Simulating Aerosol Chemistry with Graph Neural Networks

Fabiana Ferracina

April 25, 2024

WASHINGTON STATE UNIVERSITY

Motivation for Studying Aerosol Chemistry

Why should we care about aerosol dynamics?

Motivation for Studying Aerosol Chemistry

Why should we care about aerosol dynamics?

- Radioactive forcing

Motivation for Studying Aerosol Chemistry

Why should we care about aerosol dynamics?

- Radioactive forcing
- Cloud formation

Motivation for Studying Aerosol Chemistry

Why should we care about aerosol dynamics?

- Radioactive forcing
- Cloud formation
- Regional variations

Motivation for Studying Aerosol Chemistry

Why should we care about aerosol dynamics?

- Radioactive forcing
- Cloud formation
- Regional variations
- Feedback mechanisms

Motivation for Simulation

Why use simulation?

Motivation for Simulation

Why use simulation?

- Aerosols are difficult to measure

Motivation for Simulation

Why use simulation?

- Aerosols are difficult to measure
- Study behavior and interactions for different scenarios

Motivation for Simulation

Why use simulation?

- Aerosols are difficult to measure
- Study behavior and interactions for different scenarios

What about existing simulators?

Motivation for Simulation

Why use simulation?

- Aerosols are difficult to measure
- Study behavior and interactions for different scenarios

What about existing simulators?

- MOSAIC: Model for Simulating Aerosol Interactions and Chemistry

Motivation for Simulation

Why use simulation?

- Aerosols are difficult to measure
- Study behavior and interactions for different scenarios

What about existing simulators?

- MOSAIC: Model for Simulating Aerosol Interactions and Chemistry
- Computational complexity associated with the detailed representations of aerosol

Motivation for Simulation

Why use simulation?

- Aerosols are difficult to measure
- Study behavior and interactions for different scenarios

What about existing simulators?

- MOSAIC: Model for Simulating Aerosol Interactions and Chemistry
- Computational complexity associated with the detailed representations of aerosol
- Closed-source

Can we do better?

Maybe

Can we do better?

Maybe

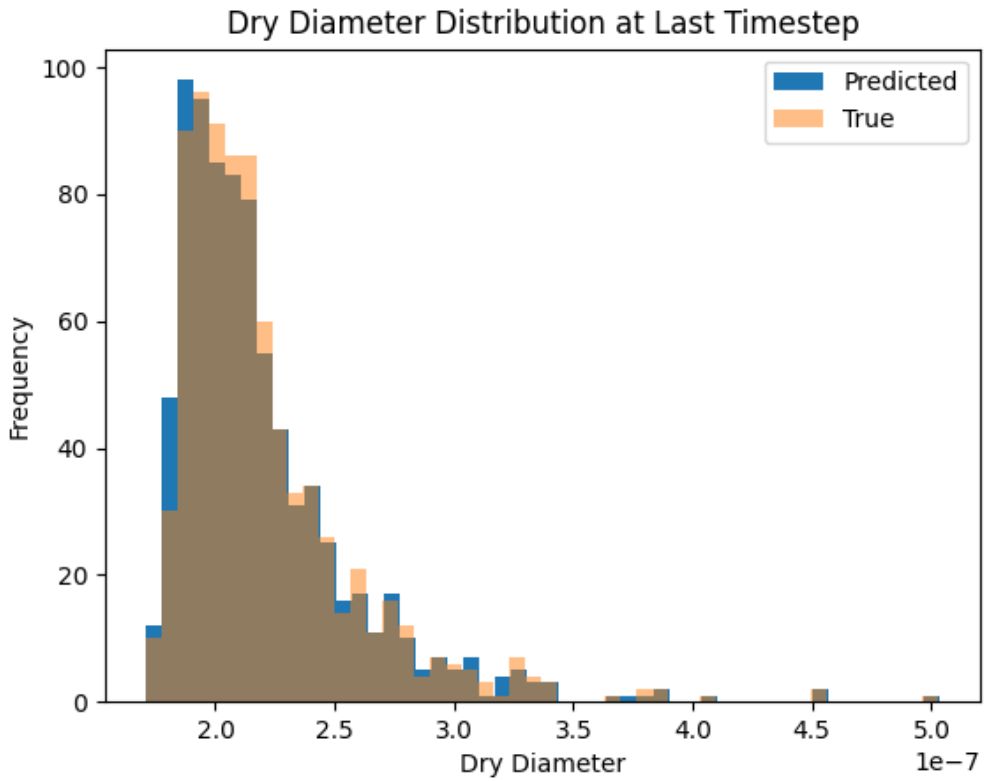


Figure: Results from Chem GNS on a simple particle-gas system. 142 timesteps, 808 particles, 3 time changing features, 3 time fixed particle properties, 2 chemical types. MSE $\approx 8.126 \times 10^{-7}$ in rate of change rate. Time to predict ≈ 0.4 seconds

Graph Neural Networks (Scarselli et al. 2008)

Let $G = (V, E)$. Each node $v_i \in V$ has an associated feature vector x_i, and each edge $(v_i, v_j) \in E$ has an associated weight w_{ij}. At each message passing step t, the state of each node v_i is updated based on information from its neighbors.

