Data

Category	**Element**	**Description**	**Example**
🔷 **Temporal structure**	n_steps_input	Length of the sequence used as input (temporal history).	81 steps
	n_steps_output	Subsequent steps used as output (future prediction).	
	Sliding windows	Samples are generated by sliding through the temporal sequence with a 1-frame step.	If N_t=10, we obtain 81-10+1 = 72 samples
	Sample form	Data Block (n_{input}, H, W, C).	81, 256, 256, 11

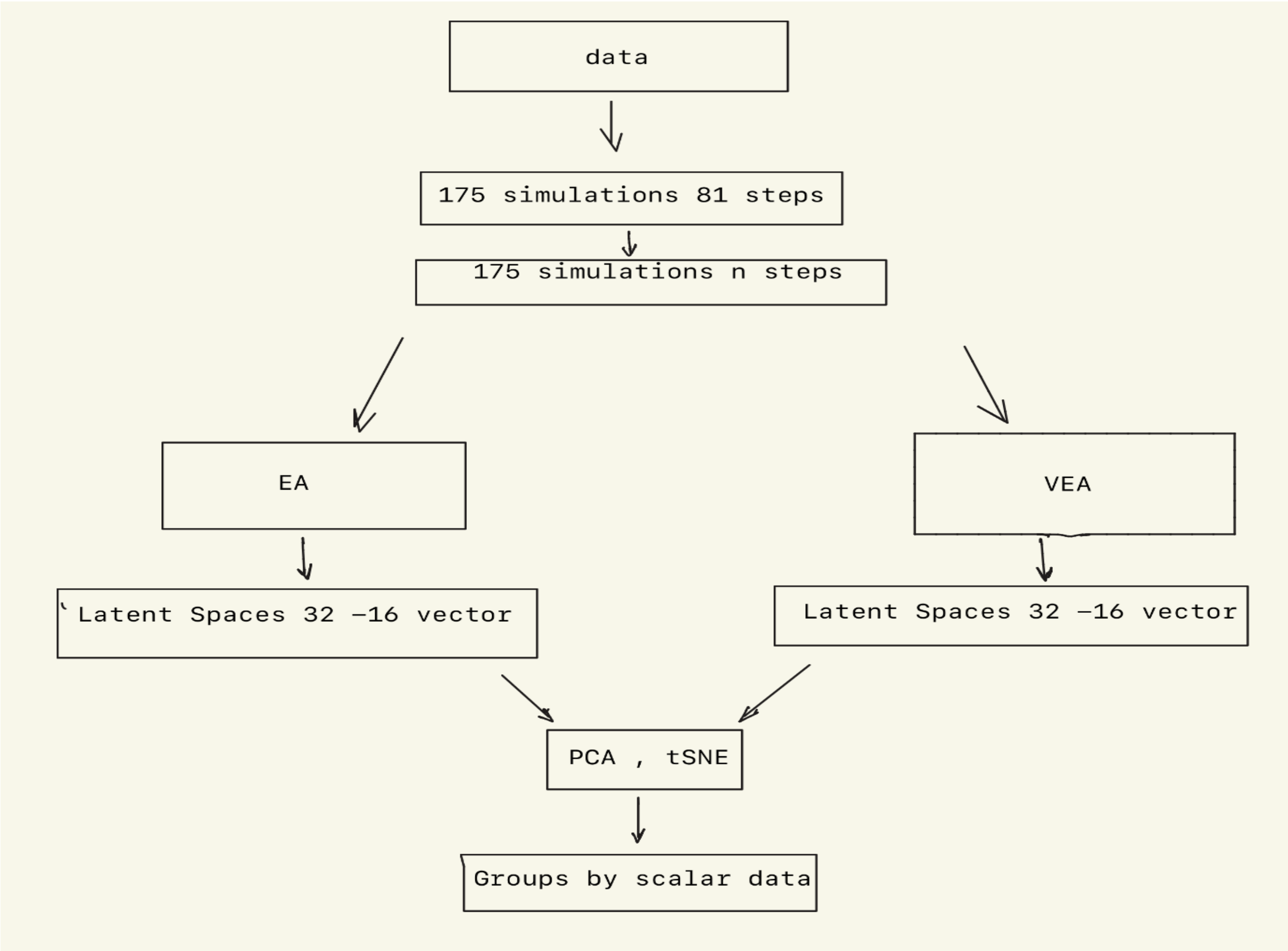
Category	Element	Description	Example
◆ **Physical parameters**	α (activity)	Intensidad de las fuerzas activas del sistema.	Negativo: pushers Positivo: pullers
		β (alineamiento)	Tendencia de las partículas a alinearse con su entorno.
	ζ (acoplamiento flujo–orientación)	Interacción entre el flujo y la orientación de las partículas.	ζ alto → vórtices grandes
◆ **Simulated Data**	Chanels (11)	Magnitudes físicas calculadas en cada punto del campo.	

Chanel	Physical meaning
c	Density or local concentration
u_x, u_y	Velocity field components
D_xx, D_xy, D_yy	Deformation tensor components
S_xxxx, S_xxxy, S_xxyy, S_xyyy, S_yyyy	Fourth-order moments (anisotropy and spatial correlations)

Project: Use autoencoders and variational autoencoders to learn low-dimensional latent representations that capture key features and phase transitions in active matter systems.

By analyzing the latent space with PCA and t-SNE, reveals transitions between dynamic regimes, meaningful clustering of physical phases, and interpretable organization related to the system’s underlying parameters.

Identify low-dimensional latent structures revealing active-matter dynamic phases.



Algorithm 1 Resumen del pipeline cVAE para `active_matter`

Input: Dataset $\mathcal{D}$ (`active_matter`), parámetros (α, β, ζ)
Output: Latentes $\mathbf{Z}_\mu$, proyecciones PCA/t-SNE

function LoadDataset
  Cargar $\mathcal{D}$ con $n\_steps\_input = 81$, normalización z-score
end function
function Subsample($\mathcal{D}$)
  Seleccionar frames $\{0, step, 2step, \ldots\}$ y condiciones (α, ζ)
  Normalizar condiciones y asignar `class_id`
end function
function TrainCVAE(dataset)
  for epoch $= 1$ to N **do**
    Calcular β (KL annealing)
    $(x, cond) \to (\hat{x}, \mu, \log \sigma^2, z)$
    $\mathcal{L} = MSE(x, \hat{x}) + \beta \cdot KL(\mu, \log \sigma^2)$
    Retropropagar y actualizar pesos
  end for
end function
function AnalyzeLatent(model)
  Obtener μ para todas las muestras $\to \mathbf{Z}_\mu$
  Aplicar PCA y t-SNE, graficar por clase (α, ζ)
  Guardar $\{\mathbf{Z}_\mu, Z_{PCA}, Z_{tSNE}\}$
end function

$\mathcal{D} \leftarrow$ LoadDataset
`wrap` $\leftarrow$ Subsample($\mathcal{D}$)
TrainCVAE(`wrap`)
AnalyzeLatent(`model`)

Architecture

Part	**What it does**	**Shape / Size**
Input	A short movie (sequence) of physical fields	(11 variables, 9 time steps, 256×256 pixels)
Encoder	Compresses the movie into a small numeric code	Becomes a 256-number feature vector
Condition	Adds the physical parameters (α, ζ)	2 numbers
Latent space	Final compact representation that summarizes the system	**32 numbers (latent size = 32)**
Decoder	Uses the 32 numbers and (α, ζ) to rebuild the movie	Output same size as input
Loss function	Compares input and output to learn good reconstructions	Uses error + regularization