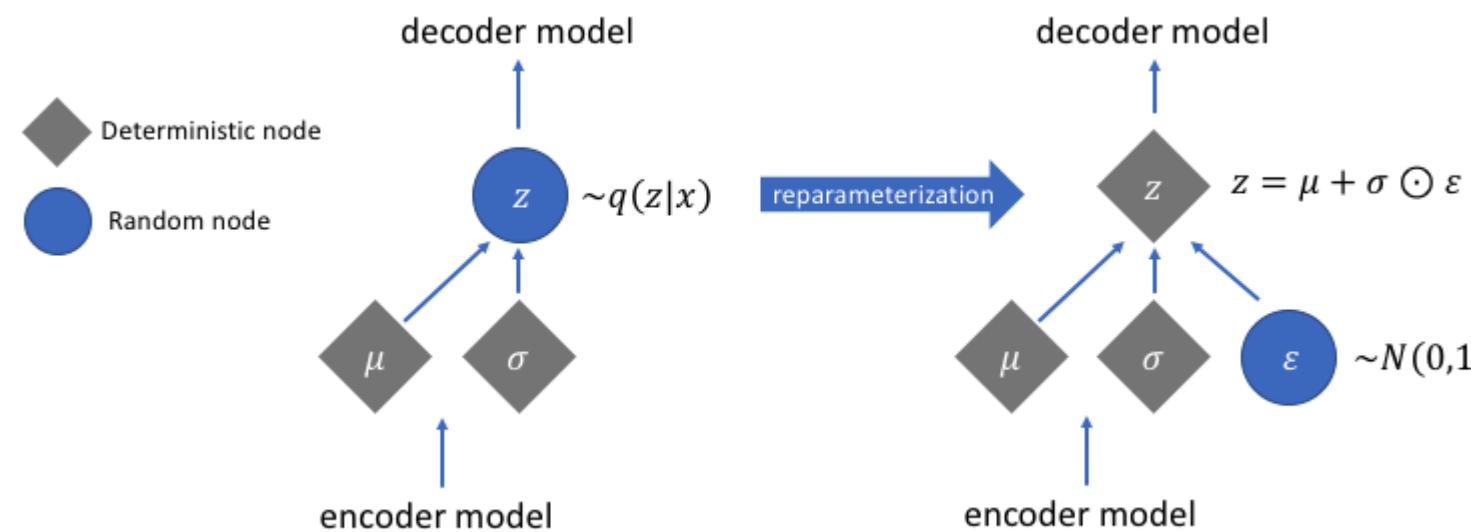
```
KLD = -0.5 * torch.sum(1 + logvar - mu.pow(2) - logvar.exp())
```



VAE

$$\max \mathcal{L}_i(\phi, \beta) = \frac{1}{L} \sum_{l=1}^L \log p_\theta(x_i|z_l) - D_{\text{KL}} \left(q_\phi(z|x_i) || p_\theta(z) \right)$$

Decoder **Encoder** **Fixed**



Se quiere muestrear z de la distribución $\mathcal{N}(\mu, \sigma^2)$. Reescribimos z como:

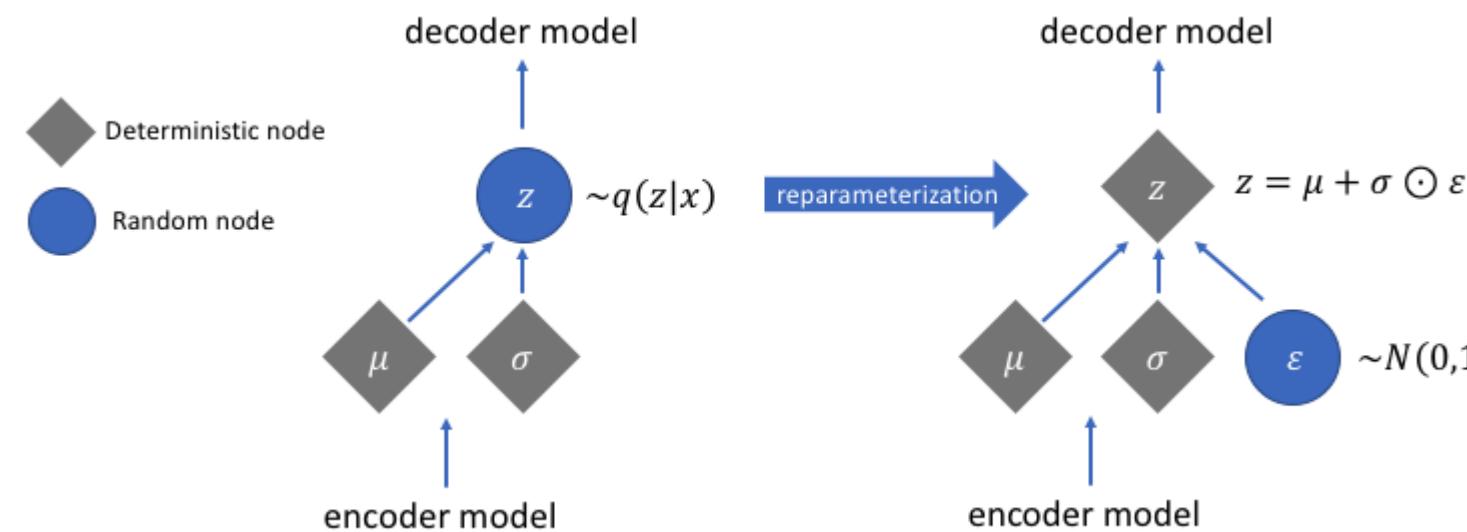
$$z = \mu + \sigma \cdot \epsilon$$
 donde $\epsilon \sim \mathcal{N}(0,1)$.



VAE

$$\max \mathcal{L}_i(\phi, \beta) = \frac{1}{L} \sum_{l=1}^L \log p_\theta(x_i|z_l) - D_{\text{KL}} \left(q_\phi(z|x_i) || p_\theta(z) \right)$$

Decoder **Encoder** **Fixed**



Se quiere muestrear z de la distribución $\mathcal{N}(\mu, \sigma^2)$. Reescribimos z como:

$$z = \mu + \sigma \cdot \epsilon$$

donde $\epsilon \sim \mathcal{N}(0,1)$.

Reescribimos la varianza en función de **log var**:

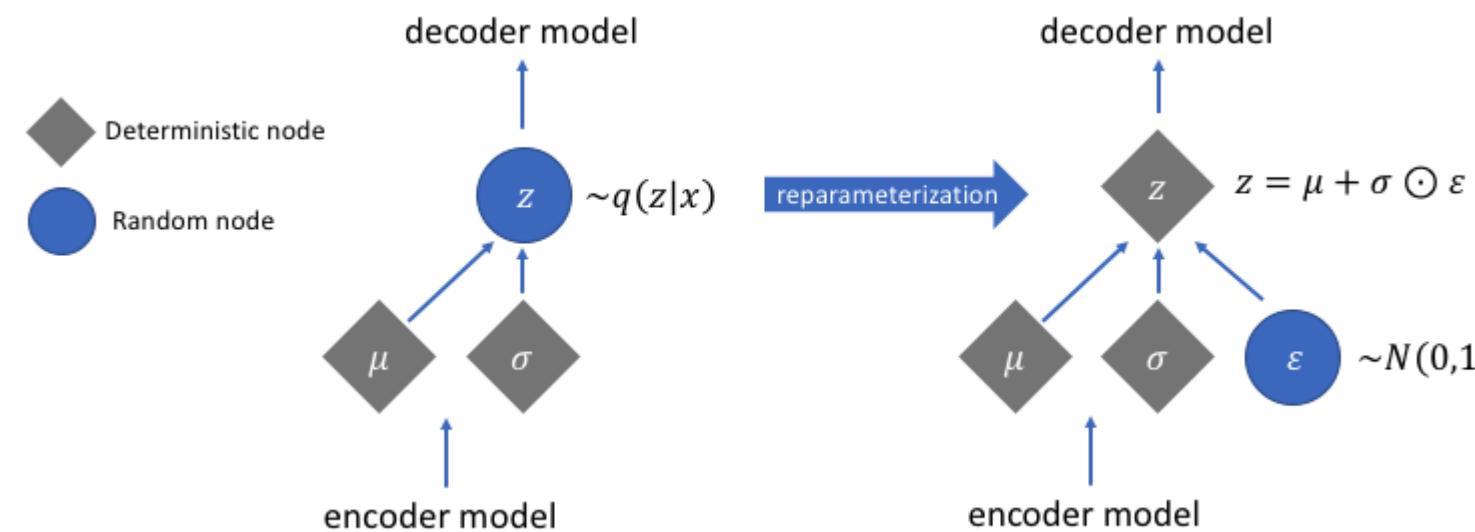
$$\sigma = \sqrt{\sigma^2} = \sqrt{\exp(\log \text{var})} = \exp\left(\frac{1}{2}\log \text{var}\right)$$



VAE

$$\max \mathcal{L}_i(\phi, \beta) = \frac{1}{L} \sum_{l=1}^L \log p_{\theta}(x_i | z_l) - D_{\text{KL}} \left(q_{\phi}(z|x_i) || p_{\theta}(z) \right)$$

Decoder **Encoder** **Fixed**



Se quiere muestrear z de la distribución $\mathcal{N}(\mu, \sigma^2)$. Reescribimos z como:

$$z = \mu + \sigma \cdot \epsilon$$

donde $\epsilon \sim \mathcal{N}(0,1)$.

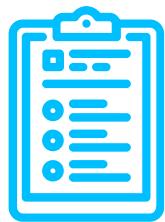
Reescribimos la varianza en función de **log var**:

$$\sigma = \sqrt{\sigma^2} = \sqrt{\exp(\log \text{var})} = \exp\left(\frac{1}{2}\log \text{var}\right)$$

```
def reparameterize(self, mu, logvar):
    std = torch.exp(0.5 * logvar)
    eps = torch.randn_like(std)
    return mu + eps * std
```



2.



β -VAE

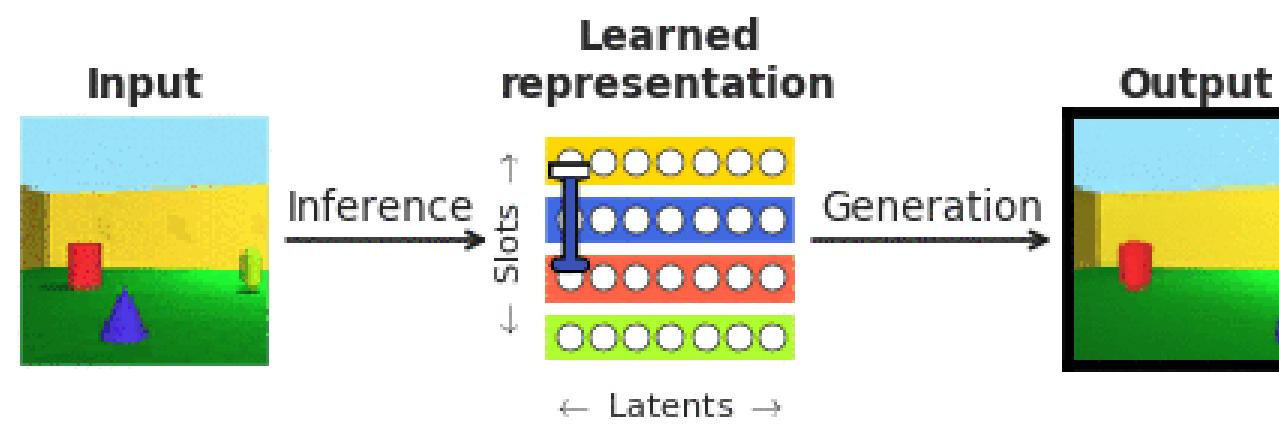
TRANSFORMATEC

> Reinventa el mundo <



β -VAE

Disentangled representation



La idea principal es aprender representaciones latentes donde cada variable capture de forma independiente uno de los factores generativos subyacentes a los datos, lo que puede facilitar tareas posteriores como clasificación, transfer learning o zero-shot inference.

Una representación bien desenredada debe tener variables que respondan de forma selectiva a cambios en factores concretos, por ejemplo, pose, iluminación o escala en imágenes.



β -VAE

Los autores proponen obtener representaciones latentes más disentangled (separadas e interpretables) haciendo: $D_{KL}(p(z|x)||p(z)) < C$



Irina Higgins et al. (2017) "β-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework".
International Conference on Learning Representations (ICLR).

β -VAE

Los autores proponen obtener representaciones latentes más disentangled (separadas e interpretables) haciendo: $D_{KL}(p(z|x)||p(z)) < C$

La divergencia KL actúa como un information bottleneck, forzando a la red a utilizar su limitada capacidad de codificación de la manera más eficiente posible.