$$m_i^{(t)} = \sum_{j \in N(i)} \text{MessageFunction}(x_i, x_j, w_{ij}) \tag{1}$$

$$h_i^{(t+1)} = \text{UpdateFunction}(h_i^{(t)}, m_i^{(t)}) \tag{2}$$

GNN Schematics

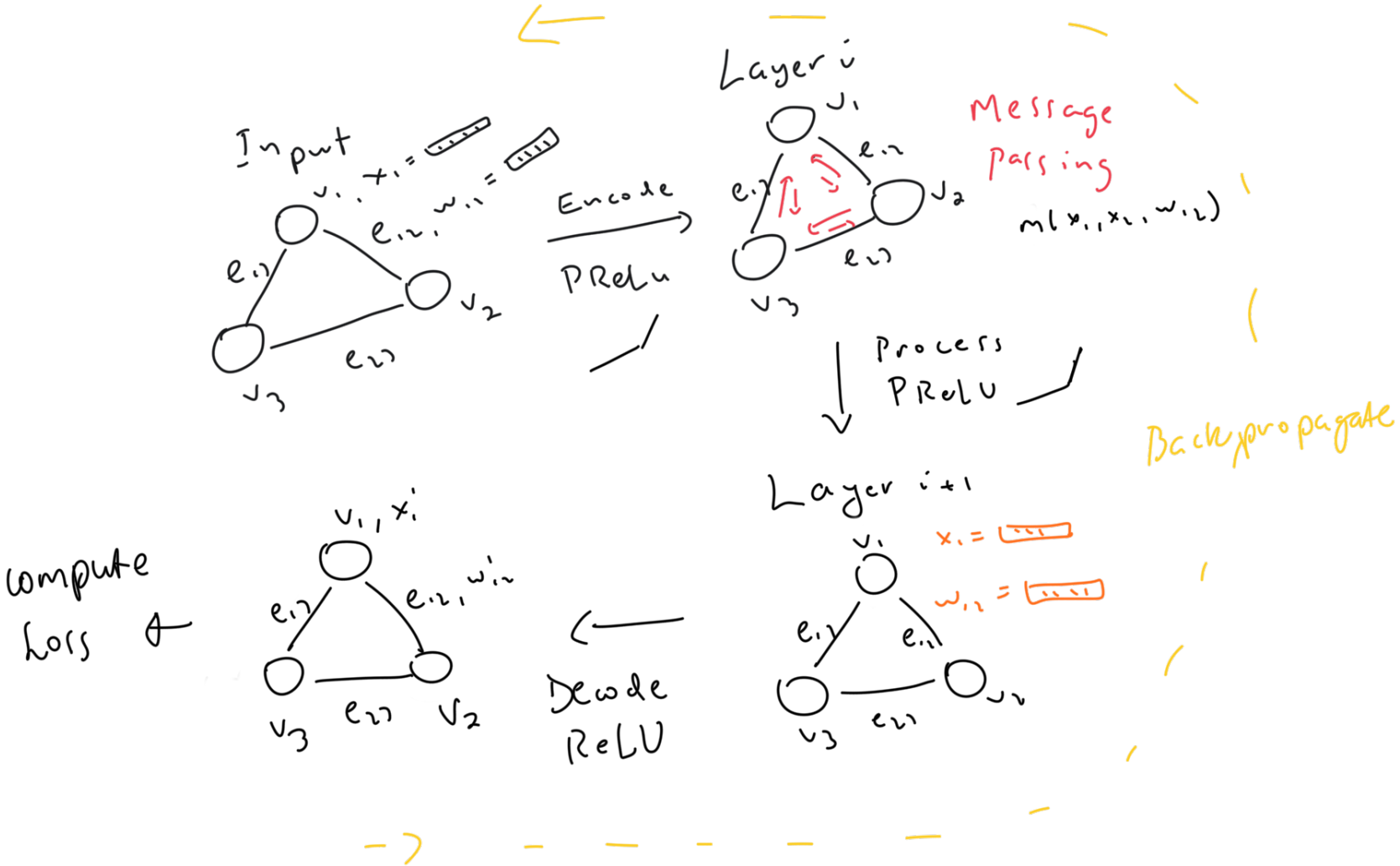


Figure: GNN scheme for Chem GNS

Parametric Rectified Linear Unit by He et al. 2015

Definition:

$$\text{PReLU}(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{otherwise} \end{cases}$$

Parametric Rectified Linear Unit by He et al. 2015

Definition:

$$\text{PReLU}(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{otherwise} \end{cases}$$

Key Points:

- PReLU introduces a learnable parameter α to capture negative slope.

Parametric Rectified Linear Unit by He et al. 2015

Definition:

$$\text{PReLU}(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{otherwise} \end{cases}$$

Key Points:

- PReLU introduces a learnable parameter α to capture negative slope.
- When $\alpha = 0$, PReLU reduces to the ReLU activation function.

Parametric Rectified Linear Unit by He et al. 2015

Definition:

$$\text{PReLU}(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{otherwise} \end{cases}$$

Key Points:

- PReLU introduces a learnable parameter α to capture negative slope.
- When $\alpha = 0$, PReLU reduces to the ReLU activation function.
- PReLU allows for learning the optimal slope for negative inputs at each channel, improving model performance.

