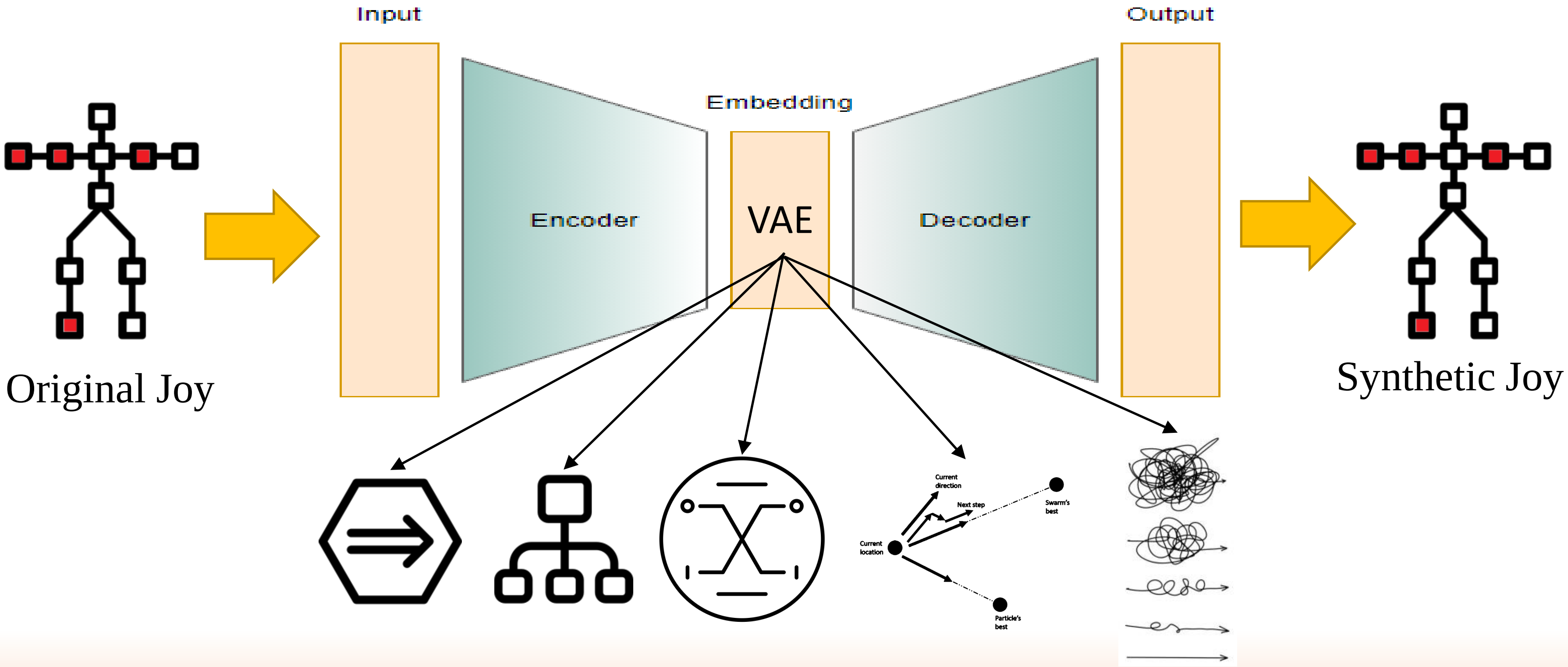
# Conditional Hierarchical Fuzzy PSO Disentangled VAE

By: Seyed Muhammad Hossein Mousavi
mosavi.a.i.buali@gmail.com
Lugano, Switzerland

September 2024



Input

Embedding

Output

Encoder

VAE

Decoder

Original Joy

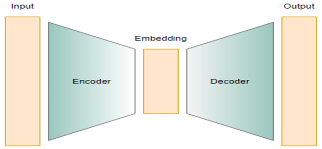
Synthetic Joy

Title

# Outline:

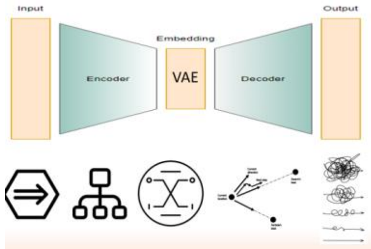
- **Auto Encoder (AE)**
  - Definition
  - Applications
  - Challenges in SDG
  - AE structure
  - More on the latent space

- **Variational Auto Encoders (VAE)**



  - Definition
  - Applications
  - Challenges In SDG
  - VAE structure
  - More on the latent space
  - Difference of AE and VAE
  - Advantages over other algorithms
  - Disadvantages over other algorithms
  - In body motion SDG

- **Conditional Hierarchical Fuzzy PSO Disentangled VAE**



  - Latent Space Body Example
  - Note
  - Conditioning
  - Hierarchical
  - Fuzzy
    - Definition
    - Applications
    - In VAE
  - Optimization
    - Definition
    - Applications
    - PSO
    - In VAE (PSO)
  - Disentanglement
  - Improvement Compared to VAE
  - Implementation and Equations

# Auto Encoder (AE)

**Definition:**

- A type of neural network used for unsupervised learning that aims to learn efficient codings of input data by compressing it into a lower-dimensional latent space and then reconstructing the original input from this compressed representation.
- Aim to retain as much of the original information as possible.
- By adding an output target layer and adding supervised loss, it could be used in a supervised manner.

