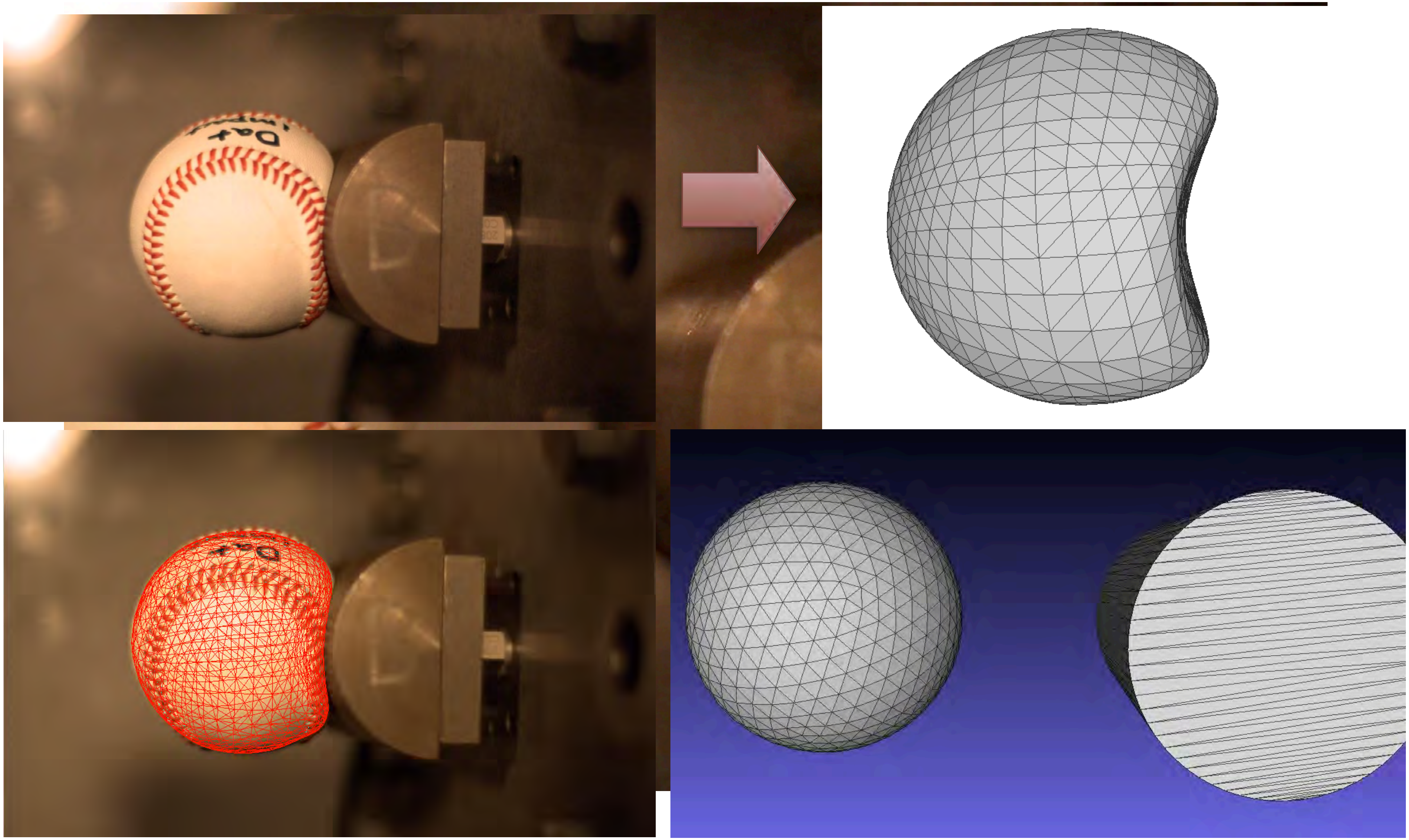
Deep 3D Surface Meshes

P. Fua

EPFL Computer Vision Lab
Lausanne
Switzerland

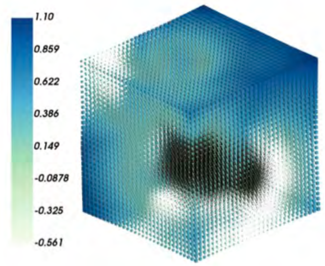
Aerial Mapping

Deforming 3D Surfaces



Ngo, PAMI'16

3D Surface Representations



	Voxels	Explicit surface mesh	Point sets	Continuous implicit fields
High frequency details?	--	++	+	++
Arbitrary topology?	+	-	+	++
Regularity?	+	+	-	++

