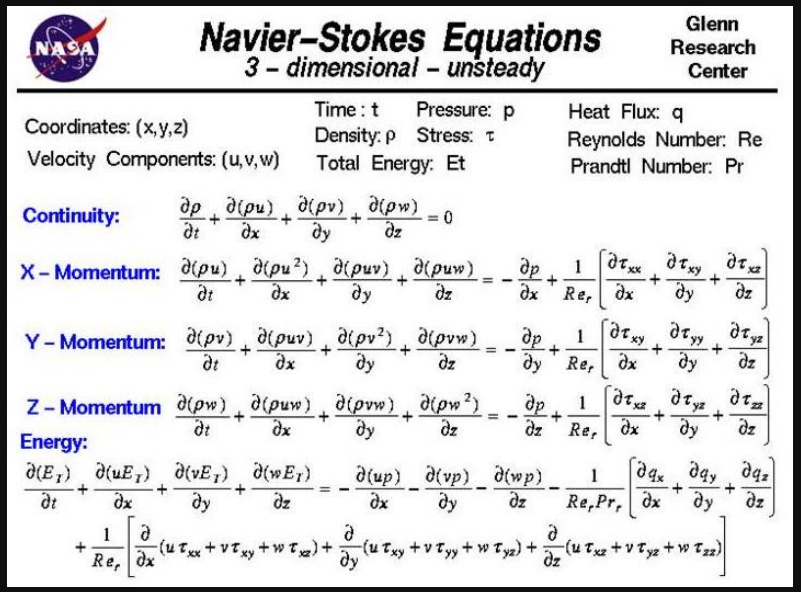
Christopher Curtis, Brent Garey, Liam Langert, Justin Feldman
CS7150: Deep Learning, Steve Schmidt
Northeastern University

CFD Surrogate Transformer

Multivariate Regression in Lower Dimensional Representations with Self-Attention, Scientific-Priors, and Physics Informed Loss

Domain Context: CFD

- What is Computational Fluid Dynamics?
  - Predicting fluid flow with Newton's Laws
  - Navier-Stokes Equations:



Navier–Stokes Equations
3 – dimensional – unsteady
Glenn Research Center

Coordinates: (x,y,z)
Velocity Components: (u,v,w)
Time : t Pressure: p
Density: ρ Stress: τ
Total Energy: Et
Heat Flux: q
Reynolds Number: Re
Prandtl Number: Pr

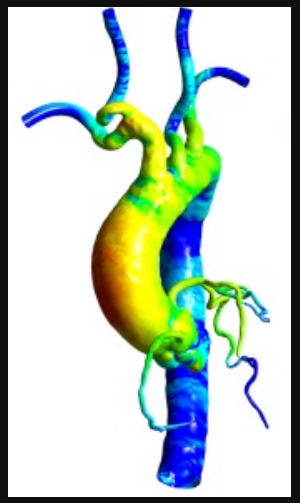
Continuity:
$$\frac{\partial \rho}{\partial t}+\frac{\partial(\rho u)}{\partial x}+\frac{\partial(\rho v)}{\partial y}+\frac{\partial(\rho w)}{\partial z}=0$$

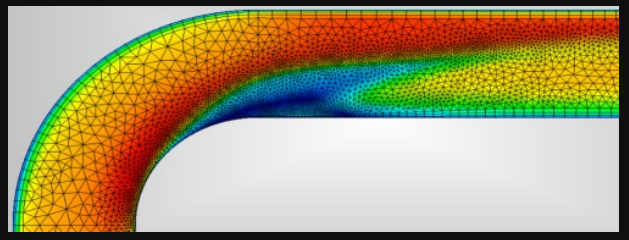
X – Momentum:
$$\frac{\partial(\rho u)}{\partial t}+\frac{\partial(\rho u^2)}{\partial x}+\frac{\partial(\rho uv)}{\partial y}+\frac{\partial(\rho uw)}{\partial z}=-\frac{\partial p}{\partial x}+\frac{1}{Re_r}\left[\frac{\partial \tau_{xx}}{\partial x}+\frac{\partial \tau_{xy}}{\partial y}+\frac{\partial \tau_{xz}}{\partial z}\right]$$

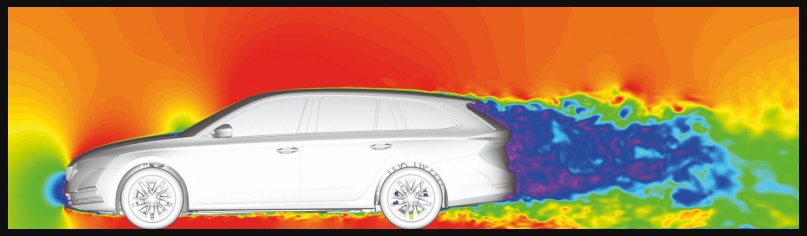
Y – Momentum:
$$\frac{\partial(\rho v)}{\partial t}+\frac{\partial(\rho uv)}{\partial x}+\frac{\partial(\rho v^2)}{\partial y}+\frac{\partial(\rho vw)}{\partial z}=-\frac{\partial p}{\partial y}+\frac{1}{Re_r}\left[\frac{\partial \tau_{xy}}{\partial x}+\frac{\partial \tau_{yy}}{\partial y}+\frac{\partial \tau_{yz}}{\partial z}\right]$$