Irina Higgins et al. (2017) "β-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework".
International Conference on Learning Representations (ICLR).

β -VAE

Los autores proponen obtener representaciones latentes más disentangled (separadas e interpretables) haciendo: $D_{KL}(p(z|x)||p(z)) < C$

La divergencia KL actúa como un information bottleneck, forzando a la red a utilizar su limitada capacidad de codificación de la manera más eficiente posible.

Formalmente:

Sea $q_\phi(z|x)$ la distribución a posteriori y $p(z)$ el prior elegido, normalmente una gaussiana isotrópica factorizable: $p(z) = \prod_{m=1}^M p(z_m)$

Al imponer la condición $D_{KL}(q_\phi(z|x)||p(z)) < C$ estamos limitando el flujo de información entre x y z .

Esta relación implica: $I(x; z) \leq E_{x \sim p_{data}(x)} [D_{KL}(q_\phi(z|x)||p(z))] < C$

Así, un canal del espacio latente sólo puede transmitir, en promedio, menos de C unidades de información (nats).

Si asumimos que la distribución a posteriori también se aproxima a una distribución que factorizable: $q_\phi(z|x) = \prod_{m=1}^M q_\phi(z_m|x)$

se tiene que $D_{KL}(q_\phi(z|x)||p(z)) = \sum_{m=1}^M D_{KL}(q_\phi(z_m|x)||p(z_m))$

Esto significa que las C unidades de información se reparte entre las M dimensiones. En presencia de una restricción tan estricta, la única forma de usar eficientemente la capacidad es asignar cada unidad latente a capturar información de un único factor generativo.



β -VAE

Bajo una restricción de capacidad, la red debe seleccionar qué información preservar para poder reconstruir x de forma adecuada. Si los factores generativos subyacentes son independientes, la asignación óptima en términos de eficiencia de información es que cada dimensión z_m codifique preferentemente uno de esos factores.

Formalmente, si se intenta codificar información redundante o enredada (entangled), se desperdicia capacidad, ya que se estarían utilizando más de C unidades para representar la misma variación. Así, la solución óptima se acerca a:

$$q_{\phi}(z|x) \approx \prod_{k=1}^K q_{\phi}(z_k|v_k)$$

donde cada z_k está alineado con un factor generativo independiente v_k . Esto se traduce en una representación disentangled (desenredada).



β -VAE

Para forzar que el modelo use una capacidad limitada en el espacio latente y, por ende, promueva representaciones disentangled, se plantea la siguiente optimización restringida:

Maximizar: $\max_{\theta, \phi} \mathbb{E}_{q_\phi(z|x)} [\log p_\theta(x|z)]$

Sujeto a: $D_{KL}(q_\phi(z|x) \| p(z)) \leq C$



β -VAE

Para forzar que el modelo use una capacidad limitada en el espacio latente y, por ende, promueva representaciones disentangled, se plantea la siguiente optimización restringida:

Maximizar: $\max_{\theta, \phi} \mathbb{E}_{q_\phi(z|x)} [\log p_\theta(x|z)]$

Sujeto a: $D_{KL}(q_\phi(z|x) \| p(z)) \leq C$

Este problema lo solucionamos con una generalización a los multiplicadores de Lagrange: las **condiciones de KKT**.



Karush-Kuhn-Tucker conditions (KKT)

$$\min \mathbf{f}(\mathbf{x}) \quad \text{sujeto a } g_i(x) \leq 0, h_j(x) = 0$$

$$\mathcal{L}(x, \mu, \lambda) = f(x) + \mu^\top g(x) + \lambda^\top h(x)$$

Stationarity

$$\frac{\partial}{\partial x} \mathcal{L}(x, \mu, \lambda) = \frac{\partial}{\partial x} f(x) + \sum_i \mu_i \frac{\partial}{\partial x} g(x) + \sum_i \lambda_i \frac{\partial}{\partial x} h(x)$$

Primal feasibility

$$g_i(x) \leq 0 \quad h_i(x) = 0$$

Dual feasibility

$$\lambda_i g_i(x) = 0$$

Complementary slackness

$$\lambda_i \geq 0$$



William Karush (1939) "Minima of functions of several variables with inequalities as side constraints".
M. Sc. Dissertation. Dept. of Mathematics, Univ. of Chicago.

Harold W. Kuhn and Albert W. Tucker (1951) "Nonlinear programming".
Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability (pp. 481–492). University of California Press.

β -VAE

Maximizar: $\max_{\theta, \phi} \mathbb{E}_{q_\phi(z|x)} [\log p_\theta(x|z)]$

Sujeto a: $D_{KL}(q_\phi(z|x) \| p_\theta(z)) \leq C$

Definimos la función Lagrangiana: $\mathcal{L}(\theta, \phi, \beta) = \mathbb{E}_{z \sim q_\phi(z|x)} \log p_\theta(x|z) - \beta (D_{KL}(q_\phi(z|x) \| p_\theta(z)) - C)$

sujeto a: $g(\theta, \phi) = D_{KL}(q_\phi(z|x) \| p_\theta(z)) - C \leq 0$



β -VAE

Maximizar: $\max_{\theta, \phi} \mathbb{E}_{q_\phi(z|x)} [\log p_\theta(x|z)]$

Sujeto a: $D_{KL}(q_\phi(z|x) \| p(z)) \leq C$

Definimos la función Lagrangiana: $\mathcal{L}(\theta, \phi, \beta) = \mathbb{E}_{z \sim q_\phi(z|x)} \log p_\theta(x|z) - \beta (D_{KL}(q_\phi(z|x) \| p_\theta(z)) - C)$

sujeto a: $g(\theta, \phi) = D_{KL}(q_\phi(z|x) \| p_\theta(z)) - C \leq 0$

Stationarity

$$\nabla_{\theta, \phi} \mathcal{L}(\theta, \phi, \beta) = 0$$

Primal feasibility

$$g(\theta, \phi) = D_{KL}(q_\phi(z|x) \| p(z)) - C \leq 0$$

Dual feasibility

$$\beta \geq 0$$

Complementary slackness $\beta \cdot g(\theta, \phi) = \beta(D_{KL}(q_\phi(z|x) \| p(z)) - C) = 0$



β -VAE

Maximizar: $\max_{\theta, \phi} \mathbb{E}_{q_\phi(z|x)} [\log p_\theta(x|z)]$

Sujeto a: $D_{KL}(q_\phi(z|x) \| p(z)) \leq C$

Definimos la función Lagrangiana: $\mathcal{L}(\theta, \phi, \beta) = \mathbb{E}_{z \sim q_\phi(z|x)} \log p_\theta(x|z) - \beta (D_{KL}(q_\phi(z|x) \| p_\theta(z)) - C)$

sujeto a: $g(\theta, \phi) = D_{KL}(q_\phi(z|x) \| p_\theta(z)) - C \leq 0$

Stationarity

$$\nabla_{\theta, \phi} \mathcal{L}(\theta, \phi, \beta) = 0$$

Primal feasibility

$$g(\theta, \phi) = D_{KL}(q_\phi(z|x) \| p(z)) - C \leq 0$$

Dual feasibility

$$\beta \geq 0$$

Complementary slackness $\beta \cdot g(\theta, \phi) = \beta(D_{KL}(q_\phi(z|x) \| p(z)) - C) = 0$

- Si la restricción es activa (es decir, $D_{KL} = C$), entonces se debe tener $\beta > 0$.
- Si la restricción es inactiva (es decir, $D_{KL} < C$), entonces $\beta = 0$.



β -VAE

Maximizar: $\max_{\theta, \phi} \mathbb{E}_{q_\phi(z|x)} [\log p_\theta(x|z)]$

Sujeto a: $D_{KL}(q_\phi(z|x) \| p(z)) \leq C$

Definimos la función Lagrangiana: $\mathcal{L}(\theta, \phi, \beta) = \mathbb{E}_{z \sim q_\phi(z|x)} \log p_\theta(x|z) - \beta (D_{KL}(q_\phi(z|x) \| p_\theta(z)) - C)$
 $= \mathbb{E}_{z \sim q_\phi(z|x)} \log p_\theta(x|z) - \beta D_{KL}(q_\phi(z|x) \| p_\theta(z)) - C\beta$



β -VAE

Maximizar: $\max_{\theta, \phi} \mathbb{E}_{q_\phi(z|x)} [\log p_\theta(x|z)]$

Sujeto a: $D_{KL}(q_\phi(z|x) \| p(z)) \leq C$

$$\begin{aligned}\text{Definimos la función Lagrangiana: } \mathcal{L}(\theta, \phi, \beta) &= \mathbb{E}_{z \sim q_\phi(z|x)} \log p_\theta(x|z) - \beta \left(D_{KL}(q_\phi(z|x) \| p_\theta(z)) - C \right) \\ &= \mathbb{E}_{z \sim q_\phi(z|x)} \log p_\theta(x|z) - \beta D_{KL}(q_\phi(z|x) \| p_\theta(z)) - C\beta\end{aligned}$$

Como nosotros vamos a optimizar θ, ϕ ;
entonces C, β son constantes



β -VAE

Maximizar: $\max_{\theta, \phi} \mathbb{E}_{q_\phi(z|x)} [\log p_\theta(x|z)]$

Sujeto a: $D_{KL}(q_\phi(z|x) || p_\theta(z)) \leq C$

Definimos la función Lagrangiana: $\mathcal{L}(\theta, \phi, \beta) = \mathbb{E}_{z \sim q_\phi(z|x)} \log p_\theta(x|z) - \beta (D_{KL}(q_\phi(z|x) || p_\theta(z)) - C)$
 $= \mathbb{E}_{z \sim q_\phi(z|x)} \log p_\theta(x|z) - \beta D_{KL}(q_\phi(z|x) || p_\theta(z)) - C\beta$

Entonces, la función a optimizar es: $\max_{\theta, \phi} \mathbb{E}_{z \sim q_\phi(z|x)} \log p_\theta(x|z) - \beta D_{KL}(q_\phi(z|x) || p_\theta(z))$



β -VAE

Maximizar: $\max_{\theta, \phi} \mathbb{E}_{q_\phi(z|x)} [\log p_\theta(x|z)]$

Sujeto a: $D_{KL}(q_\phi(z|x) \| p(z)) \leq C$

Definimos la función Lagrangiana: $\mathcal{L}(\theta, \phi, \beta) = \mathbb{E}_{z \sim q_\phi(z|x)} \log p_\theta(x|z) - \beta (D_{KL}(q_\phi(z|x) \| p_\theta(z)) - C)$
 $= \mathbb{E}_{z \sim q_\phi(z|x)} \log p_\theta(x|z) - \beta D_{KL}(q_\phi(z|x) \| p_\theta(z)) - C\beta$

Entonces, la función a optimizar es: $\min_{\theta, \phi} -\mathbb{E}_{z \sim q_\phi(z|x)} \log p_\theta(x|z) + \beta D_{KL}(q_\phi(z|x) \| p_\theta(z))$



β -VAE

Maximizar: $\max_{\theta, \phi} \mathbb{E}_{q_\phi(z|x)} [\log p_\theta(x|z)]$

Sujeto a: $D_{KL}(q_\phi(z|x) \| p(z)) \leq C$

Definimos la función Lagrangiana: $\mathcal{L}(\theta, \phi, \beta) = \mathbb{E}_{z \sim q_\phi(z|x)} \log p_\theta(x|z) - \beta (D_{KL}(q_\phi(z|x) \| p_\theta(z)) - C)$
 $= \mathbb{E}_{z \sim q_\phi(z|x)} \log p_\theta(x|z) - \beta D_{KL}(q_\phi(z|x) \| p_\theta(z)) - C\beta$

