import numpy as np

import IPython.display as display from matplotlib import pyplot as plt import io

import base64

ys = 200 + np.random.randn(100) x = [x for x in range(len(ys))]

fig = plt.figure(figsize=(4, 3), facecolor='w') plt.plot(x, ys, '-')

plt.fill\_between(x, ys, 195, where=(ys > 195), facecolor='g', alpha=0.6)

plt.title("Sample Visualization", fontsize=10)

data = io.BytesIO() plt.savefig(data)

image = F"data:image/png;base64,

{base64.b64encode(data.getvalue()).decode()}" alt = "Sample Visualization"

display.display(display.Markdown(F"""![{alt}]({image})""")) plt.close(fig)

<IPython.core.display.Markdown object>

*# Install dependencies (if needed)*

!pip install tensorflow numpy matplotlib seaborn

*# Import necessary libraries*

import numpy as np import tensorflow as tf

from tensorflow import keras

from tensorflow.keras.layers import Input, Dense, Flatten, Reshape,

Conv2D, Conv2DTranspose, Lambda

from tensorflow.keras.models import Model from tensorflow.keras import backend as K import matplotlib.pyplot as plt

import seaborn as sns

*# Load Fashion MNIST Dataset*

(x\_train, \_), (x\_test, \_) = keras.datasets.fashion\_mnist.load\_data()

*# Normalize and reshape the dataset*

x\_train = x\_train.astype("float32") / 255.0 x\_test = x\_test.astype("float32") / 255.0

x\_train = np.expand\_dims(x\_train, axis=-1) *# (28, 28, 1)*

x\_test = np.expand\_dims(x\_test, axis=-1)

*# Define Latent Space Dimension*

latent\_dim = 2 *# Keeping it 2D for easy visualization*

*# Encoder*

def build\_encoder():

inputs = Input(shape=(28, 28, 1))

x = Conv2D(32, (3, 3), strides=2, activation="relu", padding="same")(inputs)

x = Conv2D(64, (3, 3), strides=2, activation="relu", padding="same")(x)

x = Flatten()(x)

x = Dense(16, activation="relu")(x)

z\_mean = Dense(latent\_dim, name="z\_mean")(x) z\_log\_var = Dense(latent\_dim, name="z\_log\_var")(x)

def sampling(args):

z\_mean, z\_log\_var = args batch = K.shape(z\_mean)[0]

epsilon = K.random\_normal(shape=(batch, latent\_dim)) return z\_mean + K.exp(0.5 \* z\_log\_var) \* epsilon

z = Lambda(sampling, name="z")([z\_mean, z\_log\_var])

return Model(inputs, [z\_mean, z\_log\_var, z], name="Encoder")

*# Decoder*

def build\_decoder():

latent\_inputs = Input(shape=(latent\_dim,))

x = Dense(7 \* 7 \* 64, activation="relu")(latent\_inputs) x = Reshape((7, 7, 64))(x)

x = Conv2DTranspose(64, (3, 3), strides=2, activation="relu", padding="same")(x)

x = Conv2DTranspose(32, (3, 3), strides=2, activation="relu", padding="same")(x)

outputs = Conv2DTranspose(1, (3, 3), activation="sigmoid", padding="same")(x)

return Model(latent\_inputs, outputs, name="Decoder")

*# \*VAE Model as a Custom Keras Model\**

class VAE(Model):

def \_init\_(self, encoder, decoder, \*\*kwargs): super(VAE, self).\_init\_(\*\*kwargs) self.encoder = encoder

self.decoder = decoder

def call(self, inputs):

z\_mean, z\_log\_var, z = self.encoder(inputs) reconstructed = self.decoder(z)

*# Compute VAE Loss*

reconstruction\_loss = tf.reduce\_mean( keras.losses.binary\_crossentropy(K.flatten(inputs),

K.flatten(reconstructed))

)

kl\_loss = -0.5 \* tf.reduce\_mean(1 + z\_log\_var - tf.square(z\_mean) - tf.exp(z\_log\_var))

total\_loss = reconstruction\_loss + kl\_loss

self.add\_loss(total\_loss) return reconstructed

*# Initialize VAE*

encoder = build\_encoder() decoder = build\_decoder() vae = VAE(encoder, decoder)