Graph Network Simulator (GNS) by Sanchez-Gonzalez et al. 2020

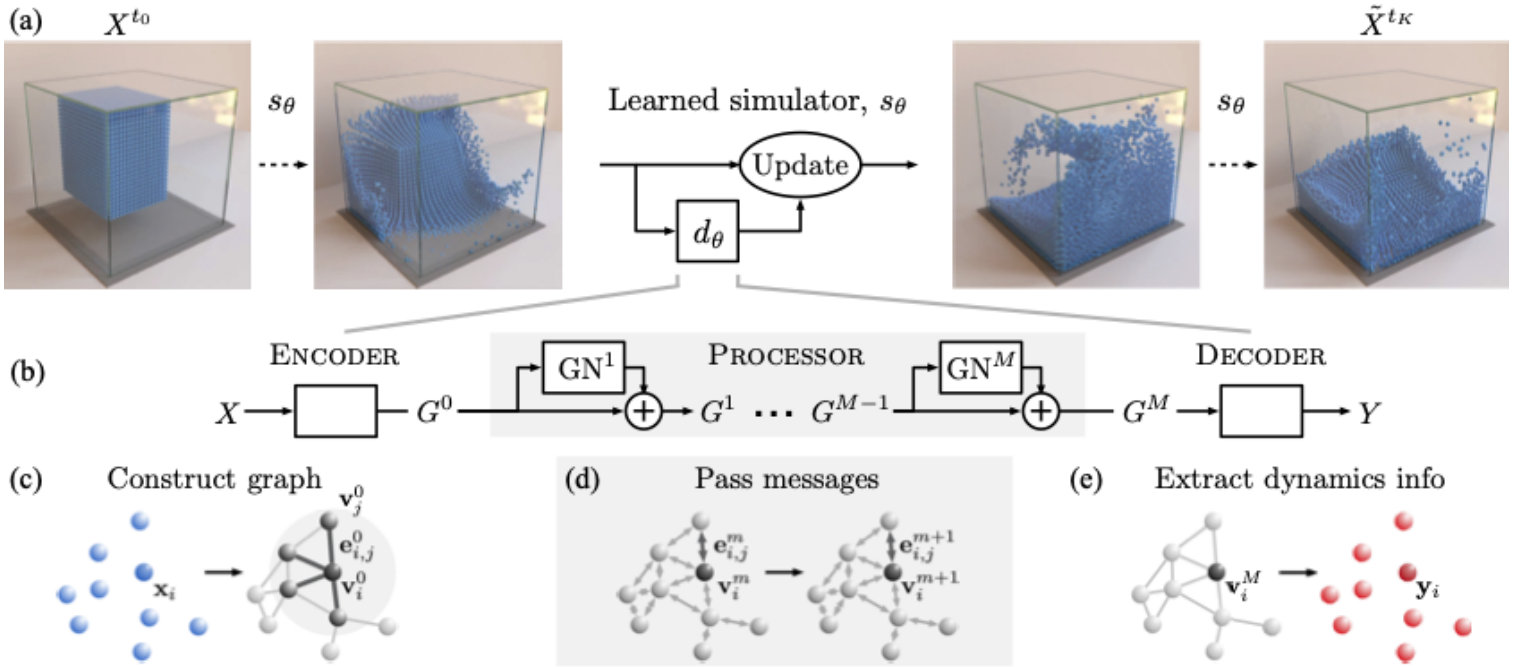
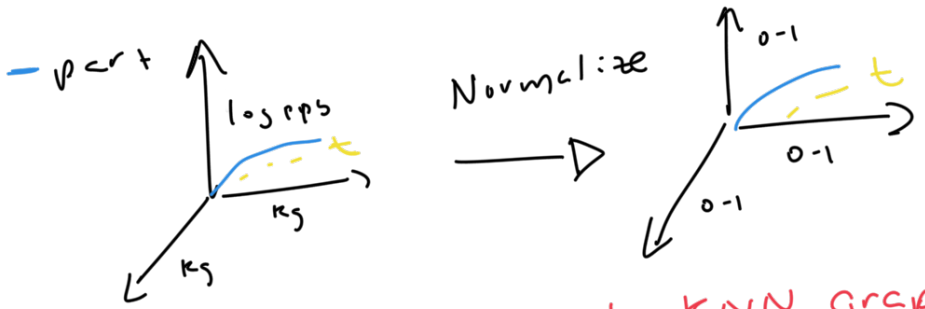


Figure: d_θ uses an "encode-process-decode" scheme, which computes dynamics information, Y, from input state, X

Chem GNS

Input at t per particle



Construct KNN graph k=2

node ◯ → "particle"

node feature → change from t-1 → t, differences from know minimax, node type, normalized material properties

edge → "displacements" and "distance" from neighbors Euclidean!