**Application:**

- ➢ Dimensionality reduction (feature selection)
- ➢ Denoising
- ➢ Outlier detection
- ➢ Feature extraction
- ➢ Data compression
- ➢ SDG

# Auto Encoder (AE)

**Challenges in SDG:**

- **Lack of Diversity**
  - Converge towards average representations
- **Overfitting**
  - Generated data that is too similar to the training examples and not general enough to represent real-world variations. Basically, it makes noise.
- **Feature Loss**
  - Important features may be lost during compression in the latent space, particularly if the dimensionality reduction is too aggressive or the network architecture is not suitable
- **Control Over Generated Attributes**
  - Traditional autoencoders provide limited control over specific attributes of the generated data, making it challenging to direct the generation process toward desired characteristics.

- **Solution**:
  - As AE generates similar samples to the original one, latent space should be manipulated by **interpolation**.
    - For example, by taking two latent space representations of face images and interpolating between them, the decoder can generate new faces that possess features from both original images.

# Auto Encoder (AE)

**AE Structure:**

1. **Encoder**: Compresses the input data into a lower-dimensional representation. The data is encoding deterministic.
    - An input layer, hidden layers of fully connected layers or convolutional layers (from big to small), and ReLU, a sigmoid activation function for introducing non-linearity and learning complex patterns.
2. **Latent space**: Holds the compressed representation of the input data - A lower-dimensional vector that encodes the essential features of the input data.
3. **Decoder**: Reconstructs the input data from the lower-dimensional latent representation. Just like Encode but backward (from small to big)
    - By interaction between latent vector and weights and biases of each layer, reconstruction happens.
4. **Loss function**: To determine how well the decoder is reconstructing the original data from the compressed latent representation.
    - The training process aims to minimize this loss, thereby improving the accuracy of the reconstruction.
        - Normally, **MSE** or **Binary Cross-Entropy** will be used.
5. **Training**:
    - **Backpropagation**: Used to **optimize the weights and biases** within the encoder and decoder.
        - This process involves calculating the gradient of the loss function with respect to each weight and bias and then **adjusting those weights to minimize the loss**.

# Auto Encoder (AE)

**More on the latent space:**

- If the data is image and the content is a face, the **latent space** stores:
  - **Facial Symmetry -** Variations in symmetry could be encoded in the latent space.
  - **Facial Proportions -** Distances between major facial landmarks (eyes, nose, mouth).
  - **Eye Size and Spacing -** Relative size and distance between the eyes.
  - **Mouth Shape -** Shapes related to different expressions like smiling or frowning.
  - **Nose Shape -** Width and length of the nose.
  - **Cheekbone Structure -** Prominence and position of cheekbones.
  - **Eyebrow Thickness and Arch -** Shape and size of the eyebrows.
  - **Skin Texture -** Elements like smoothness or visible age signs.
  - **Hair Style and Hairline -** Outline and style of hair might be abstractly captured.
  - **Chin and Jawline Shape -** Contours of the chin and jawline.

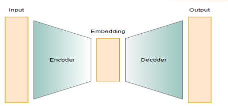
# Variational Auto Encoder (VAE)

**Definition:**

- It is a type of generative model that builds upon the architecture of a traditional autoencoder to produce complex, generative models.
- VAEs are particularly well-known for their ability to learn latent structures and distributions within data, enabling them to generate new data points that are similar to the training data but with higher diversity.
- The higher diversity comes with data distribution in the VAE, which is Gaussian distribution, but AE is crisp.

**Application:**

- ➢ SDG (various modalities)
- ➢ Data denoising (by reconstructing clean data from the corrupt)
- ➢ Feature learning and extraction (the latent space)
- ➢ Data compression (by latent space and weights/biases in different layers)
- ➢ And way more…

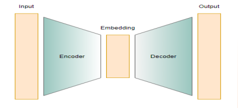


# Variational Auto Encoder (VAE)

## Challenges in SDG:

- **Reconstruction vs Regularization**:
    - VAEs balance two main components in their loss function: the reconstruction loss and the KL divergence (**regularization** term). Tuning this balance is critical because a model focused too much on reconstruction might ignore the latent space's structure, while too much focus on regularization can lead to the underfitting and poor reconstruction of the data.
        - **Regularization**: The techniques used to prevent overfitting. Overfitting occurs when a model learns not only the underlying pattern but also the noise in the training data, making it perform well on training data but poorly on unseen data. Regularization techniques help improve a model's ability to generalize from the training data to unseen data.