*# Compile & Train*

vae.compile(optimizer="adam")

vae.fit(x\_train, epochs=5, batch\_size=128, validation\_data=(x\_test, None))

*# Save Model*

vae.save("vae\_fashion\_mnist.h5")

*# \*Latent Space Exploration\**

def plot\_latent\_space(encoder, x\_test): z\_mean, \_, \_ = encoder.predict(x\_test) plt.figure(figsize=(8, 6))

sns.scatterplot(x=z\_mean[:, 0], y=z\_mean[:, 1])

plt.xlabel("Latent Dimension 1")

plt.ylabel("Latent Dimension 2") plt.title("Latent Space Distribution") plt.show()

plot\_latent\_space(encoder, x\_test)

*# \*New Sample Generation\**

def generate\_samples(decoder, num\_samples=10): random\_latent\_vectors = np.random.normal(size=(num\_samples,

latent\_dim))

generated\_images = decoder.predict(random\_latent\_vectors)

fig, axes = plt.subplots(1, num\_samples, figsize=(20, 2))

for i in range(num\_samples): axes[i].imshow(generated\_images[i].squeeze(), cmap="gray") axes[i].axis("off")

plt.show()

generate\_samples(decoder)

*# \*Manipulating the Latent Space\**

def manipulate\_latent\_space(decoder): z1 = np.array([[-2, -2]])

z2 = np.array([[2, 2]])

img1 = decoder.predict(z1)[0].squeeze() img2 = decoder.predict(z2)[0].squeeze()

fig, ax = plt.subplots(1, 2, figsize=(6, 3)) ax[0].imshow(img1, cmap="gray") ax[0].set\_title("z = [-2, -2]") ax[0].axis("off")

ax[1].imshow(img2, cmap="gray") ax[1].set\_title("z = [2, 2]") ax[1].axis("off")

plt.show() manipulate\_latent\_space(decoder)

*# \*Performance Metrics\**

def reconstruction\_error(vae, x\_test): reconstructed = vae.predict(x\_test)

mse = np.mean(np.square(x\_test - reconstructed)) print(f"Reconstruction MSE: {mse:.4f}")

reconstruction\_error(vae, x\_test) Requirement already satisfied: tensorflow in

/usr/local/lib/python3.11/dist-packages (2.18.0)

Requirement already satisfied: numpy in

/usr/local/lib/python3.11/dist-packages (1.26.4) Requirement already satisfied: matplotlib in

/usr/local/lib/python3.11/dist-packages (3.10.0) Requirement already satisfied: seaborn in

/usr/local/lib/python3.11/dist-packages (0.13.2) Requirement already satisfied: absl-py>=1.0.0 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (1.4.0) Requirement already satisfied: astunparse>=1.6.0 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (1.6.3) Requirement already satisfied: flatbuffers>=24.3.25 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (25.1.24)

Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (0.6.0) Requirement already satisfied: google-pasta>=0.1.1 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (0.2.0) Requirement already satisfied: libclang>=13.0.0 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (18.1.1) Requirement already satisfied: opt-einsum>=2.3.2 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (3.4.0) Requirement already satisfied: packaging in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (24.2) Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!

=4.21.3,!=4.21.4,!=4.21.5,<6.0.0dev,>=3.20.3 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (4.25.6) Requirement already satisfied: requests<3,>=2.21.0 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (2.32.3) Requirement already satisfied: setuptools in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (75.1.0) Requirement already satisfied: six>=1.12.0 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (1.17.0) Requirement already satisfied: termcolor>=1.1.0 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (2.5.0) Requirement already satisfied: typing-extensions>=3.6.6 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (4.12.2) Requirement already satisfied: wrapt>=1.11.0 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (1.17.2) Requirement already satisfied: grpcio<2.0,>=1.24.3 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (1.70.0) Requirement already satisfied: tensorboard<2.19,>=2.18 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (2.18.0) Requirement already satisfied: keras>=3.5.0 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (3.8.0) Requirement already satisfied: h5py>=3.11.0 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (3.12.1) Requirement already satisfied: ml-dtypes<0.5.0,>=0.4.0 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (0.4.1) Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (0.37.1) Requirement already satisfied: contourpy>=1.0.1 in

/usr/local/lib/python3.11/dist-packages (from matplotlib) (1.3.1) Requirement already satisfied: cycler>=0.10 in