→ learned GNN to get input at t+1

Results from GNS Simulations: Water

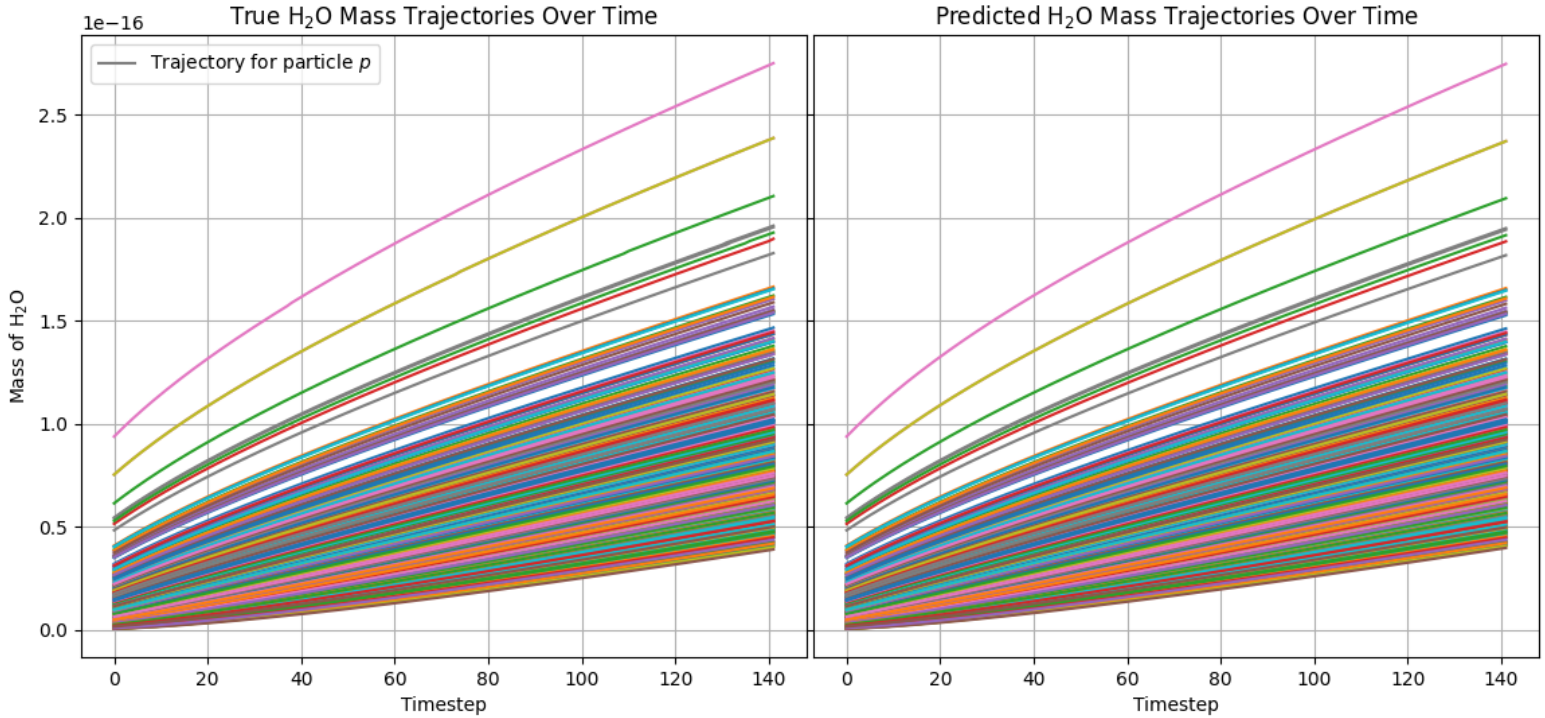


Figure: Results from Chem GNS on a simple particle-gas system. 142 timesteps, 808 particles, 3 time changing features, 3 time fixed particle properties, 2 chemical types. NMAE ≈ 0.004. Time to predict ≈ 0.4 seconds