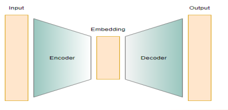
- **Vanishing Latent Space**:
    - In VAEs, particularly when used with powerful decoders such as LSTM or deep convolutional networks, there's a risk that the model might **ignore the latent variables** (known as "**posterior collapse**"). This happens if the decoder becomes too good at reconstructing the input **without needing to use the latent** codes effectively, rendering the latent space uninformative.
        - **Posterior**: It is the distribution of latent variables given the input data. This is an approximate posterior computed by the encoder part of the VAE, which learns to map input data to a distribution over latent variables.



# Variational Auto Encoder (AE)

## VAE Structure:

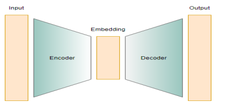
- **Encoder**: Just like AE, the encoder part of a VAE takes input data and transforms it into a distribution in a latent (hidden) space. Typically, this involves a neural network that outputs parameters of a probability distribution, usually the mean ($\mu$) and variance ($\sigma^2$) of a Gaussian distribution. The encoding is probabilistic, unlike AE, which was deterministic.
  - An input layer, hidden layers of fully connected layers or convolutional layers (from big to small), and ReLU, a sigmoid activation function for introducing non-linearity and learning complex patterns.
- **Latent space**: Holds the compressed representation of the input data - A lower-dimensional vector that encodes the essential features of the input data. This is after the final layer.
  - This is where the encoder compresses the input data into a condensed form. The latent space is represented by a **probability distribution**, typically Gaussian, characterized by mean and variance vectors. From this distribution, latent variables are sampled using a technique called reparameterization to allow for gradient-based optimization.
- **Reparameterization Trick**: Instead of sampling from the distribution directly, which can't be backpropagated through, the model samples from a standard normal distribution and then shifts and scales these samples according to the learned parameters ($\mu$ and $\sigma$).
  - This allows the model to backpropagate through the random sampling process, making it possible to train the model using standard stochastic gradient descent.



# Variational Auto Encoder (AE)

**VAE Structure:**

- **Decoder**: This is done by transforming the latent variables back into the data space, typically aiming to maximize the likelihood of the data given the latent variables. The decoder mirrors the encoder in architecture but works in reverse, gradually upsampling the latent representations back to the dimensionality of the original input data.
    - Just like Encode but backward (from small to big). By interaction between latent vectors and weights and biases of each layer, reconstruction happens.
- **Loss Function**: The VAE is trained using a loss function that consists of two parts.
    - **Reconstruction Loss**: This measures the difference between the original input data and the reconstructed data from the decoder, encouraging accurate reconstructions. To determine how well the decoder is reconstructing the original data from the compressed latent representation. Normally, MSE or Binary Cross-Entropy will be used.
    - **Kullback-Leibler (KL) Divergence**: This is a regularizer that measures the difference between the learned latent distribution and the prior distribution of the latent variables (usually a standard Gaussian). This term ensures that the latent space does not deviate too much from the assumed prior. Also, This encourages the learned distribution to be close to the prior, ensuring that the model generalizes well and does not just memorize the training data.
        - **Prior**: Probability distribution of the latent variables before any observation of the actual data.



# Variational Auto Encoder (AE)

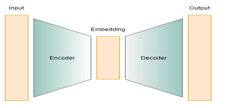
**More on the latent space:**

- The latent variables can capture key features and attributes like facial expressions, orientation, age, hairstyle, and other facial characteristics. The probabilistic nature of the VAE enables these features to be modeled as distributions rather than fixed values, providing a rich representation that can be sampled and manipulated.
- **Interpolation**: By taking two latent space representations of different face images and interpolating between them, we can generate a sequence of images that smoothly transition from one face to another.
- **Attribute Vector Arithmetic**: It's possible to find vectors in latent space that correspond to changes in specific attributes. For instance, by averaging the latent vectors of images with and without glasses and then subtracting these averages, you can isolate the "glasses" vector. Adding this vector to the latent representation of a face without glasses could add glasses to the reconstructed image.
- **Controlled Manipulation**: If certain dimensions of the latent space are known to correspond to specific attributes (which can be discovered through experimentation or more systematic study like supervised learning on labeled data), we can directly manipulate these dimensions to alter those attributes in the generated images. For example, increasing a value along a dimension that controls smile intensity can make the face smile more in the reconstructed image.
- **Exploration and Discovery**: By systematically varying latent variables and observing changes in the output, one can explore what each dimension of the latent space controls. This is often done in a controlled experimental setup where we vary one dimension while keeping others constant.
- **Conditional VAEs**: For more directed manipulation, Conditional VAEs (CVAEs) can be trained by conditioning on specific attributes (like age, gender, etc.). This allows direct control over these attributes during generation by setting the conditions along with the latent variables.