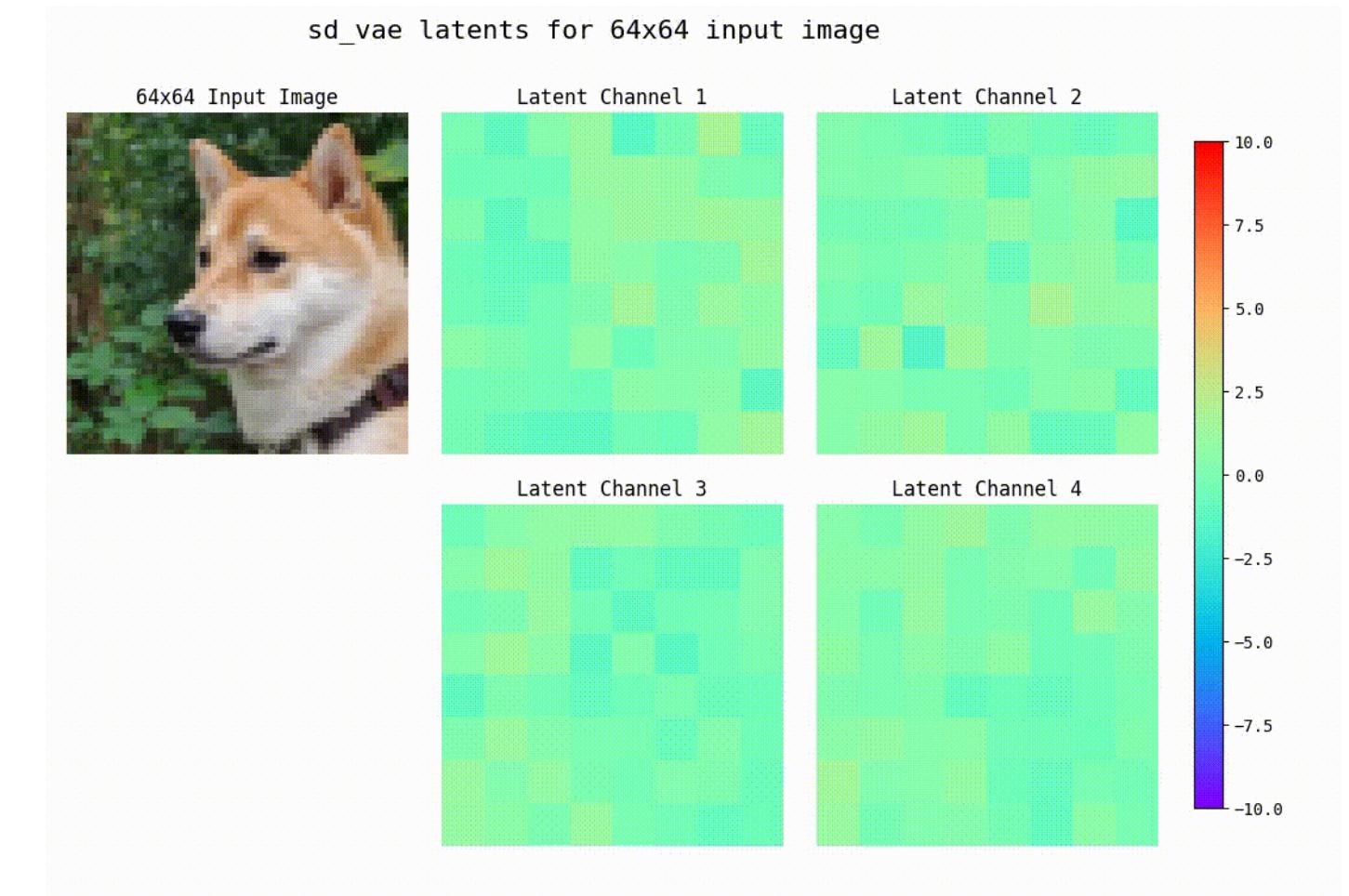
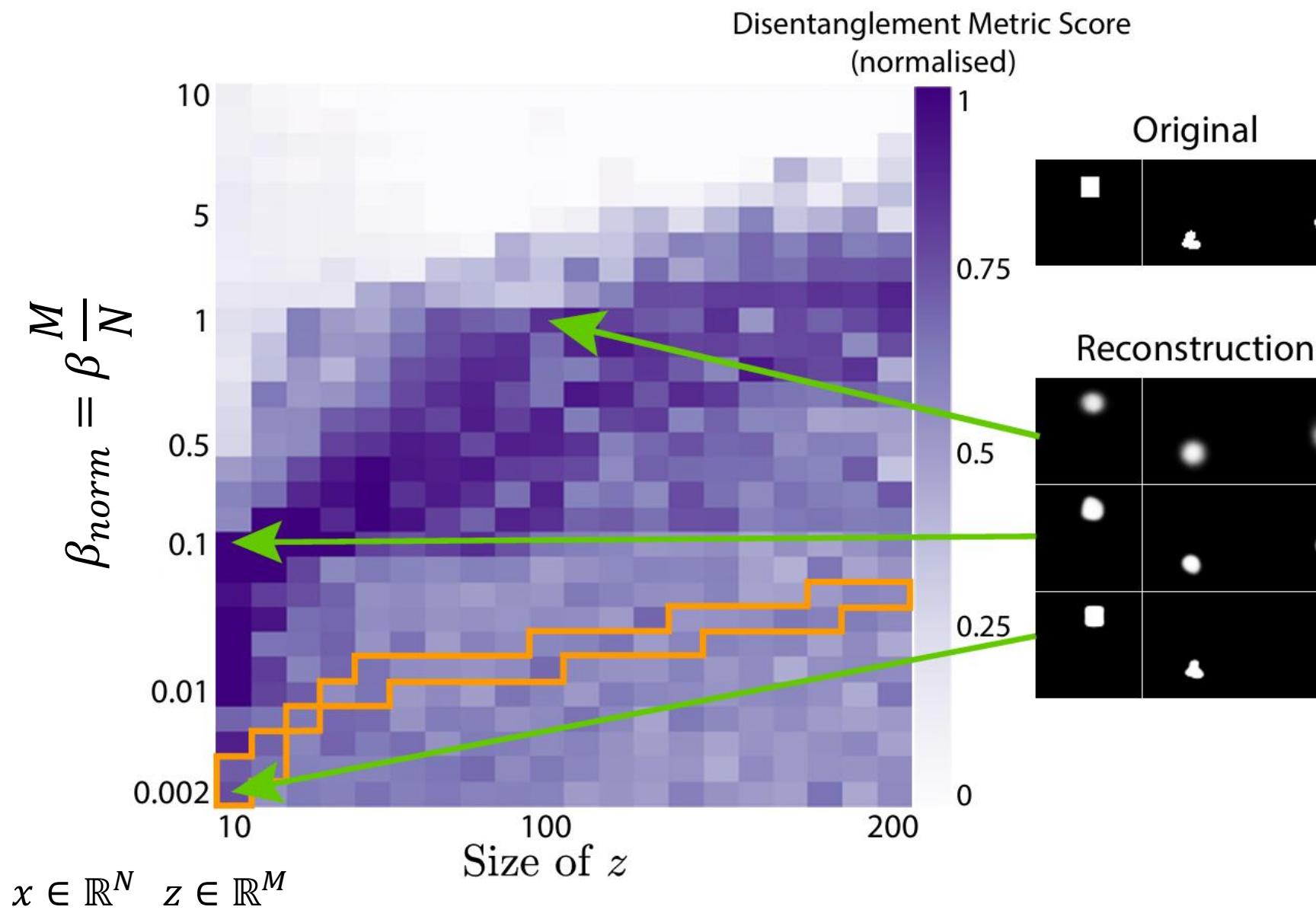
Entonces, la función a optimizar es: $\min_{\theta, \phi} -\mathbb{E}_{z \sim q_\phi(z|x)} \log p_\theta(x|z) + \beta D_{KL}(q_\phi(z|x) \| p_\theta(z))$

En el VAE clásico, el equilibrio entre la reconstrucción y la regularización se da para $\beta = 1$. Al aumentar β , se refuerza la penalización sobre el término de divergencia KL, lo que obliga a la distribución aproximada $q_\phi(z|x)$ a acercarse aún más a la distribución a priori $p(z)$. Con $\beta > 1$:

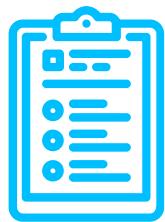
- **Reducción de capacidad latente:** Al imponer una penalización mayor sobre la divergencia KL, se restringe la cantidad de información que puede fluir en el espacio latente.
- **Disentanglement:** La limitación de la capacidad favorece que cada dimensión del latente capture factores de variación específicos y, en muchos casos, se observe que estos factores se vuelven más independientes entre sí.
- **Trade-off:** Un β demasiado alto puede degradar la calidad de la reconstrucción, ya que el modelo se centra excesivamente en cumplir la restricción de la divergencia KL.



β -VAE



3.



vQ-VAE

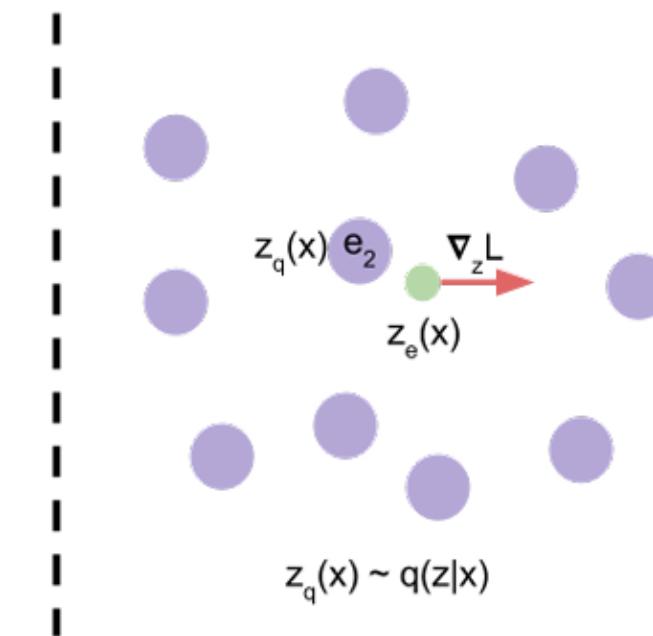
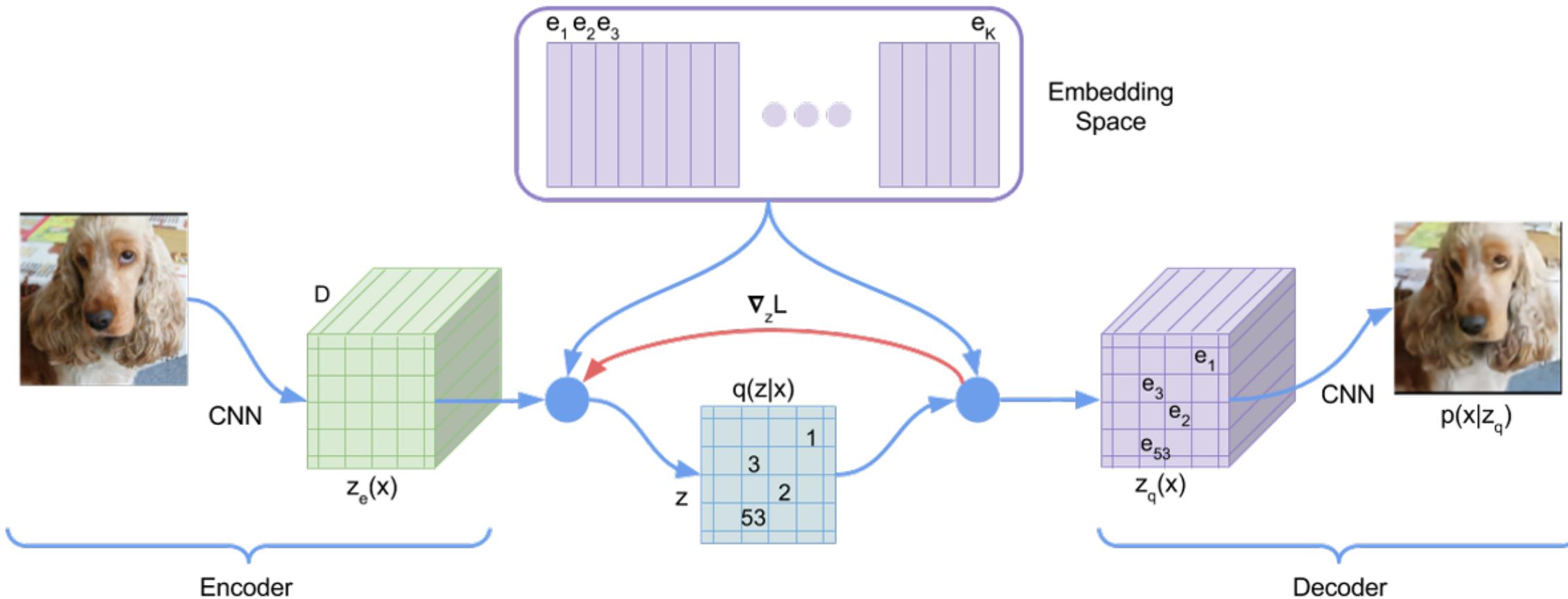
TRANSFORMATEC

> Reinventa el mundo <



vQ-VAE

(Vector Quantised-Variational AutoEncoder)



vQ-VAE

(Vector Quantised-Variational AutoEncoder)

$$\mathcal{L} = \underbrace{\|x - \hat{x}\|^2}_{\text{Reconstruction Loss}} + \underbrace{\|\text{detach}[z_{e(x)}] - e\|^2}_{\text{Codebook Loss}} + \beta \underbrace{\|z_{e(x)} - \text{detach}[e]\|^2}_{\text{Commitment Loss}}$$

donde:

- x es la entrada original.
- \hat{x} es la reconstrucción generada por el decodificador.
- $z_{e(x)}$ es la salida del encoder.
- e es el embedding del codebook más cercano.
- $\text{detach}[\cdot]$ representa a stop-gradient.
- β es un hiperparámetro que modula la importancia de Commitment Loss.



vQ-VAE

(Vector Quantised-Variational AutoEncoder)

$$\mathcal{L} = \underbrace{\|x - \hat{x}\|^2}_{\text{Reconstruction Loss}} + \underbrace{\|\text{detach}[z_{e(x)}] - e\|^2}_{\text{Codebook Loss}} + \beta \underbrace{\|z_{e(x)} - \text{detach}[e]\|^2}_{\text{Commitment Loss}}$$

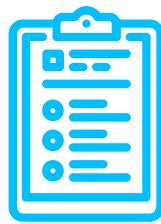
Reconstruction Loss Mide la capacidad del decodificador de reconstruir la entrada original a partir de la representación latente cuantizada.