/usr/local/lib/python3.11/dist-packages (from matplotlib) (0.12.1) Requirement already satisfied: fonttools>=4.22.0 in

/usr/local/lib/python3.11/dist-packages (from matplotlib) (4.55.7) Requirement already satisfied: kiwisolver>=1.3.1 in

/usr/local/lib/python3.11/dist-packages (from matplotlib) (1.4.8) Requirement already satisfied: pillow>=8 in

/usr/local/lib/python3.11/dist-packages (from matplotlib) (11.1.0) Requirement already satisfied: pyparsing>=2.3.1 in

/usr/local/lib/python3.11/dist-packages (from matplotlib) (3.2.1)

Requirement already satisfied: python-dateutil>=2.7 in

/usr/local/lib/python3.11/dist-packages (from matplotlib) (2.8.2) Requirement already satisfied: pandas>=1.2 in

/usr/local/lib/python3.11/dist-packages (from seaborn) (2.2.2) Requirement already satisfied: wheel<1.0,>=0.23.0 in

/usr/local/lib/python3.11/dist-packages (from astunparse>=1.6.0-

>tensorflow) (0.45.1)

Requirement already satisfied: rich in /usr/local/lib/python3.11/dist- packages (from keras>=3.5.0->tensorflow) (13.9.4)

Requirement already satisfied: namex in

/usr/local/lib/python3.11/dist-packages (from keras>=3.5.0-

>tensorflow) (0.0.8)

Requirement already satisfied: optree in

/usr/local/lib/python3.11/dist-packages (from keras>=3.5.0-

>tensorflow) (0.14.0)

Requirement already satisfied: pytz>=2020.1 in

/usr/local/lib/python3.11/dist-packages (from pandas>=1.2->seaborn) (2024.2)

Requirement already satisfied: tzdata>=2022.7 in

/usr/local/lib/python3.11/dist-packages (from pandas>=1.2->seaborn) (2025.1)

Requirement already satisfied: charset-normalizer<4,>=2 in

/usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0-

>tensorflow) (3.4.1)

Requirement already satisfied: idna<4,>=2.5 in

/usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0-

>tensorflow) (3.10)

Requirement already satisfied: urllib3<3,>=1.21.1 in

/usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0-

>tensorflow) (2.3.0)

Requirement already satisfied: certifi>=2017.4.17 in

/usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0-

>tensorflow) (2024.12.14)

Requirement already satisfied: markdown>=2.6.8 in

/usr/local/lib/python3.11/dist-packages (from tensorboard<2.19,>=2.18-

>tensorflow) (3.7)

Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /usr/local/lib/python3.11/dist-packages (from tensorboard<2.19,>=2.18->tensorflow) (0.7.2)

Requirement already satisfied: werkzeug>=1.0.1 in

/usr/local/lib/python3.11/dist-packages (from tensorboard<2.19,>=2.18-

>tensorflow) (3.1.3)

Requirement already satisfied: MarkupSafe>=2.1.1 in

/usr/local/lib/python3.11/dist-packages (from werkzeug>=1.0.1-

>tensorboard<2.19,>=2.18->tensorflow) (3.0.2) Requirement already satisfied: markdown-it-py>=2.2.0 in

/usr/local/lib/python3.11/dist-packages (from rich->keras>=3.5.0-

>tensorflow) (3.0.0)

Requirement already satisfied: pygments<3.0.0,>=2.13.0 in

/usr/local/lib/python3.11/dist-packages (from rich->keras>=3.5.0-

>tensorflow) (2.18.0)

Requirement already satisfied: mdurl~=0.1 in

/usr/local/lib/python3.11/dist-packages (from markdown-it-py>=2.2.0-

>rich->keras>=3.5.0->tensorflow) (0.1.2)

ValueError Traceback (most recent call last)

<ipython-input-2-b8ebdb6d0edc> in <cell line: 0>()

---> 82 vae = VAE(encoder, decoder)

1. encoder = build\_encoder()
2. decoder = build\_decoder()

83

84 # Compile & Train

/usr/local/lib/python3.11/dist-packages/keras/src/models/model.py in

init (self, \*args, \*\*kwargs)

154 if functional\_init\_arguments(args, kwargs):

