Anthony Gruber

Nonlinear PDEs in Computer Graphics and Data Science

Outline

- Introduction
- Examples in computer graphics
  - p-Willmore flow; quasiconformal mappings; grid generation
  - Partially funded by NSF DMS #1912902, 1912705
- Examples in data science
  - ANN function approximation; reduced order modeling
  - Partially funded by DOE DE-SC0020418

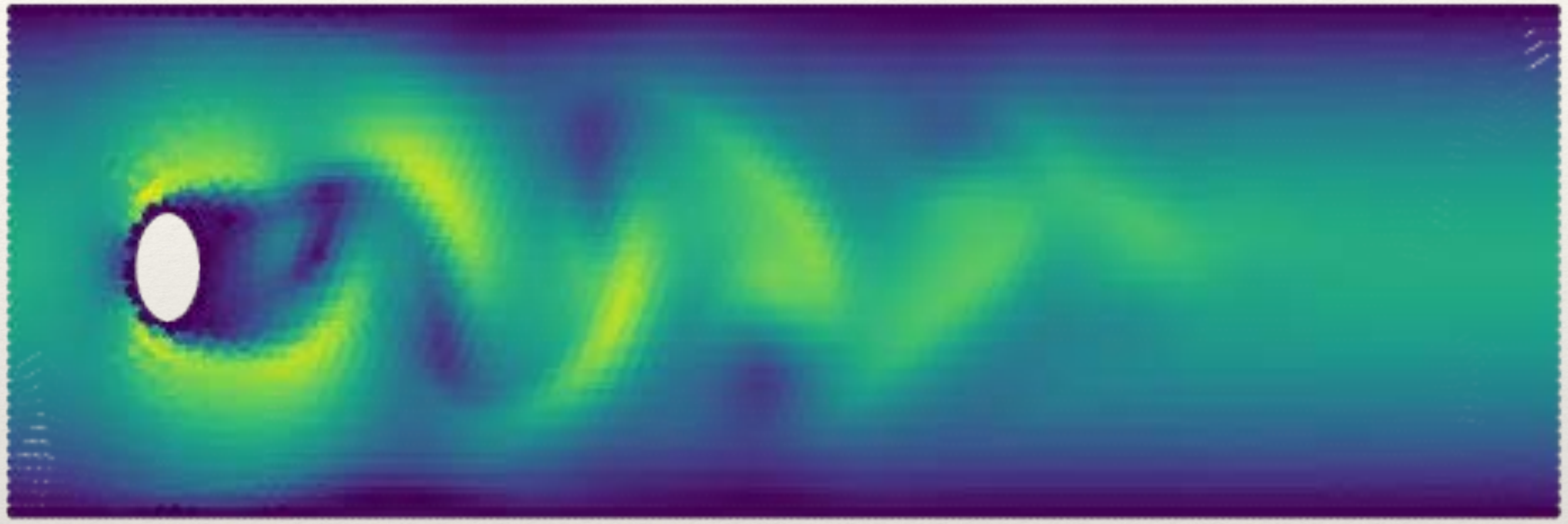
Papers since 2019

- Journal articles
  - A. Gruber. ``Planar Immersions with Prescribed Curl and Jacobian Determinant are Unique", *Bull. Aust. Math. Soc.*, (in press).
  - A. Gruber, M. Gunzburger, L. Ju, Y. Teng, Z. Wang. ``Nonlinear Level Set Learning for Function Approximation on Sparse Data with Applications to Parametric Differential Equations", *Numer. Math. Theor. Meth. Appl.*, (2021).
  - A. Gruber, A. Pámpano, M. Toda. ``Regarding the Euler-Plateau Problem with Elastic Modulus’’, *Ann. Mat. Pura Appl.*, (2021).
  - A. Gruber, E. Aulisa. ``Computational p-Willmore Flow with Conformal Penalty", ***ACM Trans. Graph***. 39, 5, Article 161 (September 2020), 16 pages.
  - A. Gruber, M. Toda, H. Tran. ``On the variation of curvature functionals in a space form with application to a generalized Willmore energy’’, *Ann. Glob. Anal. Geom.* 56, 147–165 (2019).
- Conference papers
  - A. Gruber, E. Aulisa. ``Quaternionic remeshing during surface evolution", *Proceedings of the 18th ICNAAM*, Rhodes, Greece, 2020, (in press).
  - A. Gruber, M. Toda, H. Tran. ``Willmore-stable minimal surfaces", *Proceedings of the 18th ICNAAM*, Rhodes, Greece, 2020, (in press).
  - E. Aulisa, A. Gruber, M. Toda, H. Tran. ``New Developments on the p-Willmore Energy of Surfaces", *Proceedings of 21st ICGIQ*, Sofia: Avangard Prima, 2020.
  - R. Bridges, A. Gruber, C. Felder, M. Verma, C. Hoff. ``Active Manifolds: Reducing high dimensional functions to 1-D; A non-linear analogue to Active Subspaces". ***Proceedings of ICML***, 9-15 June 2019, Long Beach, California, USA. PMLR 97:764-772.
- Submitted articles
  - A. Gruber, M. Gunzburger, L. Ju, Z. Wang. ``A Comparison of Neural Network Architectures for Data-Driven Reduced-Order Modeling’’, (submitted).
  - A. Gruber, E. Aulisa. ``Quasiconformal Mappings for Surface Mesh Optimization", (under review).
  - A. Gruber, A. Pámpano, M. Toda. ``On p-Willmore Disks with Boundary Energies", (under review).
  - A. Gruber. ``Parallel Codazzi Tensors with Submanifold Applications", (under review).
  - A. Gruber, M. Toda, H. Tran. ``Stationary Surfaces with Boundaries", (under review).

Why are PDEs so ubiquitous?

- Much easier to write down *relationships* than *solutions*.
- **Ex)** $I = \int_a^b L(t, \mathbf{q}(t), \dot{\mathbf{q}}(t))\,.$

  $\dot{I} = 0 \implies \partial_t L_{\dot{\mathbf{q}}} = L_{\mathbf{q}}$

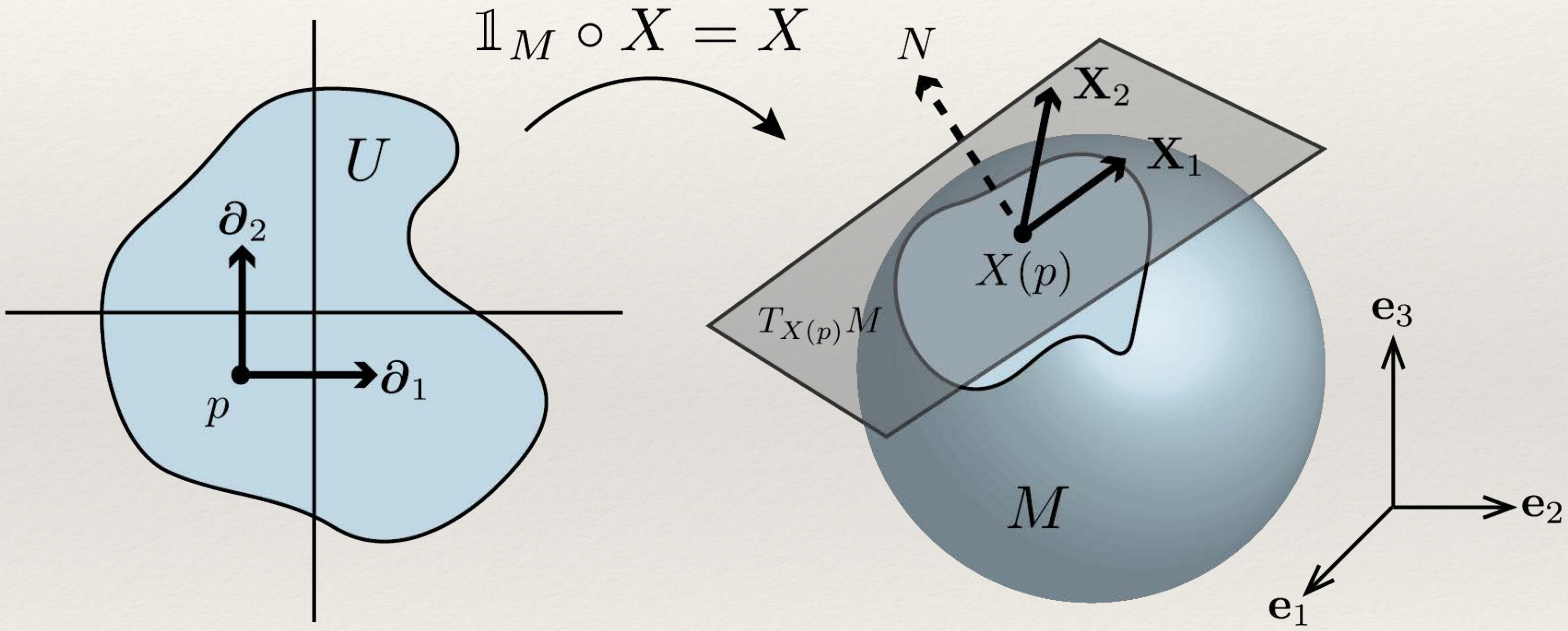


- Heavily linked to geometry!
  - Important relationships expressed as PDEs.

$$\mathbf{u}_t - \nu\Delta\mathbf{u} + (\mathbf{u}\cdot\nabla)\,\mathbf{u} + \nabla p = 0,$$
$$\nabla\cdot\mathbf{u} = 0,$$
$$\mathbf{u}\,\big|_{t=0} = \mathbf{u}_0,$$

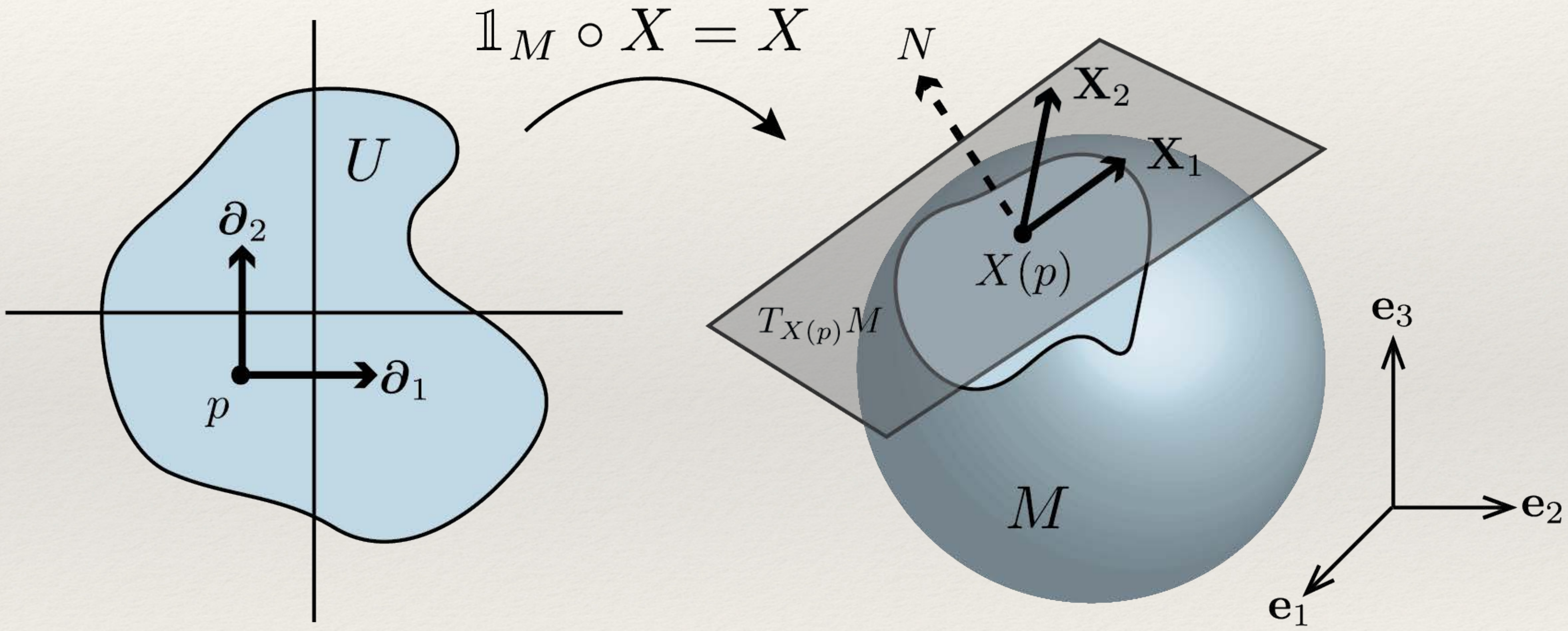
What is a Riemannian geometry?

- "Smooth" manifold M equipped with "smooth" metric g.
- Local charts cover M and "agree on overlaps".
- Example)
  $X : U \subset \mathbb{R}^2 \to \mathbb{R}^3$,
  $g(u, v) = \langle dX(u), dX(v) \rangle$.



What is a Riemannian geometry?

- Metric g determines intrinsic behavior (curvature, conformal structure).
- Change in normal N determines extrinsic behavior (shape, bending).

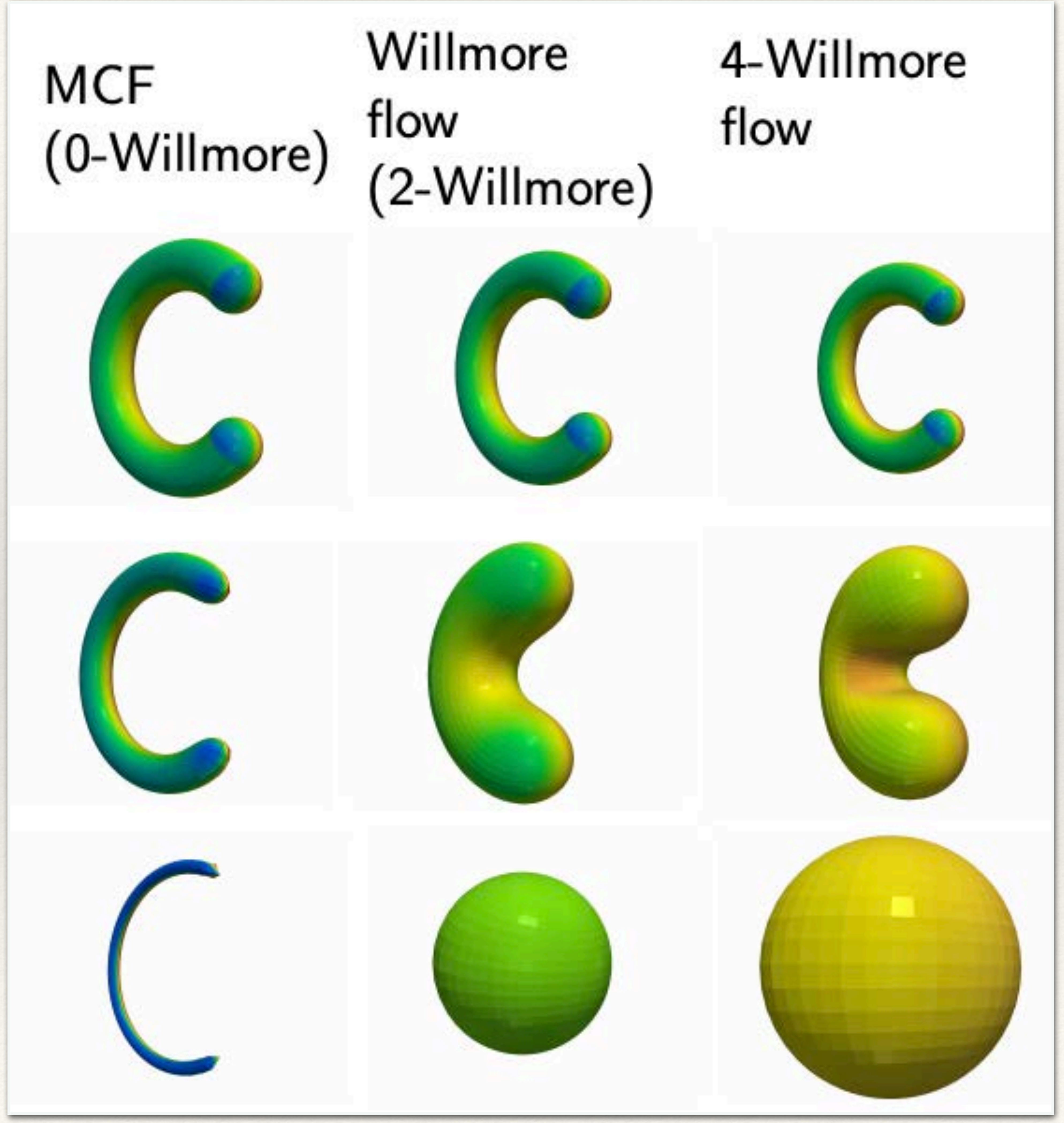


Where do NPDEs meet geometry?

- *Everywhere!*
- **Ex)** Finding X s.t. $g_{ij} = \langle X_i, X_j \rangle$ for given functions g_{ij} .
- **Ex)** Gauss, Codazzi compatibility equations e.g $\tilde{R}_{ijkl} = R_{ijkl} + h_{ik}h_{jl} - h_{jk}h_{il}$.
- **Ex)** Mean curvature flow $\dot{X} = -\Delta_g X = -2HN$
  - p-Willmore flow: $\langle \dot{X}, N \rangle = -\frac{p}{2}\Delta_g H^{p-1} - pH^{p-1}\left(2H^2 - K\right) + 2H^{p+1}$

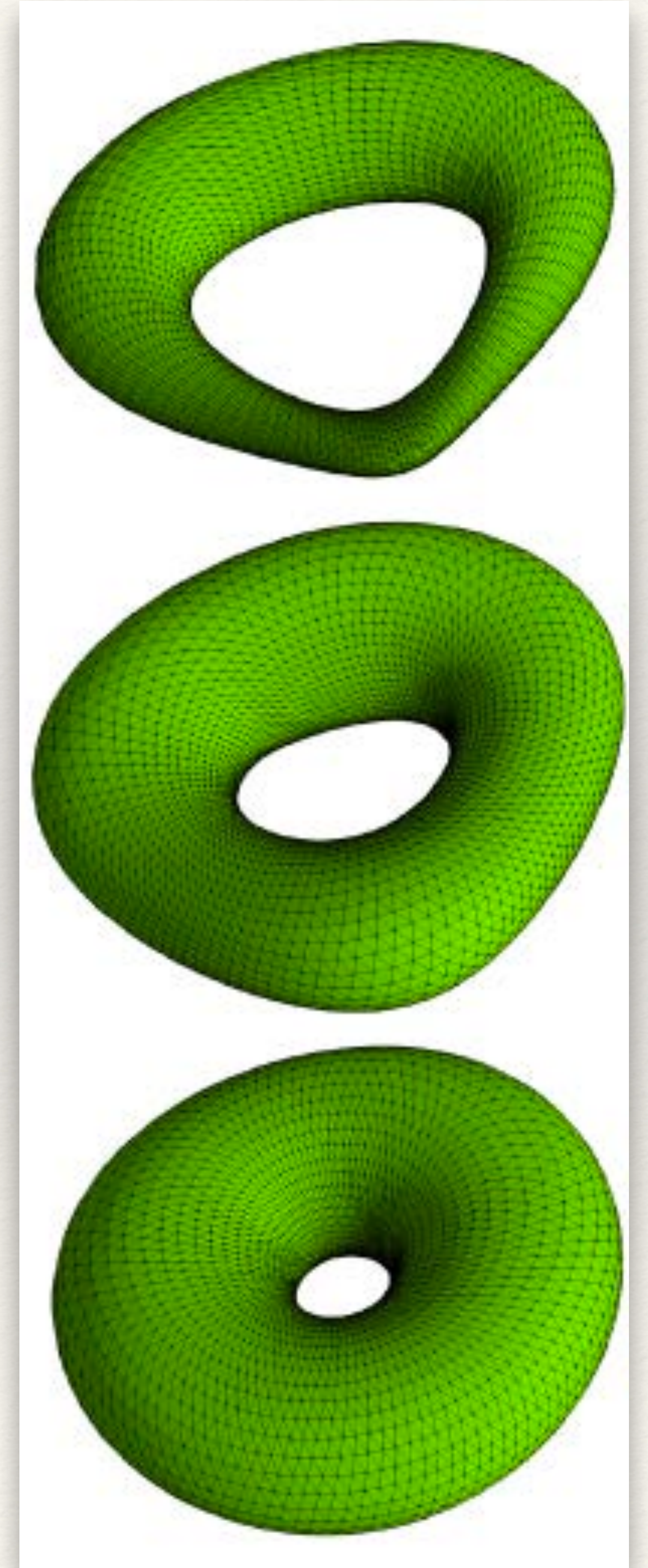
Outline

- Introduction
- **Examples in computer graphics**
  - p-Willmore flow; quasiconformal mappings; grid generation
  - Partially funded by NSF DMS #1912902, 1912705
- Examples in data science
  - ANN function approximation; reduced order modeling
  - Partially funded by DOE DE-SC0020418



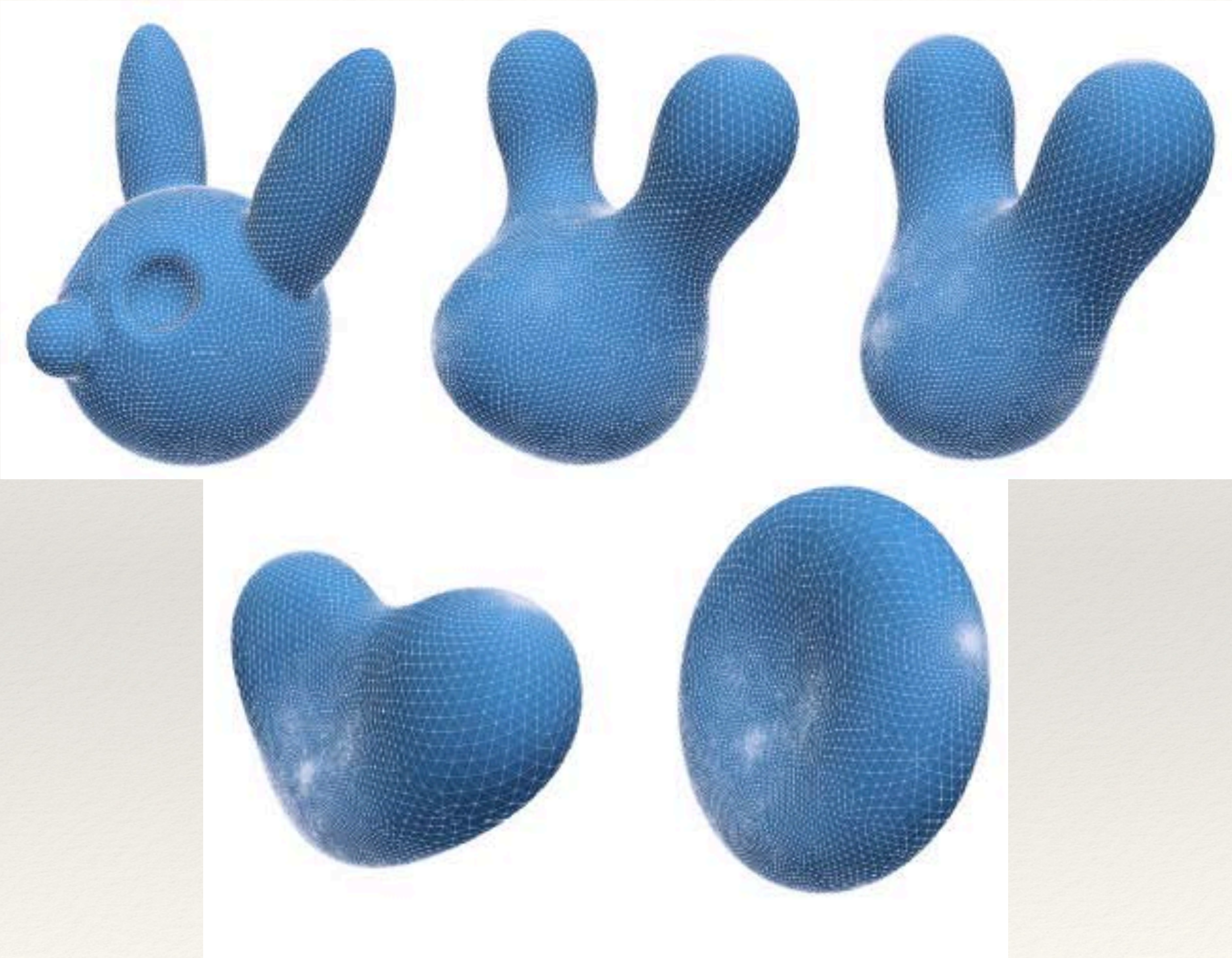
Modeling p-Willmore Flow

- p-Willmore energy: $\mathscr{W}^p(X) = \int_M |H|^p \, d\mu_g$.
- Suppose M is closed. New variable $Y := \Delta_g X$ (G. Dziuk, 2012).
  - Weak definition $\int_M Y \cdot \psi + \langle dX \cdot d\psi \rangle_g = 0.$
- Willmore flow becomes coupled pair of 2^{nd}-order PDEs for X.