There are many applications at which explicit representations excel:

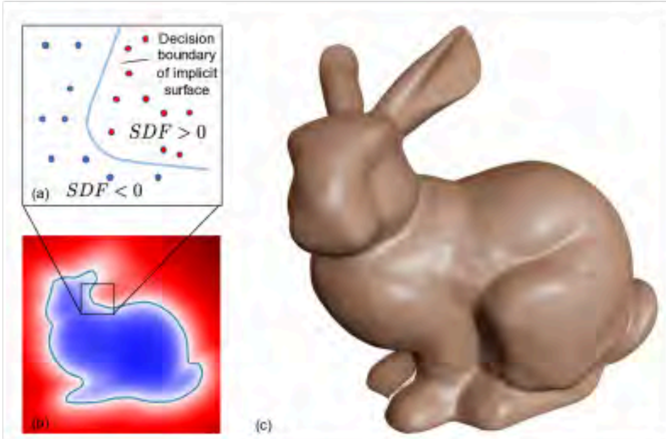
- High-quality rendering in computer graphics.
- Precise modeling of biological structures from biomedical data.
- Computational fluid dynamics in computer assisted design.

But:

- Their topology is fixed.
- They are not particularly deep learning friendly.

—> Implicit Surface Representations

Signed Distance Fields (SDF)



- Represent a 3D surface S by the zero crossings of a signed distance function

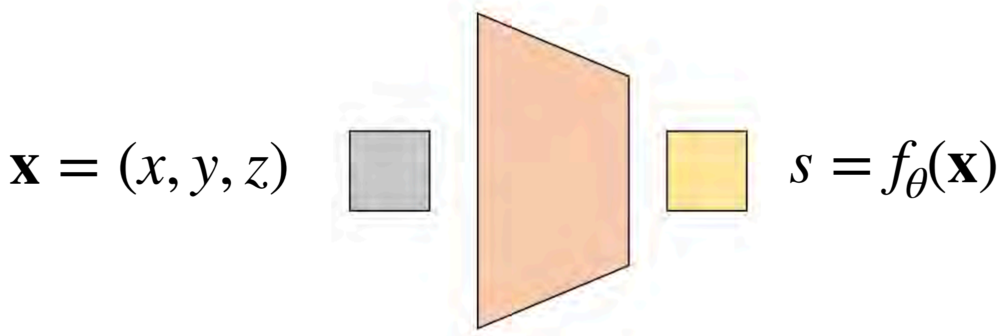
$$f: \mathbb{R}^3 \rightarrow \mathbb{R}$$

$$\forall \mathbf{x} \in \mathbb{R}^3,\ f(\mathbf{x}) \text{ is the signed distance to the surface.}$$

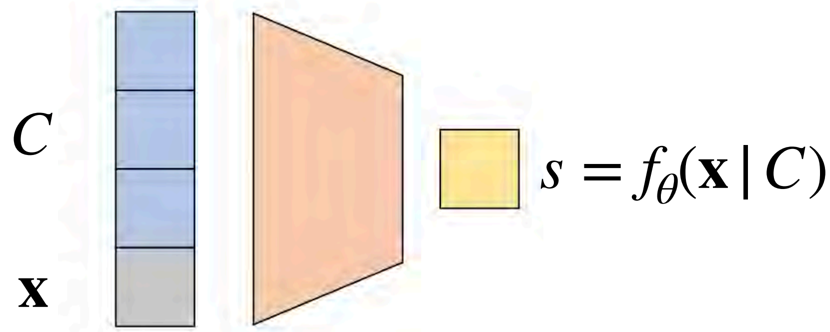
- Such surfaces can easily change topology, which is harder to do with explicit surface representations.

- SDFs have long been appealing in theory but hard to use in practice because it it was necessary to store the 3D values of f in a cube like structure until ….

