Análisis Exploratorio del Dataset *ActiveMatter* (The Well)

Equipo 2 – Project 02: Revealing Hidden Order

22 de octubre de 2025

1. Introducción

El dataset *ActiveMatter*, parte de la colección **The Well** (`arXiv:2412.00568`), contiene simulaciones de materia activa que presentan transiciones de fase desde estados isotrópicos hacia estados con alineamiento colectivo (nemático o polar). Este conjunto de datos es ideal para estudiar estructuras latentes de baja dimensión mediante autoencoders o VAEs, en el contexto del proyecto *Revealing Hidden Order*.

A continuación se detallan los pasos para instalar las dependencias, descargar el conjunto de datos, cargarlo en Python y realizar un primer análisis exploratorio (EDA).

—

2. Instalación de librerías necesarias

Antes de comenzar, se deben instalar las librerías requeridas para la descarga y el análisis del conjunto de datos:

```
# --- instalar el loader oficial ---
pip install the_well

# --- librer as para an lisis y visualizaci n ---
pip install numpy matplotlib h5py torch tqdm scikit-learn
```

Estas librerías permiten descargar los datos, manipularlos en memoria y generar visualizaciones básicas.

—

3. Carga del dataset *ActiveMatter*

El paquete `the_well` proporciona una interfaz directa para acceder a los datos desde Hugging Face o descargarlos localmente.

Opción A: descarga local

```
the-well-download --base-path ./data \
                  --dataset ActiveMatter \
                  --split train
```

Esto descargará los archivos HDF5 en la ruta `./data/ActiveMatter/train/`.

Opción B: acceso por streaming (sin descarga completa)

```python
from the_well.data import WellDataset

trainset = WellDataset(
    well_base_path="hf://datasets/polymathic-ai/",
    well_dataset_name="ActiveMatter",
    well_split_name="train"
)
```

Esta opción permite leer los datos directamente desde la nube sin almacenarlos completamente en disco.

—

4. Exploración inicial de la estructura

```python
import numpy as np
from torch.utils.data import DataLoader

# Crear DataLoader
loader = DataLoader(trainset, batch_size=1, shuffle=True)

# Tomar una muestra
batch = next(iter(loader))

print(type(batch))
print(batch.shape)
```

La estructura típica es de la forma $(1, 81, 256, 256, n_{\text{fields}})$, donde los 81 pasos corresponden al tiempo y la malla es de 256×256.

—

5. Identificación de los campos físicos

Los campos registrados en el dataset son los siguientes:

Campo	Tipo	Descripción
c	Escalar	Concentración
U	Vectorial	Velocidad (componentes u_x, u_y)
Q	Tensorial	Tensor de orientación
S	Tensorial	Tasa de deformación

Cuadro 1: Campos físicos disponibles en el conjunto *ActiveMatter*.

Ejemplo de separación de los campos en Python:

```python
c_field   = batch[0, :, :, :, 0]  # concentraci n
u_x_field = batch[0, :, :, :, 1]  # velocidad x
u_y_field = batch[0, :, :, :, 2]  # velocidad y
```

—

6. Visualización de un instante

Para graficar una instantánea del campo de concentración:

```
import matplotlib.pyplot as plt

t = 0  # primer paso temporal
plt.figure(figsize=(6,5))
plt.imshow(c_field[t], cmap='viridis')
plt.title(f"Concentraci n - paso {t}")
plt.colorbar()
plt.show()
```

Visualización del campo de velocidad (submuestreado para claridad):

```
plt.figure(figsize=(6,6))
plt.quiver(u_x_field[t][::8, ::8], u_y_field[t][::8, ::8])
plt.title("Campo de velocidad (submuestreado)")
plt.show()
```

Estas visualizaciones permiten identificar la evolución de estructuras coherentes y patrones de alineamiento en el sistema de materia activa.

—