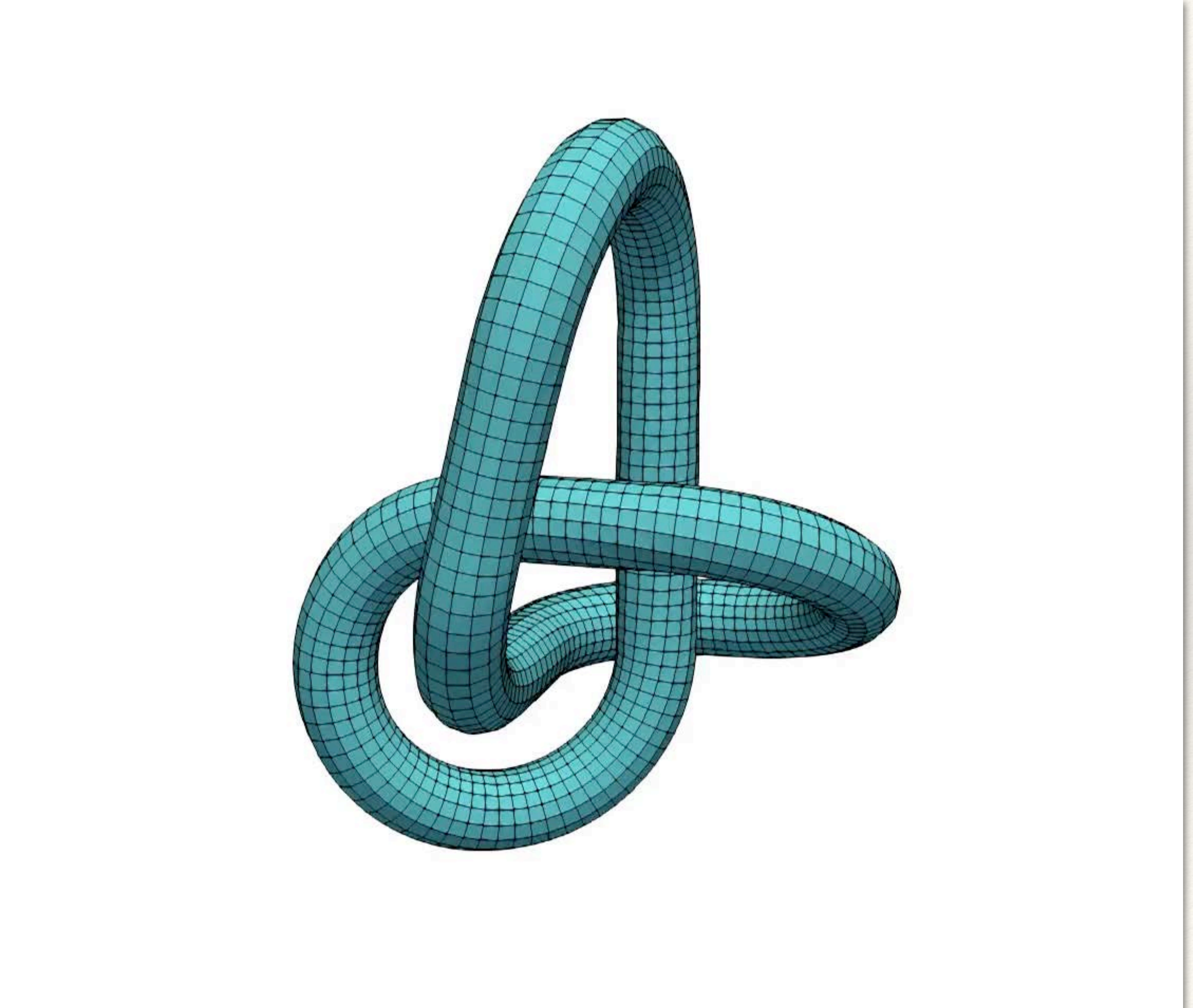
Modeling p-Willmore Flow

- This trick works for p-Willmore flows, too.
- Developed algorithm for p-Willmore flow with area/volume constraints.
- Conformal regularization procedure — keeps mesh nice.



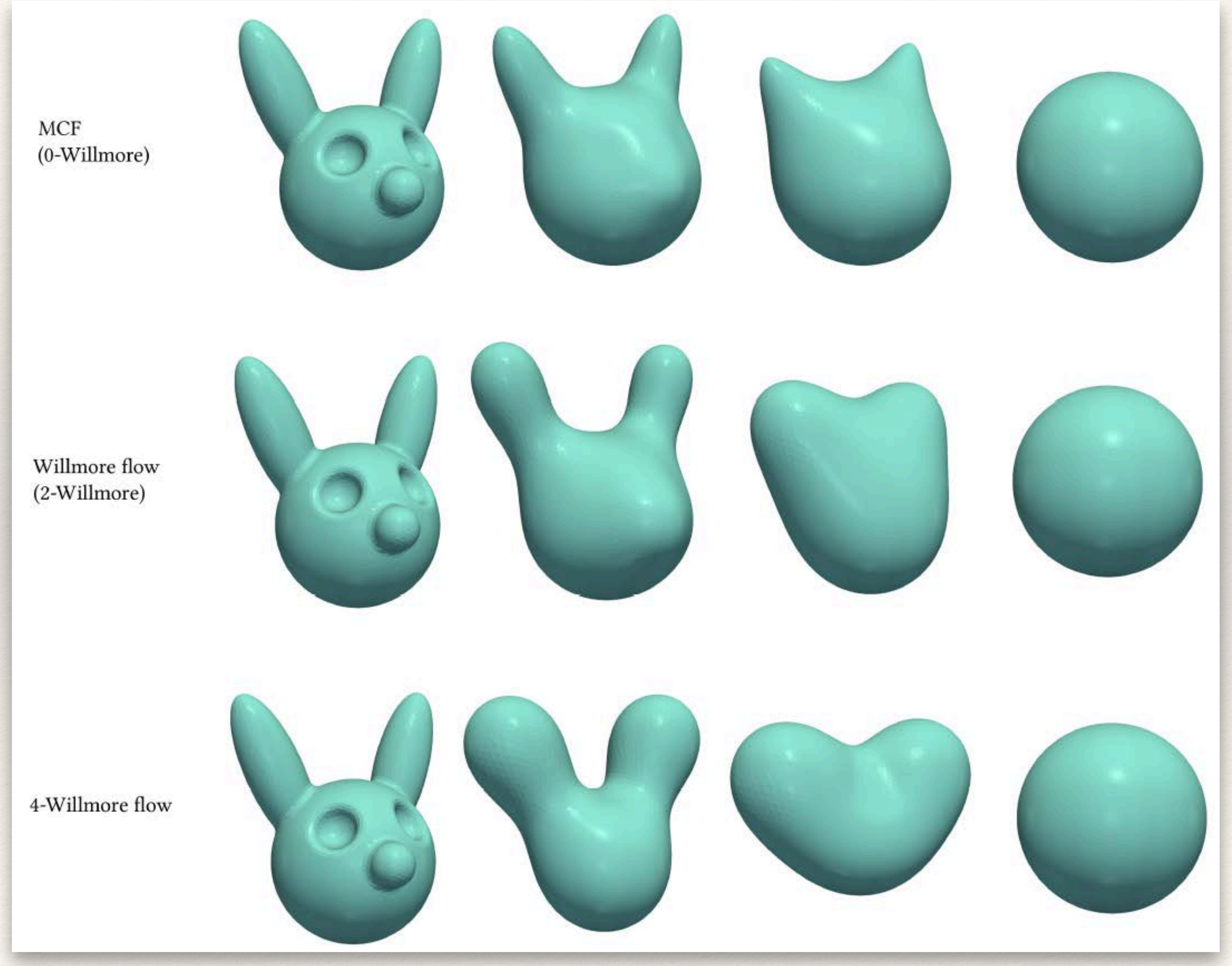
- Gruber, A. and Aulisa, E. *ACM Trans. Graph.* (2020)

Trefoil Knot Unwinding



Comparison: p-Willmore

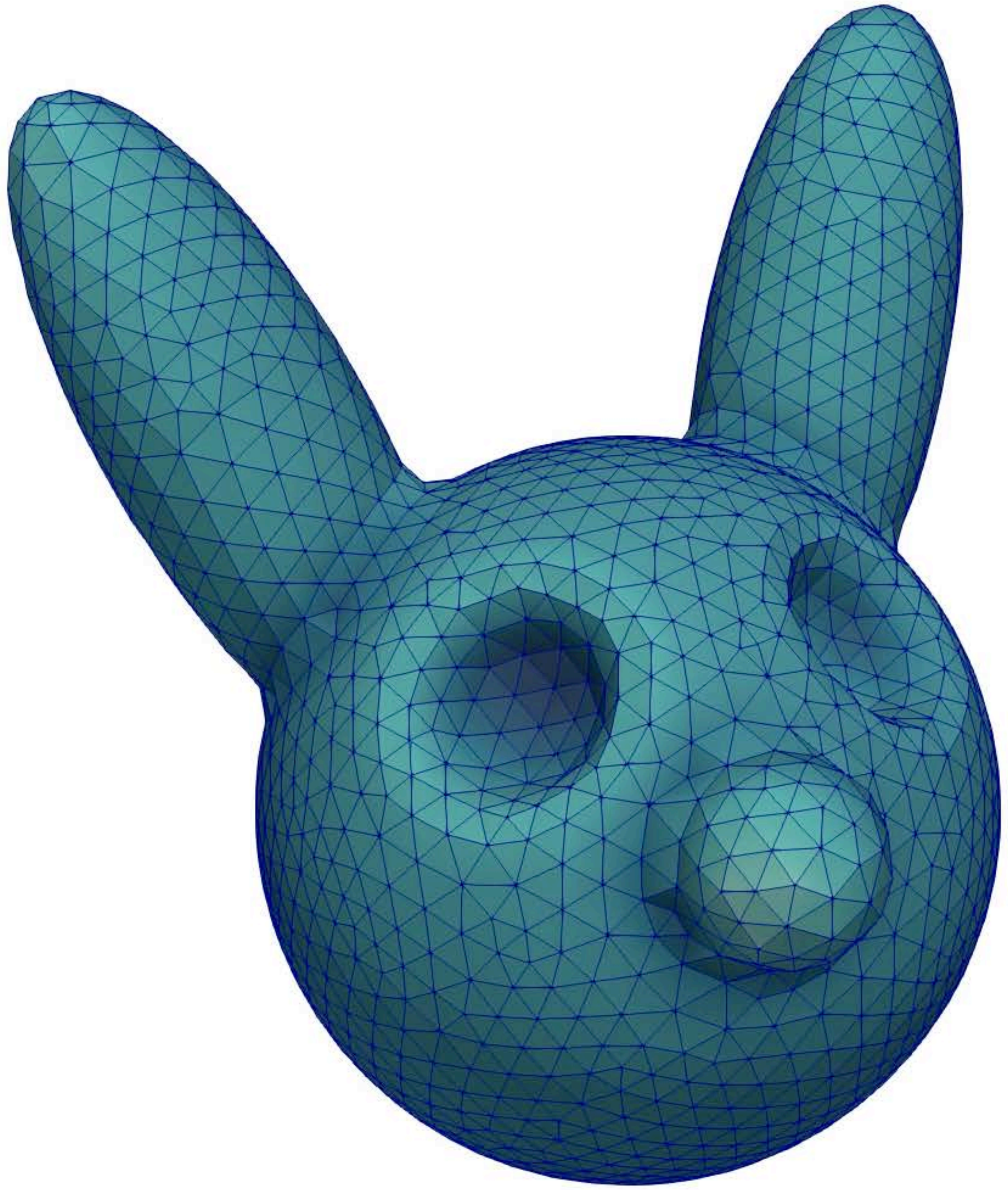
- Volume-constrained flows.
- Higher p — more rounding behavior.
- Can show that $p > 2$ Willmore minimizers resemble minimal surfaces in some aspects. *



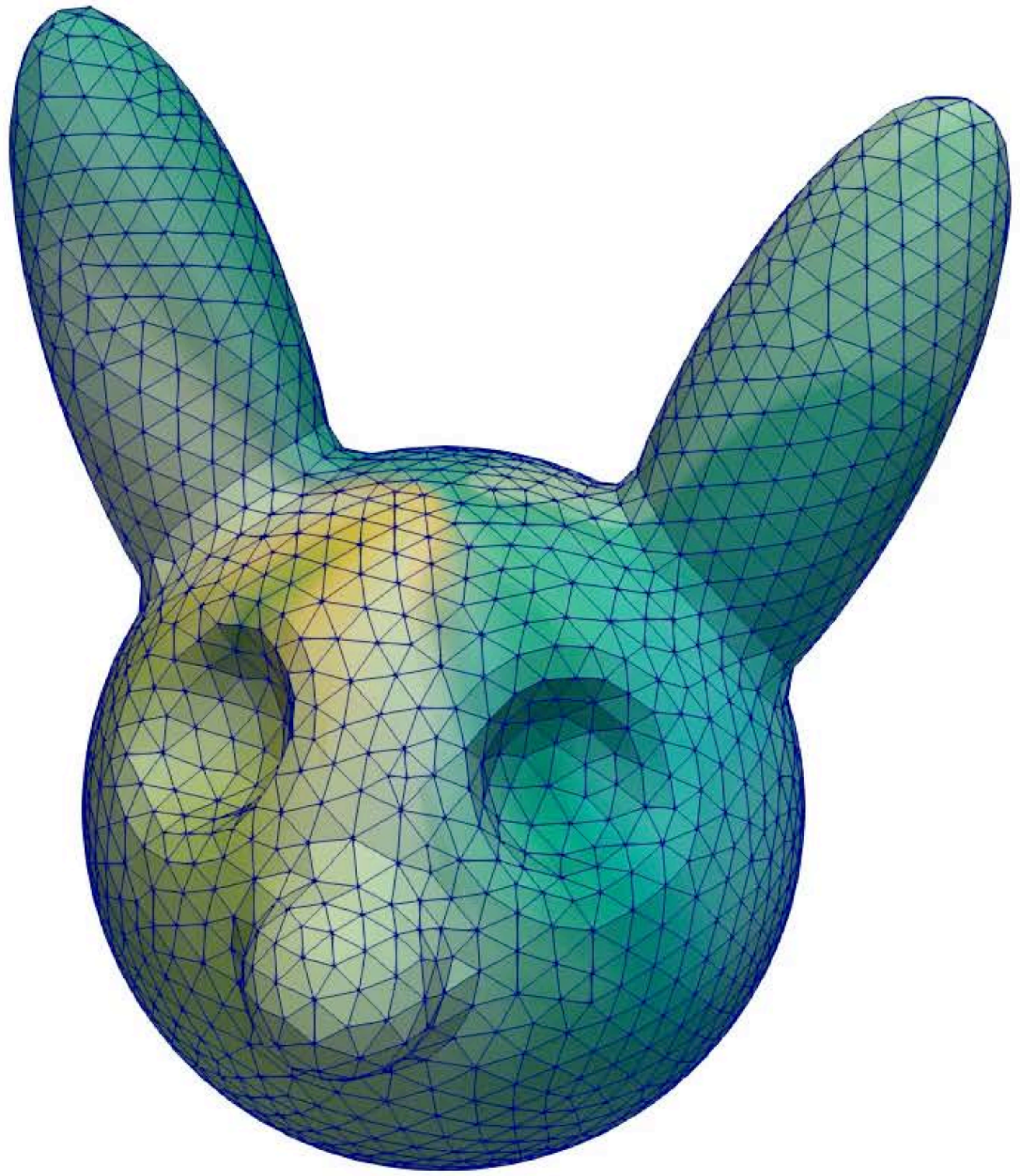
* Gruber, A., Toda M., Tran, H. *Ann. Glob. Anal. Geom.* (2019).

* Gruber, A., Pampano, A., Toda, M. *Ann. Mat. Pura. Appl.* (2021).

Influence of the Constraint



Volume preserving 2-Willmore flow

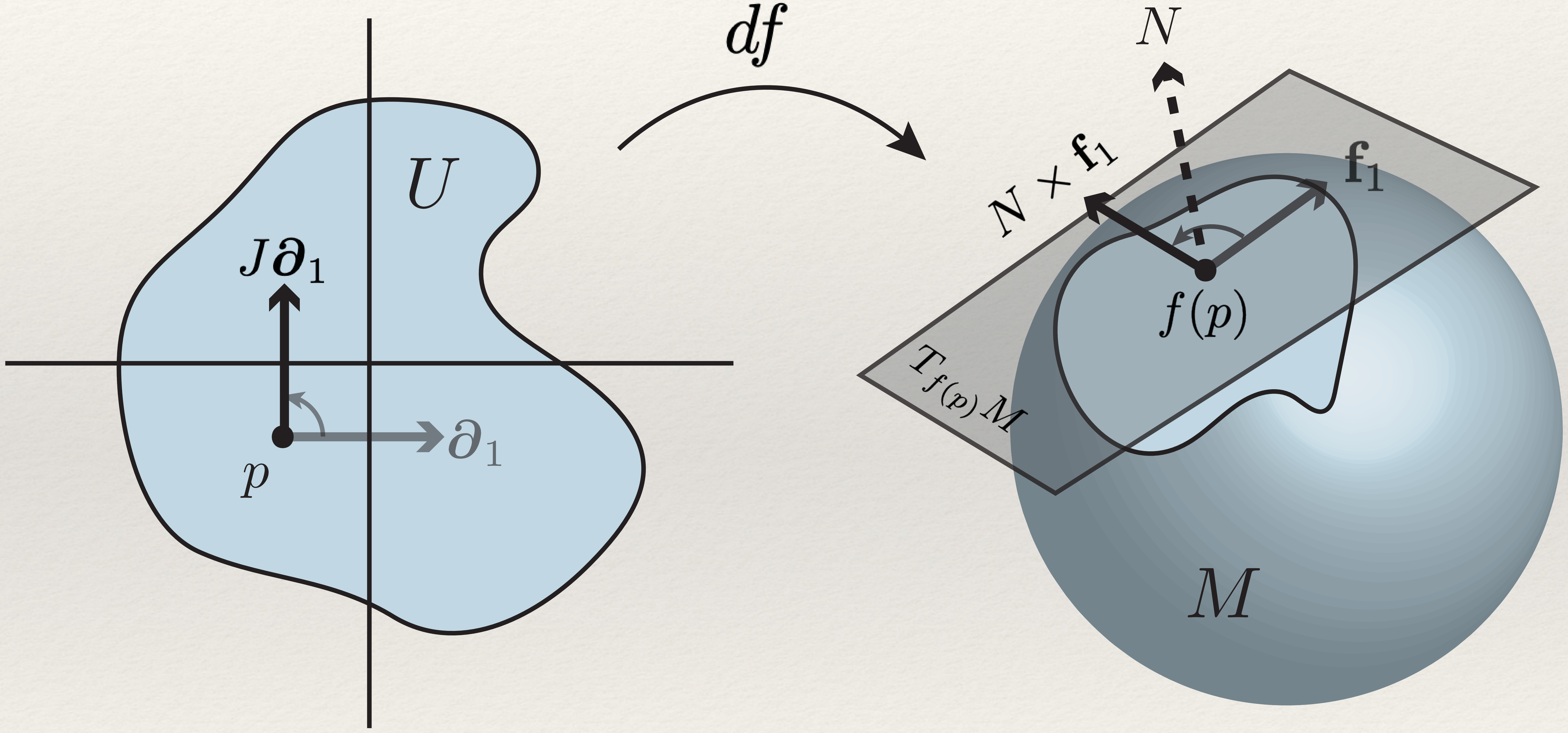


Volume and area preserving 2-Willmore flow

What about the Mesh?

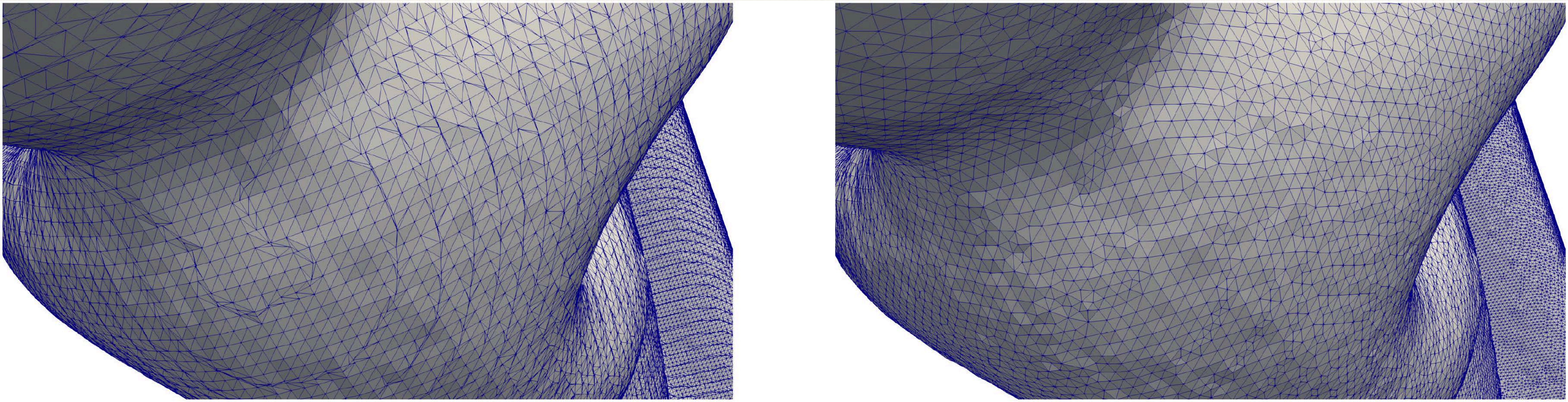
- Regularization involves least-squares conformal mapping
- Map $f:(M,g) \to (N,h)$ is conformal if $f^*h = e^{2\phi}g$.
- When $f: M \to \mathbb{R}^3$, equiv. to $\exists N$ s.t. $\star\, df = N \times df$

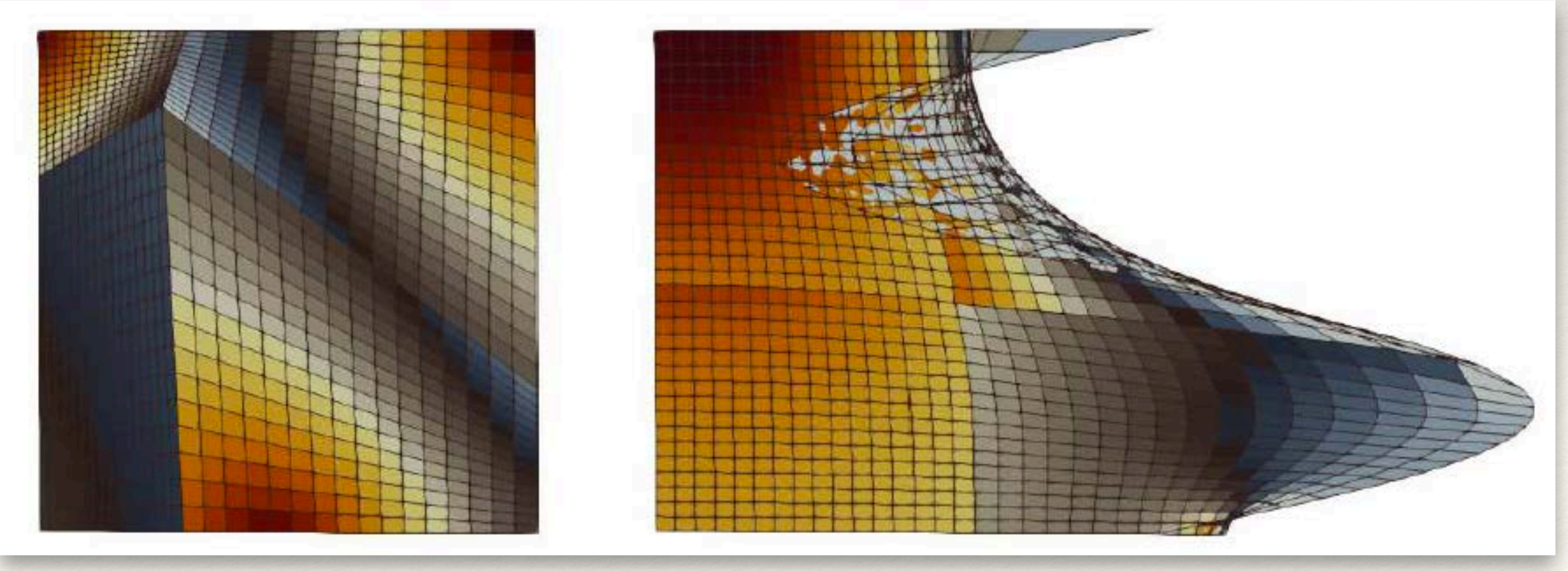
  (Kamberov, Pedit, Pinkall 1996).
- No explicit reference to metric!! (wrapped in ★)



Results

- Can produce least-squares conformal maps with this idea.
- Great! Until the need for a point-wise boundary correspondence.
- Conformality is **too restrictive** for many applications.