Z – Momentum
$$\frac{\partial(\rho w)}{\partial t}+\frac{\partial(\rho uw)}{\partial x}+\frac{\partial(\rho vw)}{\partial y}+\frac{\partial(\rho w^2)}{\partial z}=-\frac{\partial p}{\partial z}+\frac{1}{Re_r}\left[\frac{\partial \tau_{xz}}{\partial x}+\frac{\partial \tau_{yz}}{\partial y}+\frac{\partial \tau_{zz}}{\partial z}\right]$$

Energy:
$$\frac{\partial(E_T)}{\partial t}+\frac{\partial(uE_T)}{\partial x}+\frac{\partial(vE_T)}{\partial y}+\frac{\partial(wE_T)}{\partial z}=-\frac{\partial(up)}{\partial x}-\frac{\partial(vp)}{\partial y}-\frac{\partial(wp)}{\partial z}-\frac{1}{Re_rPr_r}\left[\frac{\partial q_x}{\partial x}+\frac{\partial q_y}{\partial y}+\frac{\partial q_z}{\partial z}\right]$$
$$+\frac{1}{Re_r}\left[\frac{\partial}{\partial x}(u\tau_{xx}+v\tau_{xy}+w\tau_{xz})+\frac{\partial}{\partial y}(u\tau_{xy}+v\tau_{yy}+w\tau_{yz})+\frac{\partial}{\partial z}(u\tau_{xz}+v\tau_{yz}+w\tau_{zz})\right]$$

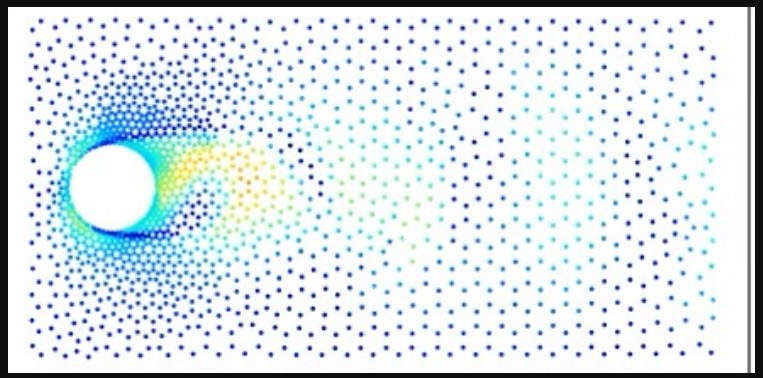






Problem Statement

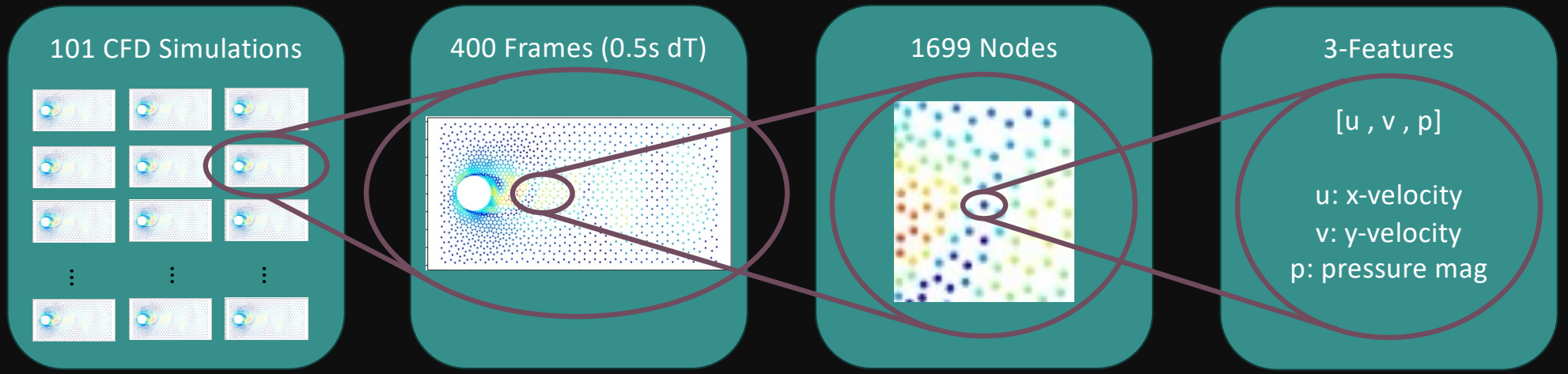
- Why does CFD need a surrogate and… what is a surrogate?
  - Mesh generation → extreme resolution requirements
  - Expensive → massive memory requirements
  - Physics formulas → highly entropic behaviors