# Variational Auto Encoder (AE)

**Difference of AE and VAE:**

- **Encoding**:
  - **Autoencoders**: Output a deterministic latent representation of the input.
  - **VAEs**: Output parameters (mean and variance) of a probabilistic distribution representing the latent space.
- **Latent Space:**
  - **Autoencoders**: Typically non-probabilistic and fixed for each input.
  - **VAEs**: Probabilistic and sampled using the reparameterization trick, allowing gradient flow during training.
- **Loss Function**:
  - **Autoencoders**: Primarily focus on minimizing reconstruction error between the input and the output.
  - **VAEs**: Loss includes both reconstruction error and the Kullback-Leibler (KL) divergence to regularize the latent space towards a prior distribution.
- **Purpose**:
  - **Autoencoders**: Aimed at effective compression and reconstruction of data.
  - **VAEs**: Designed not only for reconstruction but also for generating new data instances that are similar to the input data.
- **Generative Capability**:
  - **Autoencoders**: Not inherently generative; they aim to learn a compressed representation.
  - **VAEs**: Generative models that can create new data samples by sampling from the latent space.

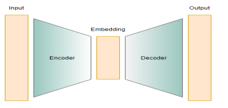


# Variational Auto Encoder (AE)

## Advantages over other algorithms

**Note**: VAE, GAN, and Diffusion models are the best for SDG tasks and surpass other algorithms.

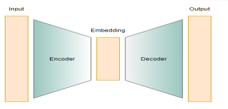
- **Stable Training**: less mode collapse. Mode collapse occurs when the generator starts producing a limited variety of outputs, even though the training data has a richer diversity.
- **Interpretable Latent Space**: The latent space of a VAE tends to be more structured and interpretable, making it easier to manipulate specific features in the generated data.
- **Ease of Training**: VAEs do not require the careful balance between a generator and discriminator that GANs need, simplifying the training process.
- **Computational Efficiency**: Training and sampling from a VAE are generally more computationally efficient than diffusion models, which require multiple forward and reverse passes through the model.
- **Simpler Implementation**: VAEs involve a less complex architectural and operational setup compared to the iterative nature of diffusion models, which require numerous time-stepping procedures.
- **Faster Sampling**: VAEs can generate new samples in a single forward pass through the decoder, whereas diffusion models typically generate samples through a lengthy iterative process, making VAEs more suitable for applications requiring real-time generation.
- **Lower Resource Requirements**: VAEs typically require less memory and computational power, making them accessible for use on devices with limited capabilities, unlike diffusion models that are resource-intensive.



# Variational Auto Encoder (AE)

## Disadvantages over other algorithms

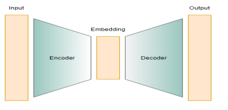
- **Lower Sample Quality**: VAEs generally produce samples with lower visual fidelity compared to GANs. GAN-generated images are often sharper and more realistic, particularly in complex domains like high-resolution photographs.
- **Blurriness in Outputs**: VAEs often result in blurrier outputs due to the use of element-wise losses like mean squared error, which averages over variations rather than capturing them precisely.
- **Detail in Generation**: Diffusion models tend to generate images with much finer details and better overall quality than VAEs. The iterative refinement process of diffusion models allows them to capture complex data distributions more effectively.
- **Expressiveness**: Diffusion models have shown a greater capacity for capturing a wider variety of data distributions and generating more diverse samples compared to VAEs, which can be somewhat limited by the Gaussian assumptions in their latent spaces.



# Variational Auto Encoder (AE)

## In body motion SDG

- **Encoder**:  An encoder uses RNN or CNN, which is trained to identify and encode the subtle differences in body language associated with various emotions. This component learns to map high-dimensional data (complex sequences of movements) into a structured latent space where similar emotional expressions are near each other.
- **Latent Space**: The latent space is designed to seperate different emotional intensities and types effectively. Manipulating the latent variables should allow smooth transitions between different emotions, such as gradually changing a motion sequence from depicting sadness to depicting happiness.
- **The reparameterization** trick involves sampling from a standard normal distribution and then transforming this sample using the learned mean and variance. This step is crucial because it converts the randomness of sampling (which isn't differentiable and thus blocks gradient backpropagation) into a deterministic and differentiable operation.
- **Decoder**: The decoder uses the latent variables to generate motion sequences that accurately reflect the intended emotions. It needs to ensure that the reconstructed motions maintain the emotional content encoded by the latent variables, preserving the integrity and recognizability of the emotional expressions.
- **Loss Function**: reconstruction loss ensures that the output motions closely match the input motions in terms of both movement and emotional expression. This might involve specialized metrics that assess emotional accuracy as well as motion fidelity. The KL divergence encourages the distribution of latent variables to stay close to a prior distribution, promoting a well-structured and generalizable latent space that is capable of varying emotions smoothly.



# Variational Auto Encoder (AE)

## Latent Space Body Example

Average joint distance Direction of movement Movement speed

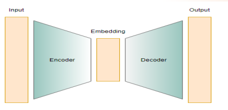
-0.89233 -1.74947 0.156183 0.218419 -0.44958 0.657947 -0.8181 -0.88357 -0.18473 -0.16203 0.690662 1.63409 1.08532

Joint speed Body symmetry

0.505999 -1.19838 -0.17305 -0.34625 -0.4284 0.675141 -1.29727 1.08986 -0.25936 0.24238 -1.77806 -1.30778 -2.04082

Energy level

1.25612 -0.41973 -0.13576 0.78871 2.65305 0.074213



# Conditional Hierarchical Fuzzy PSO Disentangled VAE

## Note

- VAEs are particularly well-known for their ability to **learn latent structures and distributions** within data, enabling them to **generate new data points that are similar to the** training data but with higher diversity.
- I propose to modify VAE as per the above title to fix the main challenge of body motion SDG:
  - ✓ **Challenge**: Capturing complex patterns of bodily emotions at a subtle level and synthesizing new samples.
- However, fixing this challenge comes with a major drawback:
  - ✓ **Drawback**: We have to increase learning **complexity**.