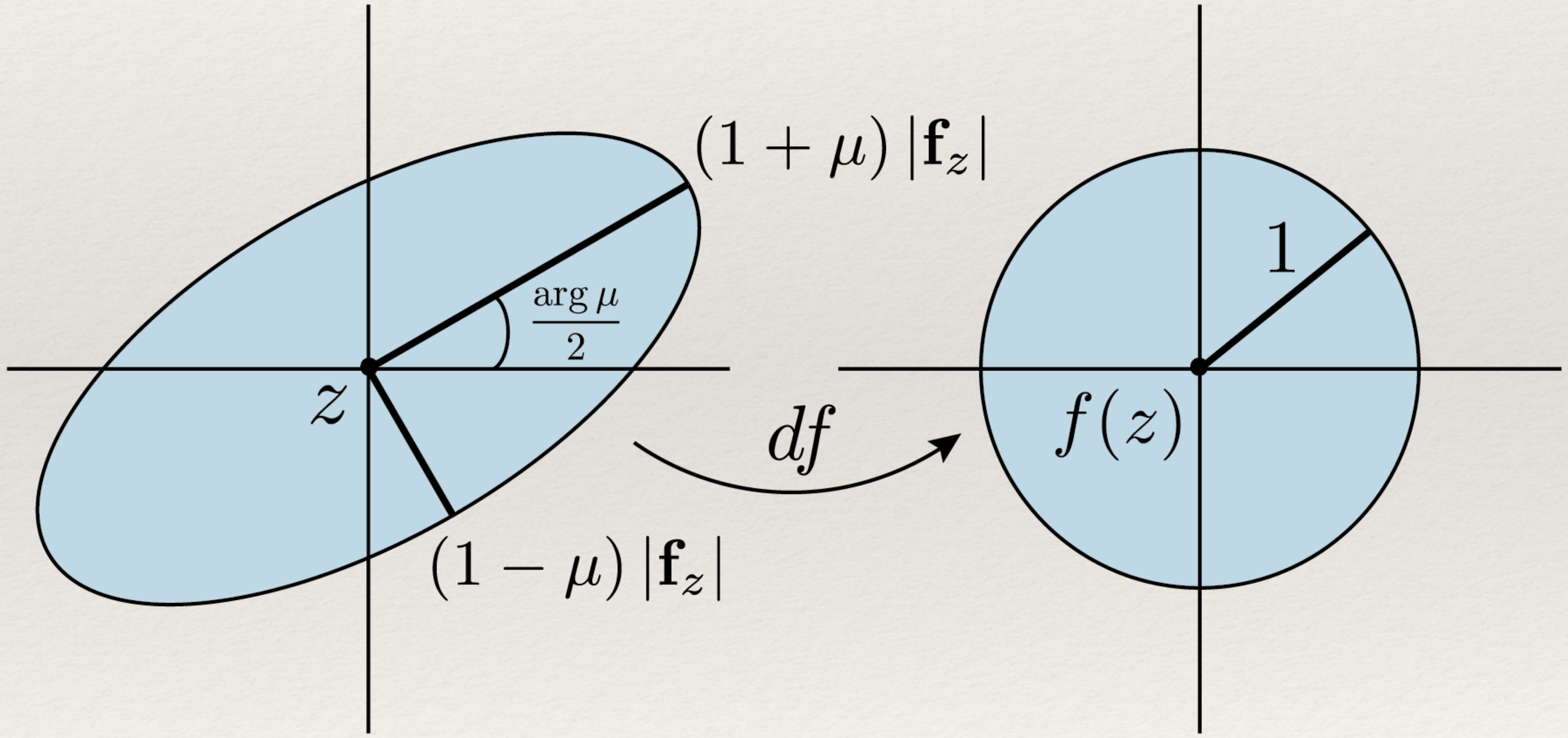




Conformal vs. Quasiconformal

- Must allow **bounded shearing distortion.**
- In quaternionic setting, this means:
  - $df^- = \mu\, df^+$ (anticonformal/conformal parts).
  - Conformal iff Beltrami coefficient $\mu = 0$.
  - μ is conjugate-dual to the *Hopf differential* $Q = df^+\overline{df^-}$.

$$df^\pm = \frac{1}{2}\left(df \mp N \star df\right)$$



Optimal Teichmuller Mappings

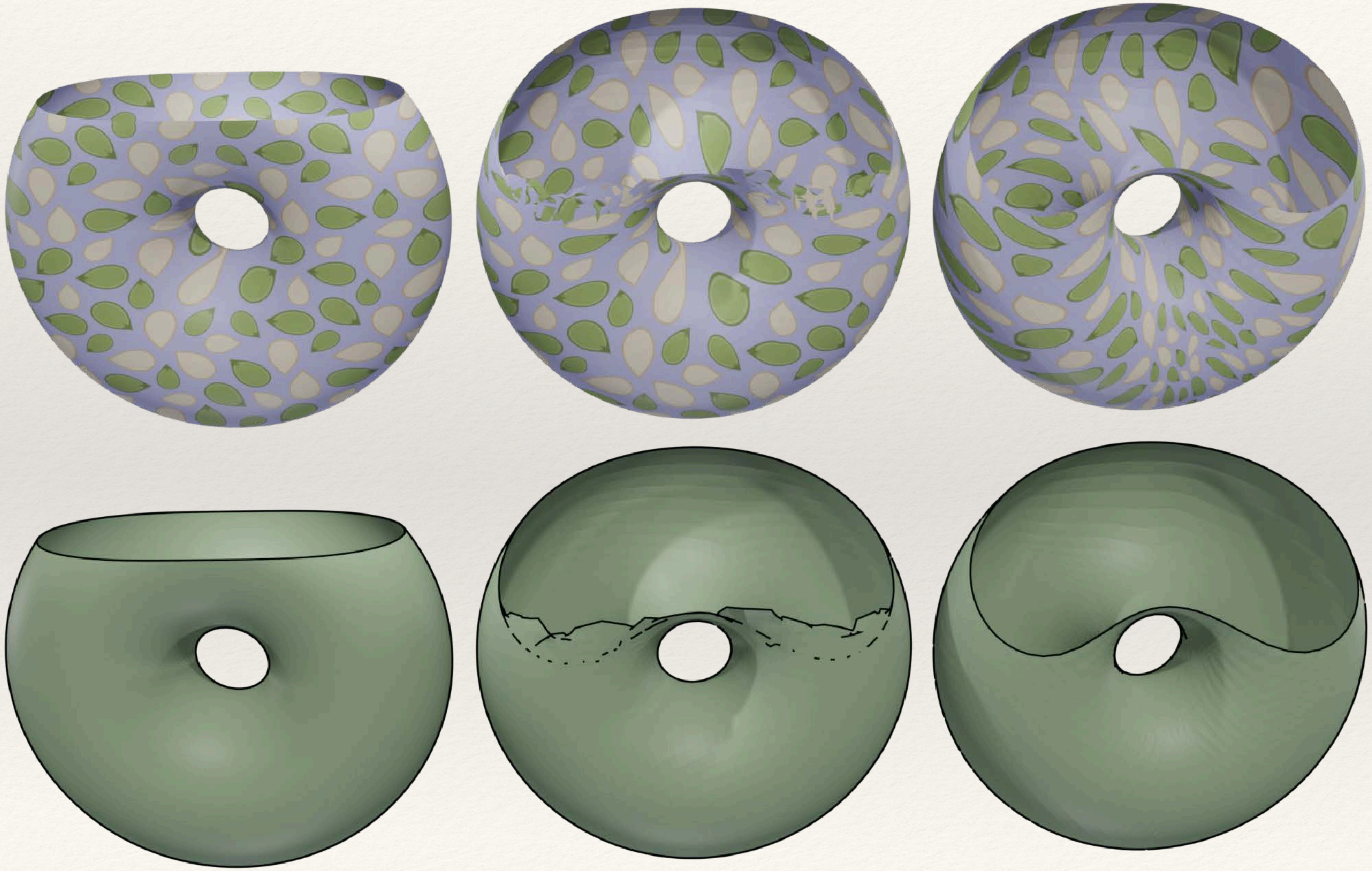
- What is the "best" QC map in a given homotopy class?

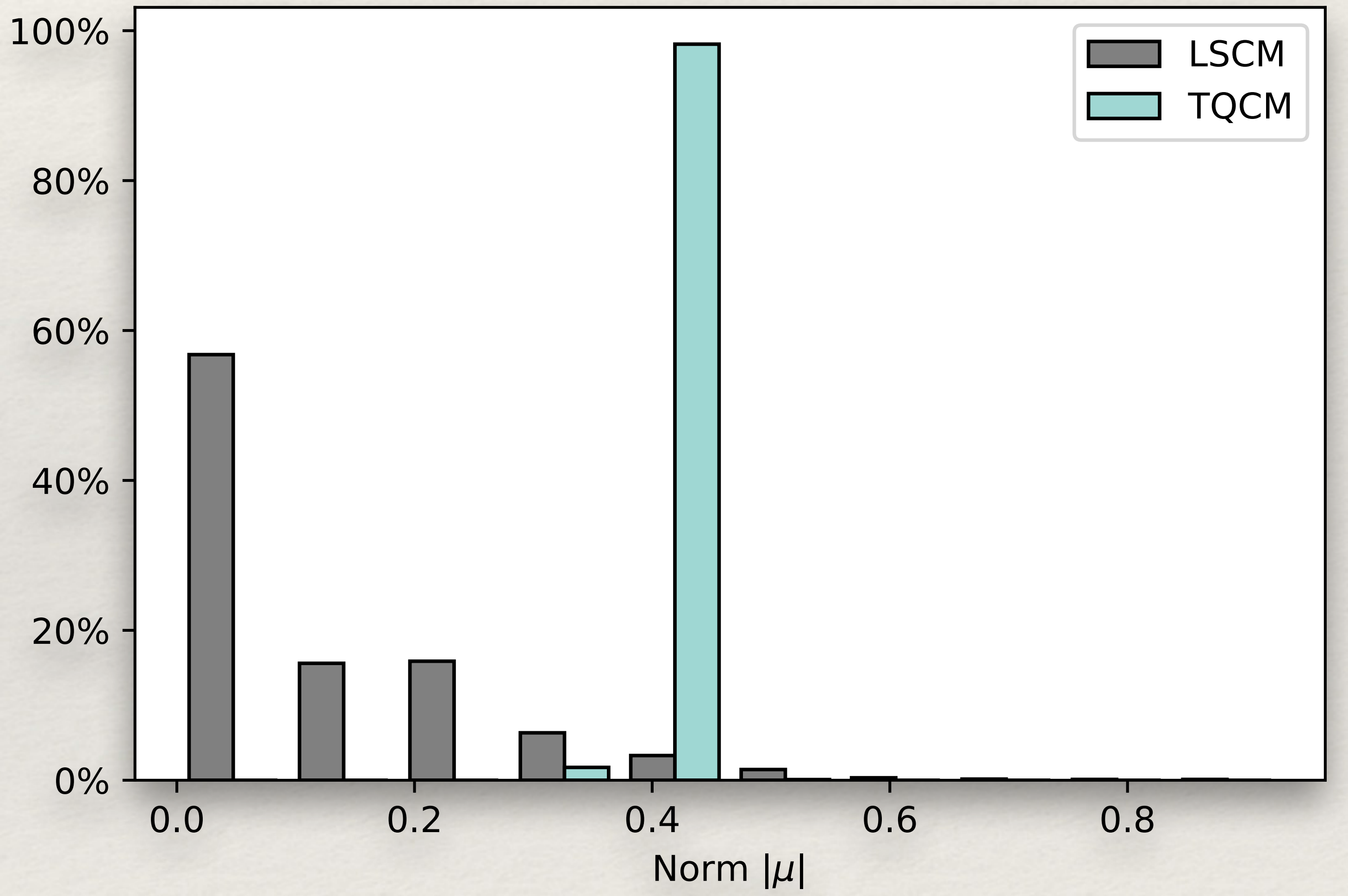
- Let $H\left([f]\right) = \inf_{h \in [f]} \left\{ \inf_{C \in M} K\left(h|_{M \backslash C}\right) \right\}$, where $K(f) = \dfrac{1 + |\mu|_\infty}{1 - |\mu|_\infty}$.

- (Strebel 1984) If $H\left([f]\right) < K(f)$ then $[f]$ contains a unique **Teichmuller** mapping.

- TM mappings have **constant** $|\mu|$ and **min-maxed conformality distortion.**

Comparison: TM vs. LSCM

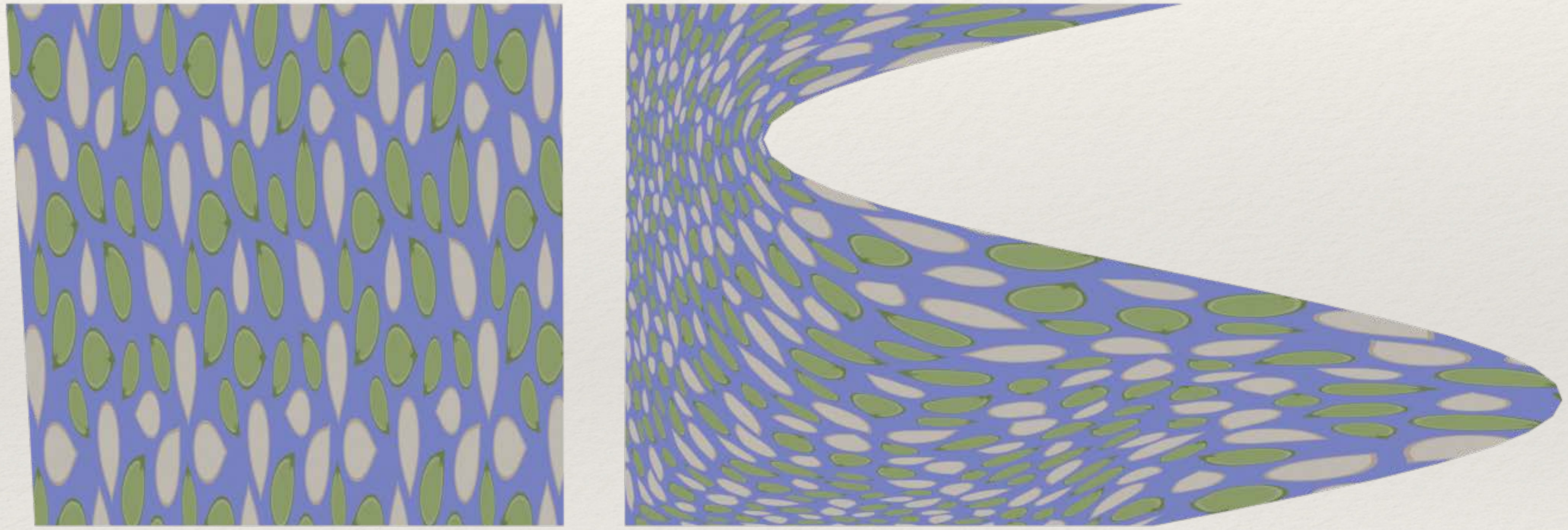




- Gruber, A. and Aulisa, E. 2021 (under review)

Computing TM mappings

- Minimize
  $$\mathcal{QC}(f) = \int_M |\, df^- - \mu\, df^+ |^2 \, d\mu_g$$
  alternatively over f, μ.
  - 1) Minimize for f given μ.
  - 2) Compute $\mu = df^- \left(df^+\right)^{-1}$.
  - 3) Locally adjust μ, moving it toward TM form (next slide)
  - Repeat steps 1-3 until convergence.



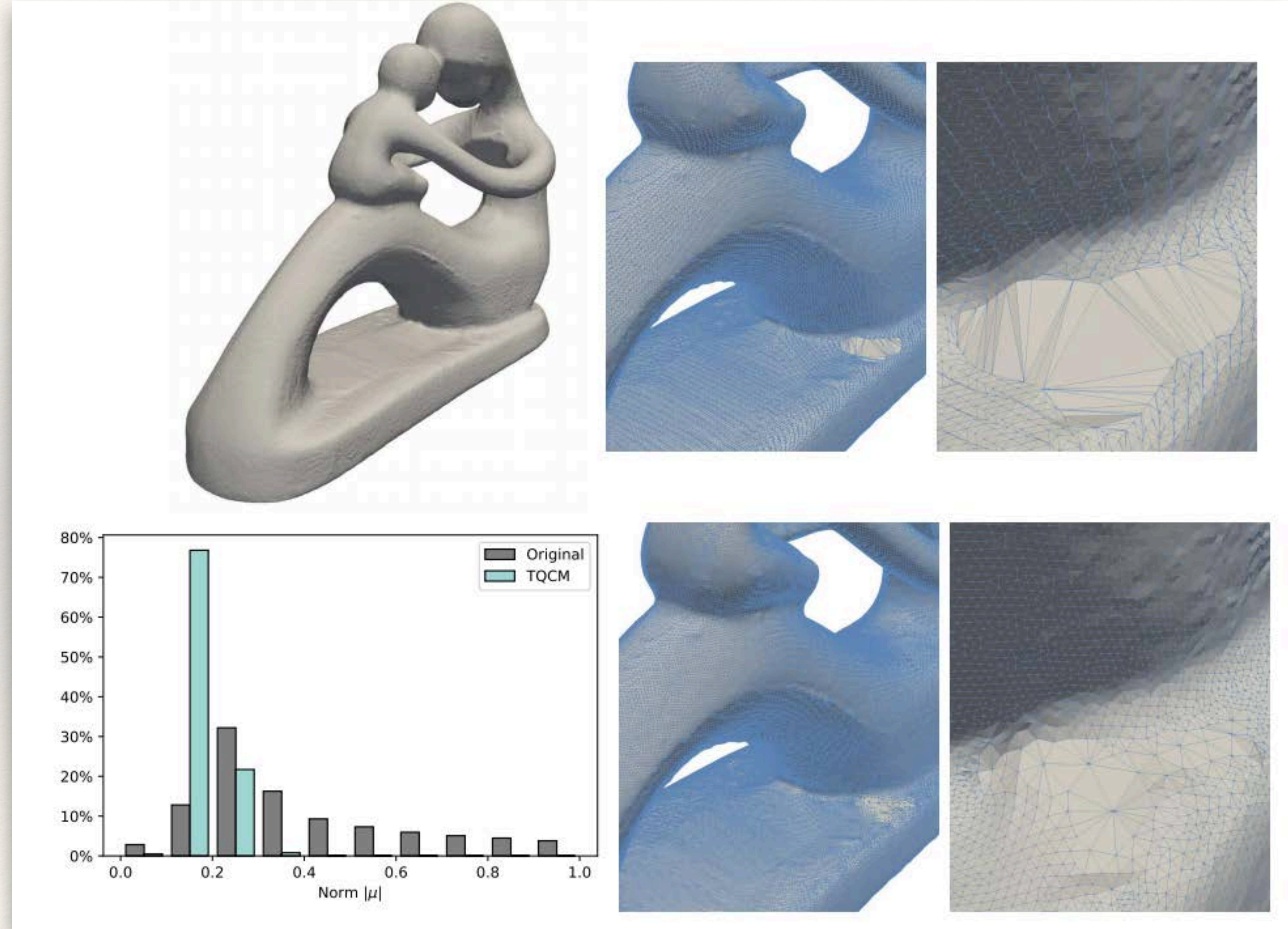
Why does it work?

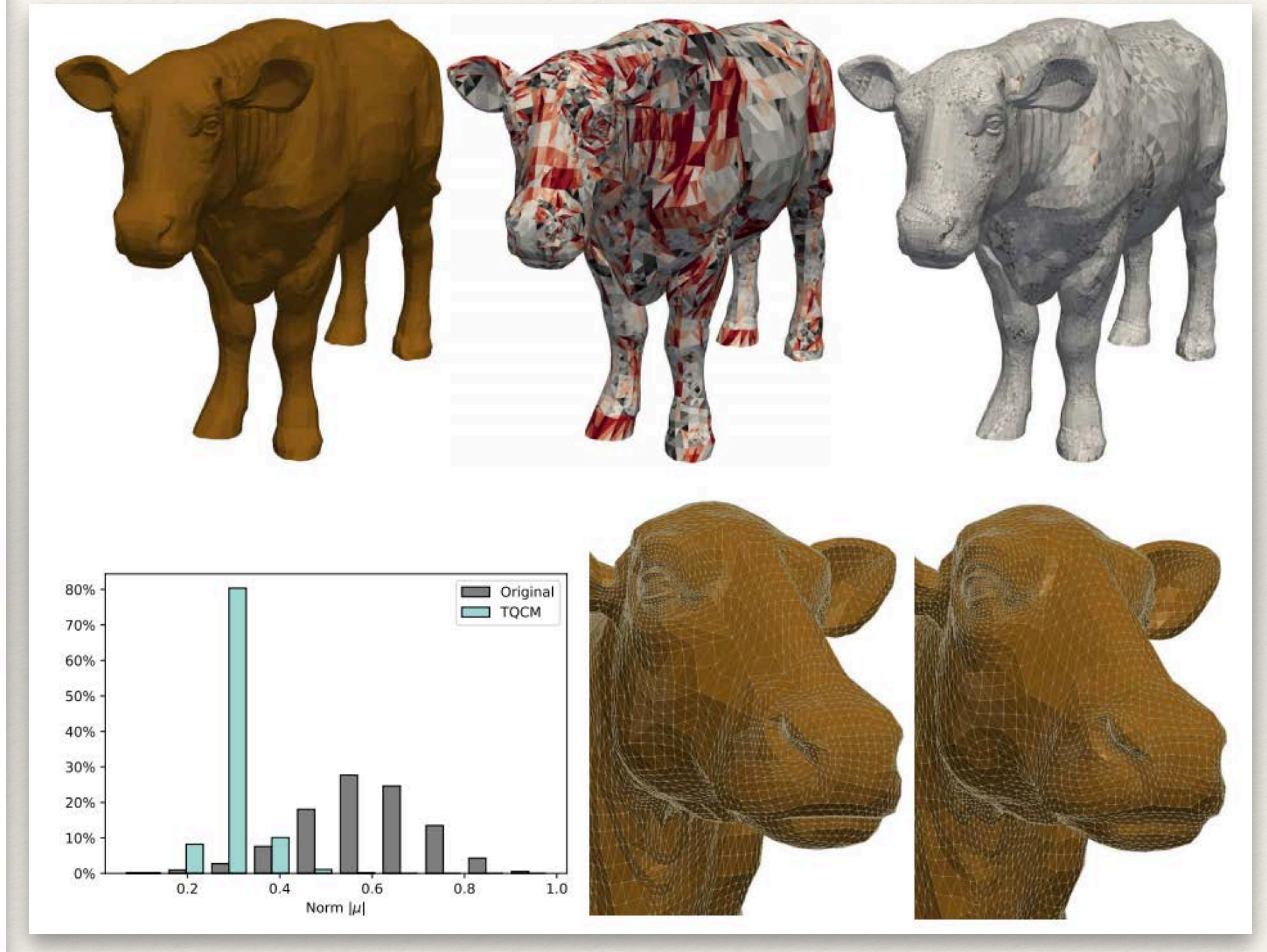
- Dirichlet energy w.r.t. any metric decomposes into **area functional** and **conformal distortion.**

$$\mathscr{D}_g(f) = \int_M |df|^2 = \int_M |df^+|^2 - |df^-|^2 + 2\int_M |df^-|^2 = \mathscr{A}(f) + 2\mathscr{C}\mathscr{D}(f)$$

- Can show that $\mathscr{Q}\mathscr{C}(f)$ is the conformal part of $\mathscr{D}_{g(\mu)}(f)$.
- (Metric $g(\mu)$ induced by QC mapping with BC μ.)