Results from GNS Simulations: Sulfate

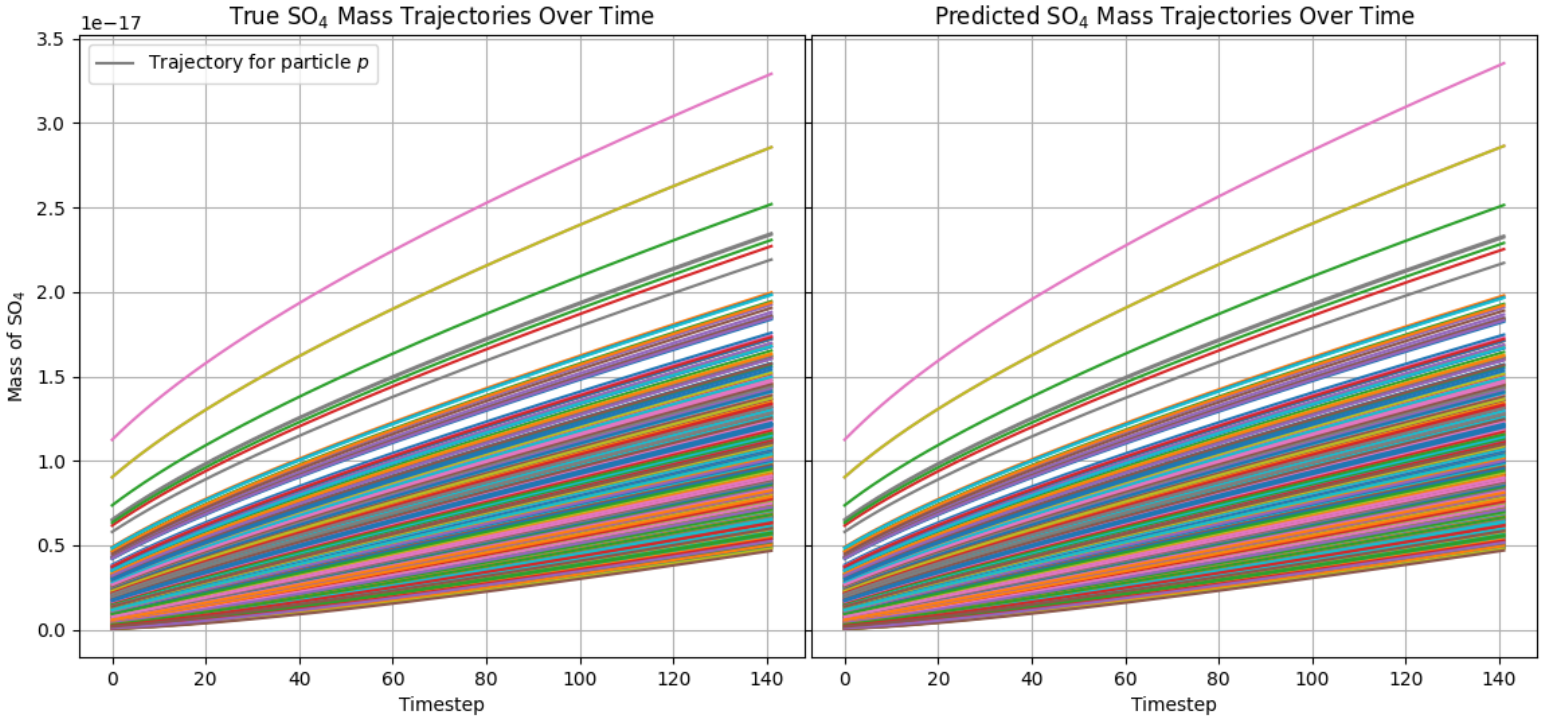


Figure: Results from Chem GNS on a simple particle-gas system. 142 timesteps, 808 particles, 3 time changing features, 3 time fixed particle properties, 2 chemical types. NMAE ≈ 0.007. Time to predict ≈ 0.4 seconds

Results from GNS Simulations: Sulfuric Acid

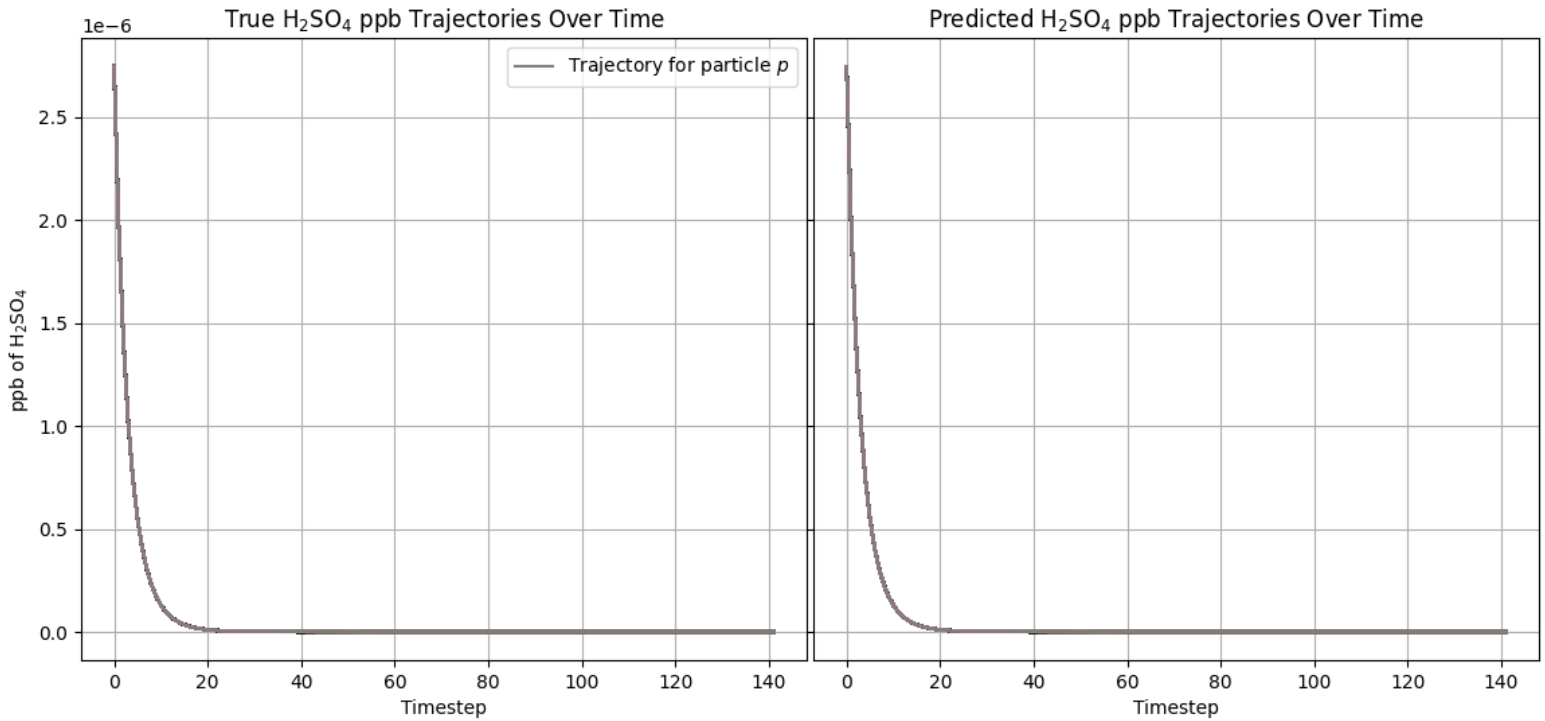


Figure: Results from Chem GNS on a simple particle-gas system. 142 timesteps, 808 particles, 3 time changing features, 3 time fixed particle properties, 2 chemical types. NMAE ≈ 0.011. Time to predict ≈ 0.4 seconds