Codebook Loss Actualiza los vectores del codebook (el conjunto de embeddings discretos) para que se acerquen a las salidas del encoder.

Commitment Loss Se quiere evitar que el encoder cambie arbitrariamente sus salidas, haciendo que sean consistentes con el embedding seleccionado.
Al penalizar la diferencia entre la salida del encoder y el embedding asignado, se compromete al encoder a trabajar en conjunto con el codebook, estabilizando así el proceso de aprendizaje.



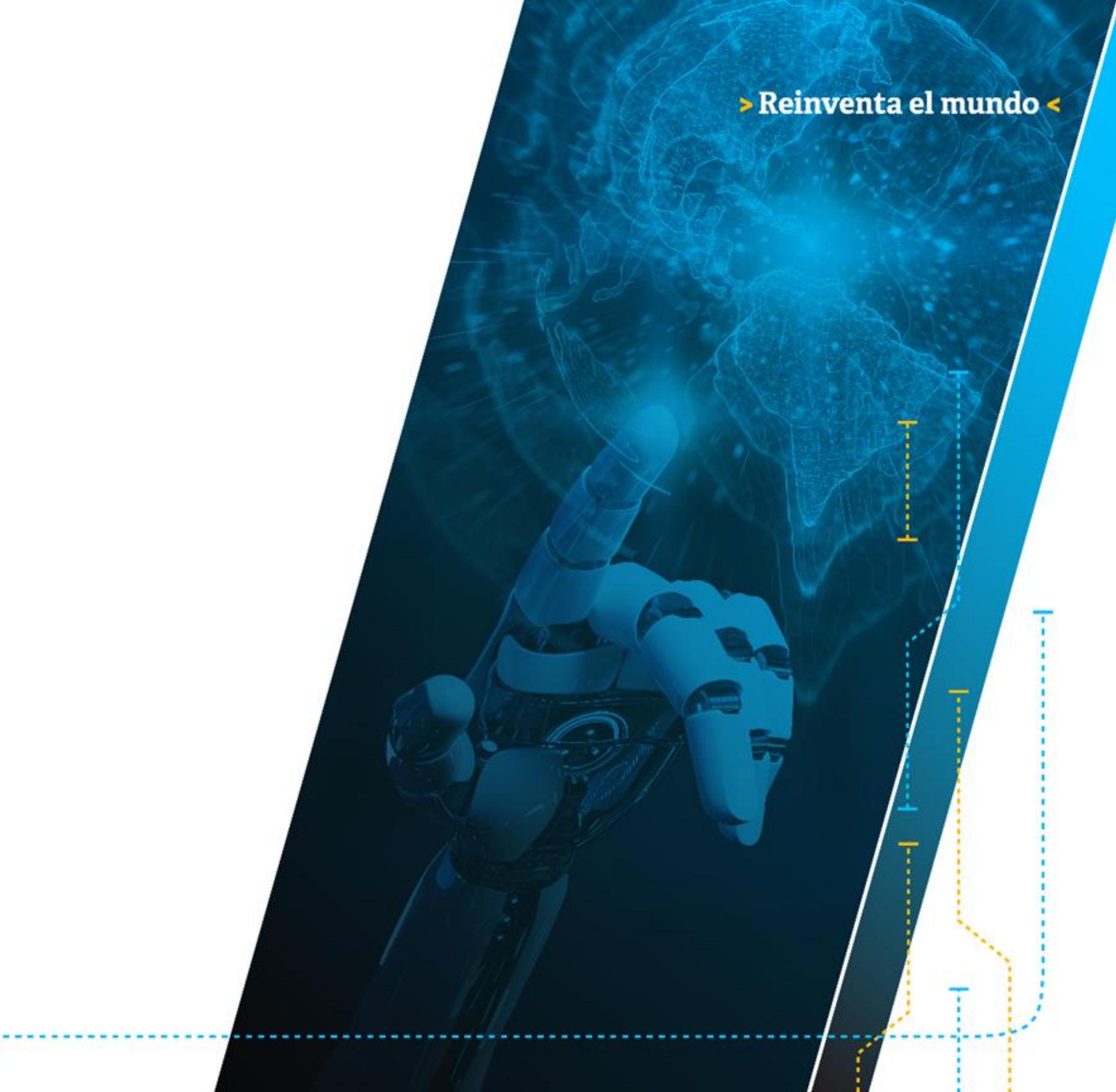
3.