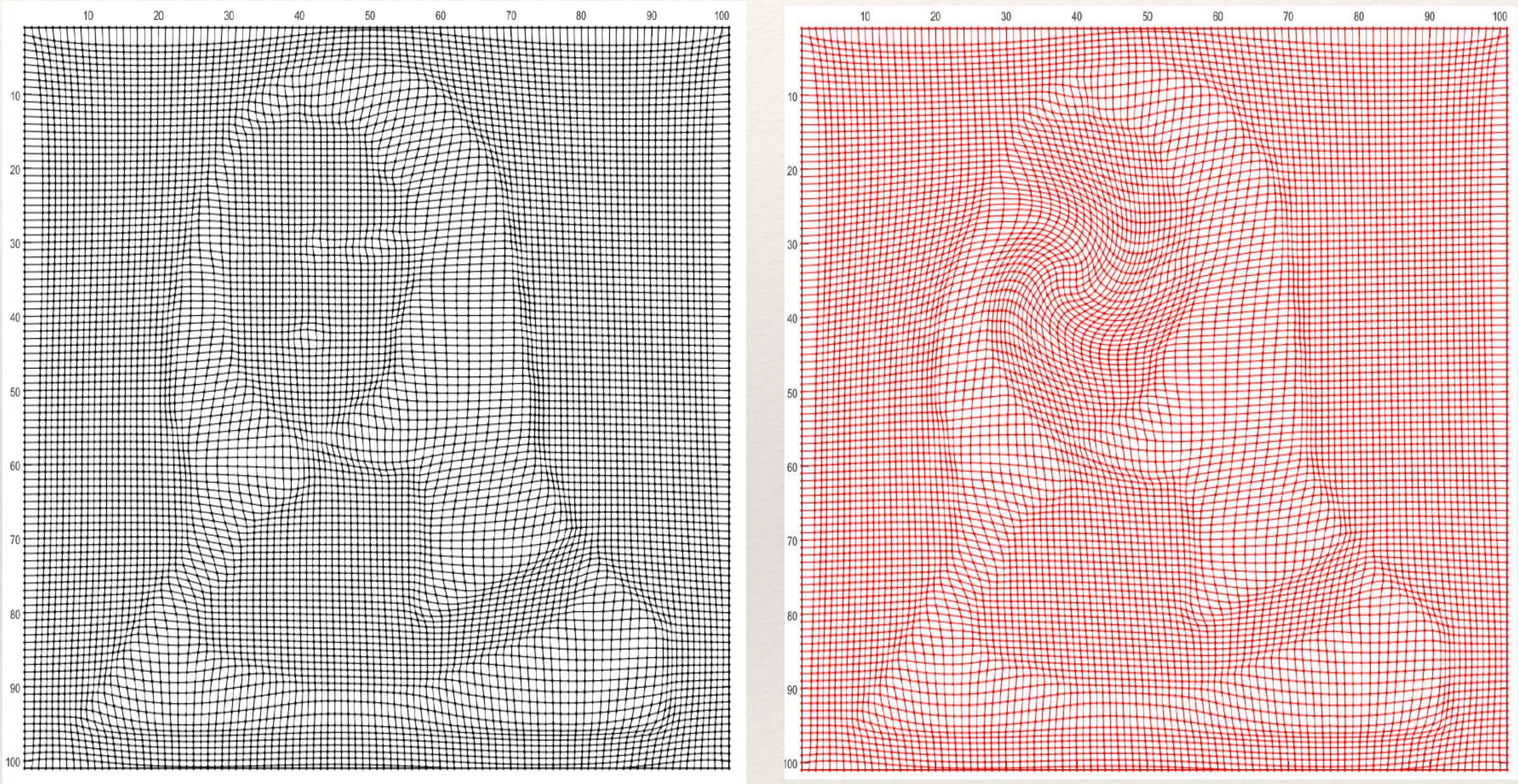
More Examples





Grid Generation

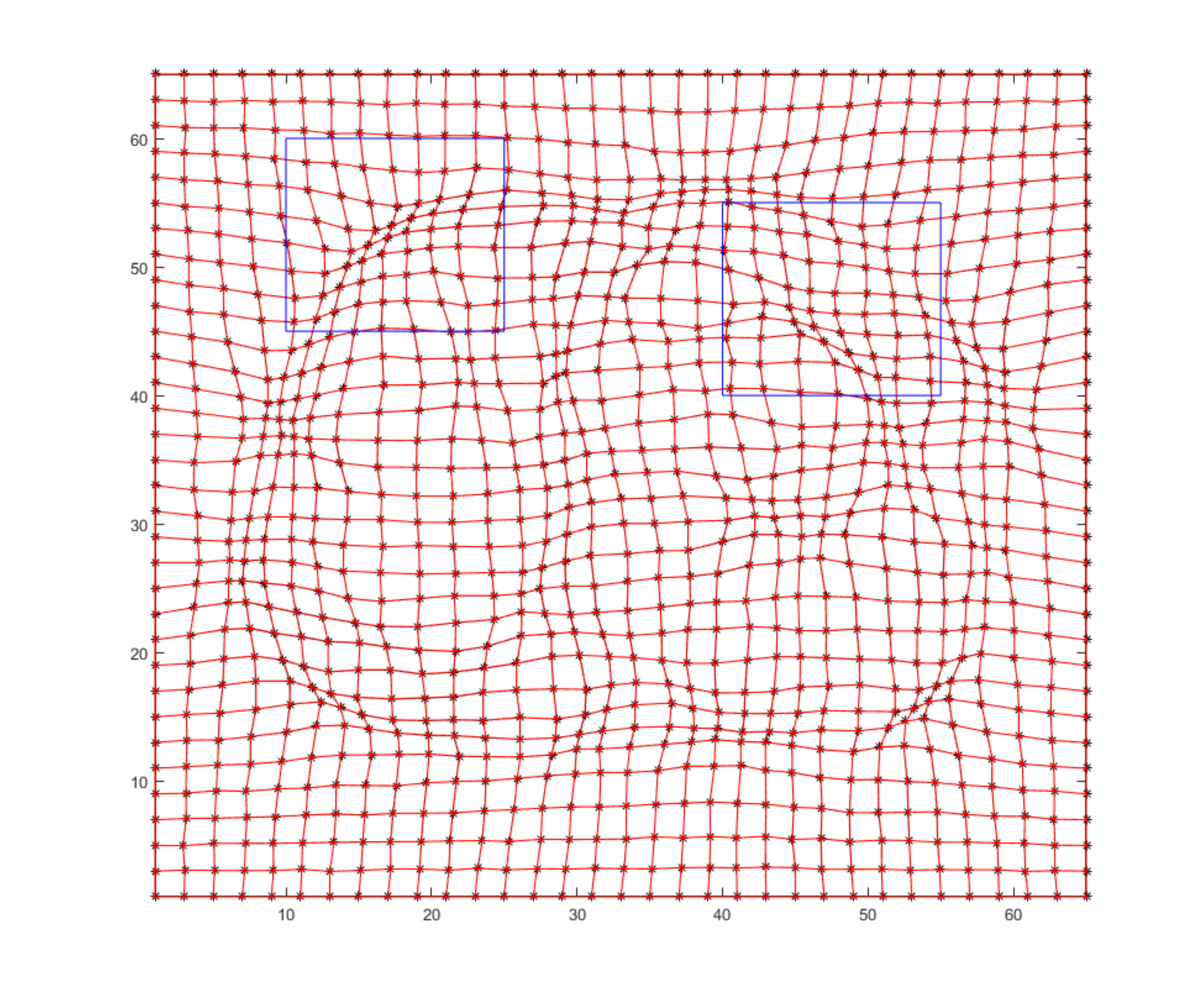
- Important for scientific computing.
- Standard is to fix **divergence, curl,** and **Jacobian determinant.**
- Prescribes **local** area and rotational distortion.

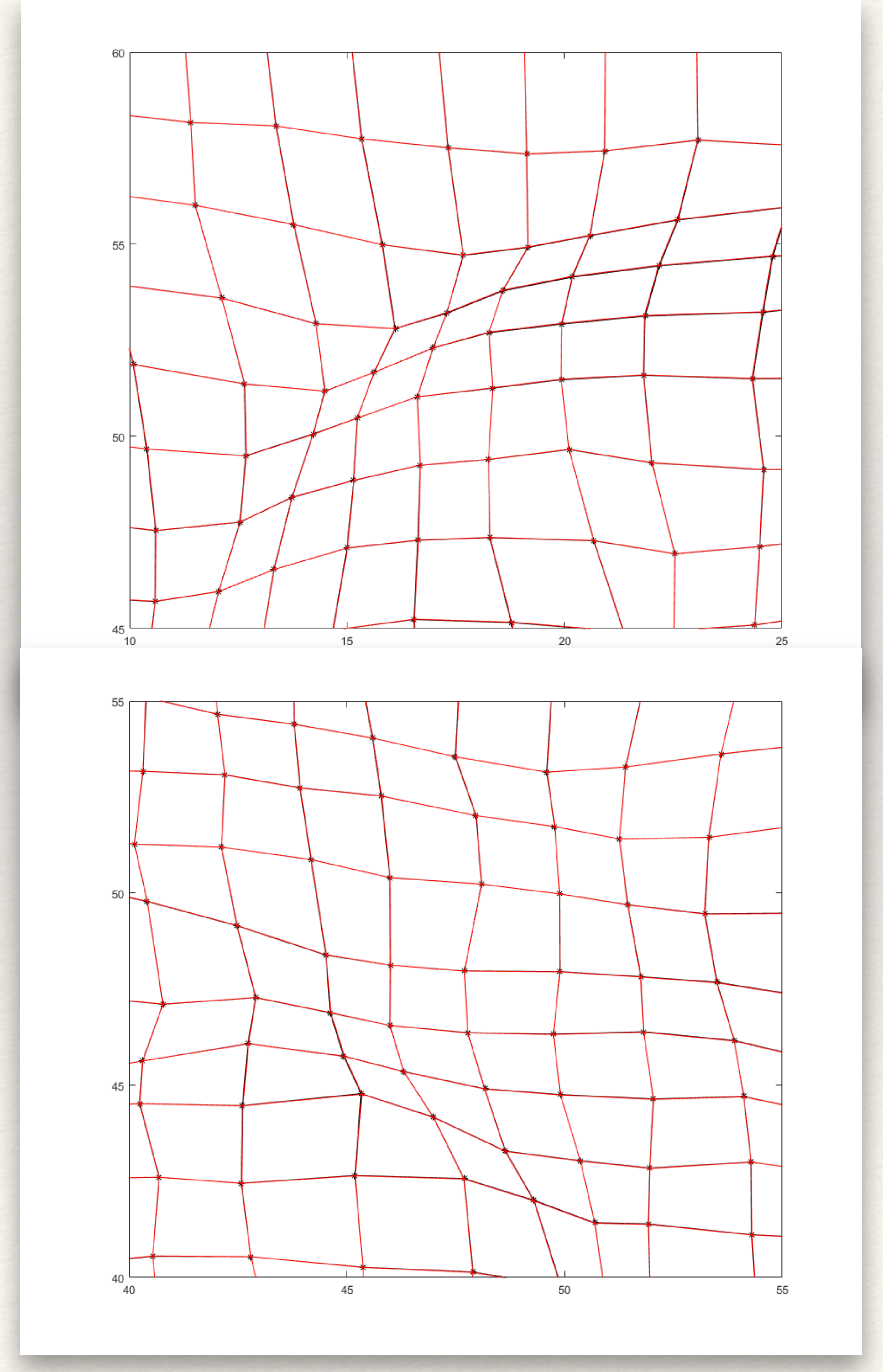


Images from Zhao et al., *Uniqueness of Transformations with Prescribed Jacobian Determinant and Curl-Vector* (2018)

Influence of Jac-det and Curl

- Researchers noticed **Jac-det + curl** sufficient for "numerical uniqueness" in 2-D.
- Z. Zhou, X. Chen, X. X. Cai, and G. Liao (2015) conjectured this holds in dimensions 2 and 3.





Images from Zhao et al., *Uniqueness of Transformations with Prescribed Jacobian Determinant and Curl-Vector* (2018)

Uniqueness Conjecture

- Consider $\phi, \psi : U \subset \mathbb{R}^n \to \mathbb{R}^n$ where $n = 2, 3$.
- Suppose:
  - $\nabla \times \phi = \nabla \times \psi \quad \text{on } U,$
  - $\operatorname{Jac} \phi = \operatorname{Jac} \psi \quad \text{on } U,$
  - $\phi = \psi \quad \text{on } \partial U.$
- Does it follow that $\phi \equiv \psi$ on U?

Dimension Two

- Conjecture is **true** in dimension two.

  A. Gruber, *Bull. Aust. Math. Soc.* (in press)

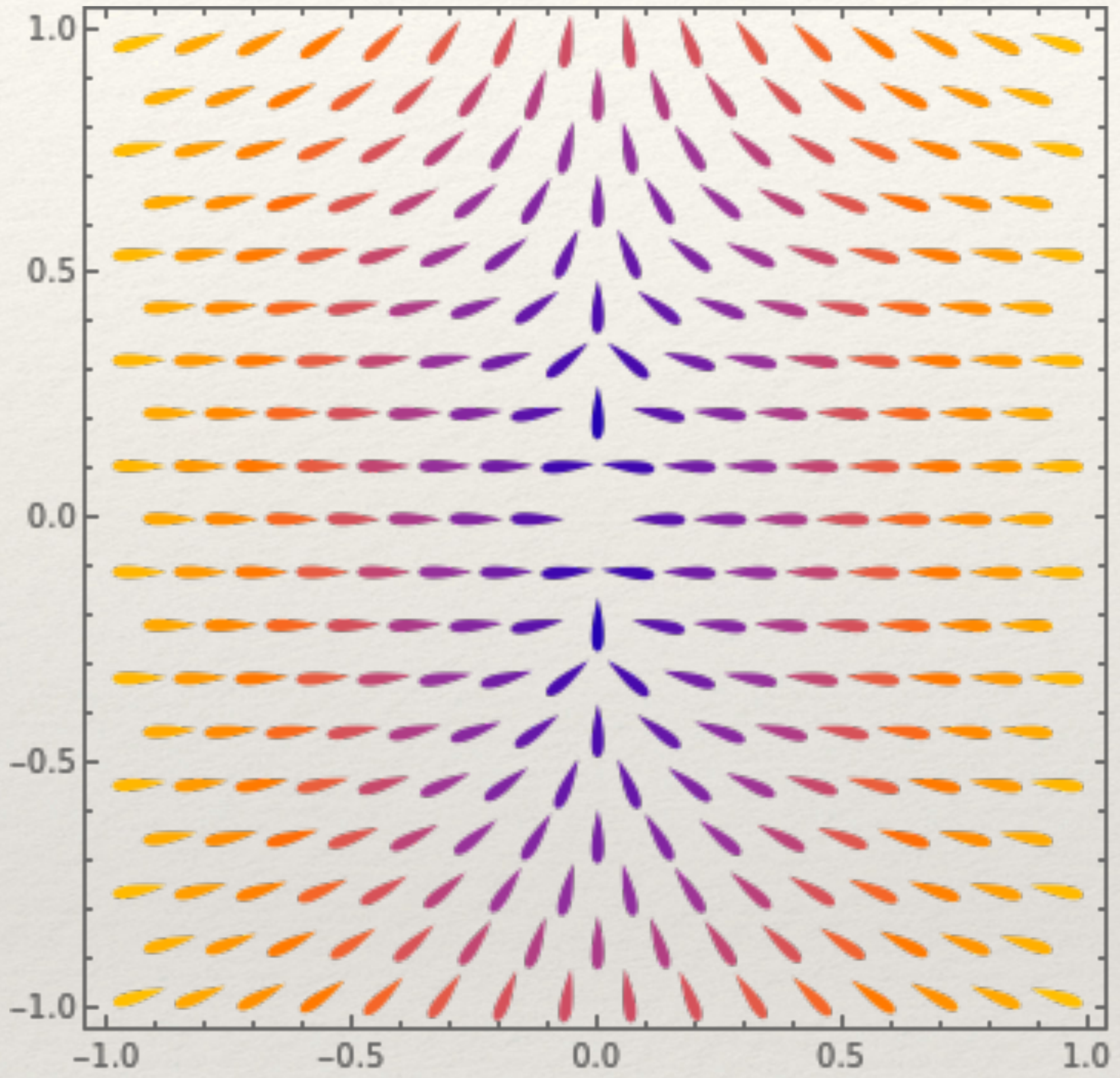
- Proof uses tools from Clifford “geometric” algebra.
- Generated by symbols $\{1, \mathbf{e}_1, \mathbf{e}_2\}$ with relations $\mathbf{e}_1^2 = \mathbf{e}_2^2 = 1, \; J := \mathbf{e}_1\mathbf{e}_2 = -\mathbf{e}_2\mathbf{e}_1.$
- (Presentation isomorphic to 2x2 matrices over $\mathbb{R}$)

Potential
Copyright
Issue

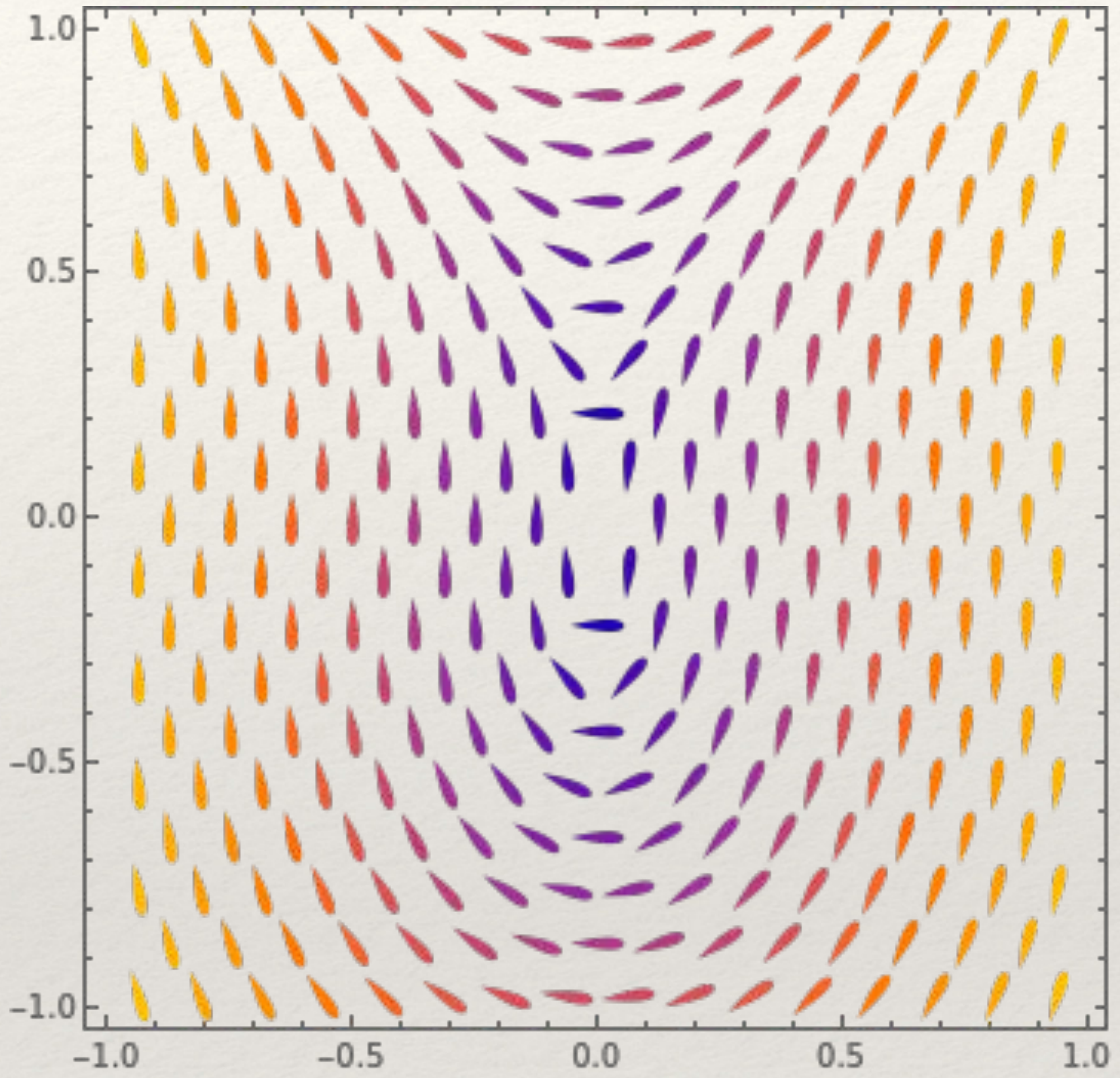
Images: https://www.euclideanspace.com/maths/algebra/clifford/d2/index.htm

Dual immersions

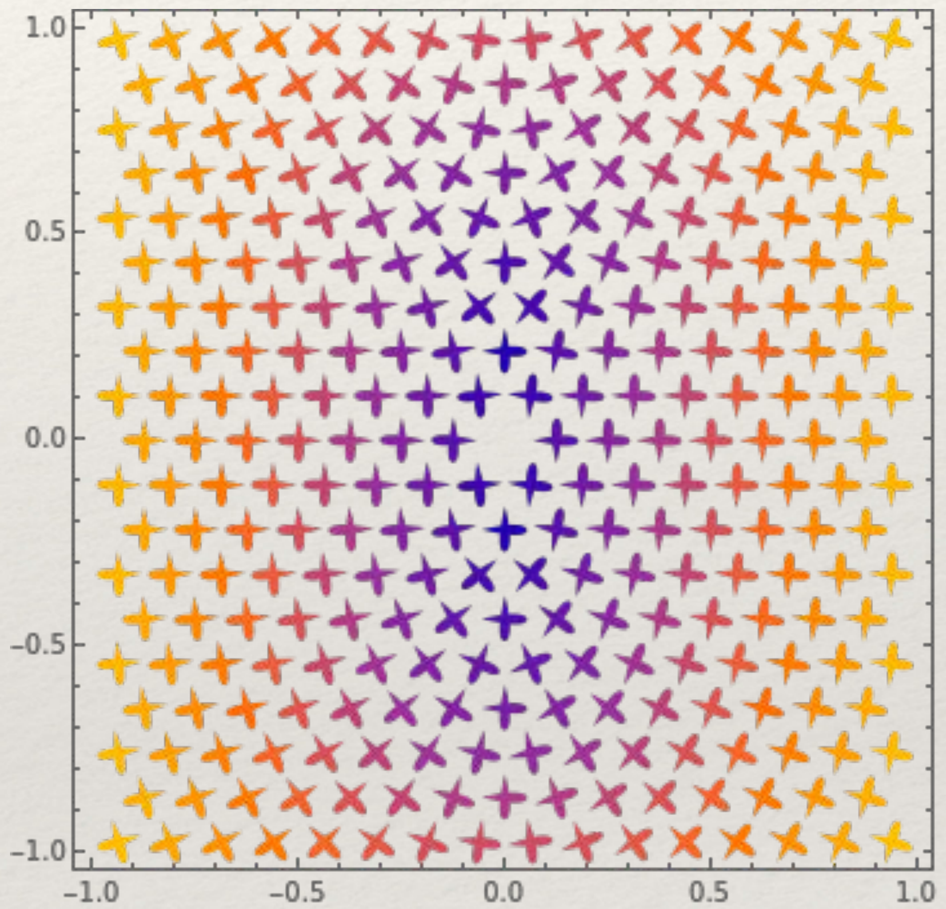
- Given immersion $\phi : M \subset \mathbb{R}^2 \to \mathbb{R}^2$.
- Then, $J\phi$ is immersive since $\ker J = 0$.
- Why do this? Converts curl into something useful…



ϕ



$J\phi$



(Pictured mappings not immersive when y=0)

Duality trick

- Notice that:

$$J(\mathbf{v} \cdot \mathbf{w}) = \mathbf{v} \wedge (J\mathbf{w}),$$
$$J(\mathbf{v} \wedge \mathbf{w}) = \mathbf{v} \cdot (J\mathbf{w}) .$$

- It follows that:

$$\det d(J\phi) = -J\left((J\phi)_1 \wedge (J\phi)_2\right) = (J\phi_1) \cdot \phi_2 = J\left(\phi_2 \wedge \phi_1\right) = \operatorname{Jac}\phi .$$
$$\operatorname{tr} d(J\phi) = \nabla \cdot (J\phi) = J\left(\nabla \wedge \phi\right) = \nabla \times \phi$$

- Gives characteristic polynomial:

$$p(\lambda) = \det(\lambda I - d(J\phi)) = \lambda^2 - (\nabla \times \phi)\lambda + \operatorname{Jac}\phi$$

Duality trick

- Can compute EWs/EVs!
- Metric is computable, and same as g:

$$d(J\phi) \cdot d(J\phi) = J\,d\phi \cdot J\,d\phi = d\phi \cdot d\phi$$

- Problem is converted to **uniqueness given metric and BVs**.
- Can show with more argument that this is true.