## Conditioning

- Making sure that the **latent space is aligned with specific conditions like emotions and intensity**.

Increasing controlled generation, such as:

- ➢ Using a sample with the highest emotion for that category to generate all other emotions based on that sample. This is called **sample conditioning**.

## Hierarchical Structure

- It organizes the latent space into different levels, capturing both macro and micro details of motion in a categorized manner and saving them in different layers.
- Data is processed through these levels sequentially, allowing each level to refine or add to the representations formed by the previous ones.
- The hierarchical organization **can help in disentangling** different levels of variation naturally, with different layers specializing in different aspects of the data.

# Conditional Hierarchical Fuzzy PSO Disentangled VAE

## Fuzzy Logic

- **Definition:**
  - Fuzzy logic is a form of many-valued logic where truth values range between 0 and 1, reflecting the degree of truth of variables (**degrees of membership**) rather than a strict true/false like in Boolean logic.
  - The key difference between fuzzy logic and simply using a range of values is that fuzzy logic allows for **degrees of truth** and can handle uncertainty by applying rules that consider these degrees.
    - Uncertainty refers to situations where information is incomplete, ambiguous, or lacks binary clarity.
      - Example: If the temperature is 30°C, you might consider it 0.7 true (or "hot" to a 70% degree).
  - This enables **complex decision-making** based on subtle interpretations of data, unlike methods that use static ranges without situational analysis.

- **Application:**
  - **Decision-Making**: Supports complex decision-making in uncertain environments, like stock market analysis.
  - **Consumer Electronics**: Found in cameras for autofocus and in air conditioners for temperature control.
  - **Healthcare**: Utilized in diagnostic systems and treatment-level assessments.
  - **Environmental Control**: Helps manage water quality and air conditioning systems.

# Conditional Hierarchical Fuzzy PSO Disentangled VAE

## Fuzzy Logic

- **In VAE**
  - ➢ Fuzzy sampling adds **variability and randomness** to reduce overfitting.
    - ✓ Fuzzy sampling involves adding a degree of randomness to the generation of new data points from the latent space.
    - ✓ By introducing this variability, the model can generate outputs that are not just the most probable but also nearby possibilities.
    - ✓ This prevents the model from too closely fitting to the training data specifics, promoting better generalization.
    - ✓ Fuzzy sampling can help the model explore parts of the latent space that might not be frequently visited if sampling were strictly deterministic.
    - ✓ In generating data like human emotions or motions, slight randomness can lead to more lifelike and believable outputs.
    - ✓ Fuzzy logic operates through the **Fuzzy Sampling layer**, which introduces **controlled randomness** into the **generation of latent vectors** during the model's sampling process (in the training).
      - ❖ Controlled here means we control the amount of randomness.

# Conditional Hierarchical Fuzzy PSO Disentangled VAE

## Optimization

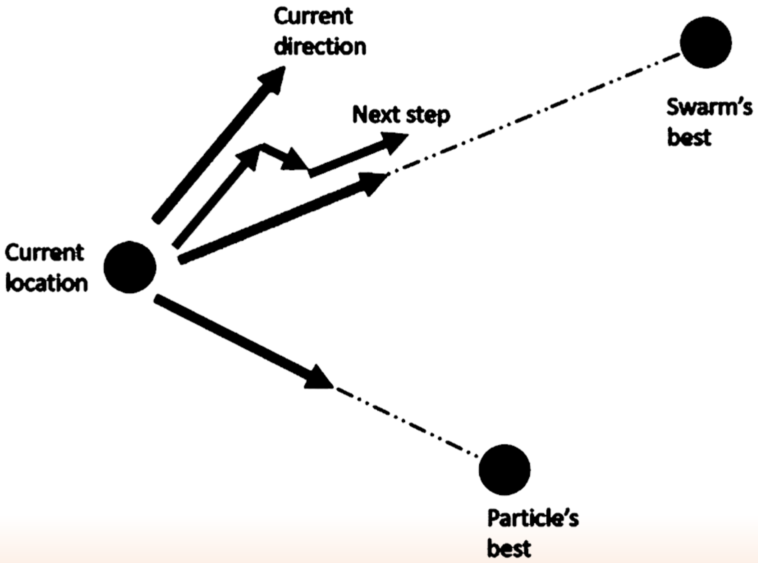
- **Definition:**
  - Optimization refers to the systematic process of improving an AI model's performance by iteratively adjusting its parameters or structure to minimize or maximize a specific objective function.
  - Optimization in machine learning refers to the process of adjusting a model's parameters during training to minimize a predefined loss function, which measures how well the model's predictions match the actual data.
  - Optimization across disciplines like machine learning and operations research aims to find the best solution by tuning parameters to minimize or maximize an objective function using strategies like optimization algorithms. This process involves exploring solution spaces to achieve optimal results within specific constraints or goals.