155 inject\_functional\_model\_class(self. class )

--> 156 functional.Functional. init (self, \*args,

\*\*kwargs)

157 else:

158 Layer. init (self, \*args, \*\*kwargs)

/usr/local/lib/python3.11/dist-packages/keras/src/utils/tracking.py in wrapper(\*args, \*\*kwargs)

24 def wrapper(\*args, \*\*kwargs):

25 with DotNotTrackScope():

---> 26 return fn(\*args, \*\*kwargs)

27

28 return wrapper

/usr/local/lib/python3.11/dist-packages/keras/src/models/functional.py in init (self, inputs, outputs, name, \*\*kwargs)

117 for x in flat\_inputs:

118 if not isinstance(x, backend.KerasTensor):

--> 119 raise ValueError(

120 "All `inputs` values must be KerasTensors.

Received: "

121 f"inputs={inputs} including invalid value

{x} of "

ValueError: All `inputs` values must be KerasTensors. Received: inputs=<Functional name=Encoder, built=True> including invalid value

<Functional name=Encoder, built=True> of type <class 'keras.src.models.functional.Functional'>

*# Install dependencies (if needed)*

!pip install tensorflow numpy matplotlib seaborn

*# Import necessary libraries*

import numpy as np import tensorflow as tf

from tensorflow import keras

from tensorflow.keras.layers import Input, Dense, Flatten, Reshape,

Conv2D, Conv2DTranspose, Lambda

from tensorflow.keras.models import Model from tensorflow.keras import backend as K import matplotlib.pyplot as plt

import seaborn as sns

*# Load Fashion MNIST Dataset*

(x\_train, \_), (x\_test, \_) = keras.datasets.fashion\_mnist.load\_data()

*# Normalize and reshape the dataset*

x\_train = x\_train.astype("float32") / 255.0 x\_test = x\_test.astype("float32") / 255.0

x\_train = np.expand\_dims(x\_train, axis=-1) *# (28, 28, 1)*

x\_test = np.expand\_dims(x\_test, axis=-1)

*# Define Latent Space Dimension*

latent\_dim = 2 *# Keeping it 2D for easy visualization*

*# Encoder*

def build\_encoder():

inputs = Input(shape=(28, 28, 1))

x = Conv2D(32, (3, 3), strides=2, activation="relu", padding="same")(inputs)

x = Conv2D(64, (3, 3), strides=2, activation="relu", padding="same")(x)

x = Flatten()(x)

x = Dense(16, activation="relu")(x)

z\_mean = Dense(latent\_dim, name="z\_mean")(x) z\_log\_var = Dense(latent\_dim, name="z\_log\_var")(x)

def sampling(args):

z\_mean, z\_log\_var = args batch = K.shape(z\_mean)[0]

epsilon = K.random\_normal(shape=(batch, latent\_dim)) return z\_mean + K.exp(0.5 \* z\_log\_var) \* epsilon

z = Lambda(sampling, name="z")([z\_mean, z\_log\_var])

return Model(inputs, [z\_mean, z\_log\_var, z], name="Encoder")

*# Decoder*

def build\_decoder():

latent\_inputs = Input(shape=(latent\_dim,))

x = Dense(7 \* 7 \* 64, activation="relu")(latent\_inputs) x = Reshape((7, 7, 64))(x)

x = Conv2DTranspose(64, (3, 3), strides=2, activation="relu", padding="same")(x)

x = Conv2DTranspose(32, (3, 3), strides=2, activation="relu", padding="same")(x)

outputs = Conv2DTranspose(1, (3, 3), activation="sigmoid", padding="same")(x)

return Model(latent\_inputs, outputs, name="Decoder")

*# VAE Model as a Custom Keras Model*

class VAE(Model):

def init (self, encoder, decoder, \*\*kwargs):

super(VAE, self). init (\*\*kwargs) *# Fix the call to `super`*

self.encoder = encoder self.decoder = decoder

def call(self, inputs):

*# Pass input through encoder*

z\_mean, z\_log\_var, z = self.encoder(inputs)

*# Generate the reconstructed image from latent code*

reconstructed = self.decoder(z)

*# Compute VAE Loss*

reconstruction\_loss = tf.reduce\_mean( keras.losses.binary\_crossentropy(K.flatten(inputs),