Consequences

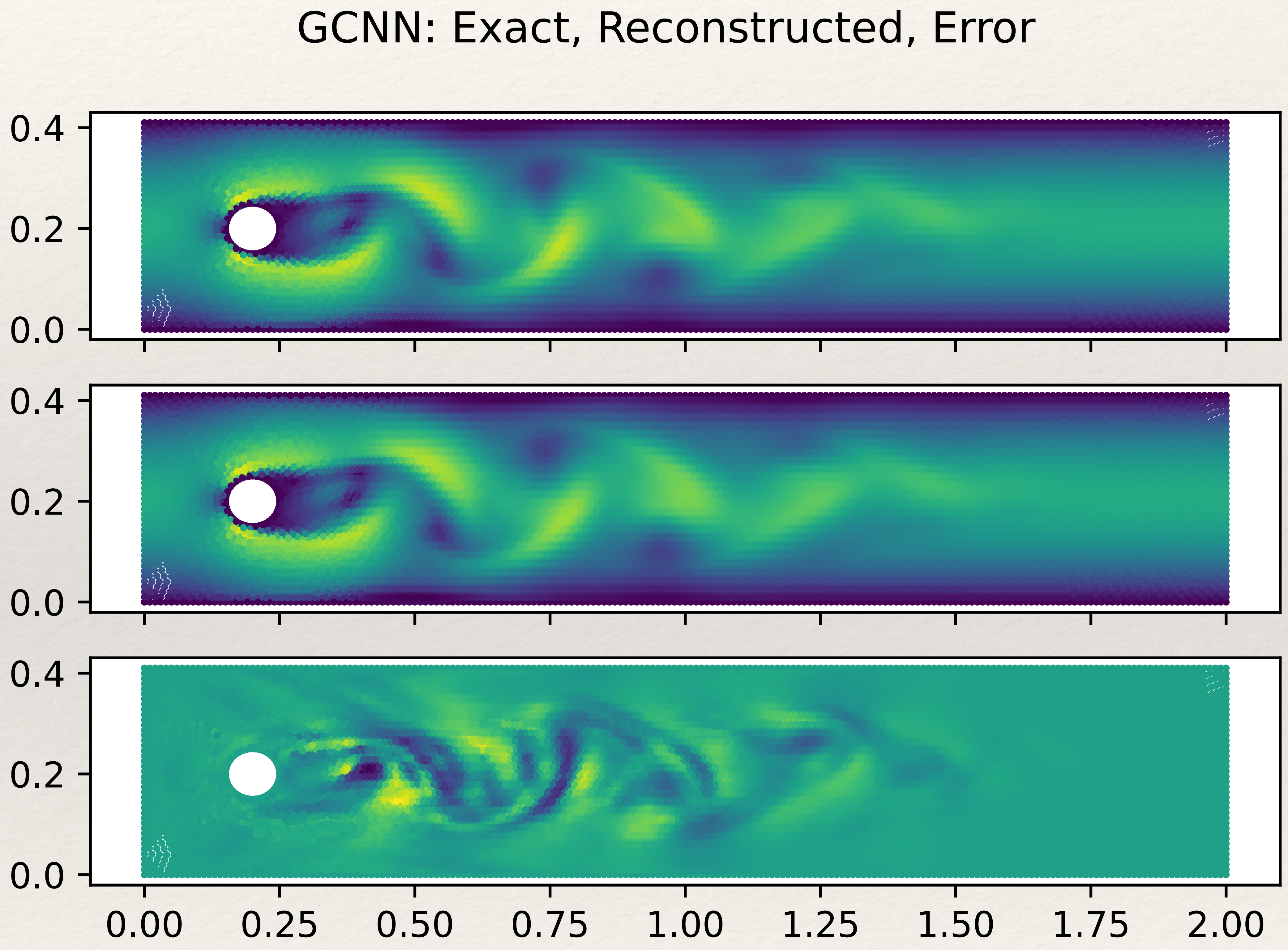
- Prescribed curl/Jac-det **even better** than div/curl in 2-D.
- Not that good in 3-D (not surprising), can't get metric…
- Example) $\phi = (y \quad z \quad x)^\top, \quad \psi = (y - x \quad z \quad x)^\top.$
  - $\operatorname{Jac}\phi = \operatorname{Jac}\psi = 1, \quad \nabla \times \phi = \nabla \times \psi = -(1 \quad 1 \quad 1).$ But…

$$d\phi \cdot d\phi = dx^2 + dy^2 + dz^2,$$
$$d\psi \cdot d\psi = 2dx^2 - 2dxdy + dy^2 + dz^2,$$

- Would be nice to (dis)prove conjecture for n=3.

Outline

- Introduction
- Examples in computer graphics
  - p-Willmore flow; quasiconformal mappings; grid generation
  - Partially funded by NSF DMS #1912902, 1912705
- **Examples in data science**
  - ANN function approximation; reduced order modeling
  - Partially funded by DOE DE-SC0020418



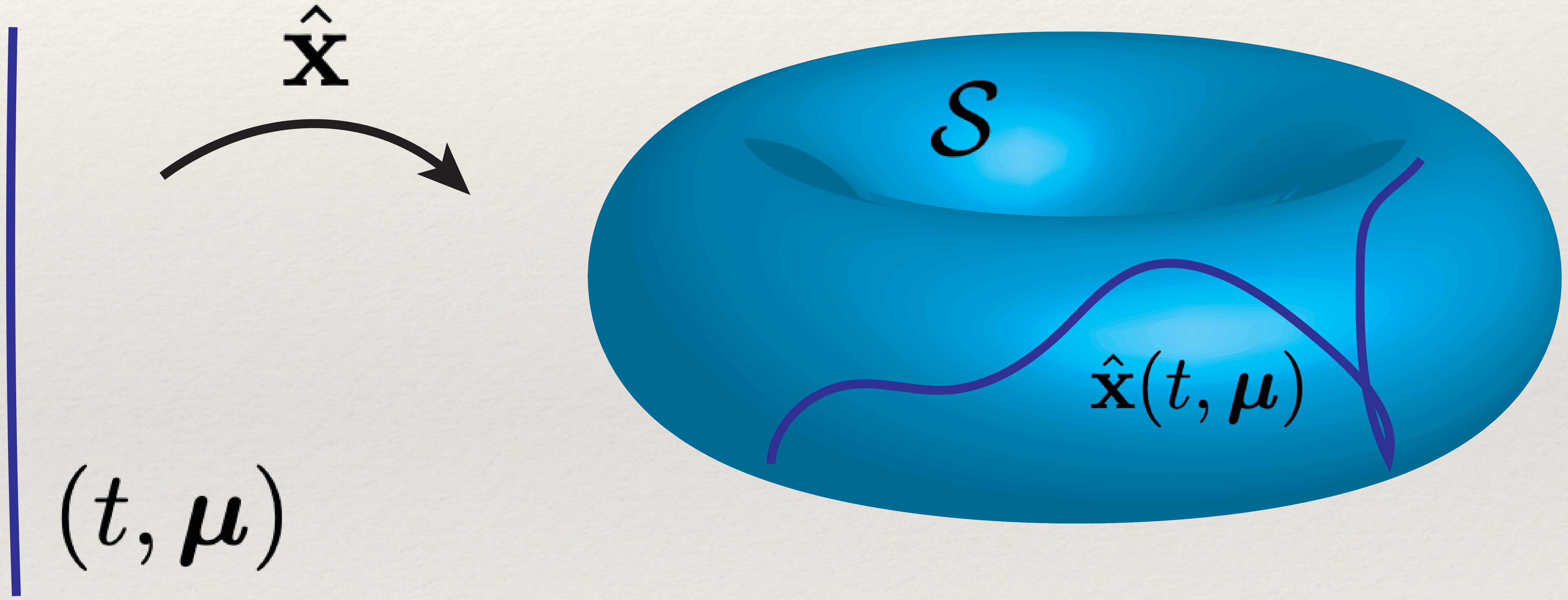
Approximation of Functions

- Often need the value of a measurement which is "expensive".
- Ex) DFT observables; infectious disease metrics, FEM consequences.
- Can depend on many parameters.
- Limited data requires informed techniques for *dimension reduction*.

Image: https://mpas-dev.github.io/ocean/ocean.html

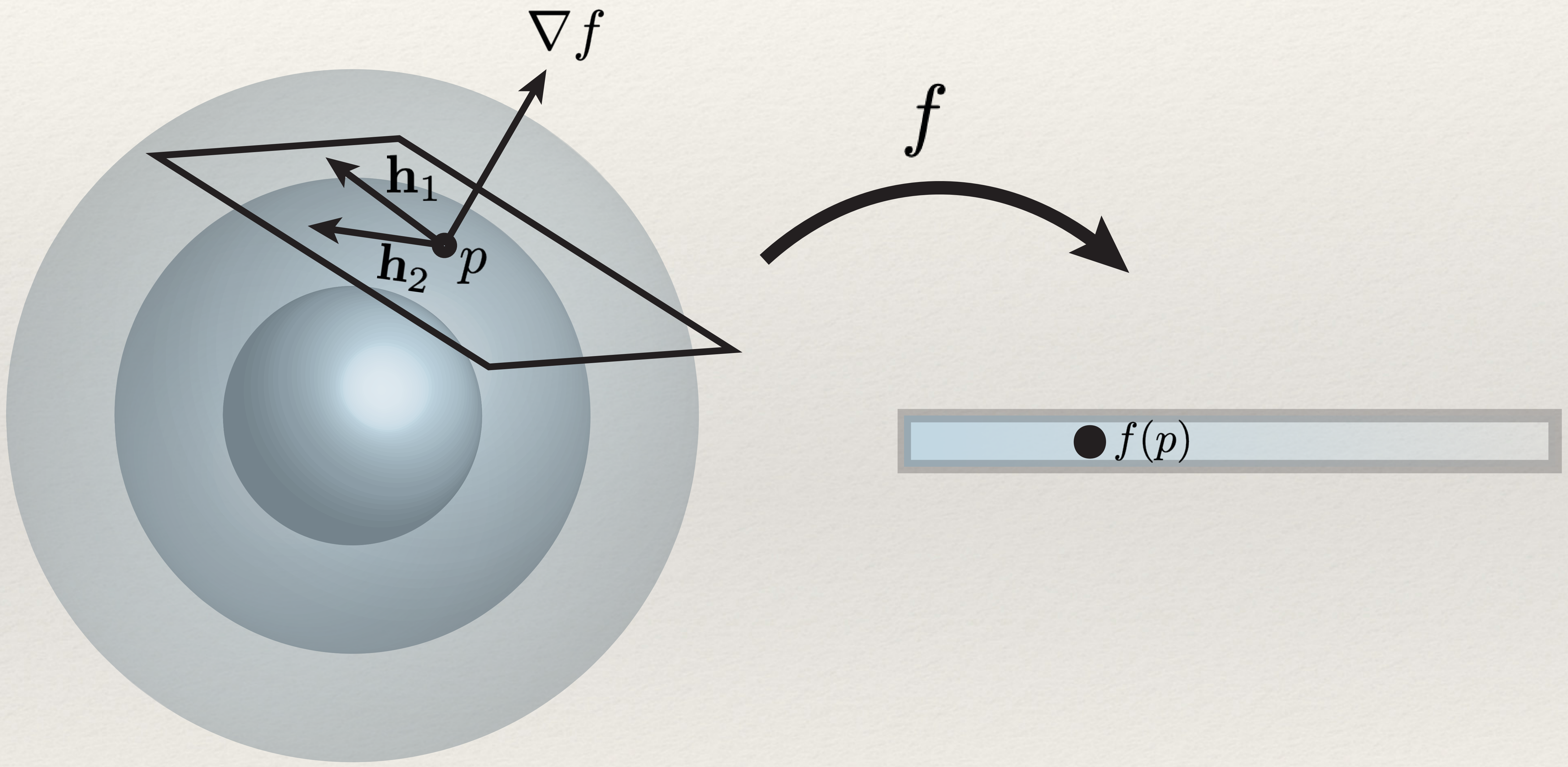
Two Broad Approaches

- 1) Data is *intrinsically* low-dimensional
  - DR should look for intrinsic features
- Clustering, reduced basis methods, etc.



Two Broad Approaches

- 2) Low-dim structure is *induced* by external mapping
  - Structure depends implicitly on objective
- Ridge regression, active manifolds, nonlinear level set learning

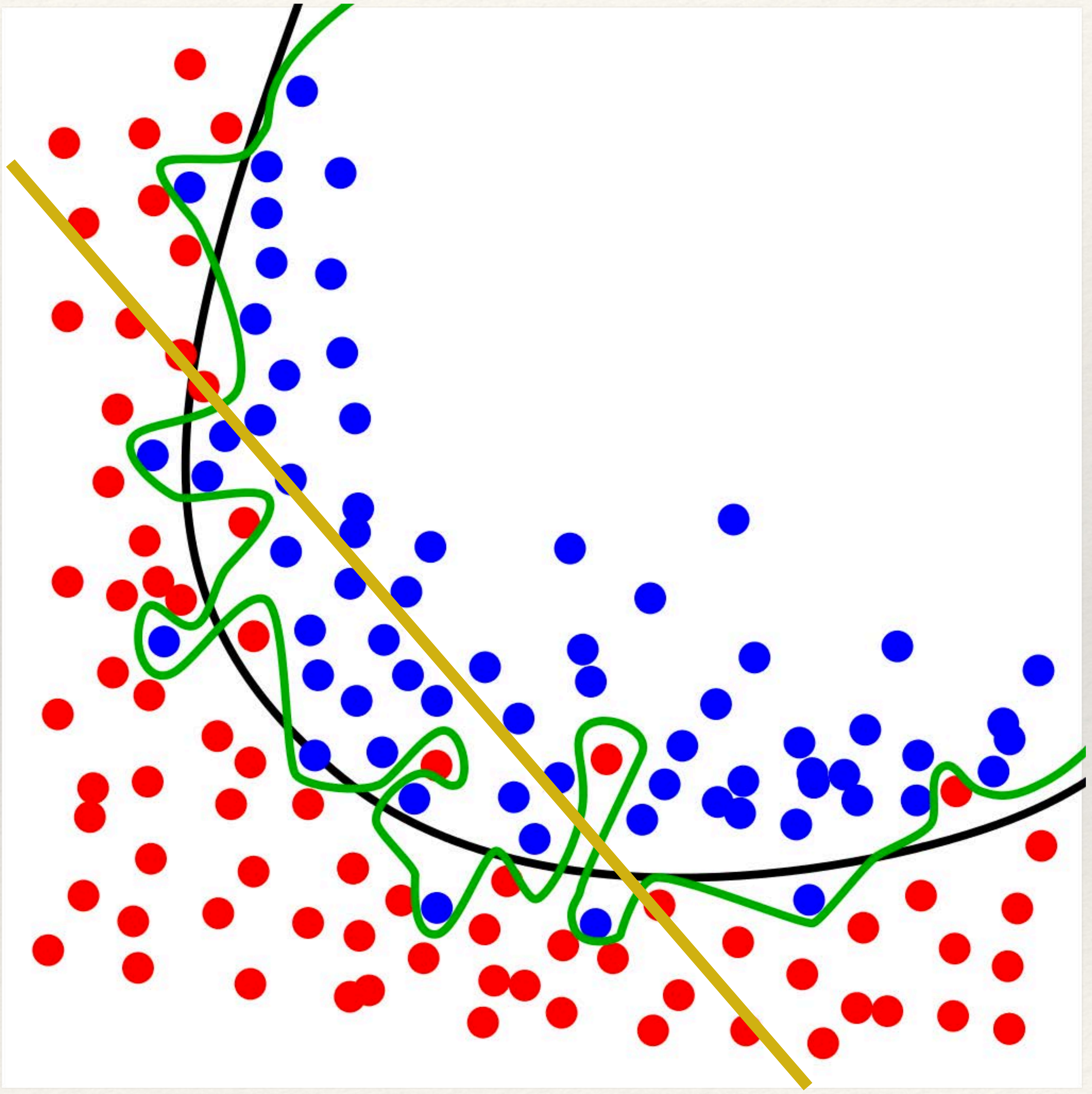


Extrinsic Approximation

- Want to approximate $f : U \subset \mathbb{R}^n \to \mathbb{R}$ for large n.
- Given samples $\{\mathbf{x}^s, f(\mathbf{x}^s), \nabla f(\mathbf{x}^s)\}_{s\in S}$ the goal is:

$$\tilde{f}(\mathbf{x}) \in \underset{g\in C^1(U)}{\arg\min} \|f(\mathbf{x}) - g(\mathbf{x})\|_2^2$$

- Classical methods *underparameterized*, e.g. least squares.
- Naive ANN method highly *overparameterized* — poor generalization ability.



Original by Chabacano - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=3610704

Extrinsic: Ridge Regression

- Instead, look for $\hat{f}(\mathbf{x}) \in \underset{g:\mathbb{R}^k \to \mathbb{R}}{\arg\min} \|f(\mathbf{x}) - g(P\mathbf{x})\|_2^2$

  where $P : \mathbb{R}^n \to \mathbb{R}^k$ is projection operator.

- Good if f is nearly constant on the kernel of P.

- Goal is optimization over f, P.

Potential Copyright Issue

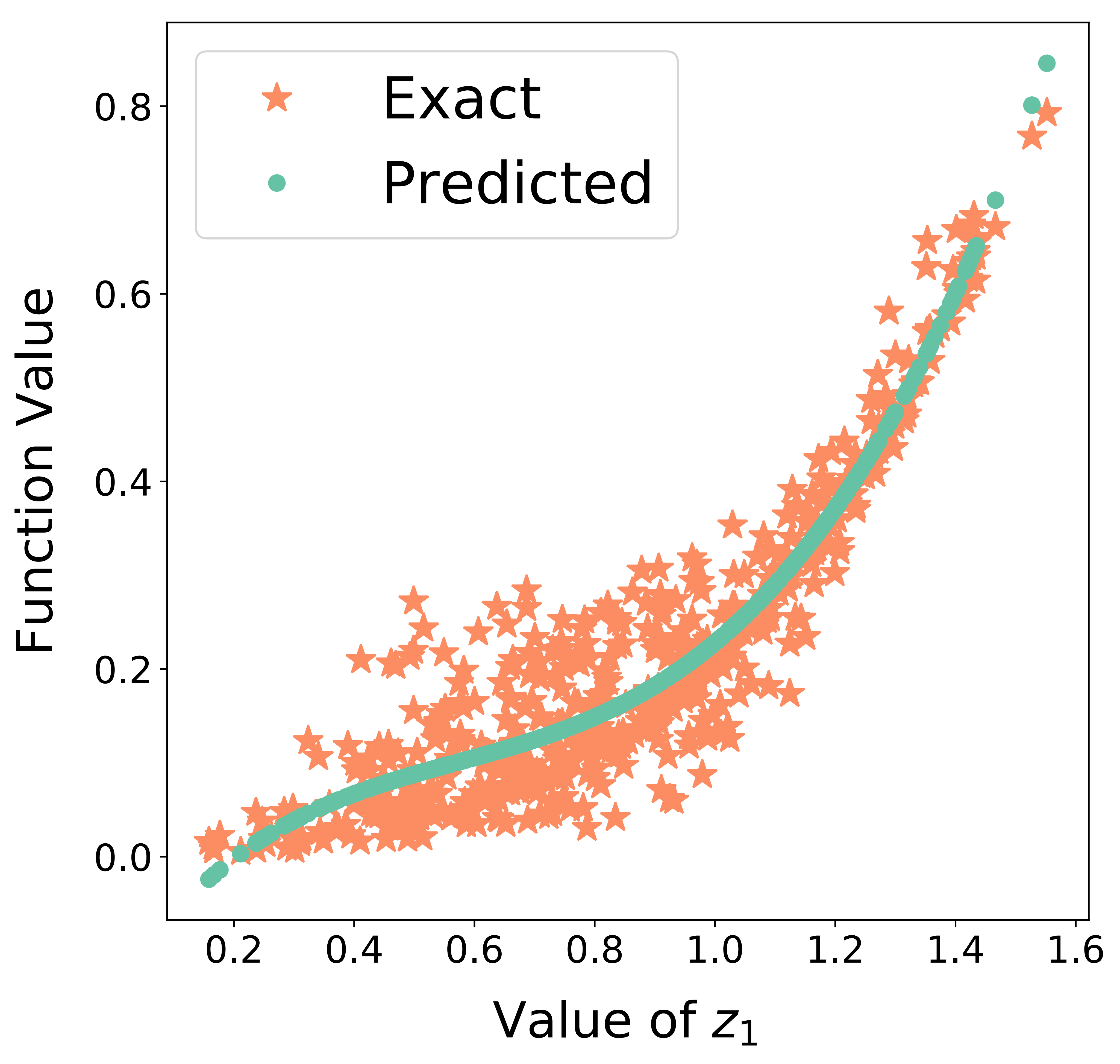
Image by Matthias Scholz: http://www.nlpca.org/pca_principal_component_analysis.html

Established Linear Methods

- Active Subspaces (P. Constantine 2014):
- Linear reduction through SVD: $\mathbf{U\Sigma V} = \int_U \nabla f (\nabla f)^\top \, d\mu$
- Columns of $\mathbf{U}$ give basis of "most important" directions.
- Projection map is a column-wise truncation of $\mathbf{U}$.