- **Application:**
  - **Machine Learning:** Fine-tuning model parameters to minimize errors and improve predictive accuracy.
  - **Engineering Design**: Enhancing product designs or processes to achieve optimal performance and cost-effectiveness.
  - **Energy Management**: Balancing production and distribution of energy resources to meet demand efficiently.
  - **Healthcare Logistics**: Scheduling staff and managing resources to improve patient care while reducing operational costs.
  - **Telecommunications**: Optimizing network routing and bandwidth allocation to enhance service quality and user experience and way more applications.

# Conditional Hierarchical Fuzzy PSO Disentangled VAE

## Optimization

- **Particle Swarm Optimization (PSO):**
  - PSO is a computational method used to optimize a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality.
  - It solves a problem by having a population of candidate solutions, here particles, and moving these particles around in the search space according to simple mathematical formulae over the particle's position and velocity.
  - Each particle's movement is influenced primarily by its local best-known position and is also guided toward the best-known positions in the search space, which are updated as better positions are found by other particles.
  - This is expected to move the swarm toward the best solutions.



# Conditional Hierarchical Fuzzy PSO Disentangled VAE

## Optimization

- **In VAE (PSO)**
  - ➢ It optimizes the exploration of the latent space, ensuring that the model can **find the most diverse and high-quality samples**.
  - ➢ PSO helps the model **explore a wide variety of potential solutions** in the latent space. This is crucial for generating diverse and high-quality samples that **capture the complexity of body motion** and emotional expressions.
  - ➢ By **evaluating different positions (samples)** in the latent space, PSO aims to find those that maximize or minimize a given objective function, which could be related to the **realism, diversity**, or emotional accuracy of the samples.
  - ➢ PSO does it by:
    - ✓ **Search Mechanism**: PSO uses particles, each representing a potential solution in the latent space, which move based on individual and neighbor experiences to find the best solutions.
    - ✓ **Velocity and Position Updates**: Particles dynamically update their positions in the search space by adjusting their velocity based on personal best positions and those of their neighbors.
    - ✓ **Objective Function Evaluation**: The quality of each position is assessed using an objective function, which checks the fidelity and emotional accuracy of the generated samples.
    - ✓ **Iterative Improvement**: Particles iteratively adjust their paths towards areas of the space with potentially better solutions, aiming for convergence to high-quality regions.
    - ✓ **Integration with Model Training**: PSO is integrated into the model’s training or post-processing phases to refine latent representations and enhance the generation of contextually appropriate outputs.

# Conditional Hierarchical Fuzzy PSO Disentangled VAE

## Optimization

- **In VAE (PSO)**
  - ➢ **Purpose**: The objective function is used to evaluate the quality of the solutions found by the particles in the latent space. Here, the objective function assesses aspects like the realism and emotional rhythms expression of the generated motion data.
  - ➢ **Criteria**: Specific criteria are the fidelity of motion to realistic human movements, the expressiveness and diversity of emotions represented, and the alignment of generated samples with given emotional intensities.

## Disentanglement

- ➢ It **separates the underlying factors of variation** in the data into **distinct parts of the latent space (features)**.
- ➢ This separation makes the **latent space more interpretable** and **easier to manipulate** for specific tasks.
- ➢ Aiming to **separate distinct features within different layers of the hierarchical structure** above.
- ➢ Allows for **precise control over specific attributes of generated outputs**, such as the type or intensity of an emotion being expressed through motion, without altering other aspects like the overall style or type of motion.
- ➢ By ensuring that the model learns to distinguish between and independently manipulate fundamental factors of variation, it can better generalize to new, unseen examples that vary along these dimensions.
  - ✓ I can adjust, for instance, the latent variables responsible for **'sadness'** to **vary the intensity of sadness** in a character's movements

# Conditional Hierarchical Fuzzy PSO Disentangled VAE

## Improvement Compared to VAE

- **Diverse Outputs**: Incorporates PSO to enhance the exploration of latent space, yielding a broader range of plausible motions and emotions.
- **Controlled Variability**: Utilizes fuzzy logic to add controlled randomness to the sampling process, increasing the realism and diversity of the generated samples.
- **Targeted Generation**: Uses **conditional** inputs to ensure that generated motions align specifically with desired emotional states and intensities.
- **Improved Disentanglement**: Separates different aspects of motion and emotion in the latent space, facilitating easier manipulation and better understanding of how specific factors influence the generated outcomes.
- **Hierarchical Detailing**: Structures the latent space hierarchically to capture both global and fine-grained details of motion, enhancing the model's ability to handle complex data interrelations effectively.