- Our Solution: Transformer Model
  - Next-frame regression predictions of fluid features
  - Expand to auto-regressive simulation
  - Significantly lower dimensional representation of input

- Our Challenges:
  - Dataset requires high-dimensionality to capture states
  - Underlying mechanics are non-linear and interdependent
  - Efficient computation is high-stakes, and time-sensitive

Dataset: NVIDIA Modulus Cylinder Flow

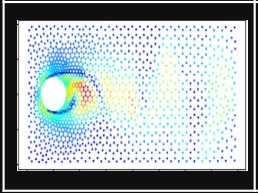
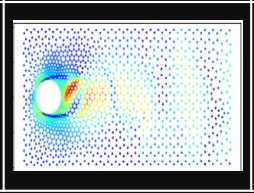
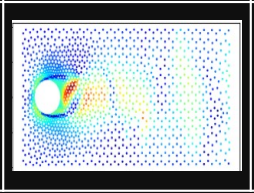
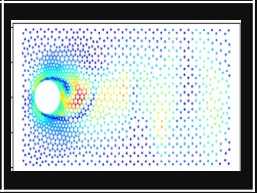


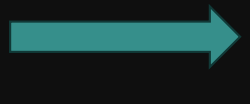
- Next-frame prediction → [u,v,p] x 1699 = 5097 Features
- Different Reynolds Number per simulation!!!
  - Reynolds Number ⇔ Viscosity! Think water vs honey. More to come on this…

Tokenization Approach

- Token, 'n', by time-step frame
  - Token n = [u,v,p] x 1699 nodes = 5097 features
  - Very large input dimension

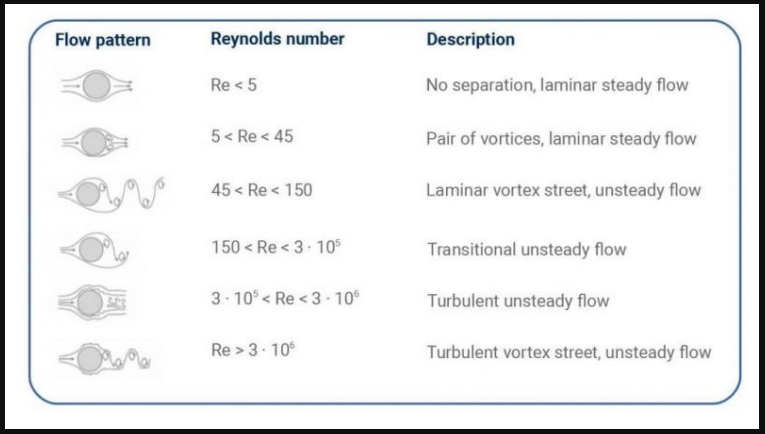
- 1 Simulation = 400 Tokens
- 1 Window = N subsequent tokens
- 101 Simulations x 400 Tokens / Window Length = Available Inputs

n	n+1	n+2	n+3
Custom	loss	functions	are…
			



Reynolds Number as Scientific Prior

- Fluid's Reynold's Number dependent on density, velocity, and molecular structure
- Reynold's Number dictates how [u,v,p] behave *over time*.
  - The Reynolds number carries a LINEAR relationship with several state values



Flow pattern	Reynolds number	Description
	$Re < 5$	No separation, laminar steady flow
	$5 < Re < 45$	Pair of vortices, laminar steady flow
	$45 < Re < 150$	Laminar vortex street, unsteady flow
	$150 < Re < 3 \cdot 10^5$	Transitional unsteady flow
	$3 \cdot 10^5 < Re < 3 \cdot 10^6$	Turbulent unsteady flow
	$Re > 3 \cdot 10^6$	Turbulent vortex street, unsteady flow

- Research Question: Can we leverage a representation of the Reynolds number which can act as an effective Positional Encoding method?

Model Overviews

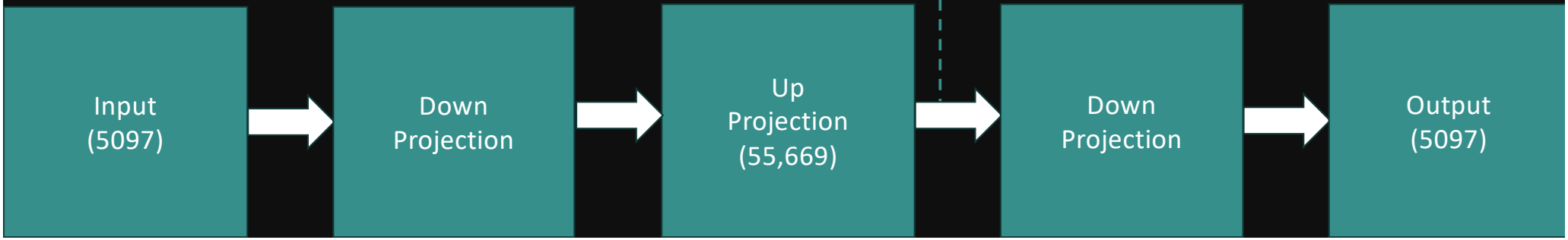
What were our model approaches? Why?