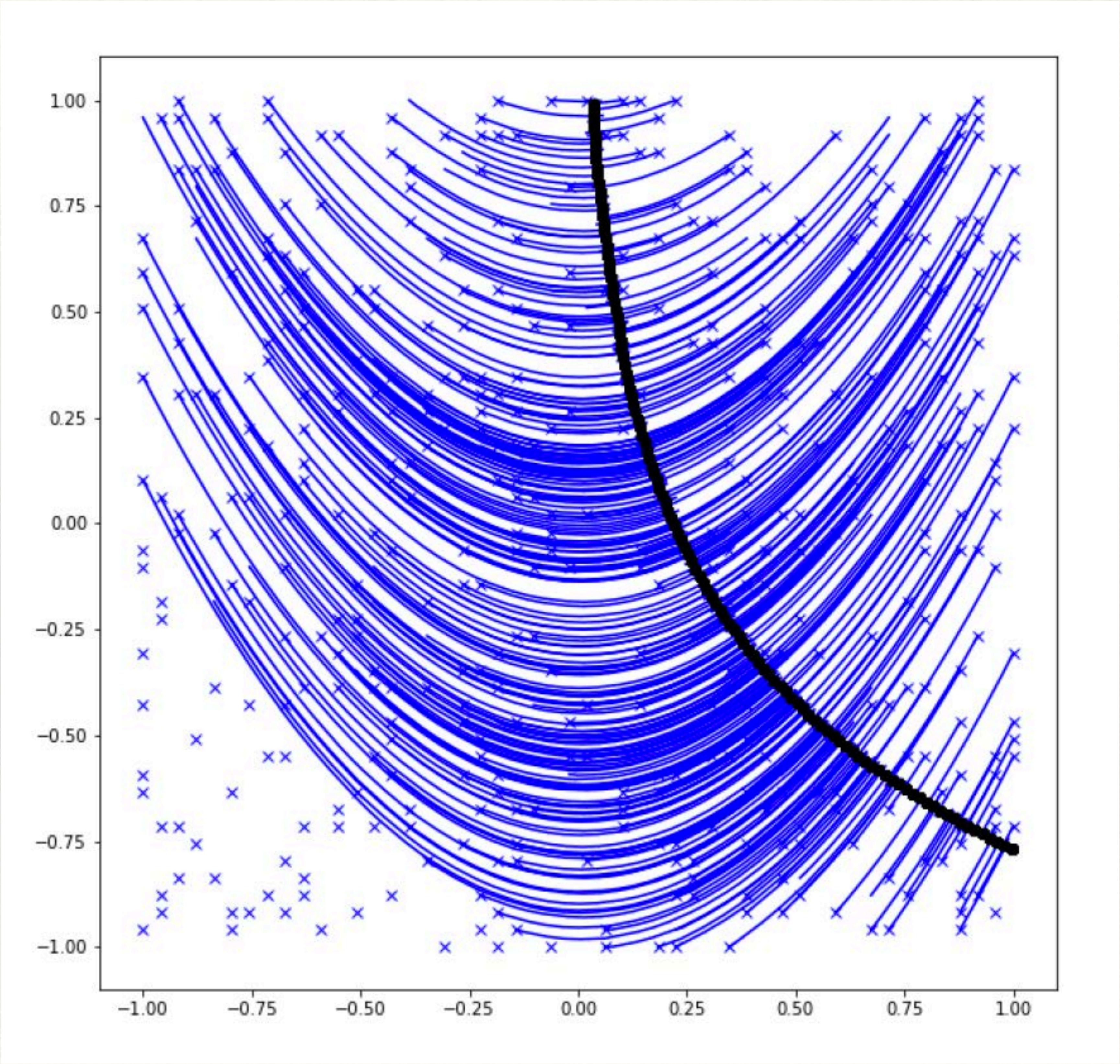
Established Linear Methods

- AS limited by linearity.
- Prediction of kinetic energy arising from parameterized 1-D Burger's equation:
- ~20% relative ℓ_2 error despite only 2-D reduction.



Nonlinear Options

- **Active Manifolds** idea of (R. Bridges, A. Gruber et al. 2019):
- Solve $\dot{\mathbf{x}} = \nabla f$, then $t \mapsto f(\mathbf{x}(t))$ characterizes f on the set $\{f^{-1}(f(\mathbf{x}(t)))\}_{t \in T}$
- Can compute projection $\pi(\mathbf{y}) = t$ by "walking level sets".
- AM works well in low dimensions, or when data is readily available.
    - Main drawback is online cost: need to solve an ODE for each evaluation.



R. Bridges, A. Gruber, C. Felder, M. Verma, C. Hoff, ICML 2019

Nonlinear Level set Learning

- Another idea due to (G. Zhang et al. 2019):
  - Improved by us: (joint with M. Gunzburger, L. Ju, Z. Wang, Y. Teng)
- Goal: extend AS to nonlinear maps through ANN technology.
- Seek invertible transformation $\mathbf{z} = \mathbf{g}(\mathbf{x})$ $\mathbf{h} \circ \mathbf{g} = \mathbf{I}$
  - Splits domain of $f \circ \mathbf{h}$ into pairs $(\mathbf{z}_A, \mathbf{z}_I)$
- Truncating the $\mathbf{z}$-domain gives low-dimensional representation.

$$\begin{tikzcd}
& X \arrow[r, shift right, "\mathbf{g}(x)"'] \arrow[dl, "f(x)"] & Z \arrow[l, shift right, "\mathbf{h}(z)"'] \arrow[r, "\pi_1"] \arrow[dl, "f \circ \mathbf{h}(z)"] & Z_{red} \arrow[dlll, bend left, "\hat{f}(z_1)"] \\
{} & {} & &
\end{tikzcd}$$

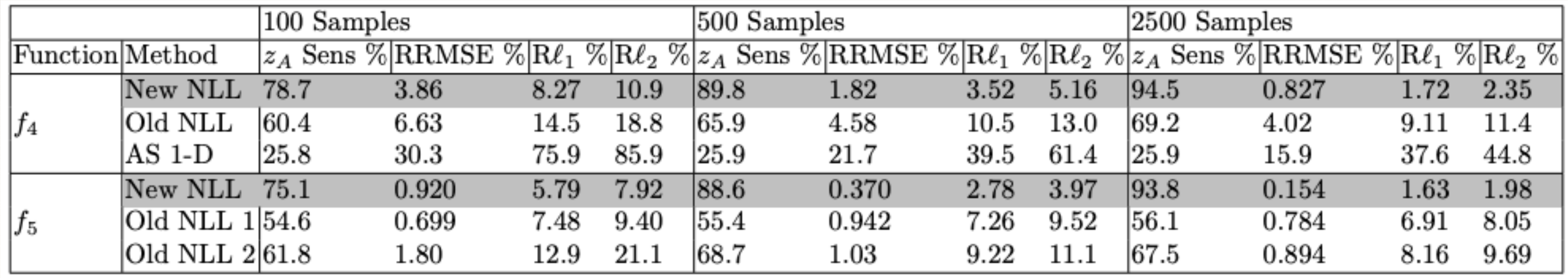
Important Ingredients

- Architecture is RevNet (Gomez et al. 2017).
  - Invertible modification of ResNet, $\mathbf{y} = \mathbf{x} + \mathbf{F}(\mathbf{x})$.
- Loss functional was significantly improved in (Gruber et al. 2021).
  - Based on observation that $\|(f \circ \mathbf{h})'(\mathbf{z})\|_{\perp}^{2} = \|\langle \nabla f(\mathbf{x}), \mathbf{h}_i(\mathbf{z})\rangle\|^{2}$.
- Can minimize:

$$L(\mathbf{h}) = \frac{1}{|S|} \sum_{s \in S} \|(f \circ \mathbf{h})'(\mathbf{z}^{s})\|_{\perp}^{2},$$

Results on Toy Examples

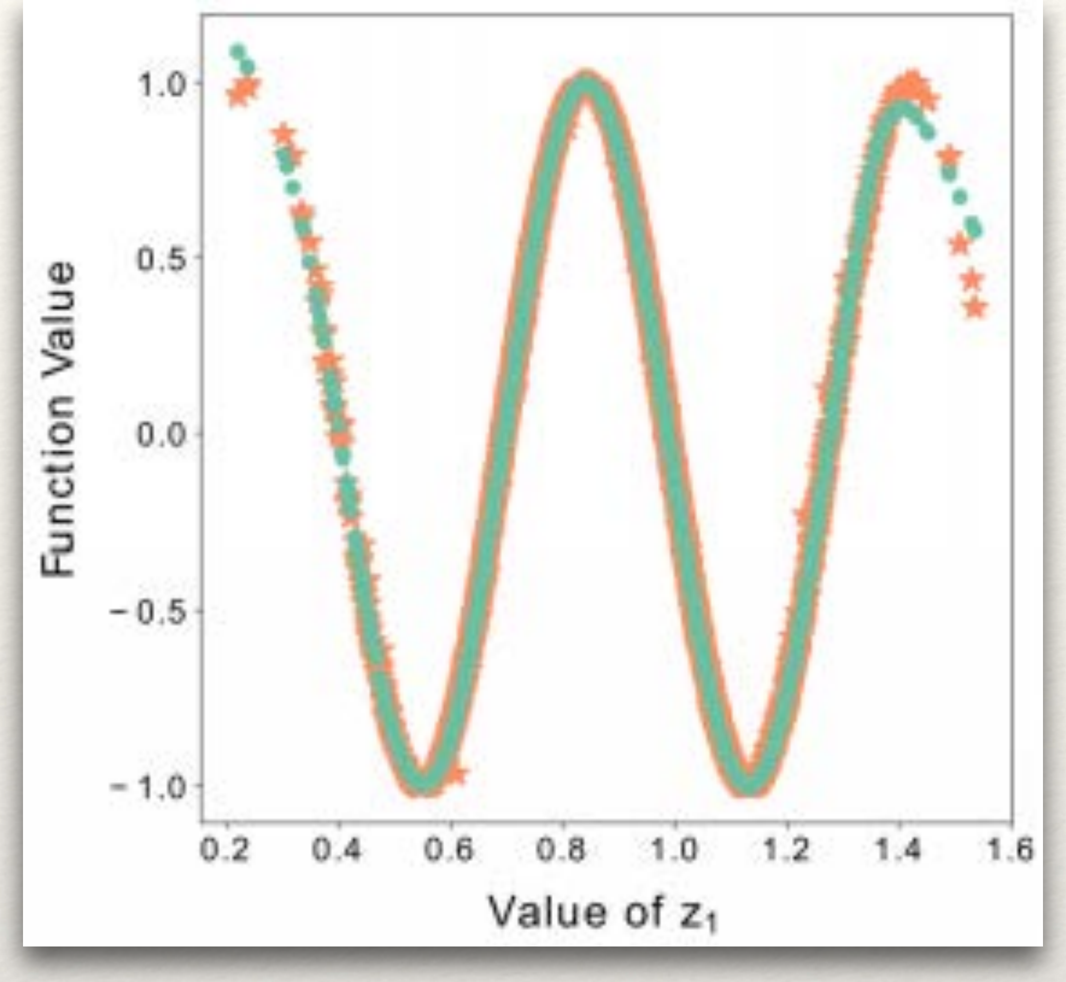
- Large improvement over AS and original NLL
- Domain: $[0,1]^n$. Functions: $f_4(\mathbf{x}) = \sin\left(\|\mathbf{x}\|^2\right)$, $\qquad f_5(\mathbf{x}) = \prod_{i=1}^{20} \frac{1}{1+x_i^2}$.
- Sensitivity: Magnitude of $(f \circ \mathbf{h})'(\mathbf{z})\mathbf{e}_1$ as percentage of total.
- Relative L^i error: $\dfrac{\|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\hat{\mathbf{x}})\|_i}{\|\mathbf{f}(\mathbf{x})\|_i}$ (**f** is vector of samples)

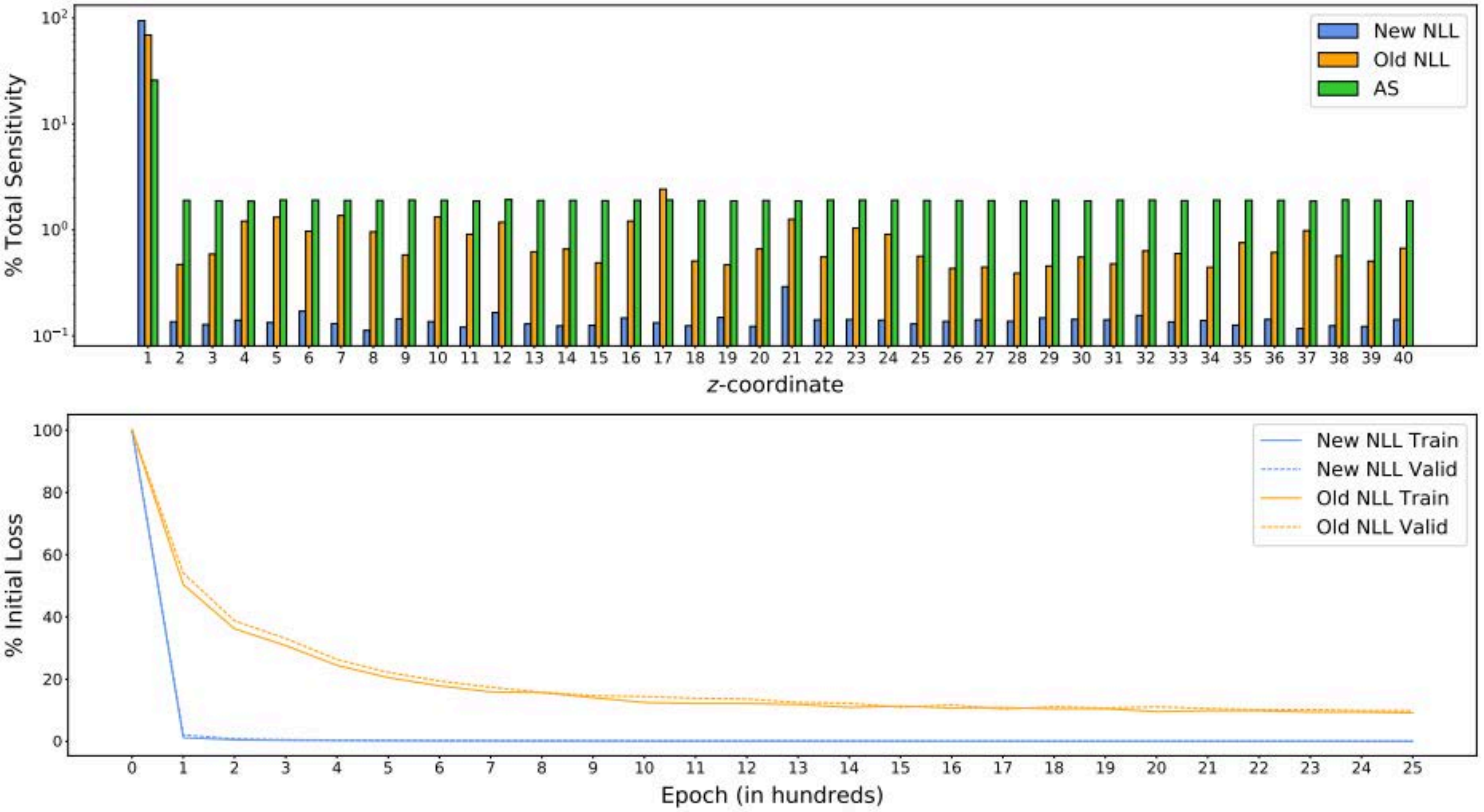


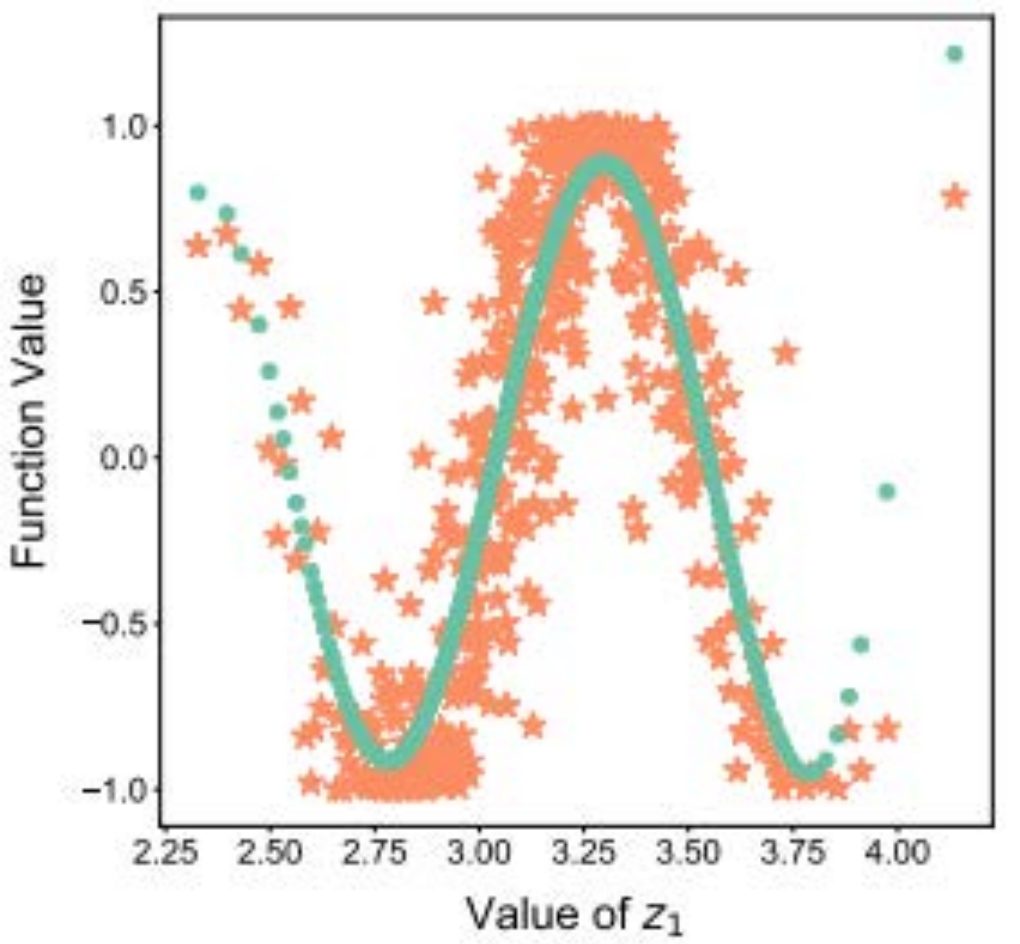
		100 Samples				500 Samples				2500 Samples			
Function	Method	z_A Sens %	RRMSE %	Rℓ_1 %	Rℓ_2 %	z_A Sens %	RRMSE %	Rℓ_1 %	Rℓ_2 %	z_A Sens %	RRMSE %	Rℓ_1 %	Rℓ_2 %
f_4	New NLL	78.7	3.86	8.27	10.9	89.8	1.82	3.52	5.16	94.5	0.827	1.72	2.35
	Old NLL	60.4	6.63	14.5	18.8	65.9	4.58	10.5	13.0	69.2	4.02	9.11	11.4
	AS 1-D	25.8	30.3	75.9	85.9	25.9	21.7	39.5	61.4	25.9	15.9	37.6	44.8
f_5	New NLL	75.1	0.920	5.79	7.92	88.6	0.370	2.78	3.97	93.8	0.154	1.63	1.98
	Old NLL 1	54.6	0.699	7.48	9.40	55.4	0.942	7.26	9.52	56.1	0.784	6.91	8.05
	Old NLL 2	61.8	1.80	12.9	21.1	68.7	1.03	9.22	11.1	67.5	0.894	8.16	9.69

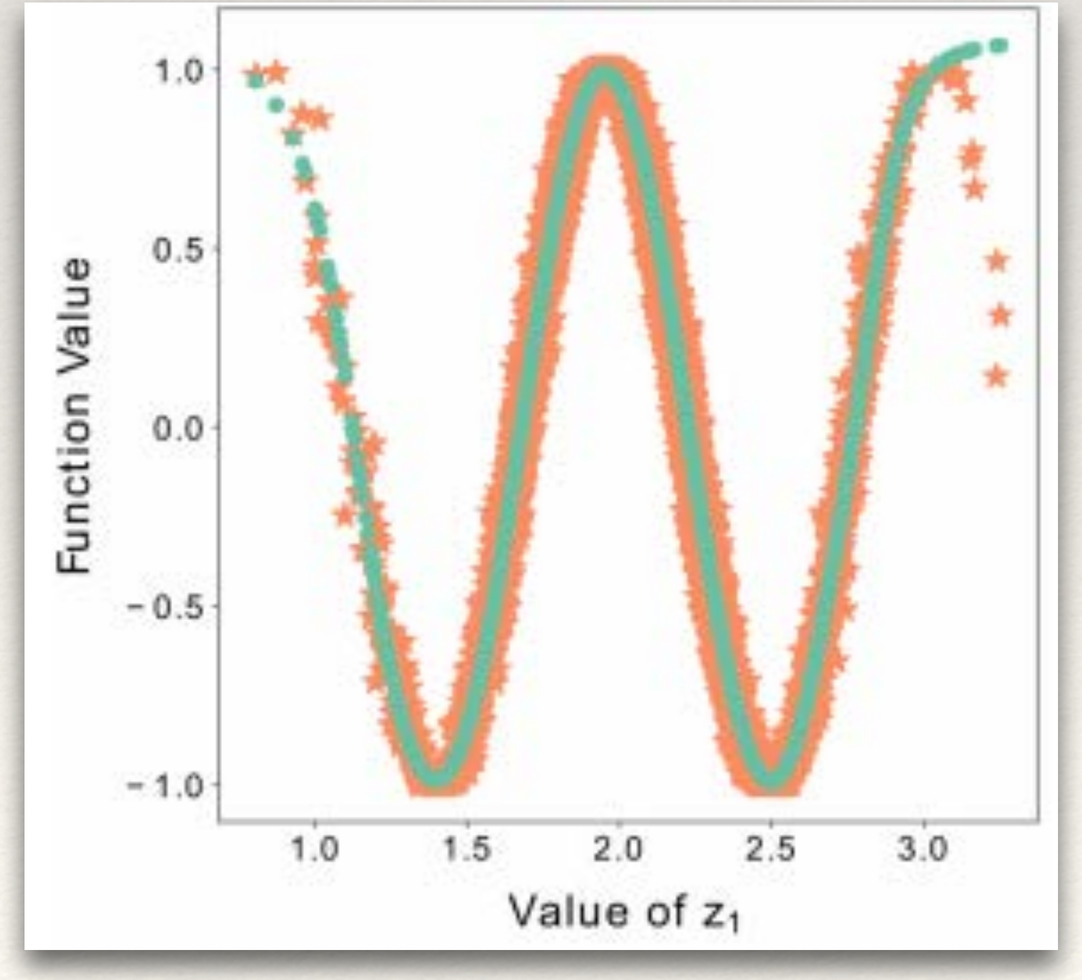
Results on Toy Examples

- On a 40 dimensional sine wave









A. Gruber, M. Gunzburger, L. Ju, Y. Teng, Z. Wang.
Numer. Math. Theor. Meth. Appl. (2021)

Predicting Kinetic Energy

- Consider the 1-D parametrized inviscid Burger's equation on [a,b], where $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3)$, $w = w(x, t, \boldsymbol{\mu})$.

$$w_t + \frac{1}{2}\left(w^2\right)_x = \mu_3 e^{\mu_2 x},$$
$$w(a, t, \boldsymbol{\mu}) = \mu_1,$$
$$w(x, 0, \boldsymbol{\mu}) = 1,$$

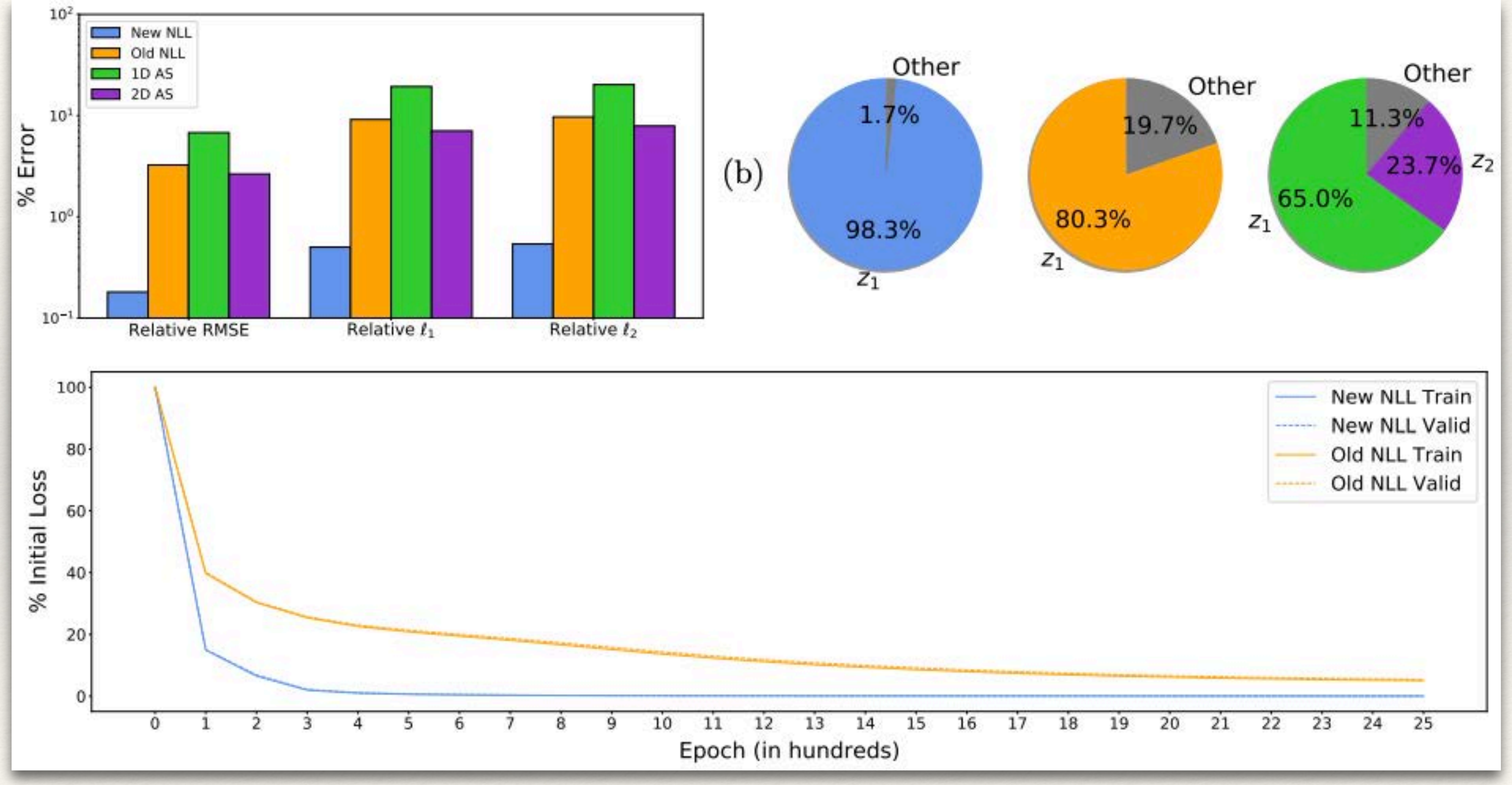
- Total kinetic energy at time t and its derivatives:

$$K(t, \boldsymbol{\mu}) = \frac{1}{2}\int_0^t \int_a^b w(x, \tau, \boldsymbol{\mu})^2 \, dx \, d\tau$$

$$\nabla K(t, \boldsymbol{\mu}) = (K_t \ K_{\boldsymbol{\mu}})^{\mathsf{T}} = \left(\frac{1}{2}\int_a^b w(x, t, \boldsymbol{\mu})^2 \, dx \quad \int_0^t \int_a^b w(x, \tau, \boldsymbol{\mu}) \, w_{\boldsymbol{\mu}}(x, \tau, \boldsymbol{\mu}) \, dx \, d\tau \right)^{\mathsf{T}}$$

Predicting Kinetic Energy

- Can compute gradients by solving sensitivity equations:
- Forward Euler with upwinding used to solve systems.