# Conditional Hierarchical Fuzzy PSO Disentangled VAE

## Implementation and Equations

**Input Data**

- Position $P_t$ : The position of each body joint at frame $t$.
- Rotation $R_t$ : The rotation (usually as Euler angles or quaternions) of each body joint at frame $t$.

The input data over $T$ frames can be written as:

$$X_t = \{P_t, R_t\},\ X = \{X_1, X_2, \dots, X_T\}$$

Where:

- $P_t$ is a vector of positions at time $t$.
- $R_t$ is a vector of rotations at time $t$.
- $T$ is the total number of frames.

Since the raw input data may have varying scales, we normalize the data to a suitable range for the neural network. This is usually done for each frame $t$ for both position $P_t$ and rotation $R_t$.

For each component $x \in \{P, R\}$, the normalization function is:

$$x_{\text{norm}} = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}}$$

**Data Normalization**

Where:

- $x_{\text{norm}}$ is the normalized value.
- $x_{\text{min}}$ and $x_{\text{max}}$ are the minimum and maximum values of the data in $X$ across all frames.

# Conditional Hierarchical Fuzzy PSO Disentangled VAE

## Implementation and Equations

In a Hierarchical VAE, the latent space is divided into top-level and bottom-level latent variables. This allows the model to capture both global motion features and fine-grained motion details.

### Top-Level Latent Space

The top-level captures coarse, global features (e.g., walking, running). The encoder first computes a Gaussian distribution from which the latent variables are sampled:

$$z_{\text{top}} \sim \mathcal{N}\left(\mu_{\text{top}}, \sigma_{\text{top}}^2\right)$$

Where:

- $\mu_{\text{top}}$ is the mean vector for the top-level latent space.
- $\sigma_{\text{top}}^2$ is the variance for the top-level latent space, typically represented as the log of the variance $\log \sigma_{\text{top}}^2$ to ensure positive variance.

The encoder computes:

$$\mu_{\text{top}} = f_{\text{enc,top}}(X_{\text{norm}})$$
$$\log \sigma_{\text{top}}^2 = g_{\text{enc, top}}(X_{\text{norm}})$$

Where $f_{\text{enc, top}}$ and $g_{\text{enc, top}}$ are neural networks that take the normalized input $X_{\text{norm}}$ and compute the mean and log-variance, respectively.

# Conditional Hierarchical Fuzzy PSO Disentangled VAE

**Implementation and Equations**

**Hierarchical Encoding**

## Bottom-Level Latent Space

The bottom-level captures finer details of the motion (e.g., subtle movements of limbs or joints). The bottom-level latent variables are conditioned on the top-level ones:

$$z_{\text{bottom}} \sim \mathcal{N}\left(\mu_{\text{bottom}}, \sigma^2_{\text{bottom}}\right)$$

Where:

- $\mu_{\text{bottom}}$ is the mean vector for the bottom-level latent space.
- $\sigma^2_{\text{bottom}}$ is the variance for the bottom-level latent space, typically represented as $\log \sigma^2_{\text{bottom}}$.

The mean and variance for the bottom-level latent space depend on the top-level latent space:

$$\mu_{\text{bottom}} = f_{\text{enc}, \text{bottom}}\left(z_{\text{top}}\right)$$
$$\log \sigma^2_{\text{bottom}} = g_{\text{enc}, \text{bottom}}\left(z_{\text{top}}\right)$$

Where $f_{\text{enc, bottom}}$ and $g_{\text{enc, bottom}}$ are neural networks conditioned on $z_{\text{top}}$.

# Conditional Hierarchical Fuzzy PSO Disentangled VAE

**Fuzzy Sampling**

## Implementation and Equations

In fuzzy sampling, we introduce controlled randomness into the latent space by sampling from a Gaussian distribution with a fuzziness factor $\eta$. This allows the model to explore diverse and varied outputs.

For both top-level and bottom-level latent spaces, we apply:

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

Where:

- $\mu$ is the mean (either $\mu_{\text{top}}$ or $\mu_{\text{bottom}}$ ).
- $\sigma$ is the standard deviation $\sigma = \exp(0.5 \cdot \log \sigma^2)$.
- $\epsilon \sim \mathcal{N}(0, I)$ is a random noise vector sampled from a standard normal distribution.
- $\eta$ is the fuzziness parameter that controls the degree of randomness.

Thus, the latent variables $z_{\text{top}}$ and $z_{\text{bottom}}$ are sampled as:

$$z_{\text{top}} = \mu_{\text{top}} + \eta_{\text{top}} \cdot \sigma_{\text{top}} \cdot \epsilon_{\text{top}}$$
$$z_{\text{bottom}} = \mu_{\text{bottom}} + \eta_{\text{bottom}} \cdot \sigma_{\text{bottom}} \cdot \epsilon_{\text{bottom}}$$

# Conditional Hierarchical Fuzzy PSO Disentangled VAE

## Disentanglement in Latent Space

### Implementation and Equations

Disentanglement ensures that different factors (e.g., emotion, intensity, general body motion) are independently represented in the latent space. This is achieved by penalizing correlations between the latent dimensions using a term called Total Correlation (TC).