World Models

TRANSFORMATEC

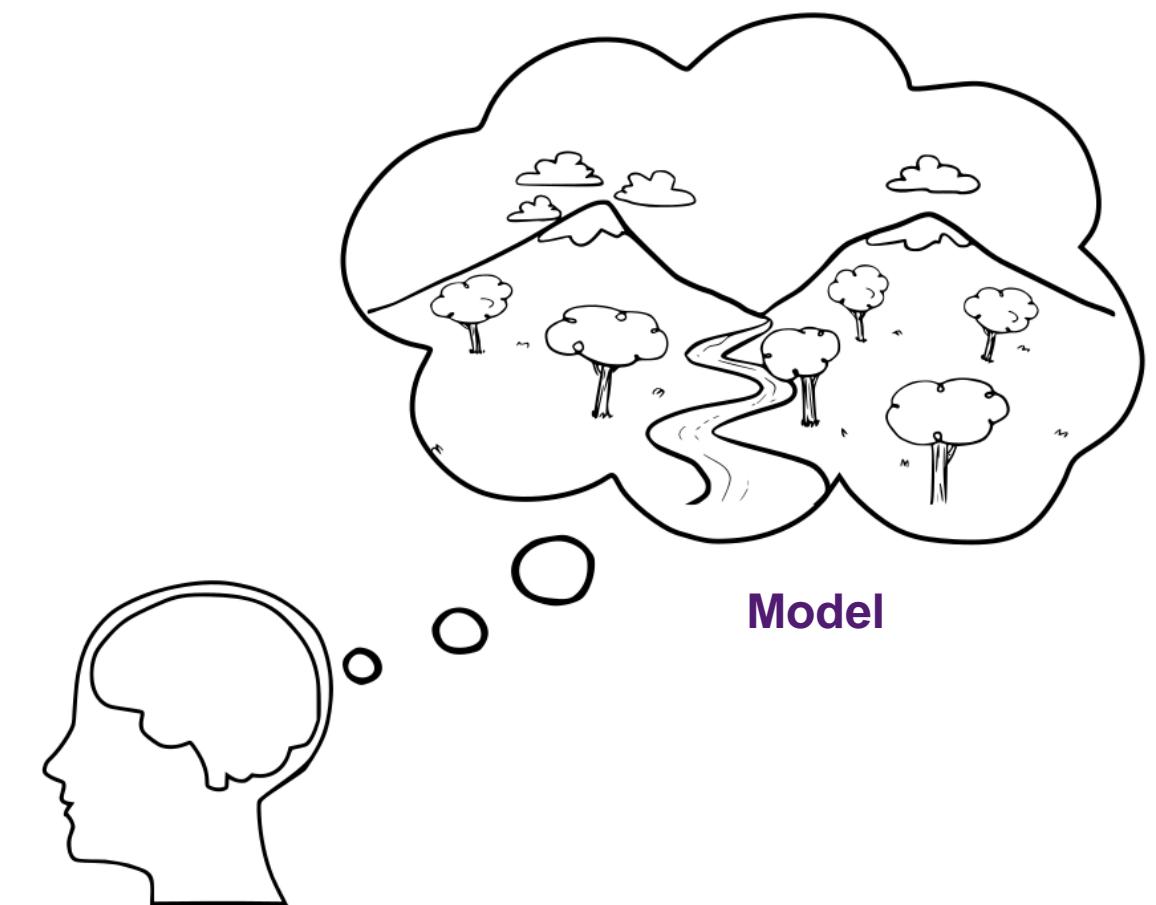
> Reinventa el mundo <



World Models



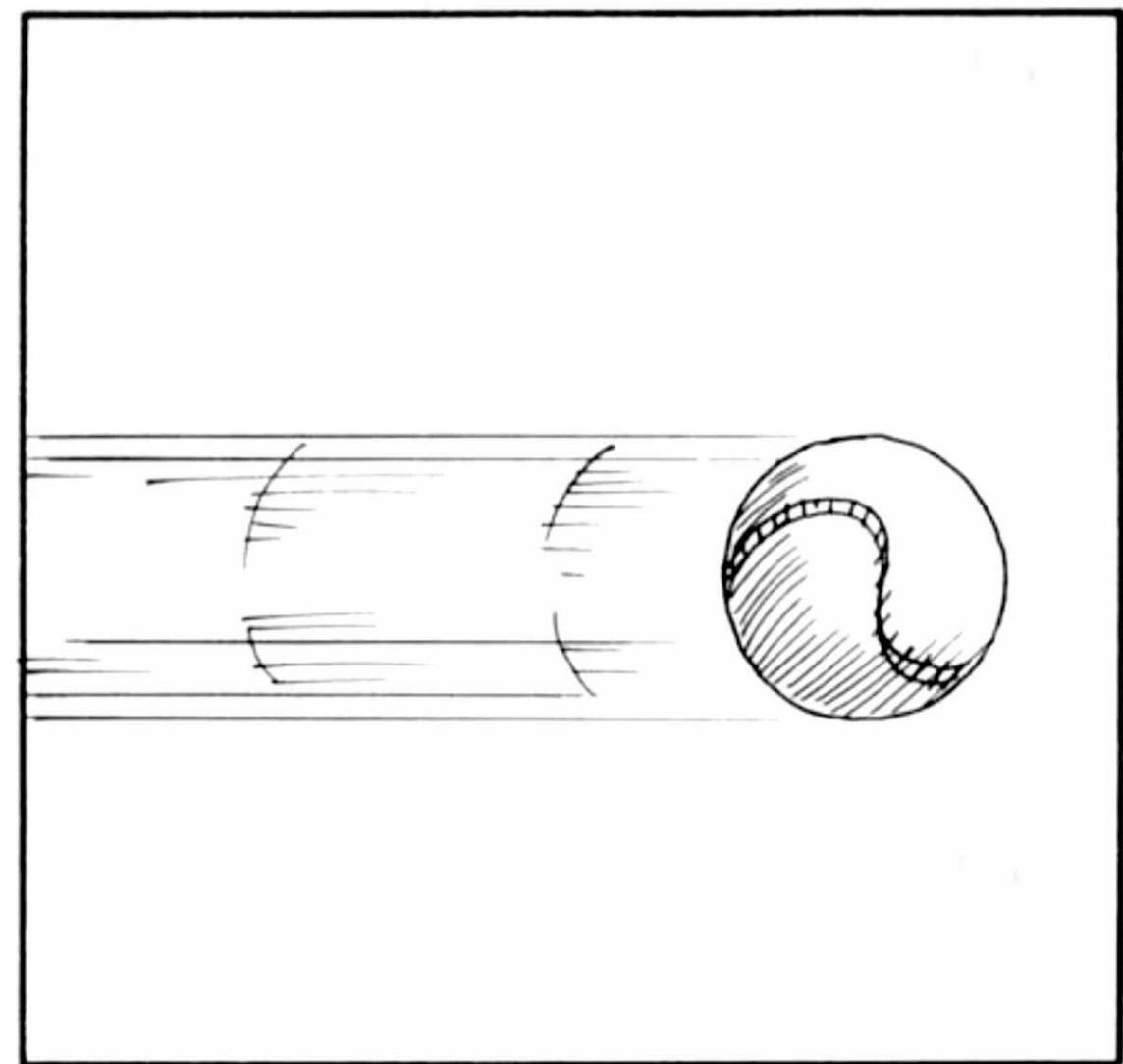
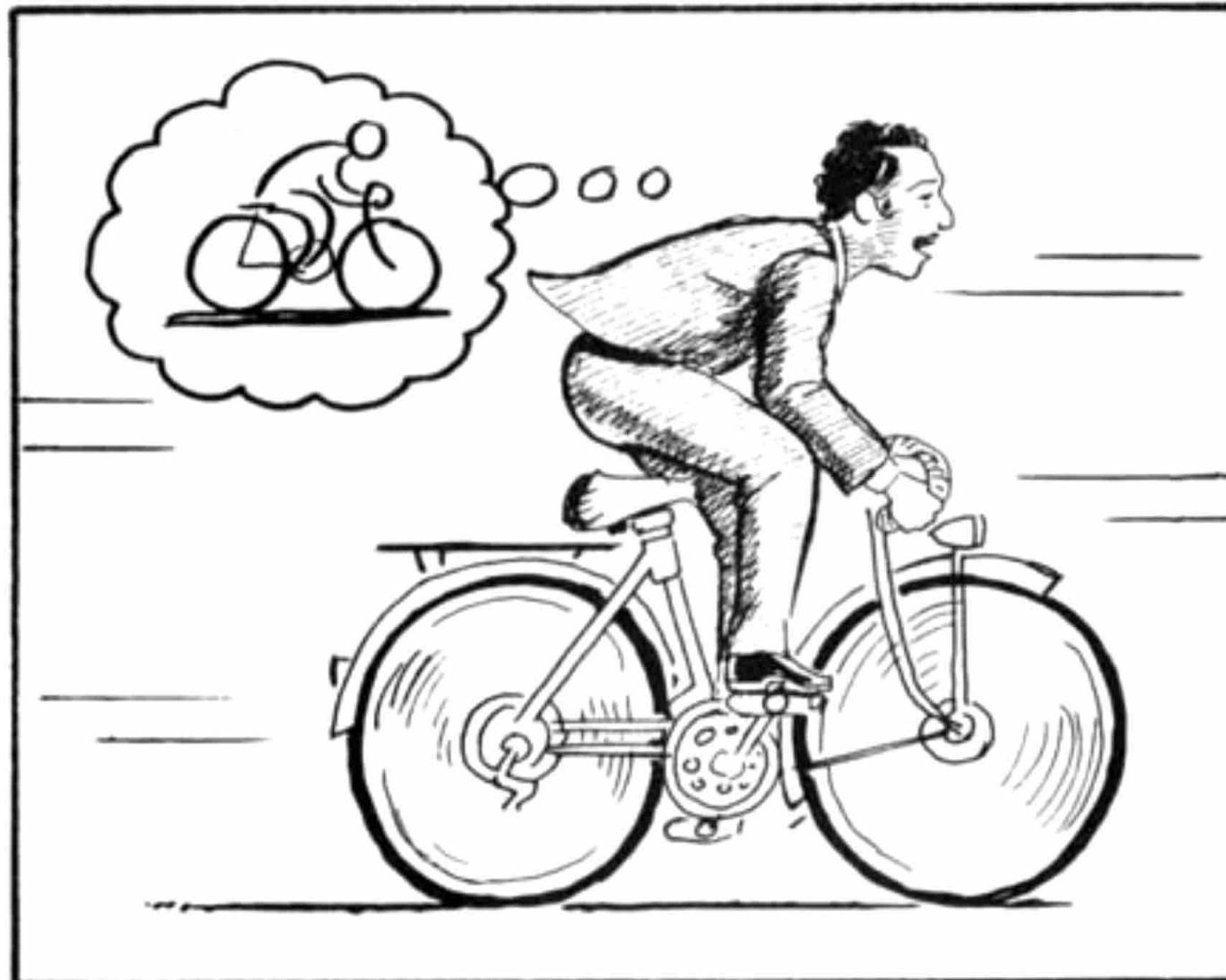
Complex world



TRANSFORMATEC

David Ha and Jürgen Schmidhuber (2018) "World Models".
arXiv preprint arXiv:1803.10122.

World Models



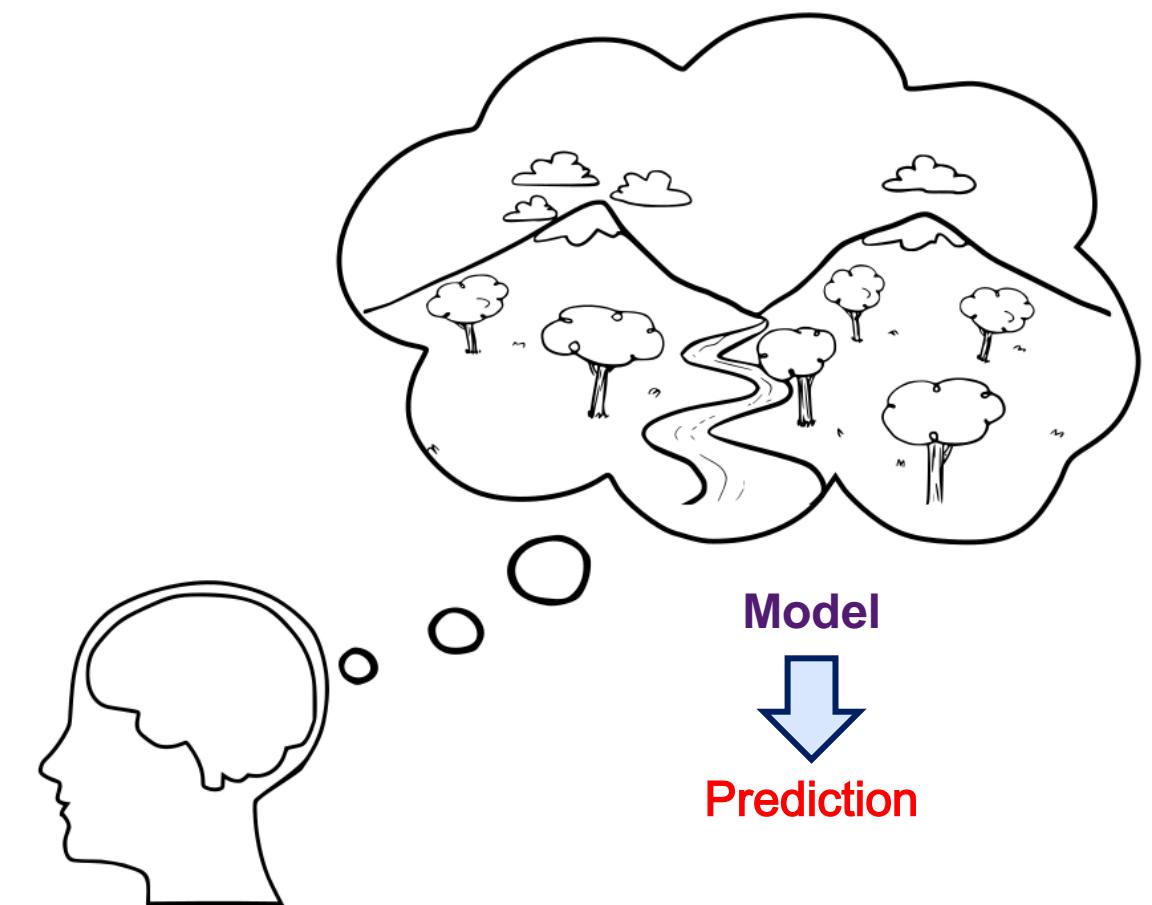
TRANSFORMATEC

David Ha and Jürgen Schmidhuber (2018) "World Models".
arXiv preprint arXiv:1803.10122.

World Models



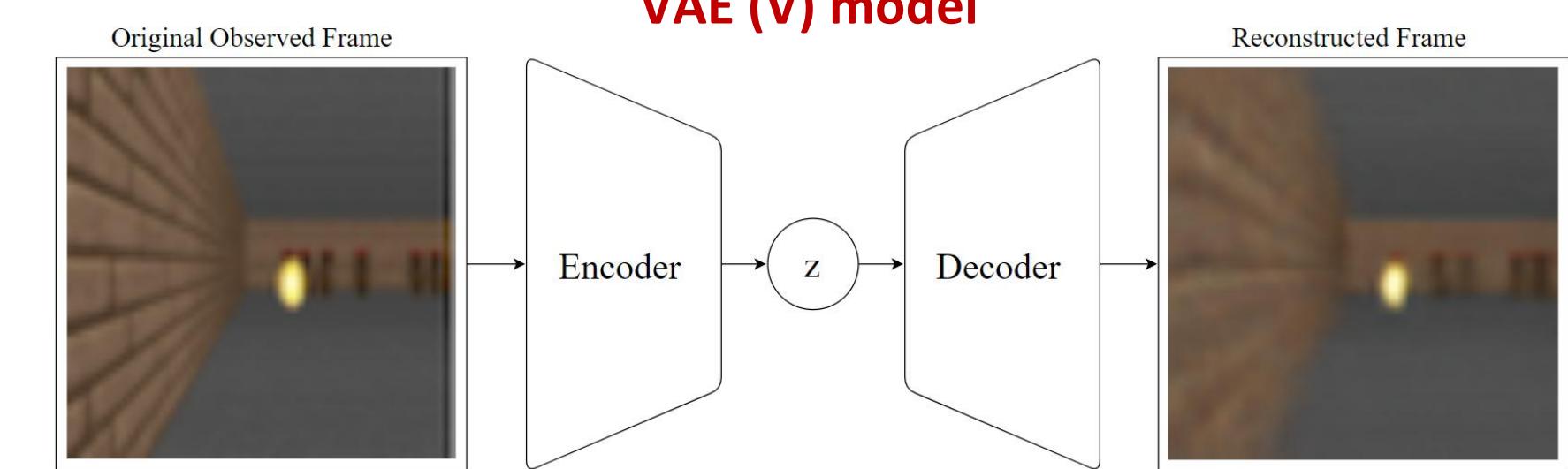
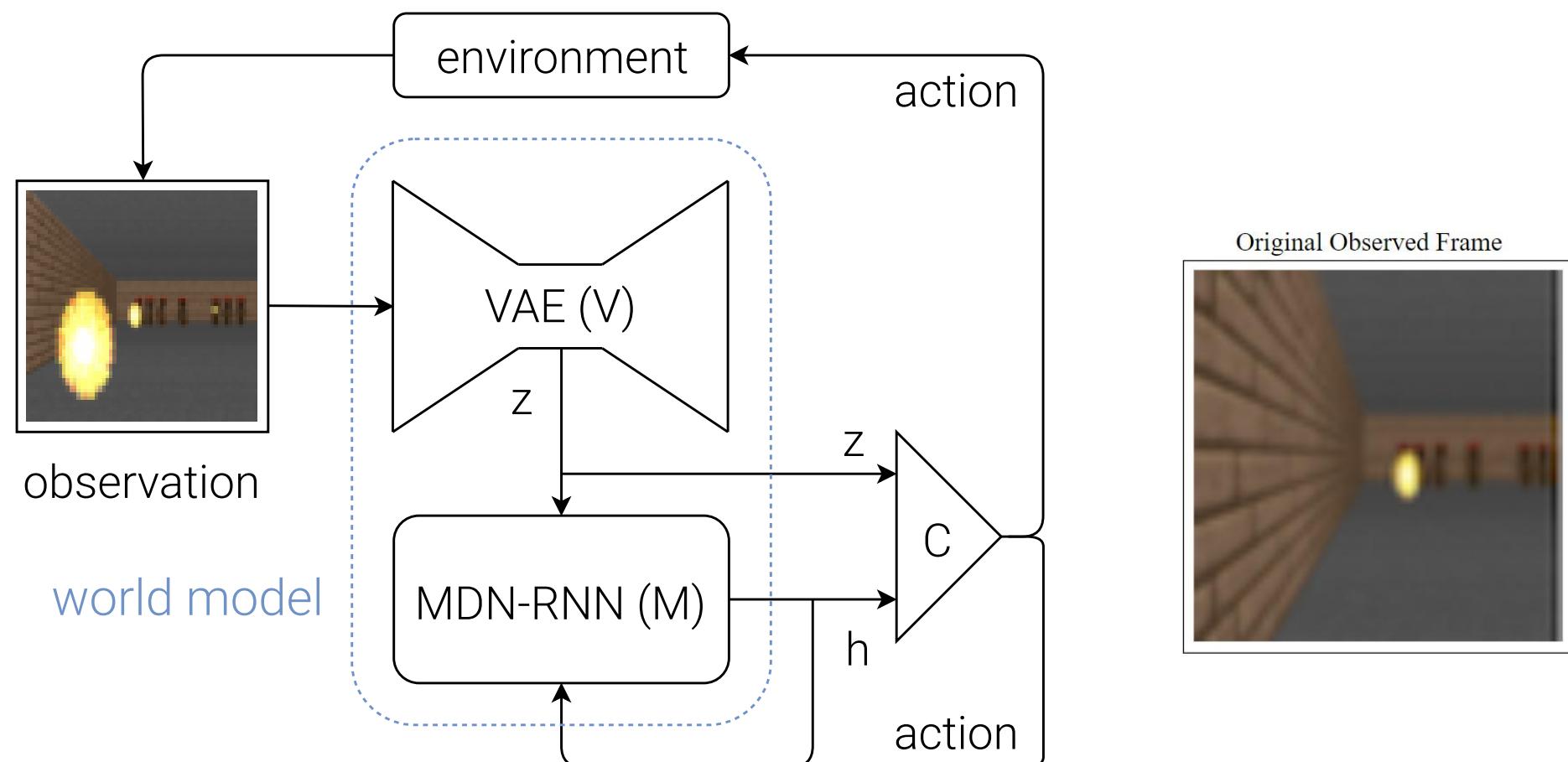
Complex world