- MLP Baseline:
  - Comparison point for Transformer
- Transformers:
  - Global Content Relationships may capture inter-sensor dynamics
  - Fixed frame length and single-step regressions are favorable for Encoders
  - Simulation, and sequential forecasting favor Decoders
- Encoder Transformer Interventions:
  - Incorporate scientific priors into state-temporal relationships
- Decoder Transformer Interventions:
  - Custom, physics informed loss functions

Baseline: MLP

- 4-Layer
- "Wide" Learning

- Sizes:
  - Up
    - [25403, 55669, 115501]
  - Down
    - [128, 64, 32]



Transformers + Common Parameters

- Built from scratch: Encoder and Decoder Transformers
- SwiGLU Activation Function
- D = 64 or 128
  - **1/7** parameter budget
  - **10x** better metrics
- Positional Encoding methods discussed later...

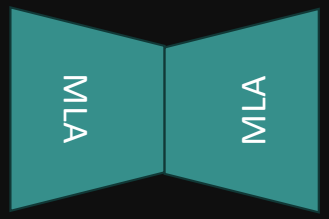
Transformers

Encoder

- Low-rank approximation for pooling R = 128
- D = 64
- Multi-Head Attention (MHA)

Decoder

- Triangular Causal Masking
- D = 128, Rank=16
- Multi-Head Latent Attention (MLA)



Designing a PE for CFD (Surrogate Model)

Inventory of our Problem Space

- Each token corresponds to a timestep...
- Each timestep is composed of the state representation containing many interdependent observations...

- Problem Space:
  - We need to encode a TEMPORAL (NxN) relationship into attention for our PE
  - We need to account for the physical relationship for our input's sensor states
  - We also want to account for how domain knowledge might influence these relationships

Intervention: Reynolds-Positional-Representation

Let $u \in \mathbb{R}^{H \times D_H}, r \in \mathbb{R}^{B}, u^{(r)} = u \odot r \in \mathbb{R}^{B \times H \times D_H}$

Cell-Level View:

$$\hat{\omega}_{[b,h,i,j]} = Q_{[b,h,i]} \cdot K_{[b,h,j]} + \sum_{d}^{D} Q_{[b,h,i,d]} u^{(r)}_{[b,h,d]} = Q_{[b,h,i]} \cdot K_{[b,h,j]} + Q_{[b,h,i]} \cdot u^{(r)}_{[b,h]}$$

Tensor-Level View: Let $\hat{R} \in \mathbb{R}^{N \times N \times D}$

$$\hat{\omega} = QK^T + einsum(Q_{bhid}, u^{(r)}_{bhd}; \rightarrow_{bhi})$$

Encoder Study

RQ1: Can our custom PE method compete with standard options?

RQ2: Can we beat a wider baseline (MLP) for single-step predictions?

Encoder Study: High-level (RQ1)

- Collect Performance results in a lossless representation of the model
  - Encoders can use Bidirectional context of the entire-tensor in it's logits, not just the final row – as with decoders.
  - This practically necessitates the use of some means of pooling for efficiency
- Run PE methods under different pooling schemes
- Isolate PE and pooling method effects
- Compare final model against non-transformer baseline

Encoder Methodology: Experimental Setups (RQ1)

- Sequence Length fixed at 10
- Each frame 1 second apart
- Goal is to predict U,V,P values of next state for ALL sensors
- A single-step multivariate regression problem

- Din=5097, D = 64 or 128, Dout = 5097

Encoder Methodology: Other Conditions (RQ1)

- RPR: Inspiration method for ReynoldsPR, uses relative token position representations rather than our Reynolds number representation
- ALiBi: Fixed linear decay signal instead of a learnable one:
  - Empirically captures local relations well
- APE: Standard method for small-fixed-context encoders
- NoPE: Just MHA, used for no PE baseline
- ANTI_METHOD: Replaces Reynolds numbers with random scalars
  - Used to determine if Reynolds numbers themselves actually play a role in performance

Encoder Study: Results (Rankings) D=128 [LOSSLESS]

Rank	Avg MSE	Std	IQR	95% CI
1st	RPR (Shaw)	APE	ReynoldsPR ★	APE
2nd	ReynoldsPR ★	ReynoldsPR ★	ALiBi	ReynoldsPR ★
3rd	ANTI_METHOD	ALiBi	APE	ALiBi
4th	ALiBi	NoPE	RPR (Shaw)	NoPE
5th	NoPE	RPR (Shaw)	ANTI_METHOD	RPR (Shaw)
6th	APE	ANTI_METHOD	NoPE	ANTI_METHOD

Results Interpretation (RQ1)

- ReynoldsPR, RPR and APE are Pareto-Optimal
  - Not surprising: RPR and APE are well known, standard methods
- **However**! RPR and APE face significant tradeoffs:
  - RPR: Best MSE, but places in bottom half for all reliability/consistency statistics
  - APE: Highly consistent, but MSE is lowest of all PE methods
- Anti-Method confirms efficacy of Reynolds number in Encoding
- **In contrast: ReynoldsPR never leaves the Top-2 ranks of performance**