K.flatten(reconstructed))

)

kl\_loss = -0.5 \* tf.reduce\_mean(1 + z\_log\_var - tf.square(z\_mean) - tf.exp(z\_log\_var))

total\_loss = reconstruction\_loss + kl\_loss

self.add\_loss(total\_loss) return reconstructed

*# Initialize VAE*

encoder = build\_encoder() decoder = build\_decoder() vae = VAE(encoder, decoder)

*# Compile & Train*

vae.compile(optimizer="adam")

vae.fit(x\_train, epochs=5, batch\_size=128, validation\_data=(x\_test, None))

*# Save Model*

vae.save("vae\_fashion\_mnist.h5")

*# Latent Space Exploration*

def plot\_latent\_space(encoder, x\_test): z\_mean, \_, \_ = encoder.predict(x\_test) plt.figure(figsize=(8, 6))

sns.scatterplot(x=z\_mean[:, 0], y=z\_mean[:, 1])

plt.xlabel("Latent Dimension 1")

plt.ylabel("Latent Dimension 2") plt.title("Latent Space Distribution") plt.show()

plot\_latent\_space(encoder, x\_test)

*# New Sample Generation*

def generate\_samples(decoder, num\_samples=10): random\_latent\_vectors = np.random.normal(size=(num\_samples,

latent\_dim))

generated\_images = decoder.predict(random\_latent\_vectors)

fig, axes = plt.subplots(1, num\_samples, figsize=(20, 2)) for i in range(num\_samples):

axes[i].imshow(generated\_images[i].squeeze(), cmap="gray") axes[i].axis("off")

plt.show()

generate\_samples(decoder)

*# Manipulating the Latent Space*

def manipulate\_latent\_space(decoder): z1 = np.array([[-2, -2]])

z2 = np.array([[2, 2]])

img1 = decoder.predict(z1)[0].squeeze() img2 = decoder.predict(z2)[0].squeeze()

fig, ax = plt.subplots(1, 2, figsize=(6, 3)) ax[0].imshow(img1, cmap="gray") ax[0].set\_title("z = [-2, -2]") ax[0].axis("off")

ax[1].imshow(img2, cmap="gray") ax[1].set\_title("z = [2, 2]") ax[1].axis("off")

plt.show() manipulate\_latent\_space(decoder)

*# Performance Metrics*

def reconstruction\_error(vae, x\_test): reconstructed = vae.predict(x\_test)

mse = np.mean(np.square(x\_test - reconstructed)) print(f"Reconstruction MSE: {mse:.4f}")

reconstruction\_error(vae, x\_test) Requirement already satisfied: tensorflow in

/usr/local/lib/python3.11/dist-packages (2.18.0)

Requirement already satisfied: numpy in

/usr/local/lib/python3.11/dist-packages (1.26.4) Requirement already satisfied: matplotlib in

/usr/local/lib/python3.11/dist-packages (3.10.0) Requirement already satisfied: seaborn in

/usr/local/lib/python3.11/dist-packages (0.13.2) Requirement already satisfied: absl-py>=1.0.0 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (1.4.0) Requirement already satisfied: astunparse>=1.6.0 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (1.6.3) Requirement already satisfied: flatbuffers>=24.3.25 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (25.1.24) Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (0.6.0) Requirement already satisfied: google-pasta>=0.1.1 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (0.2.0) Requirement already satisfied: libclang>=13.0.0 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (18.1.1) Requirement already satisfied: opt-einsum>=2.3.2 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (3.4.0) Requirement already satisfied: packaging in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (24.2) Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!

=4.21.3,!=4.21.4,!=4.21.5,<6.0.0dev,>=3.20.3 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (4.25.6) Requirement already satisfied: requests<3,>=2.21.0 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (2.32.3) Requirement already satisfied: setuptools in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (75.1.0) Requirement already satisfied: six>=1.12.0 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (1.17.0) Requirement already satisfied: termcolor>=1.1.0 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (2.5.0) Requirement already satisfied: typing-extensions>=3.6.6 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (4.12.2) Requirement already satisfied: wrapt>=1.11.0 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (1.17.2) Requirement already satisfied: grpcio<2.0,>=1.24.3 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (1.70.0) Requirement already satisfied: tensorboard<2.19,>=2.18 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (2.18.0) Requirement already satisfied: keras>=3.5.0 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (3.8.0) Requirement already satisfied: h5py>=3.11.0 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (3.12.1)