The TC Loss is added to the overall VAE loss to encourage independence between latent variables:

$$\mathrm{TC} = \mathrm{KL}\left( q(z) \parallel \prod_{i} q(z_i) \right)$$

Where:

- $q(z)$ is the joint distribution of all latent variables.
- $q(z_i)$ is the marginal distribution of the $i$-th latent variable.

By penalizing the divergence between the joint distribution and the product of the marginals, we ensure that each latent dimension independently represents a different aspect of the data.

# Conditional Hierarchical Fuzzy PSO Disentangled VAE

## Implementation and Equations

## Particle Swarm Optimization (PSO)

PSO is used to search the latent space effectively and find the optimal latent vectors $z_{\text{top}}$ and $z_{\text{bottom}}$ that lead to the best output. In PSO, each particle represents a candidate solution in the latent space. The particles update their positions based on:

1. Velocity Update:

$$v_{i,t+1} = w \cdot v_{i,t} + c_1 \cdot r_1 \cdot \left(p_{\text{best },\text{i}} - x_{i,t}\right) + c_2 \cdot r_2 \cdot \left(g_{\text{best}} - x_{i,t}\right)$$

Where:

- $v_{i,t+1}$ is the updated velocity of particle $i$ at time $t+1$.
- $w$ is the inertia weight (controls exploration vs exploitation).
- $c_1, c_2$ are acceleration coefficients (control attraction to personal and global best positions).
- $r_1, r_2$ are random factors between 0 and 1.
- $p_{\text{best },\text{i}}$ is the personal best position of particle $i$.
- $g_{\text{best}}$ is the global best position.

2. Position Update:

$$x_{i,t+1} = x_{i,t} + v_{i,t+1}$$

Where $x_{i,t}$ is the current position of particle $i$ in the latent space.

# Conditional Hierarchical Fuzzy PSO Disentangled VAE

## Implementation and Equations

We now introduce conditional inputs such as emotion $e$ and intensity $I$ into the latent space to control the style and emotional expression of the output. The final latent vector $z_{\text{final}}$ is constructed by concatenating the optimized latent vectors with the emotion and intensity inputs:

$$z_{\text{final}} = \text{Concatenate}\left(z_{\text{top}}, z_{\text{bottom}}, e, I\right)$$

Where:

- $e$ is a vector representing emotion.
- $I$ is a scalar or vector representing the intensity of the emotion.

**Conditional Inputs (Emotion and Intensity)**

# Conditional Hierarchical Fuzzy PSO Disentangled VAE

Hierarchical Decoding

## Implementation and Equations

The decoder reconstructs the motion data hierarchically from $z_{\text{final}}$.

1. Top-Level Decoding: The decoder first reconstructs the coarse, global motion features from $z_{\text{top}}$

$$\hat{X}_{\text{top}} = \text{Decoder}_{\text{top}}\left(z_{\text{top}}\right)$$

2. Bottom-Level Decoding: The decoder then reconstructs the fine-grained details from $z_{\text{bottom}}$, conditioned on the global features $\hat{X}_{\text{top}}$ :

$$\hat{X}_{\text{bottom}} = \text{Decoder}_{\text{bottom}}\left(z_{\text{bottom}} \mid \hat{X}_{\text{top}}\right)$$

The final reconstructed motion $\hat{X}$ is the combination of $\hat{X}_{\text{top}}$ and $\hat{X}_{\text{bottom.}}$.

# Conditional Hierarchical Fuzzy PSO Disentangled VAE

## Implementation and Equations

The total loss for the model consists of:

1. Reconstruction Loss: Measures the difference between the original and reconstructed motion.

$$\text{Reconstruction Loss } = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x \mid z)]$$

2. KL Divergence: Ensures that the latent variables stay close to the prior Gaussian distribution.

$$\text{KL Divergence } = \text{KL}\big(q_\phi(z \mid x) \parallel p(z)\big)$$

3. Total Correlation (TC) Loss: Ensures disentanglement by penalizing correlations between the latent variables.

$$\text{Total Correlation (TC)} = \text{KL}\left(q(z) \parallel \prod_i q(z_i)\right)$$

The total loss is:

$$\mathcal{L} = \text{ Reconstruction Loss } + \beta \cdot \text{ KL Divergence } + \lambda \cdot \text{TC}$$

Where $\beta$ and $\lambda$ are hyperparameters controlling the importance of each term.

**Reconstruction and Disentanglement Loss**

# Conditional Hierarchical Fuzzy PSO Disentangled VAE

## Implementation and Equations

## Postprocessing and Output

Finally, the output motion data $\hat{X}$ is:

1. Denormalized to match the original scale:

$$\hat{x}_{\text{denorm}} = \hat{x}_{\text{norm}} \cdot (x_{\max} - x_{\min}) + x_{\min}$$

2. Smoothed using a Gaussian filter to ensure fluid motion transitions.
3. Saved in the BVH format with the reconstructed positions $\hat{P}$ and rotations $\hat{R}$ for each frame.