However... Compression is a Practical Necessity

- Encoder can use the entire context tensor, as opposed to the last row... this is an NxD shaped tensor to create a linear map with outhead
  - With a high dimensions Dout, this is a parameter explosion!
  - 7.8 Mil in our case...
- The most common method is to simply mean pool for all tokens
- However, this is a very lossy operation!
- We instead opted for a Low Rank Approximation

Comparison (AVG MSE) (RQ1)

Sample Mean Pooling D=128

PE	Params	AVG MSE
ReyPR	~1.96 MIL	0.005675
APE	~1.96 MIL	0.003181
RPR	~1.96 MIL	0.003567
NoPE	~1.96 MIL	0.005176
ALiBi	~1.96 MIL	0.005147

Low Rank Approximation Pooling D=64,R=128

PE	Params	AVG MSE
ReyPR	~1.3 MIL	0.002905
APE	~1.3 MIL	0.002980

Encoder Transformer vs MLP Baseline (RQ2)

MLP

Params	AVG **MSE**
~1.3 MIL	0.036
~1.8 MIL	0.031
~5.6 MIL	0.028
~7.8 MIL	0.021
~10 MIL	0.015

Encoder Trans: ReyPR + LRA

Params	AVG **MSE**
~1.3 MIL	0.002905

Decoder Simulation Study

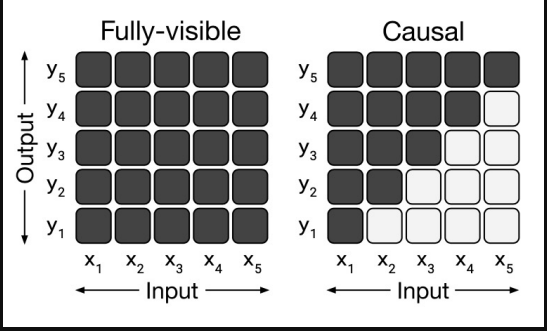
Motivation and Results

- Decoders are typically preferred for long-term forecasting
- In testing, ReyPR had almost no effect on MSE

PE	AVG MSE
ReyPR	0.003864
ReyPR + APE	0.003717
APE	0.003754

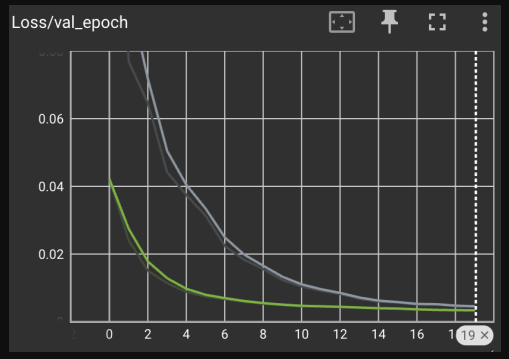
Why Decoders need a Different Solution?

- Recall that decoders use causal masking to prevent attention to future tokens
- Absolute encoding is powerful here because it gives information about future progress
- ReyPR requires an approximation of the entire context tensor to be effective, when using only the last row APE remains more effective



Physics-Informed Loss Function

- Fluid simulations use equations as a baseline
  - Could we incorporate these into our model as a weighted loss?
- Results: Higher MSE, lower physics error
- Uses node positionality
  - Potentially generalizable to other simulations



Pure MSE Loss

Navier–Stokes Equations
3 – dimensional – unsteady

Glenn Research Center

Coordinates: (x,y,z)
Velocity Components: (u,v,w)
Time : t Pressure: p
Density: ρ Stress: τ
Total Energy: Et
Heat Flux: q
Reynolds Number: Re
Prandtl Number: Pr

Continuity:
$$\frac{\partial \rho}{\partial t}+\frac{\partial(\rho u)}{\partial x}+\frac{\partial(\rho v)}{\partial y}+\frac{\partial(\rho w)}{\partial z}=0$$

X – Momentum:
$$\frac{\partial(\rho u)}{\partial t}+\frac{\partial(\rho u^2)}{\partial x}+\frac{\partial(\rho uv)}{\partial y}+\frac{\partial(\rho uw)}{\partial z}=-\frac{\partial p}{\partial x}+\frac{1}{Re_r}\left[\frac{\partial \tau_{xx}}{\partial x}+\frac{\partial \tau_{xy}}{\partial y}+\frac{\partial \tau_{xz}}{\partial z}\right]$$

Y – Momentum:
$$\frac{\partial(\rho v)}{\partial t}+\frac{\partial(\rho uv)}{\partial x}+\frac{\partial(\rho v^2)}{\partial y}+\frac{\partial(\rho vw)}{\partial z}=-\frac{\partial p}{\partial y}+\frac{1}{Re_r}\left[\frac{\partial \tau_{xy}}{\partial x}+\frac{\partial \tau_{yy}}{\partial y}+\frac{\partial \tau_{yz}}{\partial z}\right]$$