Requirement already satisfied: ml-dtypes<0.5.0,>=0.4.0 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (0.4.1) Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in

/usr/local/lib/python3.11/dist-packages (from tensorflow) (0.37.1) Requirement already satisfied: contourpy>=1.0.1 in

/usr/local/lib/python3.11/dist-packages (from matplotlib) (1.3.1) Requirement already satisfied: cycler>=0.10 in

/usr/local/lib/python3.11/dist-packages (from matplotlib) (0.12.1) Requirement already satisfied: fonttools>=4.22.0 in

/usr/local/lib/python3.11/dist-packages (from matplotlib) (4.55.7) Requirement already satisfied: kiwisolver>=1.3.1 in

/usr/local/lib/python3.11/dist-packages (from matplotlib) (1.4.8) Requirement already satisfied: pillow>=8 in

/usr/local/lib/python3.11/dist-packages (from matplotlib) (11.1.0) Requirement already satisfied: pyparsing>=2.3.1 in

/usr/local/lib/python3.11/dist-packages (from matplotlib) (3.2.1) Requirement already satisfied: python-dateutil>=2.7 in

/usr/local/lib/python3.11/dist-packages (from matplotlib) (2.8.2) Requirement already satisfied: pandas>=1.2 in

/usr/local/lib/python3.11/dist-packages (from seaborn) (2.2.2) Requirement already satisfied: wheel<1.0,>=0.23.0 in

/usr/local/lib/python3.11/dist-packages (from astunparse>=1.6.0-

>tensorflow) (0.45.1)

Requirement already satisfied: rich in /usr/local/lib/python3.11/dist- packages (from keras>=3.5.0->tensorflow) (13.9.4)

Requirement already satisfied: namex in

/usr/local/lib/python3.11/dist-packages (from keras>=3.5.0-

>tensorflow) (0.0.8)

Requirement already satisfied: optree in

/usr/local/lib/python3.11/dist-packages (from keras>=3.5.0-

>tensorflow) (0.14.0)

Requirement already satisfied: pytz>=2020.1 in

/usr/local/lib/python3.11/dist-packages (from pandas>=1.2->seaborn) (2024.2)

Requirement already satisfied: tzdata>=2022.7 in

/usr/local/lib/python3.11/dist-packages (from pandas>=1.2->seaborn) (2025.1)

Requirement already satisfied: charset-normalizer<4,>=2 in

/usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0-

>tensorflow) (3.4.1)

Requirement already satisfied: idna<4,>=2.5 in

/usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0-

>tensorflow) (3.10)

Requirement already satisfied: urllib3<3,>=1.21.1 in

/usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0-

>tensorflow) (2.3.0)

Requirement already satisfied: certifi>=2017.4.17 in

/usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0-

>tensorflow) (2024.12.14)

Requirement already satisfied: markdown>=2.6.8 in

/usr/local/lib/python3.11/dist-packages (from tensorboard<2.19,>=2.18-

>tensorflow) (3.7)

Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /usr/local/lib/python3.11/dist-packages (from tensorboard<2.19,>=2.18->tensorflow) (0.7.2)

Requirement already satisfied: werkzeug>=1.0.1 in

/usr/local/lib/python3.11/dist-packages (from tensorboard<2.19,>=2.18-

>tensorflow) (3.1.3)

Requirement already satisfied: MarkupSafe>=2.1.1 in

/usr/local/lib/python3.11/dist-packages (from werkzeug>=1.0.1-

>tensorboard<2.19,>=2.18->tensorflow) (3.0.2) Requirement already satisfied: markdown-it-py>=2.2.0 in

/usr/local/lib/python3.11/dist-packages (from rich->keras>=3.5.0-

>tensorflow) (3.0.0)

Requirement already satisfied: pygments<3.0.0,>=2.13.0 in

/usr/local/lib/python3.11/dist-packages (from rich->keras>=3.5.0-

>tensorflow) (2.18.0)