Training on Different Examples

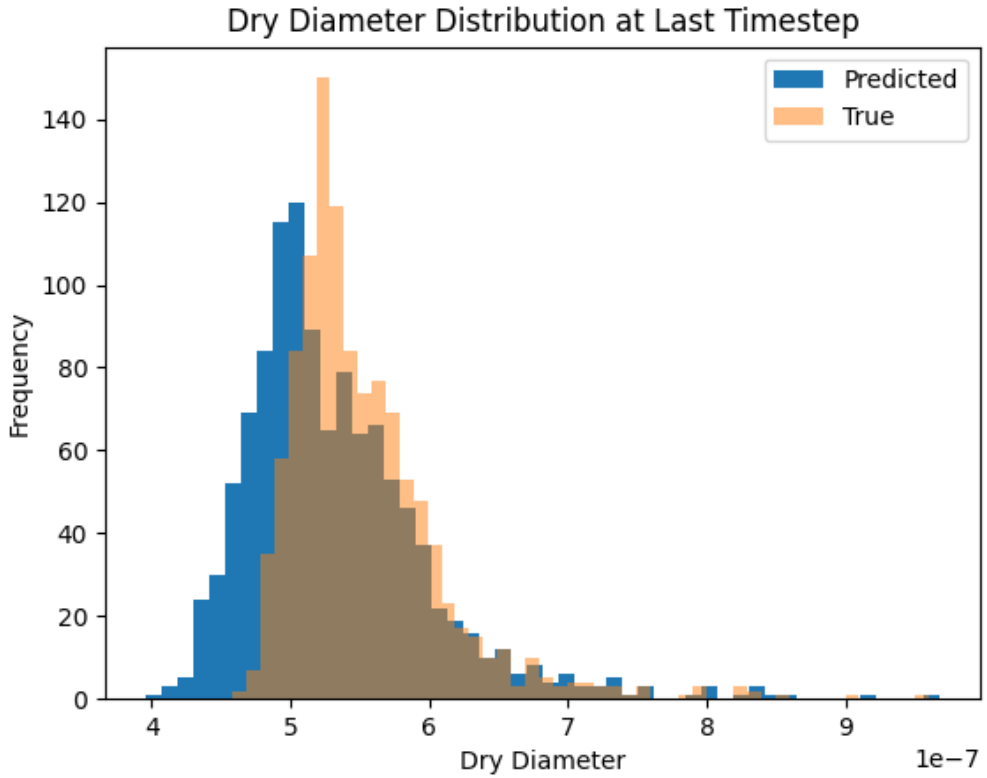


Figure: Results from Chem GNS trained on 9 different simple particle-gas systems. 142 timesteps, 1132 particles, 3 time changing features, 3 time fixed particle properties, 2 chemical types. MSE ≈ 0.03 in rate of change rate. Time to predict ≈ 0.03 seconds