Reduced Order Modeling: Intrinsic

- High-fidelity PDE simulations are expensive.
  - Consider a semi-discretization of the domain.
  - Standard identification $u(x,t) =: \mathbf{u}(\mathbf{x},t)$ creates a lot of dimensionality.
- Can we get good results without solving the full PDE?
- Standard is to **encode -> solve -> decode.**
  - This way, PDE solving is low-dimensional.

Linear Methods

- Most popular method (until recently) is **proper orthogonal decomposition.**
- Carry out PCA on solution snapshots $\{\mathbf{u}(\mathbf{x}, t_j)\}_{j=1}^{N}$, generate matrix $\mathbf{S}$.
- SVD: $\mathbf{S} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$. First n cols of $\mathbf{U}$ (say $\mathbf{A}$) are the reduced basis of **POD modes**.
- Instead of $\dot{\mathbf{u}} = \mathbf{f}(t, \mathbf{u})$, can then solve $\mathbf{A}\dot{\hat{\mathbf{u}}} = \mathbf{f}(t, \mathbf{A}\hat{\mathbf{u}})$.

Linear Methods

- POD works well until the EWs of Σ decay slowly.
- Even many modes are not enough to reliably capture behavior.
- Can we do better than linear?

Potential Copyright Issue

Lee, K. and Carlberg, K. *J. Comp. Phys.* (2019)

CNN Model Order Reduction

- Yes! Kim and Carlberg (2019) used a convolutional neural network.
  - Demonstrated greatly improved performance over POD.
- Architecture is **autoencoder-decoder:**

Potential Copyright Issue

Disadvantage of CNN

- Recall:

$$\mathbf{y}_{\ell,i} = \boldsymbol{\sigma}_{\ell}\left(\sum_{j=1}^{C_{in}} \mathbf{y}_{(\ell-1),j} \star \mathbf{W}^{j}_{\ell,i} + \mathbf{b}_{\ell,i}\right), \quad \text{where} \;\; 1 \leq i \leq C_{out}\,.$$

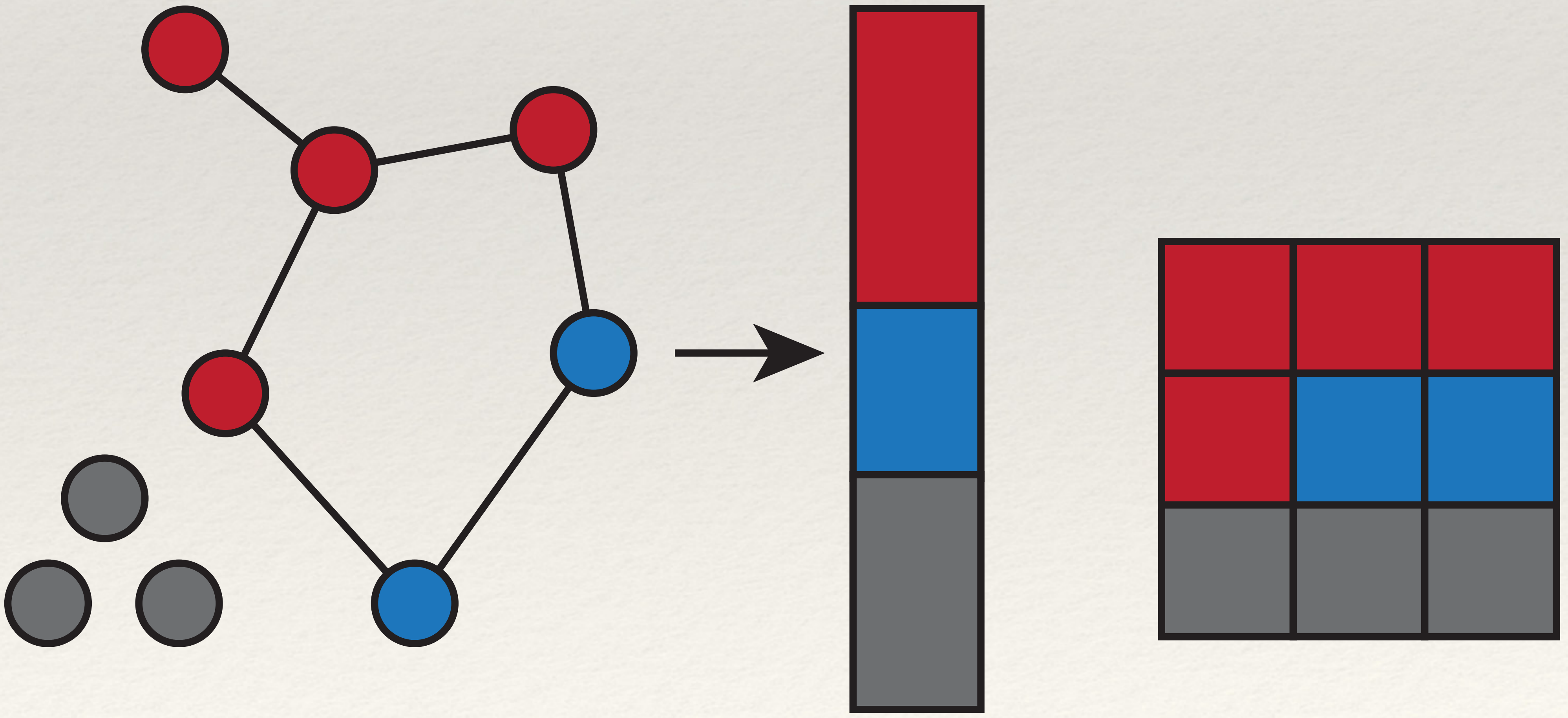
- Convolution $\star$ in 2-D:

$$(\mathbf{x} \star \mathbf{W})^{\alpha}_{\beta} = \sum_{\gamma,\delta} x^{(s\alpha+\gamma)}_{(s\beta+\delta)}\, w^{(L-1-\gamma)}_{(M-1-\delta)},$$

- Only well defined (in this form) for regular domains!

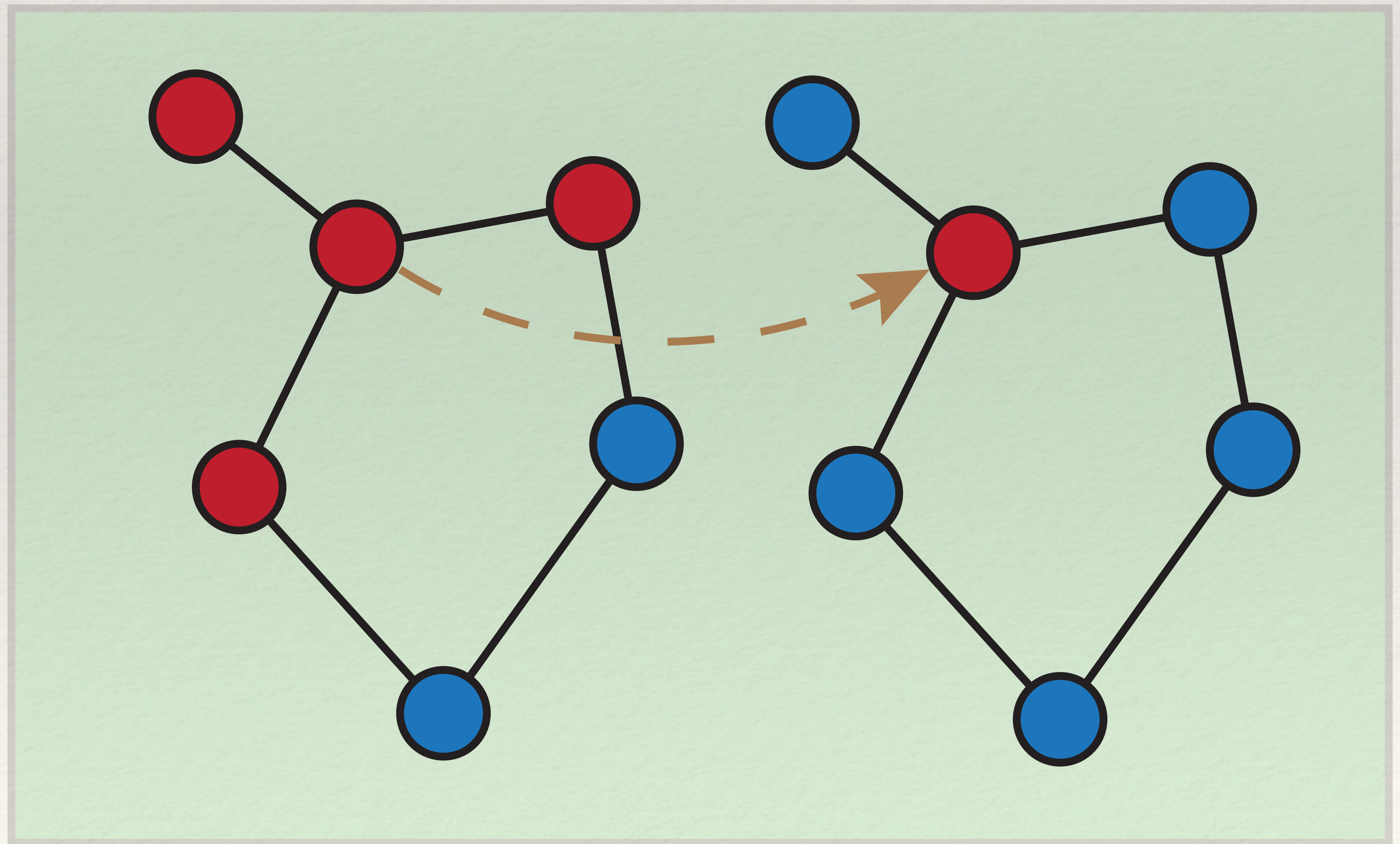
Disadvantage of CNN

- How to use CNN on irregular data?
- Current strategy is to ignore the problem:
  - inputs **y** padded with fake nodes and reshaped to a square.
  - Convolution applied to square-ified input.
  - $\tilde{\mathbf{y}}$ reassembled at end. Fake nodes ignored.
- Works surprisingly well!
  - But, not very meaningful.



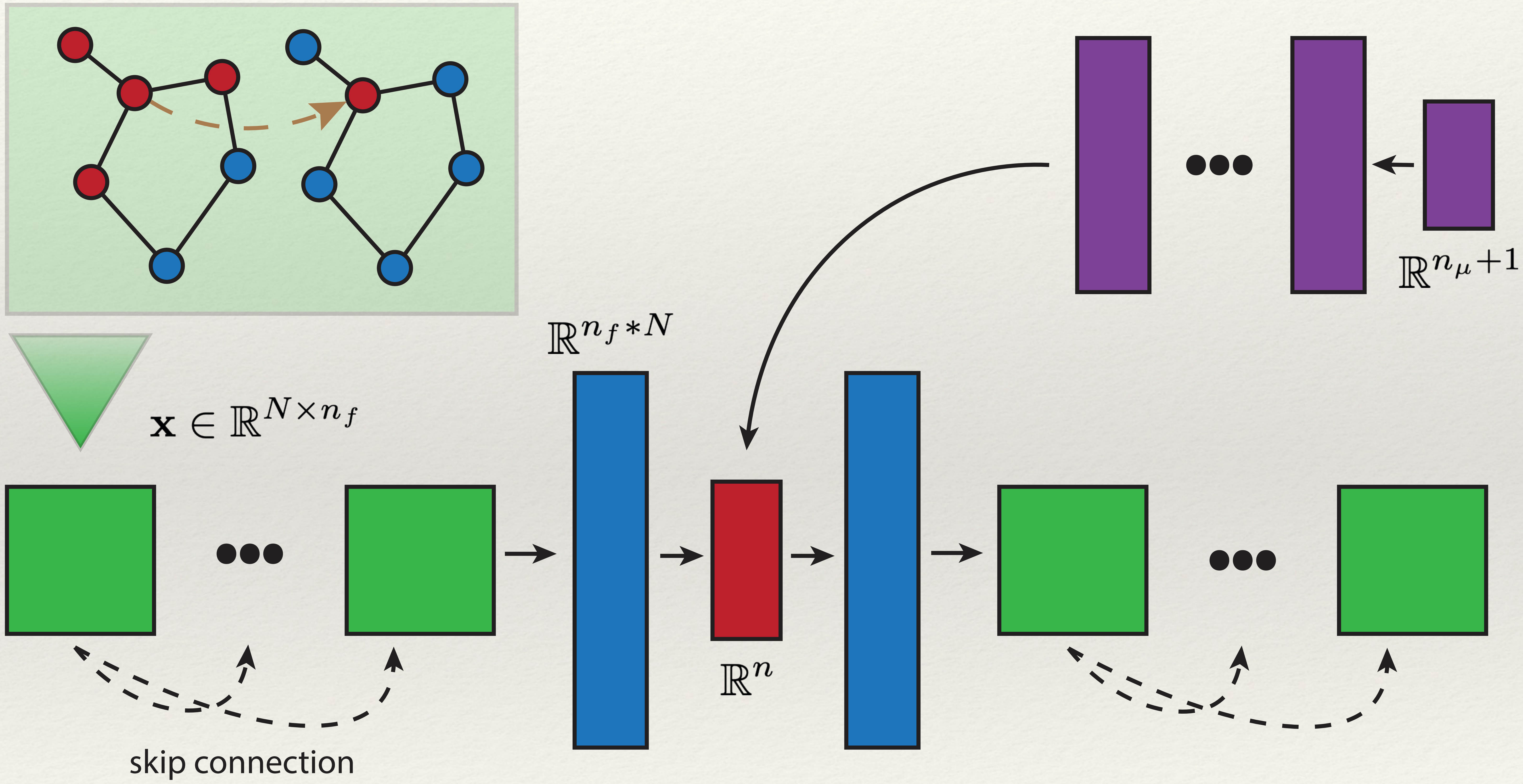
Graph Convolutional Networks

- Huge amount of recent work extending convolution to graph domains.
- $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ undirected graph with adj. matrix $\mathbf{A} \in \mathbb{R}^{|\mathscr{V}|\times|\mathscr{V}|}$.
- $\mathbf{D}$ the degree matrix $d_{ii} = \sum_j a_{ij}$.
- The Laplacian of $\mathscr{G}$: $\mathbf{L} = \mathbf{D} - \mathbf{A} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$.
  - Columns of $\mathbf{U}$ are Fourier modes of $\mathscr{G}$.
  - Discrete FT/IFT: simply multiply by $\mathbf{U}^\top/\mathbf{U}$.



GC Autoencoder Architecture

- GCN2 layers encode-decode.
- Blue layers are fully connected.
- For ROM: purple network simulates low-dim dynamics.
- Split network idea due to (Fresca et al. 2020).



A. Gruber, M. Gunzburger, L. Ju, Z. Wang, (preprint)

Experimental Details

- Want to compare performance of GCAE, CAE, FCAE for ROM applications.
- Evaluation based on two criteria:
  - Pure reconstruction ability (compression problem).
  - Ability to predict new solutions given parameters (prediction problem).
- Prediction loss used: $L(\mathbf{x}, t, \boldsymbol{\mu}) = |\mathbf{x} - \mathbf{g} \circ \hat{\mathbf{x}}|^2 + |\mathbf{h} - \hat{\mathbf{x}}|^2$.
- Compression loss: $L(\mathbf{x}, t, \boldsymbol{\mu}) = |\mathbf{x} - \mathbf{g} \circ \mathbf{h}|^2$.
- Network trained using mini-batch descent with ADAM optimizer.

2-D Parameterized Heat Equation

- Consider

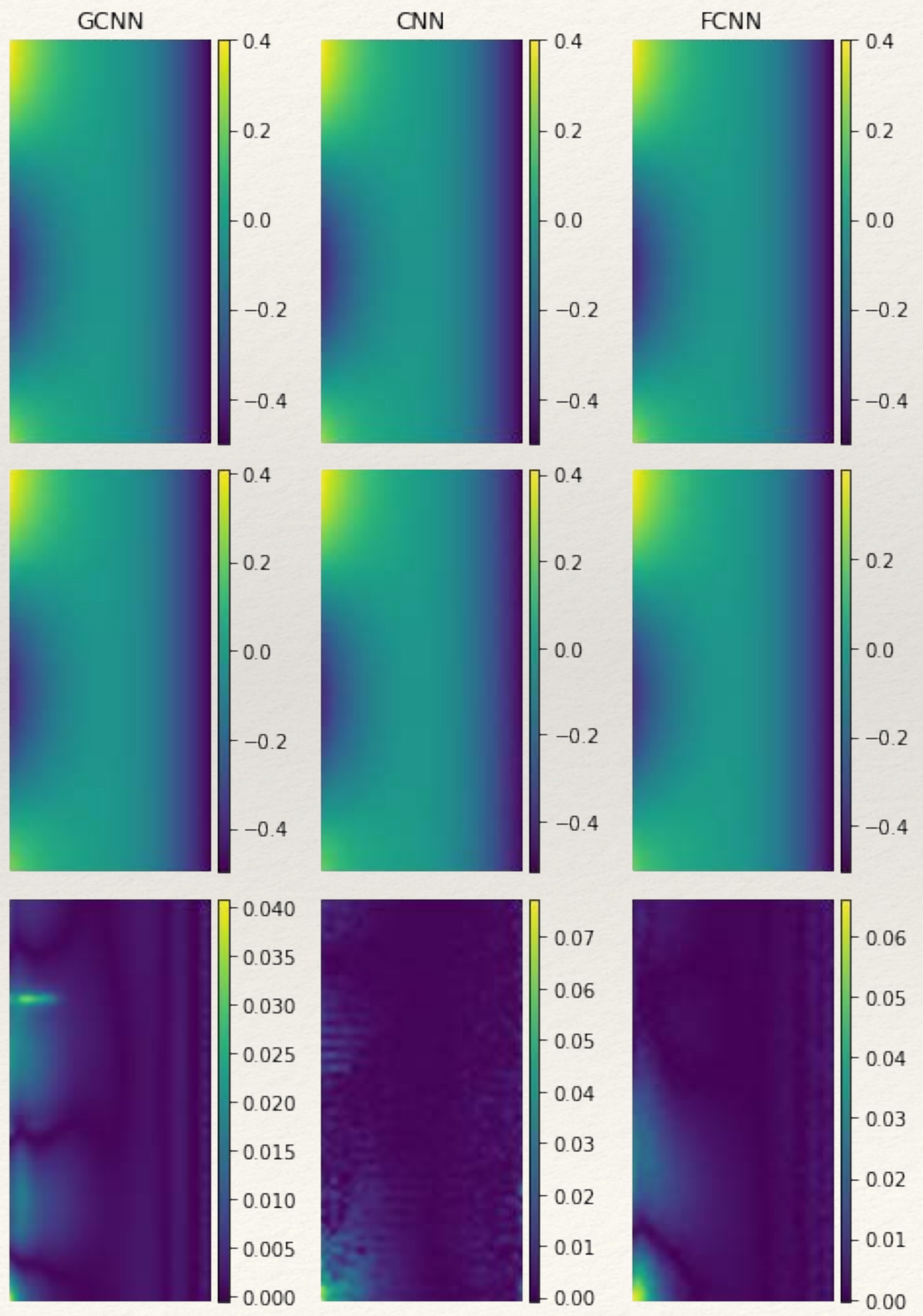
  $u = u(x, y, t, \boldsymbol{\mu}),\ U = [0,1] \times [0,2],\ \boldsymbol{\mu} \in [0.05,0.5] \times [\pi/2,\pi]$

  and solve

$$\begin{cases} u_t - \Delta u = 0 \quad \text{on } U \\ u(0,y,t) = -0.5 \\ u(1,y,t) = \mu_1 \cos(\mu_2 y) \\ u(x,y,0) = 0 \end{cases}$$

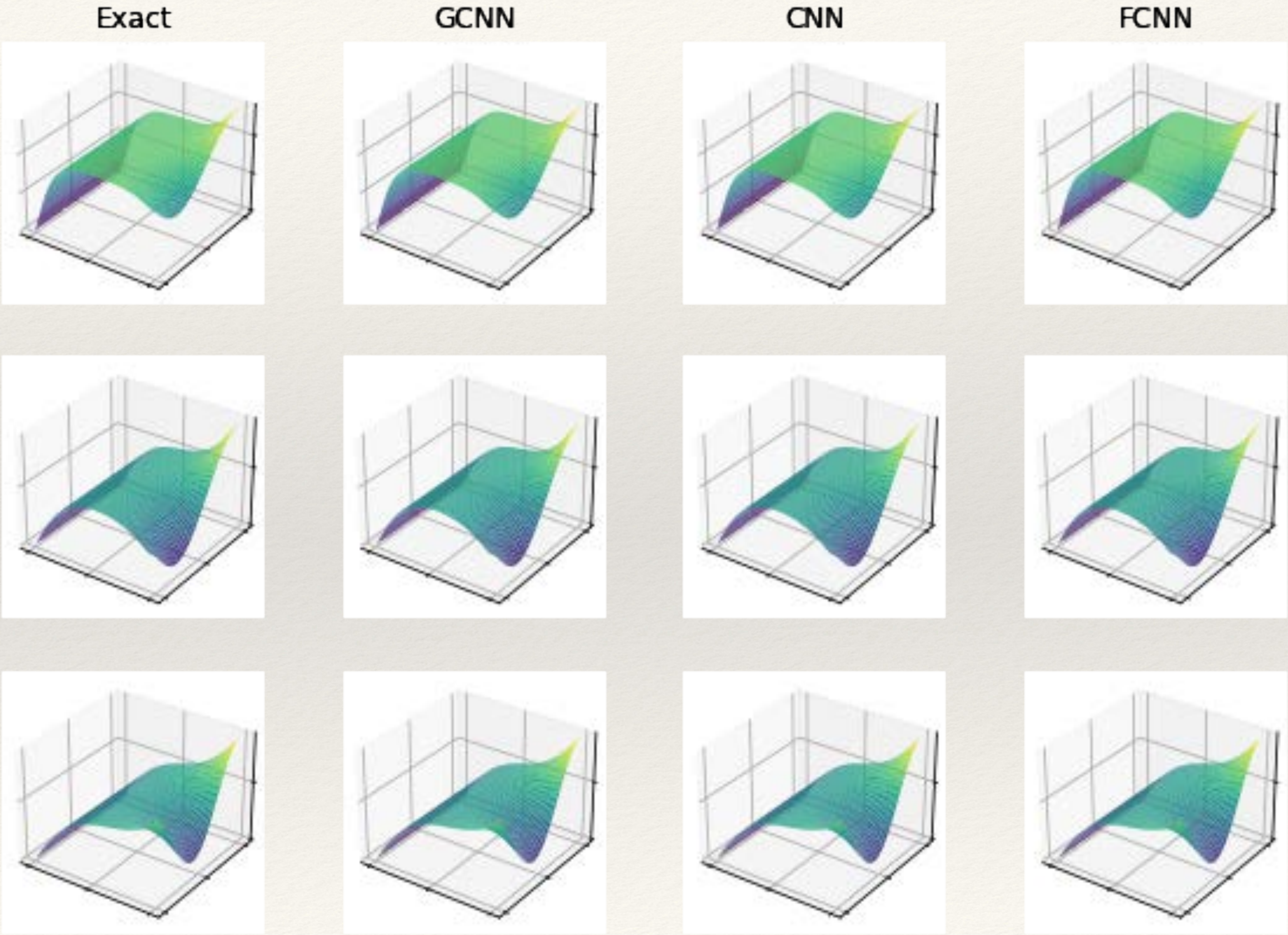
- Discretizing over (stretched) grid gives $u = \mathbf{u}(t, \boldsymbol{\mu})$.