Requirement already satisfied: mdurl~=0.1 in

/usr/local/lib/python3.11/dist-packages (from markdown-it-py>=2.2.0-

>rich->keras>=3.5.0->tensorflow) (0.1.2) Epoch 1/5

469/469 ━━━━━━━━━━━━━━━━━━━━ 103s 212ms/step - loss: 0.5447 - val\_loss: 0.4918

Epoch 2/5

469/469 ━━━━━━━━━━━━━━━━━━━━ 100s 212ms/step - loss: 0.4908 - val\_loss: 0.4914

Epoch 3/5

469/469 ━━━━━━━━━━━━━━━━━━━━ 140s 209ms/step - loss: 0.4909 - val\_loss: 0.4921

Epoch 4/5

469/469 ━━━━━━━━━━━━━━━━━━━━ 138s 201ms/step - loss: 0.4908 - val\_loss: 0.4915

Epoch 5/5

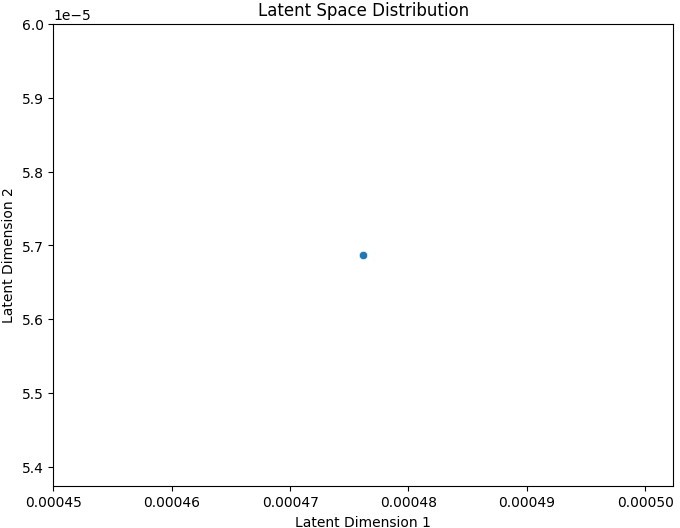
469/469 ━━━━━━━━━━━━━━━━━━━━ 100s 213ms/step - loss: 0.4907 - val\_loss: 0.4927

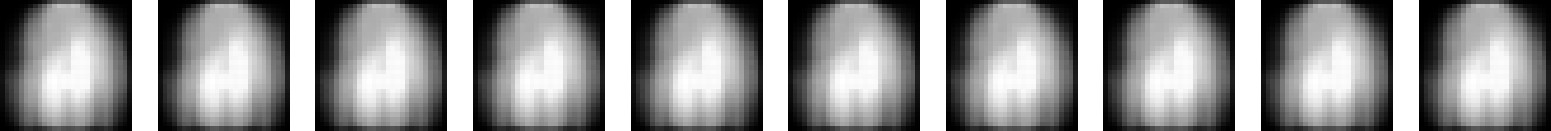
WARNING:absl:You are saving your model as an HDF5 file via

`model.save()` or `keras.saving.save\_model(model)`. This file format is considered legacy. We recommend using instead the native Keras format, e.g. `model.save('my\_model.keras')` or

`keras.saving.save\_model(model, 'my\_model.keras')`.

313/313 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step

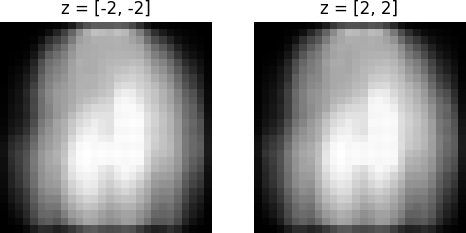




1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 121ms/step

1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step

1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step



WARNING:tensorflow:5 out of the last 317 calls to <function TensorFlowTrainer.make\_predict\_function.<locals>.one\_step\_on\_data\_dist ributed at 0x7ece77c28860> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has reduce\_retracing=True option that can avoid unnecessary retracing. For (3), please refer to [https://www.tensorflow.org/guide/function#controlling\_retracing and](http://www.tensorflow.org/guide/function#controlling_retracingand) [https://www.tensorflow.org/api\_docs/python/tf/function for more](http://www.tensorflow.org/api_docs/python/tf/functionformore) details.

313/313 ━━━━━━━━━━━━━━━━━━━━ 6s 17ms/step Reconstruction MSE: 0.0871