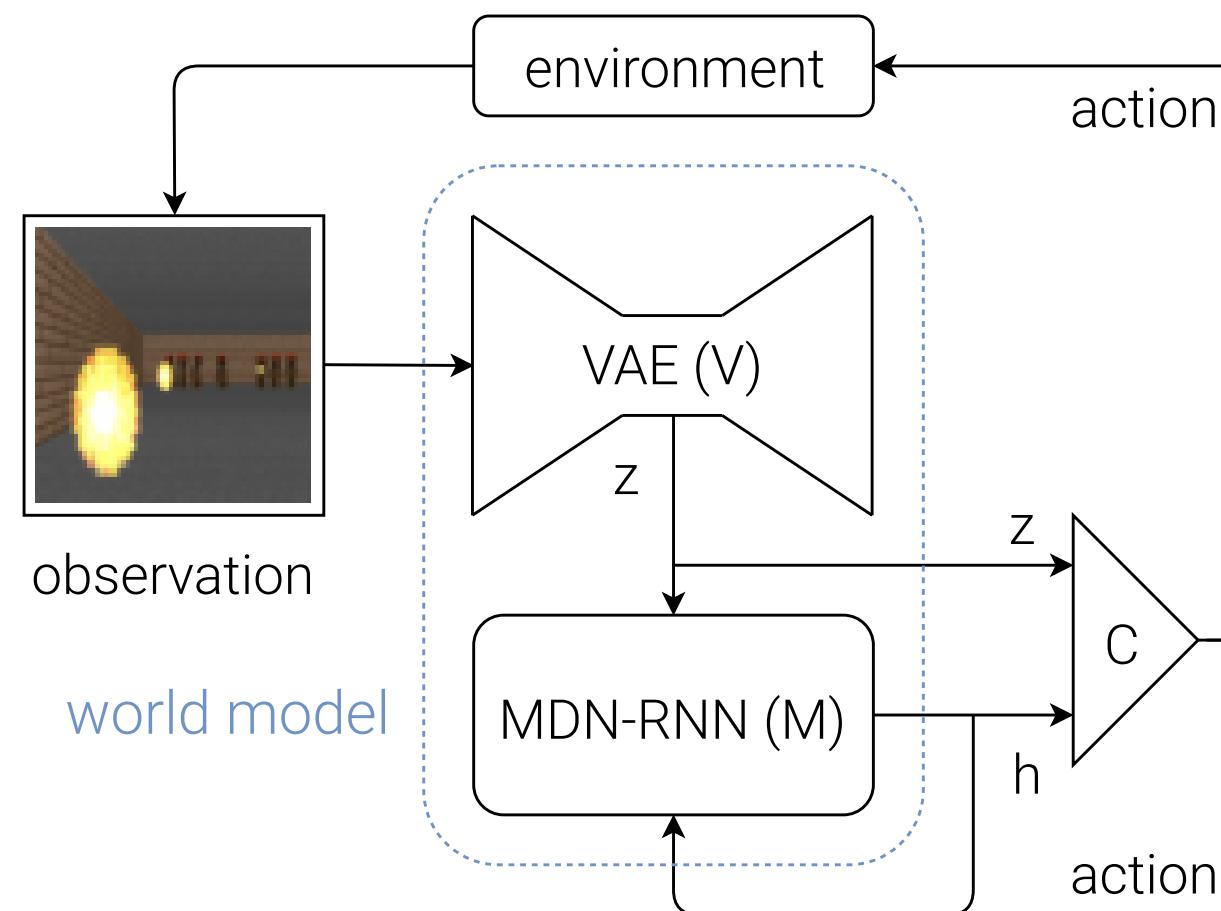
TRANSFORMATEC

David Ha and Jürgen Schmidhuber (2018) "World Models".
arXiv preprint arXiv:1803.10122.

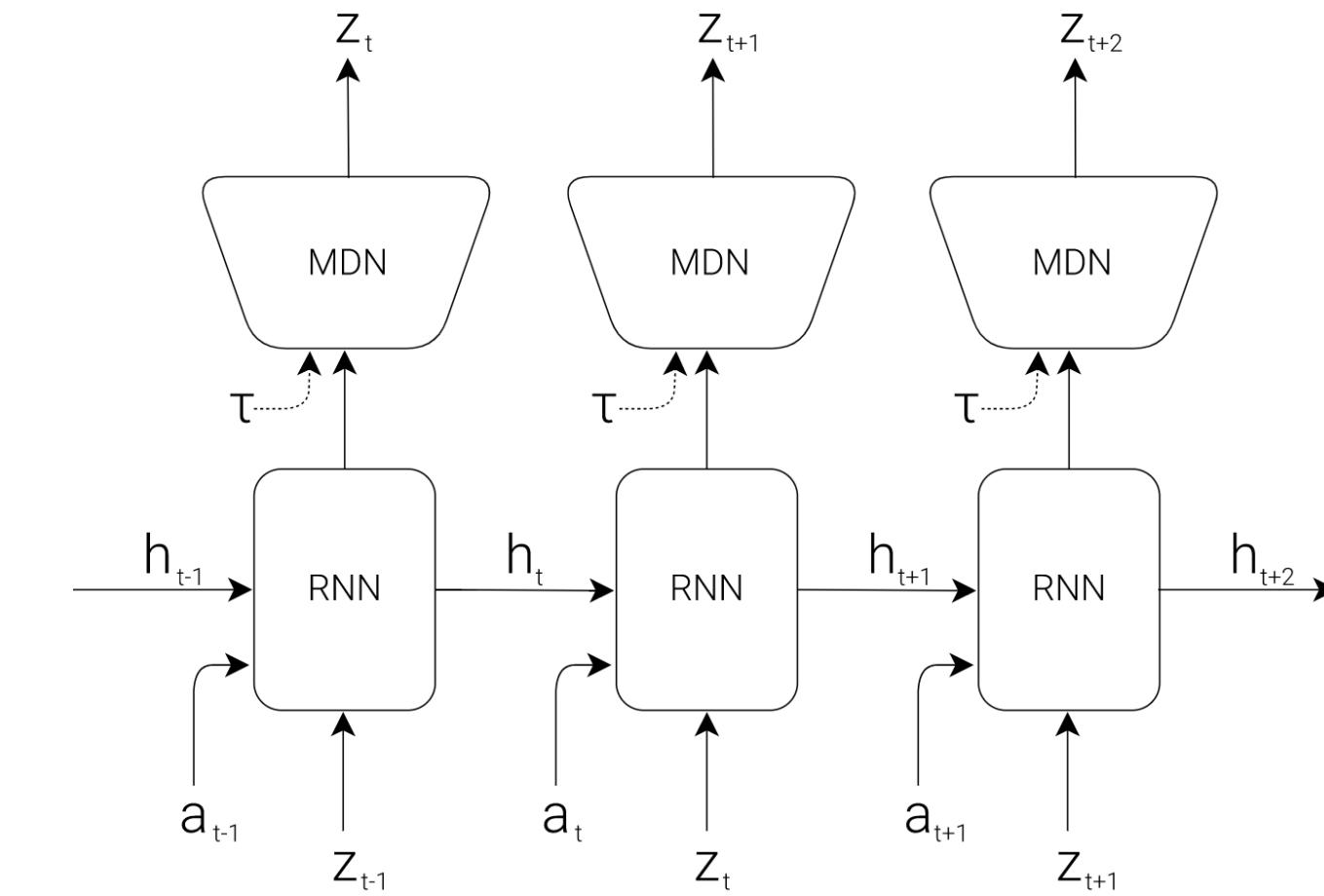
Agent Model



Agent Model



MDN-RNN (M) model



TRANSFORMATEC

David Ha and Jürgen Schmidhuber (2018) "World Models".
arXiv preprint arXiv:1803.10122.

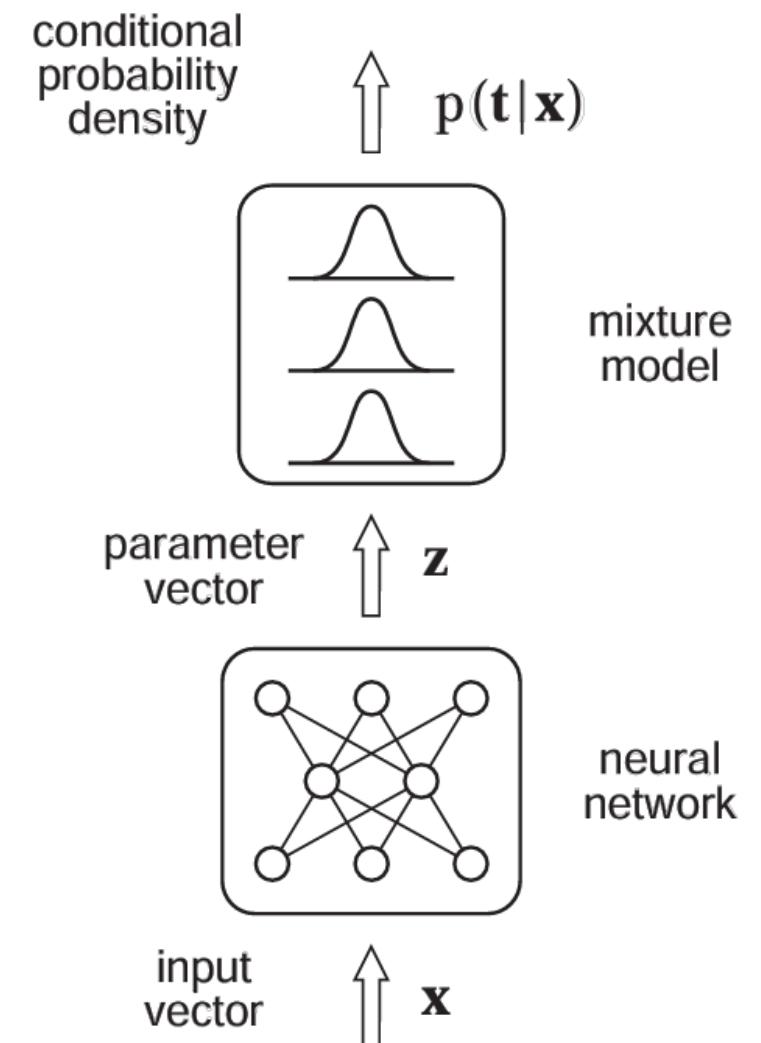
Agent Model

Mixture Density Network

sketch-rnn mosquito predictor.