Z – Momentum
$$\frac{\partial(\rho w)}{\partial t}+\frac{\partial(\rho uw)}{\partial x}+\frac{\partial(\rho vw)}{\partial y}+\frac{\partial(\rho w^2)}{\partial z}=-\frac{\partial p}{\partial z}+\frac{1}{Re_r}\left[\frac{\partial \tau_{xz}}{\partial x}+\frac{\partial \tau_{yz}}{\partial y}+\frac{\partial \tau_{zz}}{\partial z}\right]$$

Energy:
$$\frac{\partial(E_T)}{\partial t}+\frac{\partial(uE_T)}{\partial x}+\frac{\partial(vE_T)}{\partial y}+\frac{\partial(wE_T)}{\partial z}=-\frac{\partial(up)}{\partial x}-\frac{\partial(vp)}{\partial y}-\frac{\partial(wp)}{\partial z}-\frac{1}{Re_rPr_r}\left[\frac{\partial q_x}{\partial x}+\frac{\partial q_y}{\partial y}+\frac{\partial q_z}{\partial z}\right]$$
$$+\frac{1}{Re_r}\left[\frac{\partial}{\partial x}(u\tau_{xx}+v\tau_{xy}+w\tau_{xz})+\frac{\partial}{\partial y}(u\tau_{xy}+v\tau_{yy}+w\tau_{yz})+\frac{\partial}{\partial z}(u\tau_{xz}+v\tau_{yz}+w\tau_{zz})\right]$$

Possible Improvements/ Future Directions

- Generalize to other simulation environments
- Experiment with scientific priors in context-tensor computation
- Incorporate 2D spatial data structures into attention
- Use proper encoder-decoder pair for handling up and down projections for input and output

Summary of Findings

Transformers

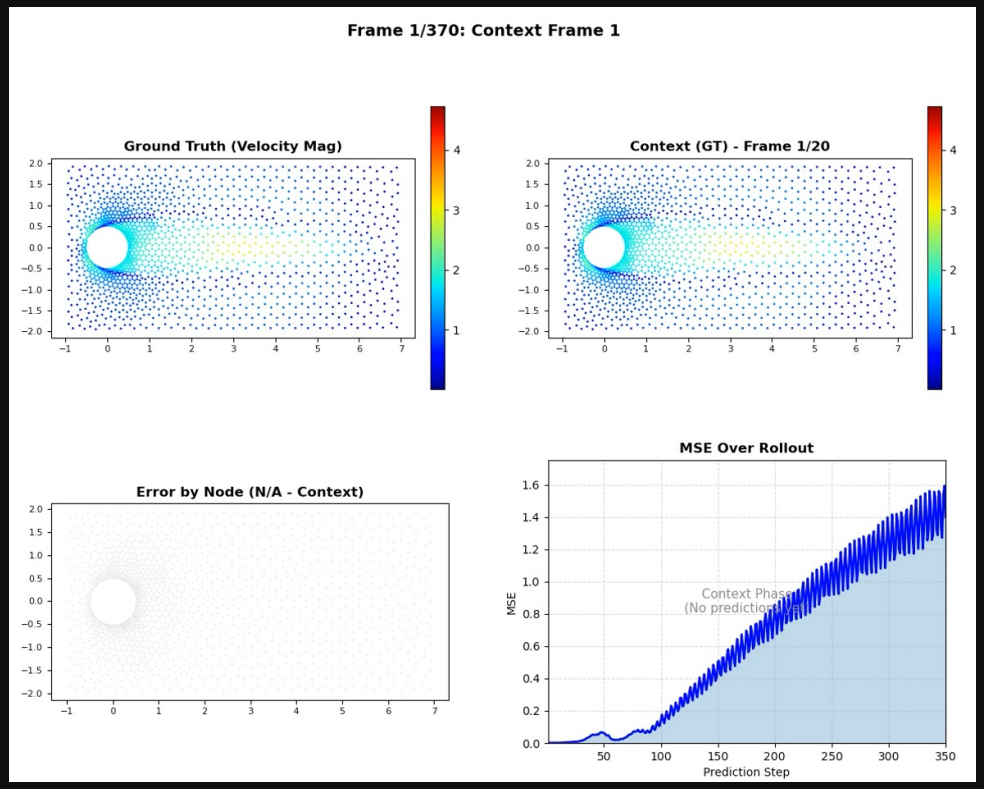
- Using a low-state representation of the problem is parameter efficient for CFD
  - 7x less parameters, almost 1000x smaller state space, increased performance by 10x