More Results from GNS Simulations: Water

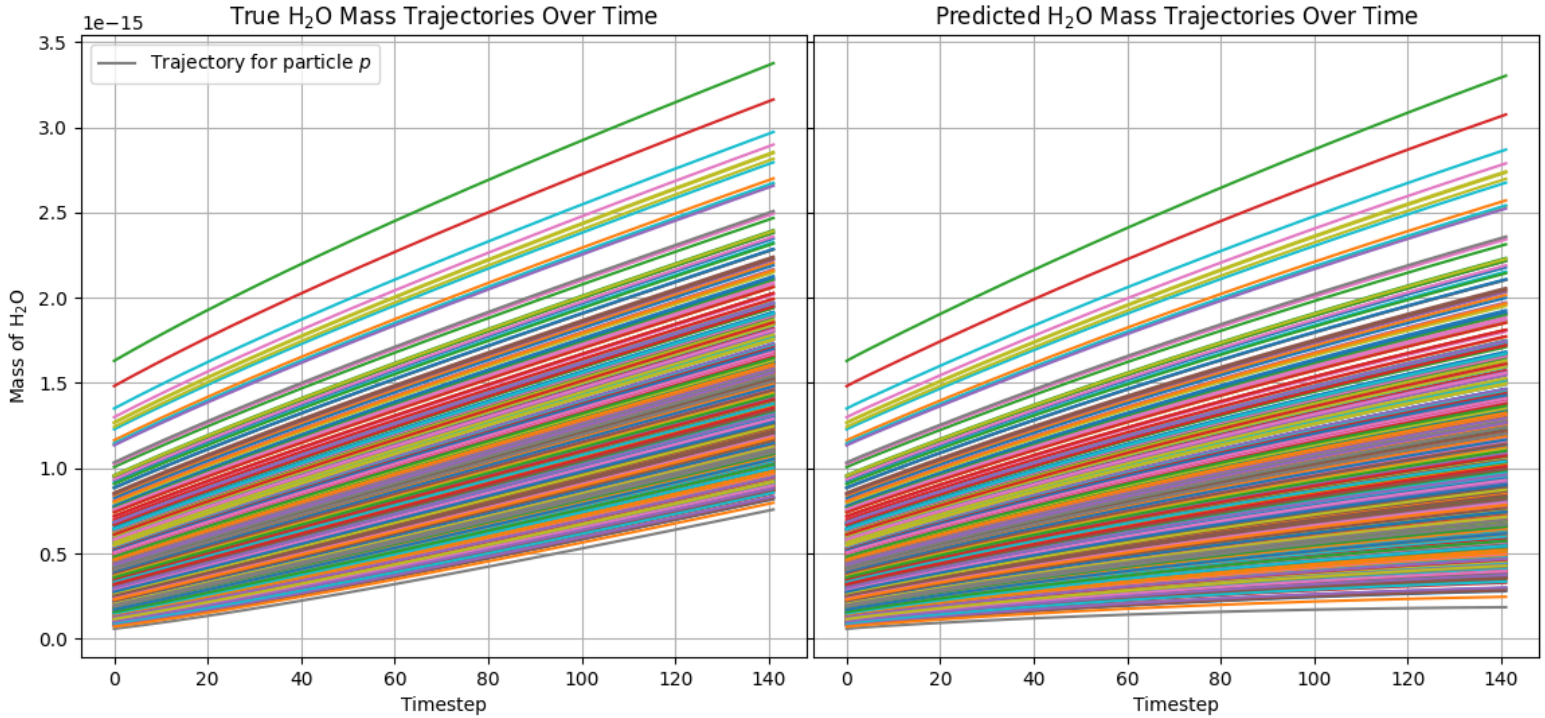


Figure: Results from Chem GNS trained on 9 different simple particle-gas systems. 142 timesteps, 1132 particles, 3 time changing features, 3 time fixed particle properties, 2 chemical types. NMAE ≈ 0.19. Time to predict ≈ 0.4 seconds

More Results from GNS Simulations: Sulfate

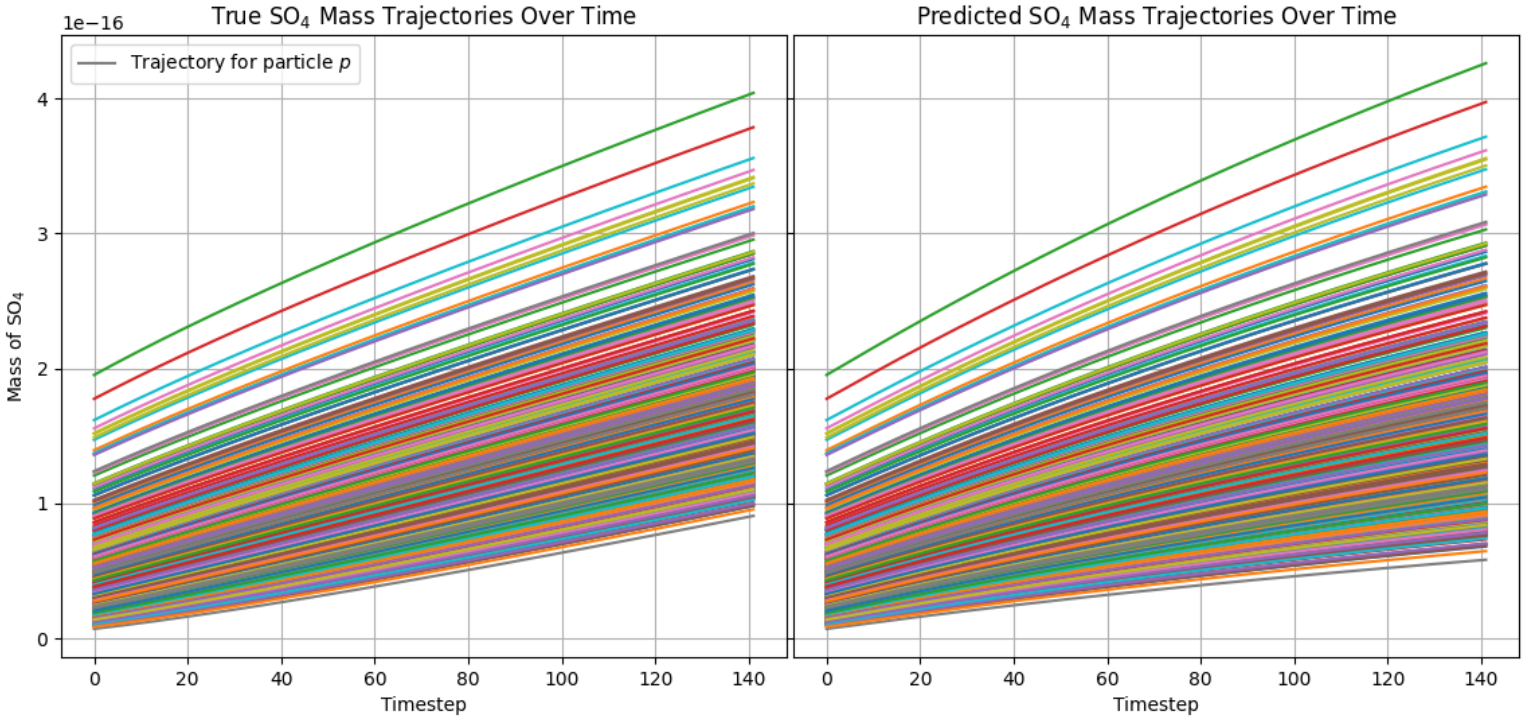


Figure: Results from Chem GNS trained on 9 different simple particle-gas systems. 142 timesteps, 1132 particles, 3 time changing features, 3 time fixed particle properties, 2 chemical types. NMAE ≈ 0.05. Time to predict ≈ 0.4 seconds