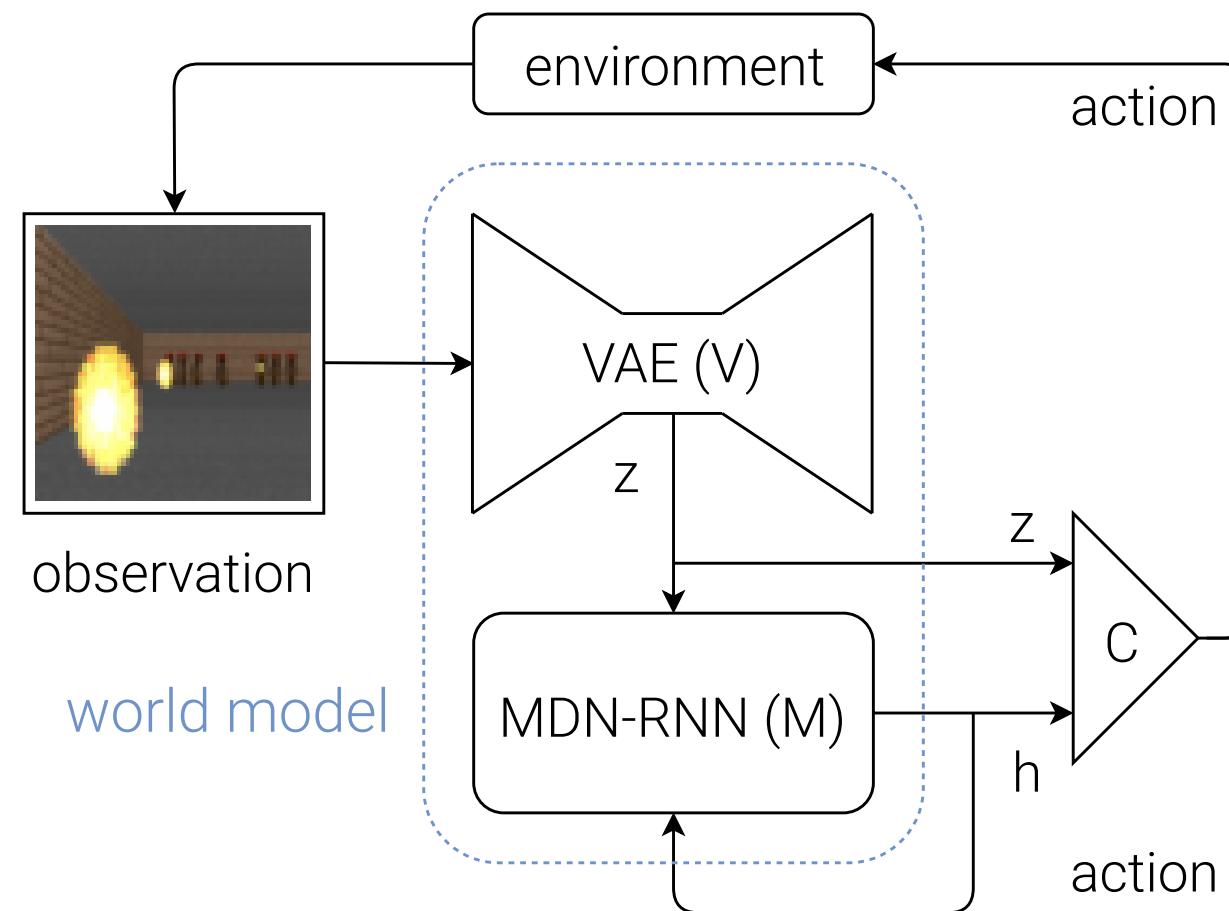
clear drawing mosquito random predict



TRANSFORMATEC

David Ha and Jürgen Schmidhuber (2018) "World Models".
arXiv preprint arXiv:1803.10122.

Agent Model

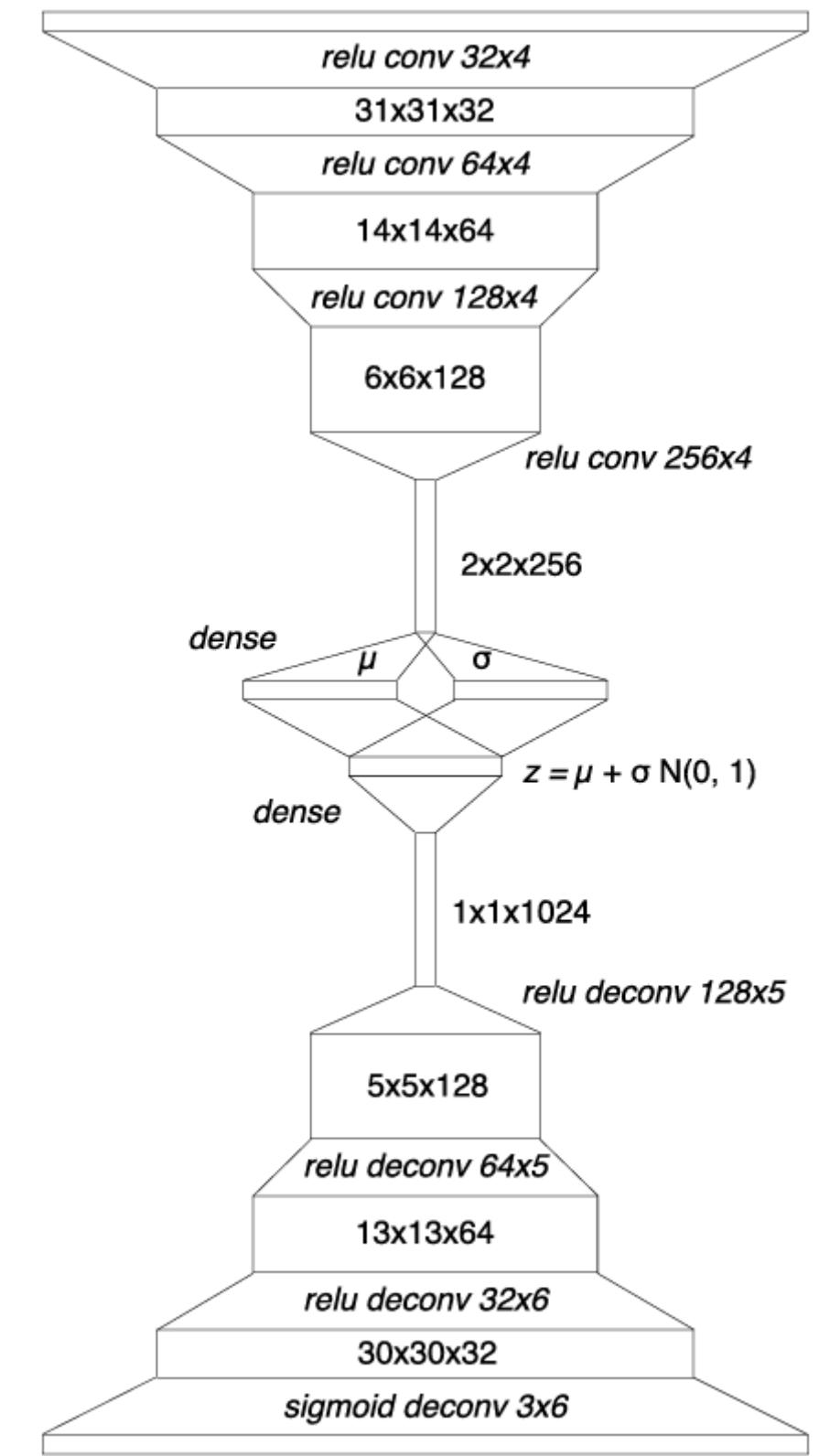


Controller (C) Model

$$a_t = W_c [z_t \ h_t] + b_c$$

where:
 W_c weight matrix
 b_c bias vector
 z_t latent vector
 h_t hidden state
 a_t action





Agent Model

At each time step, our agent receives an **observation** from the environment.

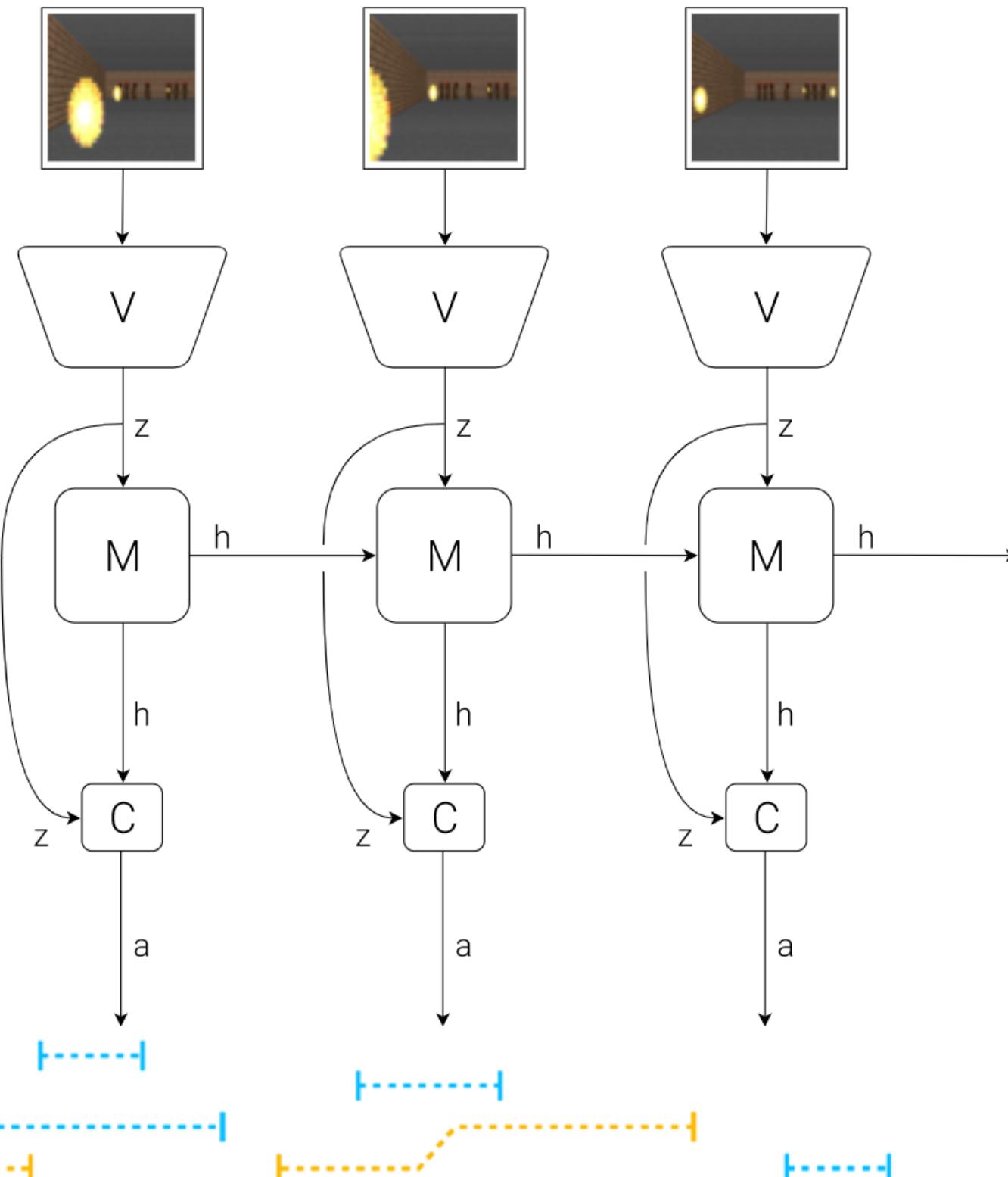
World Model

The **Vision Model (V)** encodes the high-dimensional observation into a low-dimensional latent vector.

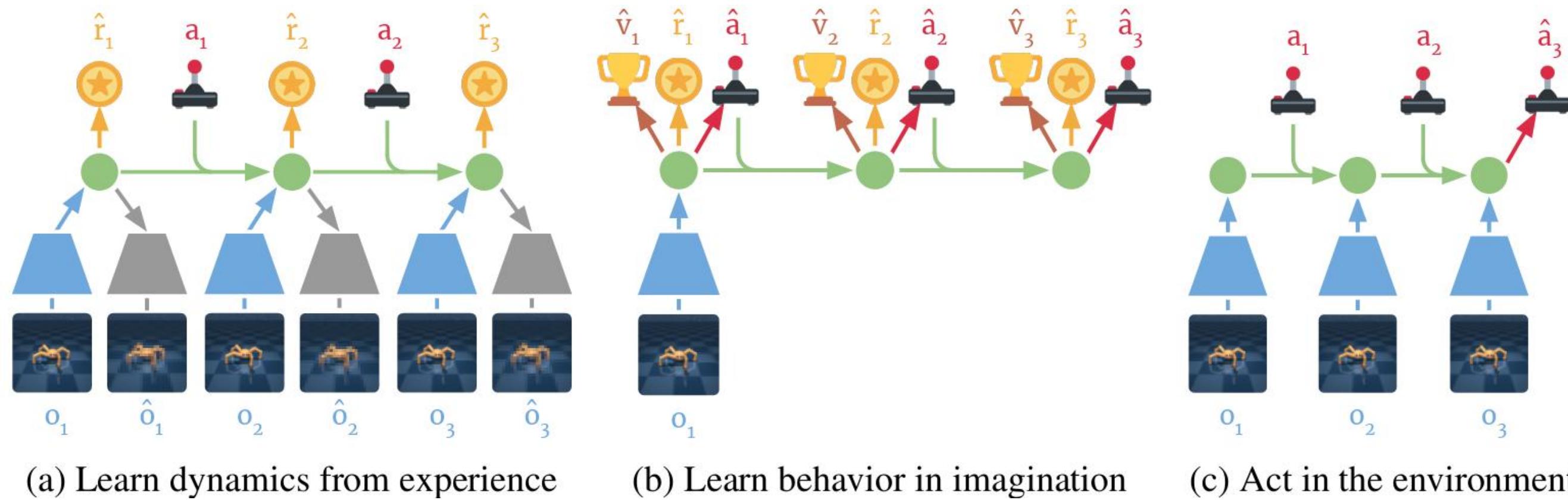
The **Memory RNN (M)** integrates the historical codes to create a representation that can predict future states.