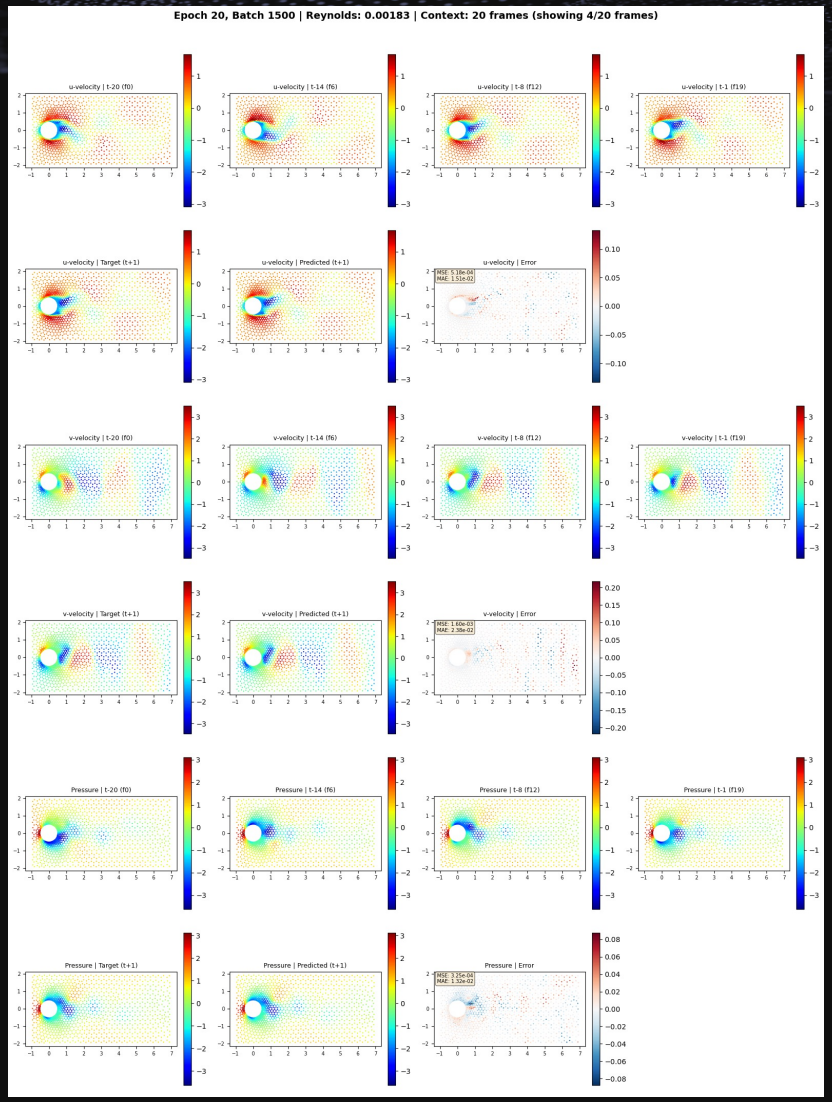
Positional Encoding

- Scientific priors can be used as effective physics informed positional encoding
  - Using ReyPR was optimal over APE for encoders
  - The choice of out-head compression/pooling affects PE influence
  - Decoder-only models limit effectiveness

Q/A + Demo!

20 Epochs: MSE Loss only





BONUS SLIDES!

Input X and Weights

$$X \in \mathbb{R}^{B \times N \times D}$$

$$W_Q, W_K, W_V \in \mathbb{R}^{D \times D}$$

- Prior to feeding our input into the attention block, *we down-project into a lower dimensional representation 'D'.*
- 'D' refers to the down-projected representation of the sensor data
  - 5097 -> 64 (or 128)
- Here, things are the same as normal, *each token is a timestep*
- Therefore: we have a matrix where each timestep is associated with a lower dimensional state-representation

Query, Key, Value Tensors

$$Q = XW_Q, K = XW_K, V = XW_V \in \mathbb{R}^{B \times N \times D}$$

- Defined in via the same tensor-level operations as in standard SA
  - (Above Equation)