More Results from GNS Simulations: Sulfuric Acid

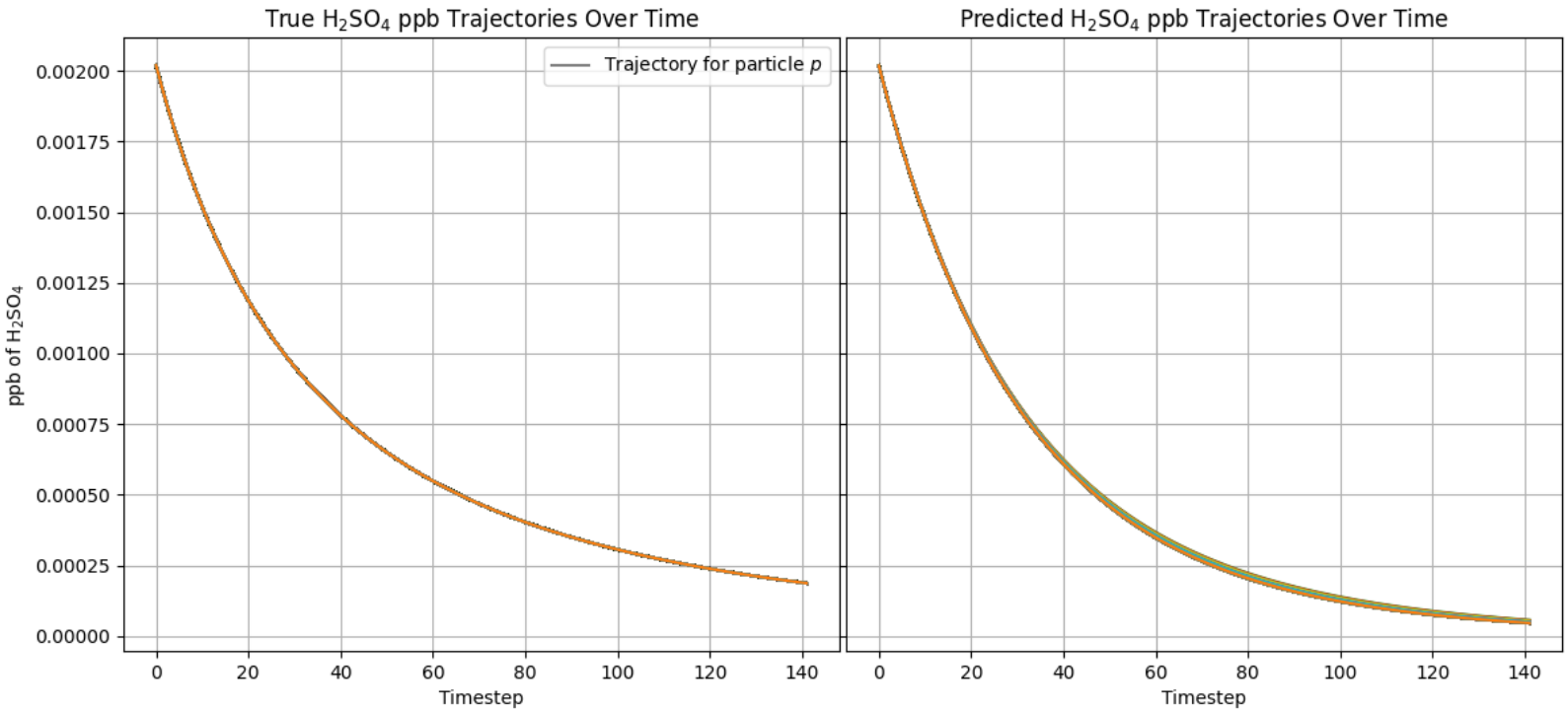


Figure: Results from Chem GNS trained on 9 different simple particle-gas systems. 142 timesteps, 1132 particles, 3 time changing features, 3 time fixed particle properties, 2 chemical types. NMAE ≈ 0.40. Time to predict ≈ 0.4 seconds

Future Directions

- Performance issues in high dimensions may require more careful selection of functions

Future Directions

- Performance issues in high dimensions may require more careful selection of functions
- Particle-particle interaction

Future Directions

- Performance issues in high dimensions may require more careful selection of functions
- Particle-particle interaction
- Global nodes with environmental information

Conclusion

- Originally proposed for physics applications, GNS can work in chemical domain

Conclusion

- Originally proposed for physics applications, GNS can work in chemical domain
- Making it bigger and better will require thinking outside Euclidean space

Conclusion

- Originally proposed for physics applications, GNS can work in chemical domain
- Making it bigger and better will require thinking outside Euclidean space
- If fasts speeds can be maintained, inclusion in climate models would be significant

References

- He, Kaiming et al. (2015). "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". In: *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034.
- Sanchez-Gonzalez, Alvaro et al. (2020). "Learning to simulate complex physics with graph networks". In: *International conference on machine learning*. PMLR, pp. 8459–8468.
- Scarselli, Franco et al. (2008). "The graph neural network model". In: *IEEE transactions on neural networks* 20.1, pp. 61–80.