A small **Controller (C)** uses the representations from both V and M to select good actions.

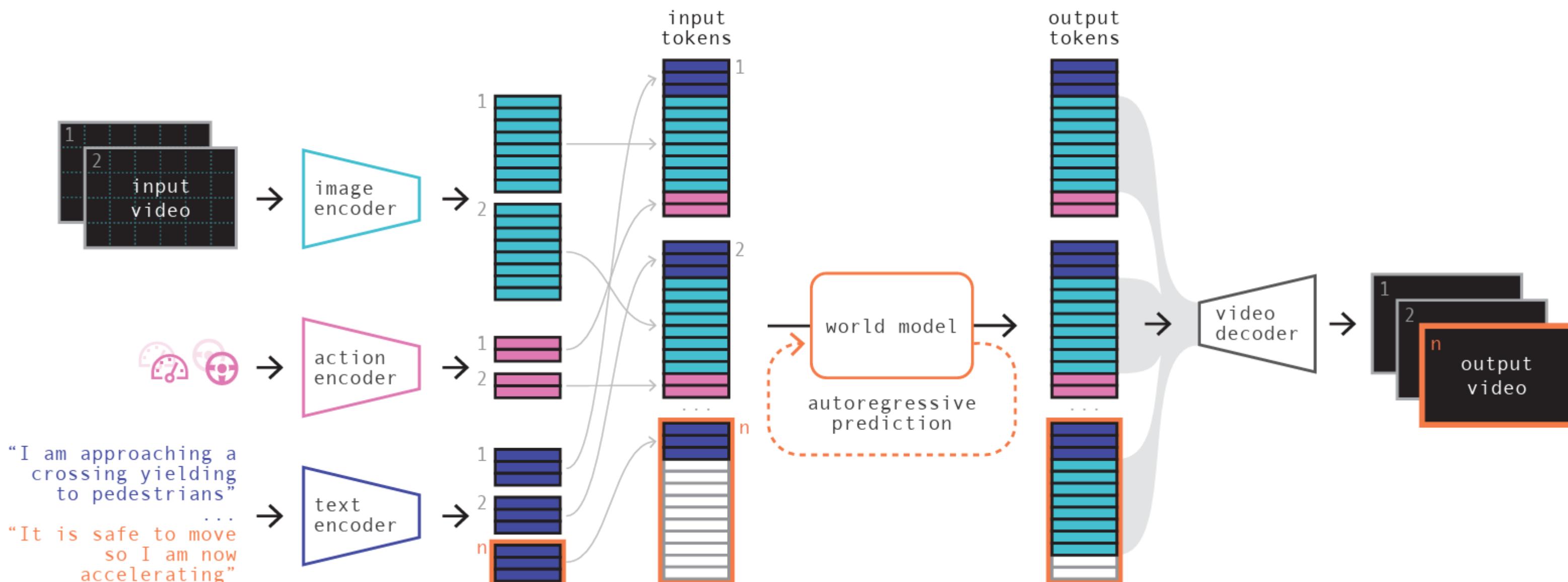
The agent performs **actions** that go back and affect the environment.



Dream to Control



GAIA-1



GAIA-1



TRANSFORMATEC

Anthony Hu et al. (2023) "GAIA-1: A Generative World Model for Autonomous Driving".
arXiv preprint arXiv:1803.10122.

GAIA-1



TRANSFORMATEC

Anthony Hu et al. (2023) "GAIA-1: A Generative World Model for Autonomous Driving".
arXiv preprint arXiv:1803.10122.

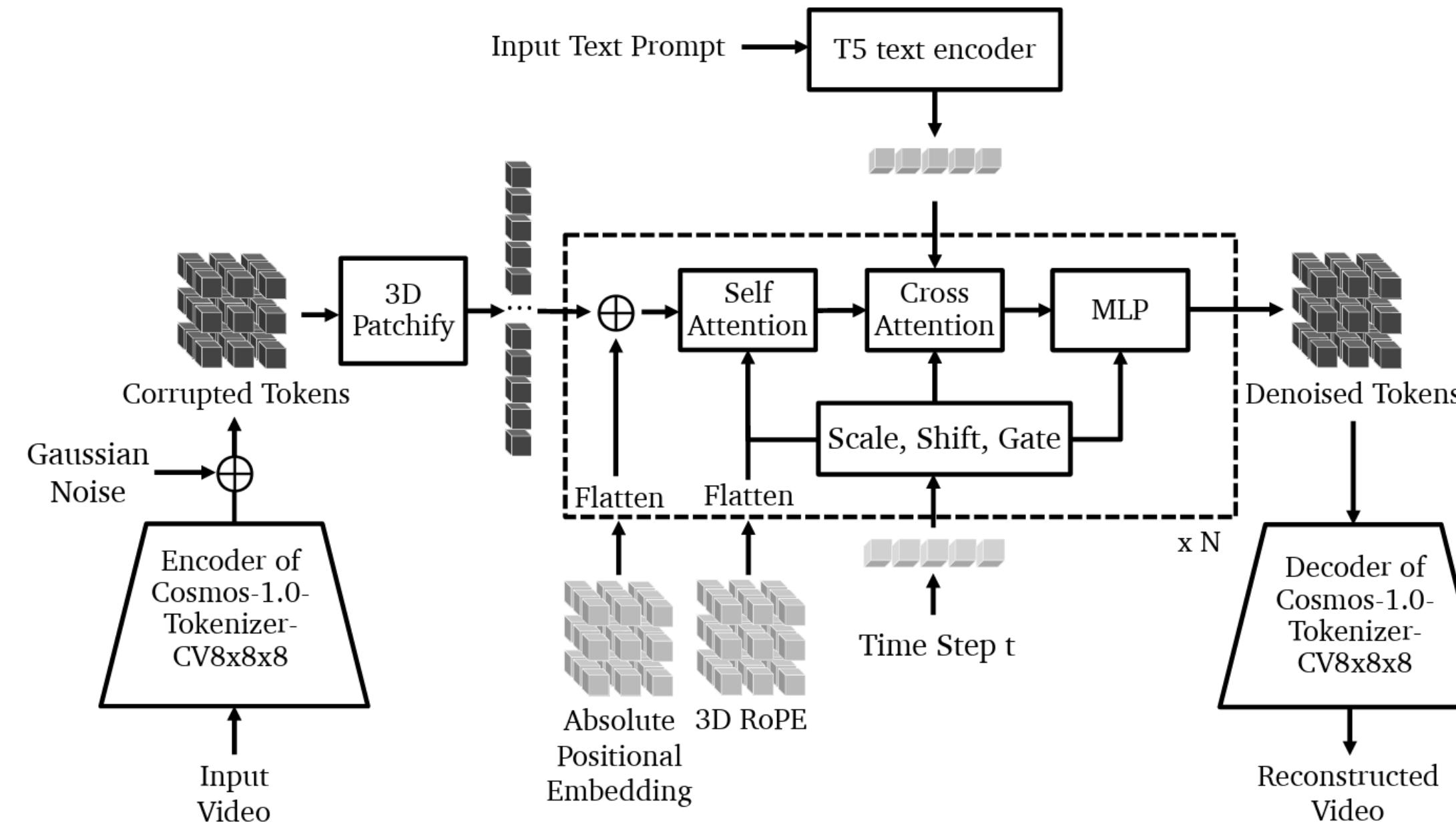
GAIA-1



TRANSFORMATEC

Anthony Hu et al. (2023) "GAIA-1: A Generative World Model for Autonomous Driving".
arXiv preprint arXiv:1803.10122.

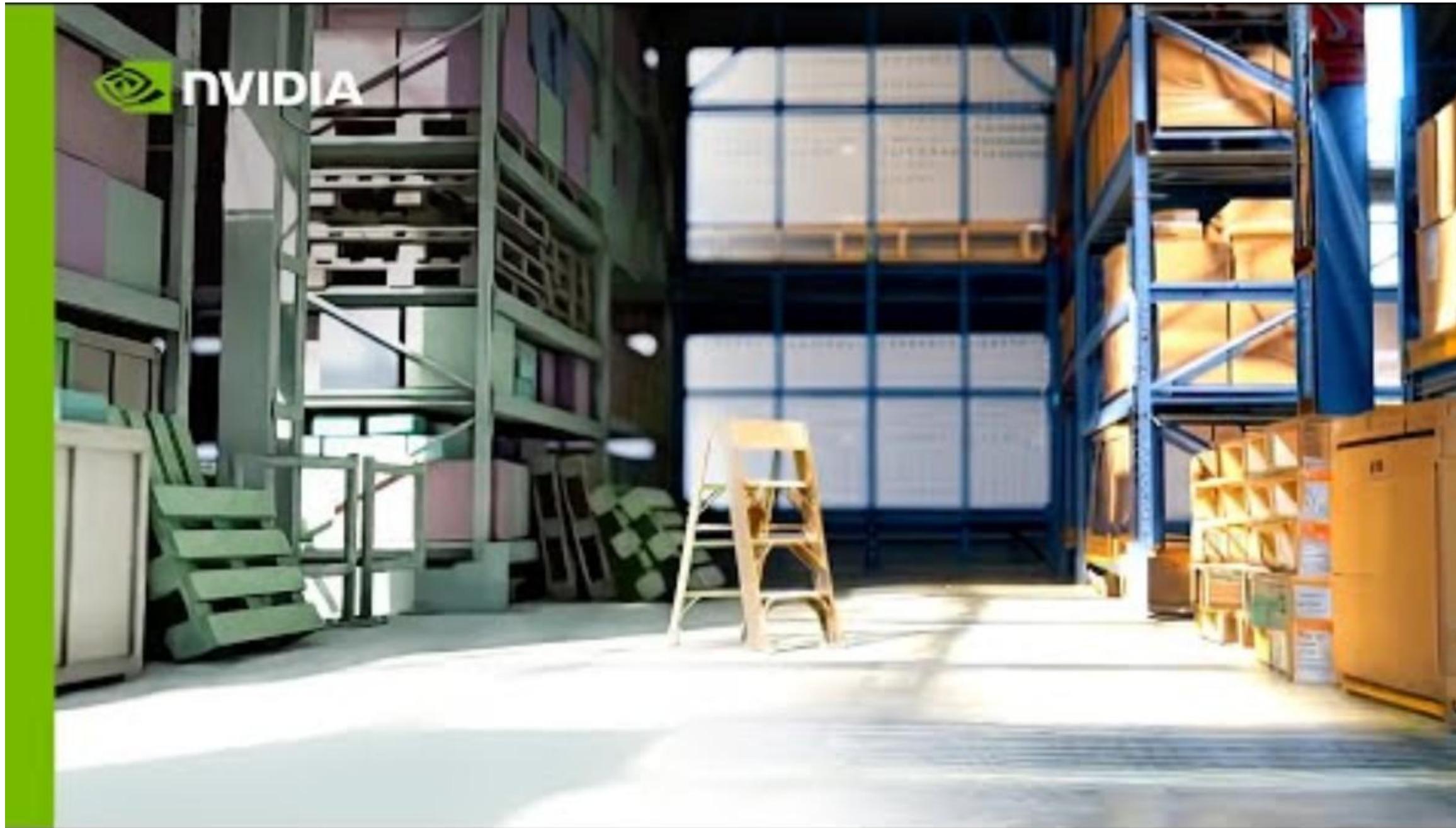
Cosmos-1.0-Diffusion *World Foundation Model*



TRANSFORMATEC

Niket Agarwal et al. (2025) "Cosmos World Foundation Model Platform for Physical AI".
arXiv preprint arXiv:2501.03575

Cosmos-1.0-Diffusion *World Foundation Model*



TRANSFORMATEC

Niket Agarwal et al. (2025) "Cosmos World Foundation Model Platform for Physical AI".
arXiv preprint arXiv:2501.03575

GRACIAS

Victor Flores Benites