Solution u: Exact, Reconstructed, Pointwise Error

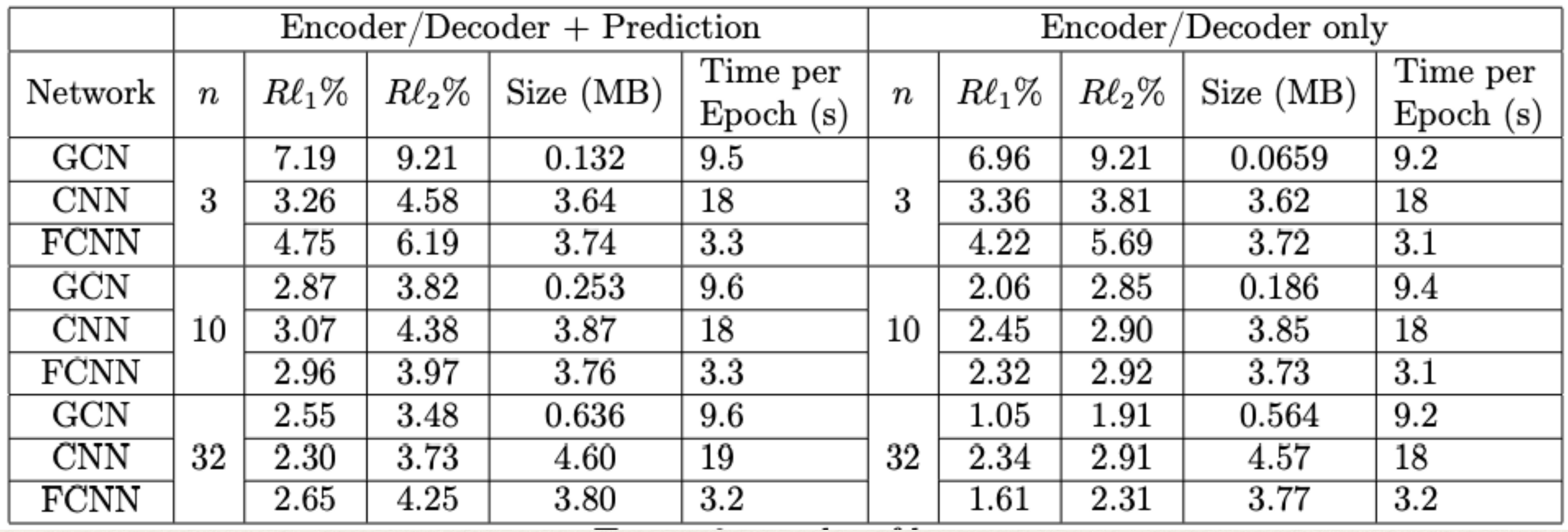


2-D Parameterized Heat Equation: Results

- Results shown for $n = 10$.
- GCNN has lowest error and least memory requirement (by >10x!)
- CNN is worst — cheap hacks have a cost.

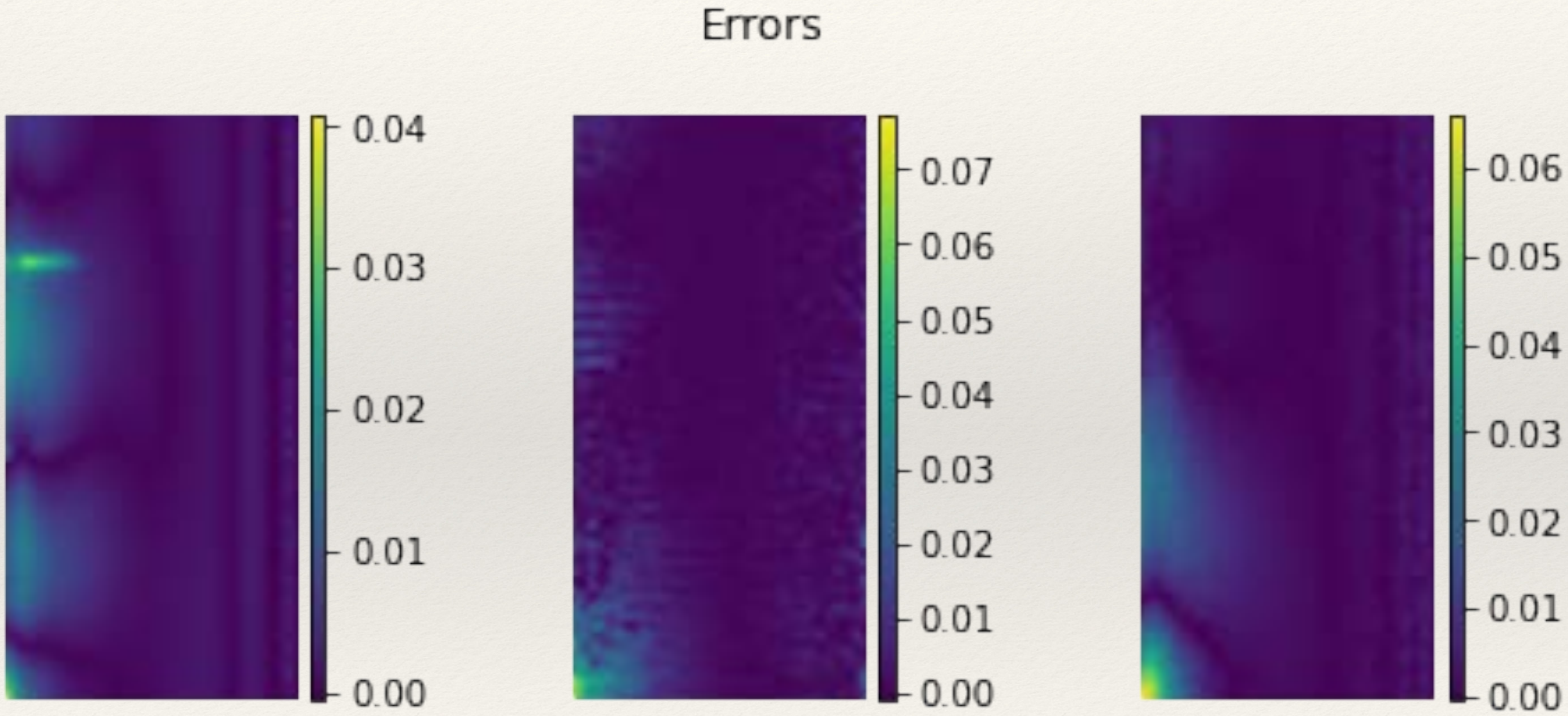


2-D Parameterized Heat Equation: Results



	Encoder/Decoder + Prediction					Encoder/Decoder only				
Network	n	$R\ell_1\%$	$R\ell_2\%$	Size (MB)	Time per Epoch (s)	n	$R\ell_1\%$	$R\ell_2\%$	Size (MB)	Time per Epoch (s)
GCN	3	7.19	9.21	0.132	9.5	3	6.96	9.21	0.0659	9.2
CNN		3.26	4.58	3.64	18		3.36	3.81	3.62	18
FCNN		4.75	6.19	3.74	3.3		4.22	5.69	3.72	3.1
GCN	10	2.87	3.82	0.253	9.6	10	2.06	2.85	0.186	9.4
CNN		3.07	4.38	3.87	18		2.45	2.90	3.85	18
FCNN		2.96	3.97	3.76	3.3		2.32	2.92	3.73	3.1
GCN	32	2.55	3.48	0.636	9.6	32	1.05	1.91	0.564	9.2
CNN		2.30	3.73	4.60	19		2.34	2.91	4.57	18
FCNN		2.65	4.25	3.80	3.2		1.61	2.31	3.77	3.2

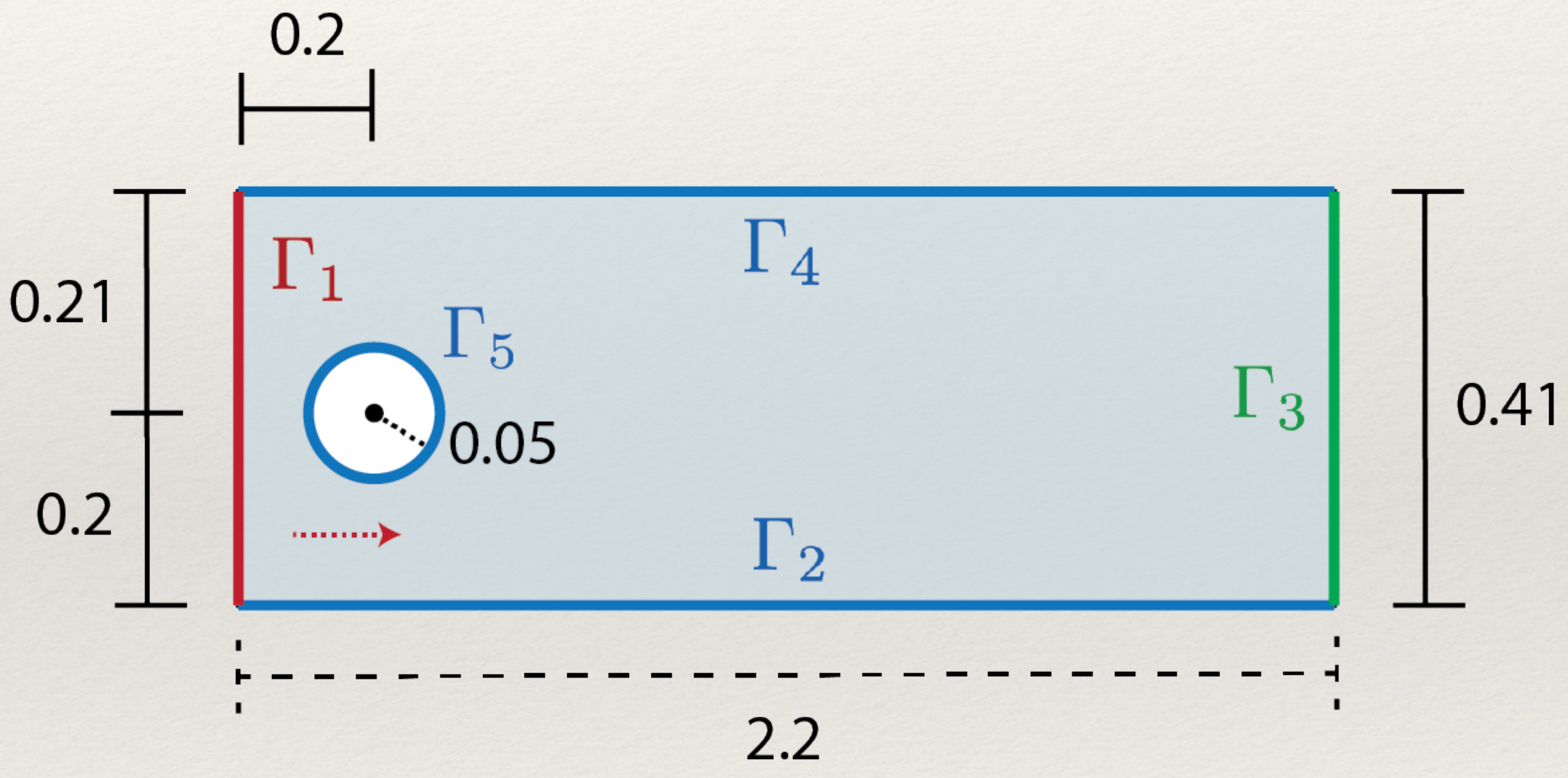
2-D Parameterized Heat Equation: Results



Unsteady Navier-Stokes Equations

- Consider the Schafer-Turek benchmark problem:

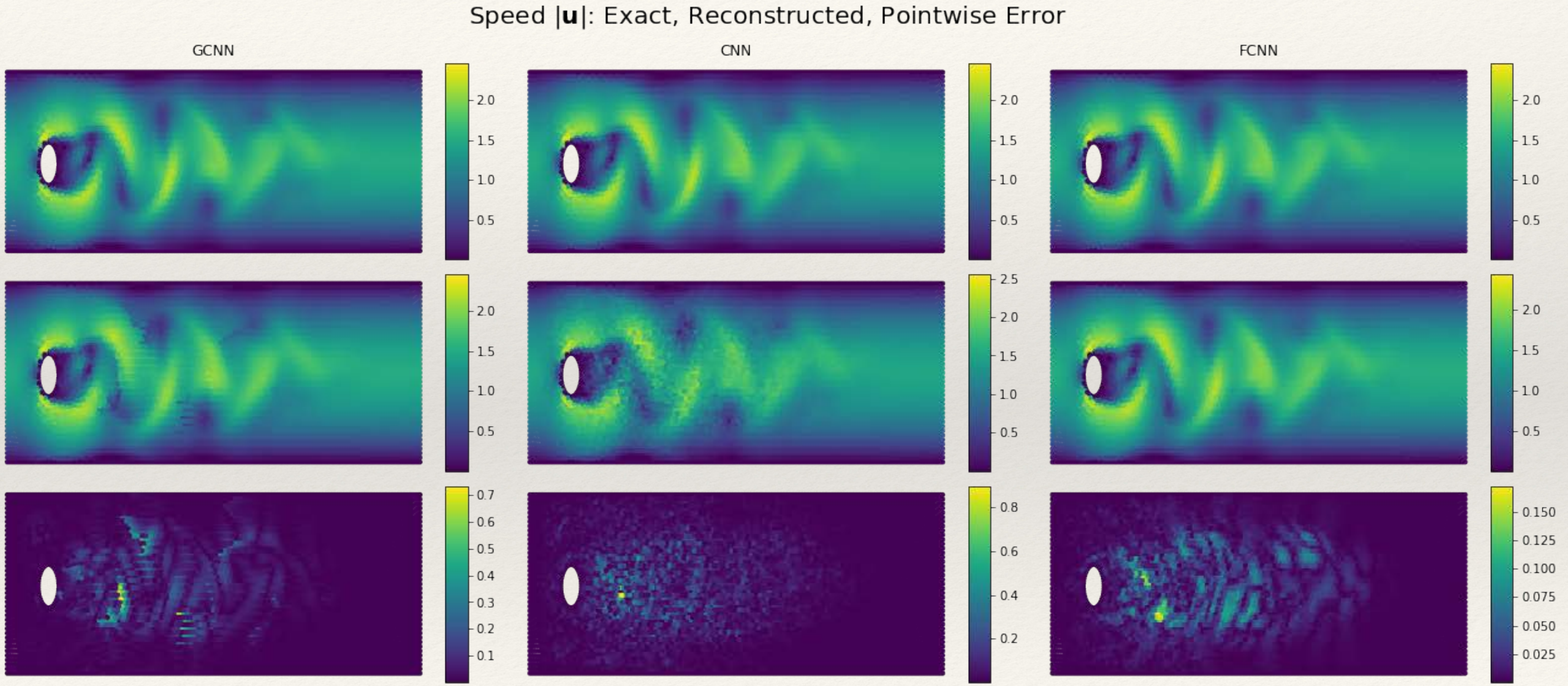
$$\dot{\mathbf{u}} - \nu \Delta \mathbf{u} + \nabla_{\mathbf{u}} \mathbf{u} + \nabla p = \mathbf{f},$$
$$\nabla \cdot \mathbf{u} = 0,$$
$$\mathbf{u}\,\big|_{t=0} = \mathbf{u}_0 \,.$$



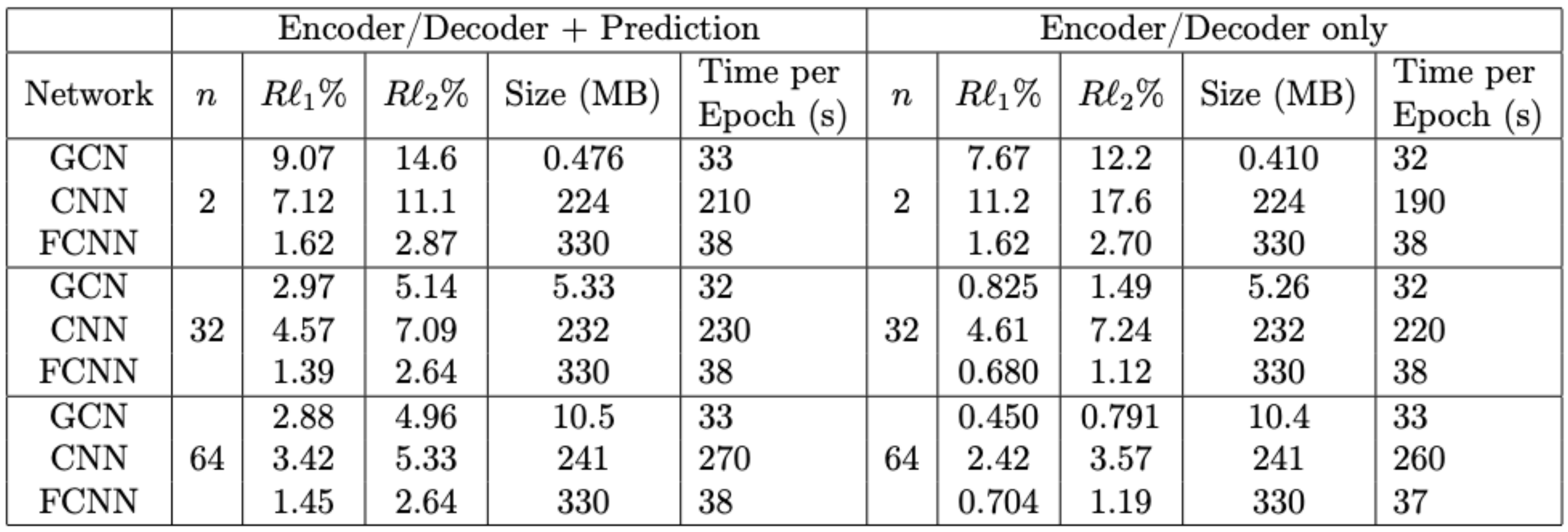
- Impose 0 boundary conditions on $\Gamma_2, \Gamma_4, \Gamma_5$. Do nothing on Γ_3. Parabolic inflow on Γ_1.

Navier-Stokes Equations: Results

- $N = 10104$
- $n = 32$
- Reynolds number 185.
- FCNN best on prediction problem.



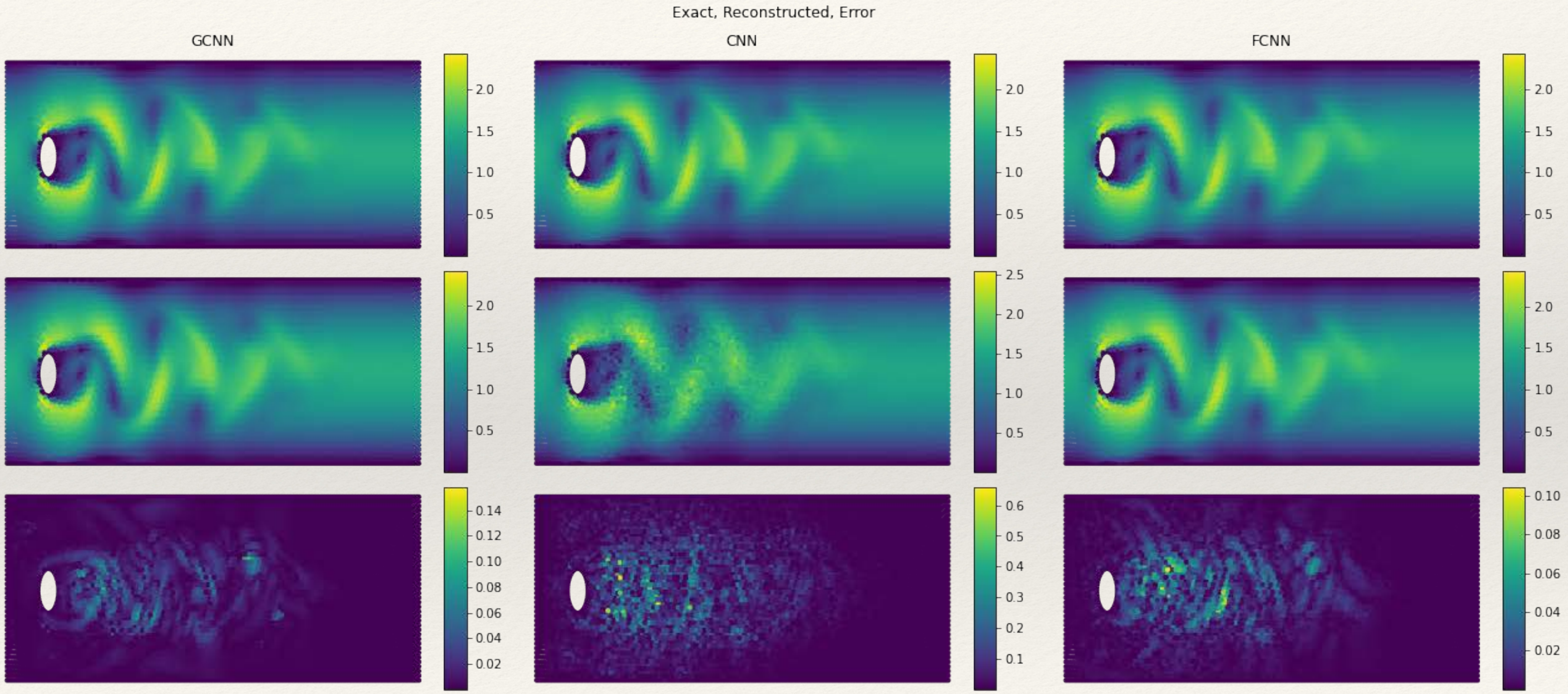
Navier-Stokes Equations: Results



	Encoder/Decoder + Prediction					Encoder/Decoder only				
Network	n	$R\ell_1\%$	$R\ell_2\%$	Size (MB)	Time per Epoch (s)	n	$R\ell_1\%$	$R\ell_2\%$	Size (MB)	Time per Epoch (s)
GCN	2	9.07	14.6	0.476	33	2	7.67	12.2	0.410	32
CNN		7.12	11.1	224	210		11.2	17.6	224	190
FCNN		1.62	2.87	330	38		1.62	2.70	330	38
GCN	32	2.97	5.14	5.33	32	32	0.825	1.49	5.26	32
CNN		4.57	7.09	232	230		4.61	7.24	232	220
FCNN		1.39	2.64	330	38		0.680	1.12	330	38
GCN	64	2.88	4.96	10.5	33	64	0.450	0.791	10.4	33
CNN		3.42	5.33	241	270		2.42	3.57	241	260
FCNN		1.45	2.64	330	38		0.704	1.19	330	37

Navier-Stokes Equations: Results

- Compression
- GCNN matches FCNN in accuracy
- GCNN memory cost is >50x less than FCNN



Conclusions

- Nonlinear PDEs and the techniques for solving/approximating their solutions are ***practical, interesting, and highly creative.***
- My work:
    - Starts from theoretical questions or abstract problems.
    - Proposes rigorous solutions validated by precise algorithms and simulations.
    - Benefits from collaboration and a diverse array of expertise.
- Most projects have a history of receiving external funding.
    - Can likely be continued.

Thank You!