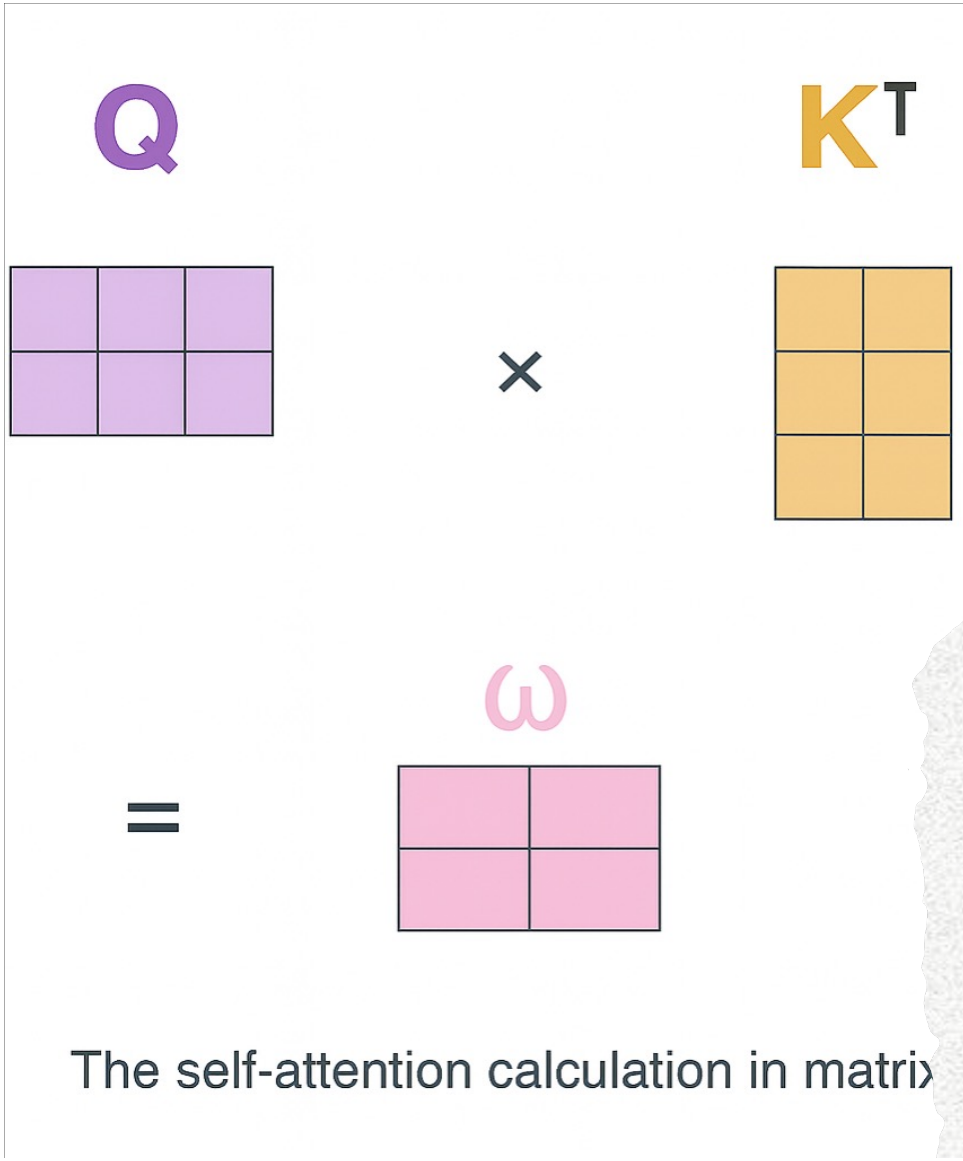
- Similarly, our Q,K,V values are simply linear reprojections of our input in order to learn different roles
- This means that each is STILL an NxD matrix containing a timestep x state-representation relationship

$$Q_{[i.j]} = (XW_q)_{[i,j]} = \sum_{d=1}^{D} X_{[i,d]} W_{q[d,j]} = X_{[i]} \cdot W_q^T{}_{[j]}$$

What does Self-Attention *Mean* Here?

Encoder Study: Results (Raw Numbers)

Method	Avg MSE ↓	Std ↓ ($\times 10^{-5}$)	IQR ↓ ($\times 10^{-5}$)	95% CI (±) ↓ ($\times 10^{-5}$)
ReynoldsPR	0.0013334	3.99	3.05	2.48
RPR (Shaw)	0.0013290	7.23	8.73	4.48
ANTI_METHOD	0.0013456	9.48	10.20	6.19
ALiBi	0.0013521	6.07	4.55	3.76
NoPE	0.0013596	6.45	11.90	4.22
APE	0.0013826	4.40	5.60	2.73



Attention Scores

- Attention Scores in NxN space
- Each token is a timestep
- Our attention learns relevance between *timesteps*
- Therefore, transformations to the attention score are affecting how each timestep is considered relevant to another

Interpreting our Attention Scores

$$\omega_{[i,j]} = (QK^T)_{[i,j]} = \sum_{d=1}^{D} Q_{[i,d]} K^T_{[d,j]} = Q_{[i]} \cdot K_{[j]}$$

- Since D in both the Q and K tensors contains a content-representation in their D-spaces...
- Our attention scores, organized by timestep, contain only a content-by-content similarity score for establishing their frame-to-frame relevance...

Pareto-Optimality

- Multiple objectives to minimize simultaneously:
  - MSE (accuracy)
  - Std / CI (variance stability)
  - IQR (robustness)

- (Definition) A method A dominates method B if:
  - A is no worse than B on all metrics, and...
  - A is strictly better than B on at least one metric.
- (Definition) A method is *Pareto-Optimal* if no other method dominates it.

- Based on our results... **ReynoldsPR is Pareto-Optimal**

Encoder Study Implications:

- Using a Transformer in a highly compressed state representation yields:
  - A massive MSE and Parameter reduction from 'wide' MLP representation
- Scientific Priors can effectively be used as alternative concepts in positional encoding in embedded representations
- Encoder compression methods greatly affect PE effectiveness
  - Can greatly impact performance on Encoders in general

Low Rank Approximation of the Lossless Linear Map

- An alternative idea is to use a latent space with the out-head
- Using a Low Rank Approximation improves performance and is substantially more efficient!
- This also restores ReyPR to a dominant performance spot!
- This is intriguing, but perhaps not surprising as this method approximates the lossless version of the problem