Deep SDF

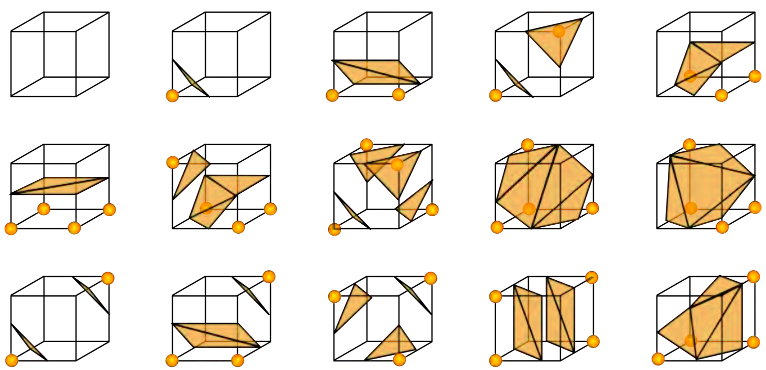


Single Shape DeepSDF



Coded Shape DeepSDF

[Park et al., CVPR'19]



But one bottleneck remains: If an explicit surface representation is required, one has to run a marching-cube style algorithm, which is **not differentiable** and often **slow**.

Deep SDF Pipeline

Vertices Facets

Loss function: $L(\mathcal{V}, \mathcal{F})$

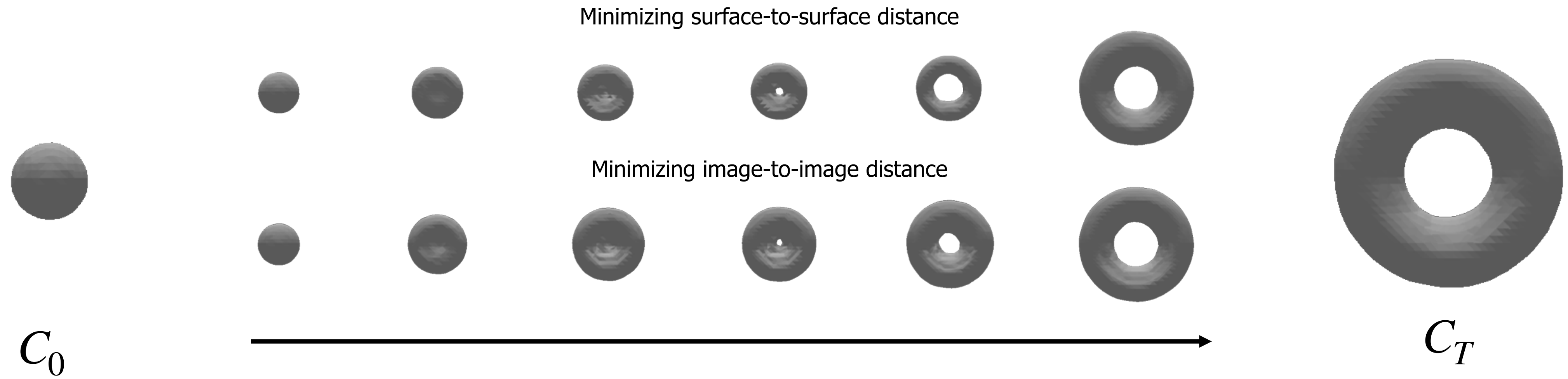
Forward pass: $\mathcal{V}, \mathcal{F} = mc(S)$, with $f_\theta(\mathbf{v}_i \mid C) = 0, \forall \mathbf{v}_i \in \mathcal{S}$.

Backward pass: $$\frac{\partial L}{\partial C} = \sum_{\mathbf{i}} \frac{\partial L}{\partial \mathbf{v}_i} \frac{\partial \mathbf{v}_i}{\partial s} \frac{\partial s}{\partial C}$$

- A priori $\frac{\partial \mathbf{v}_i}{\partial s}$ cannot be computed because mc is not differentiable.
- But, f_θ approximates a signed distance function …

- $\frac{\partial \mathbf{v}}{\partial s} = -\mathbf{n}(\mathbf{v}) = -\nabla s(\mathbf{v})$,
- $\frac{\partial \mathbf{v}}{\partial s} = -\frac{\nabla s(\mathbf{v})}{\|\nabla s(\mathbf{v})\|^2}$ is s is not a signed distance function.

End-to-End Differentiable Pipeline



1. Start with a Deep SDF code.
2. Use marching cube to compute mesh and vertices.
3. Use them for the forward pass and for backpropagation.
4. Update the SDF code and iterate.

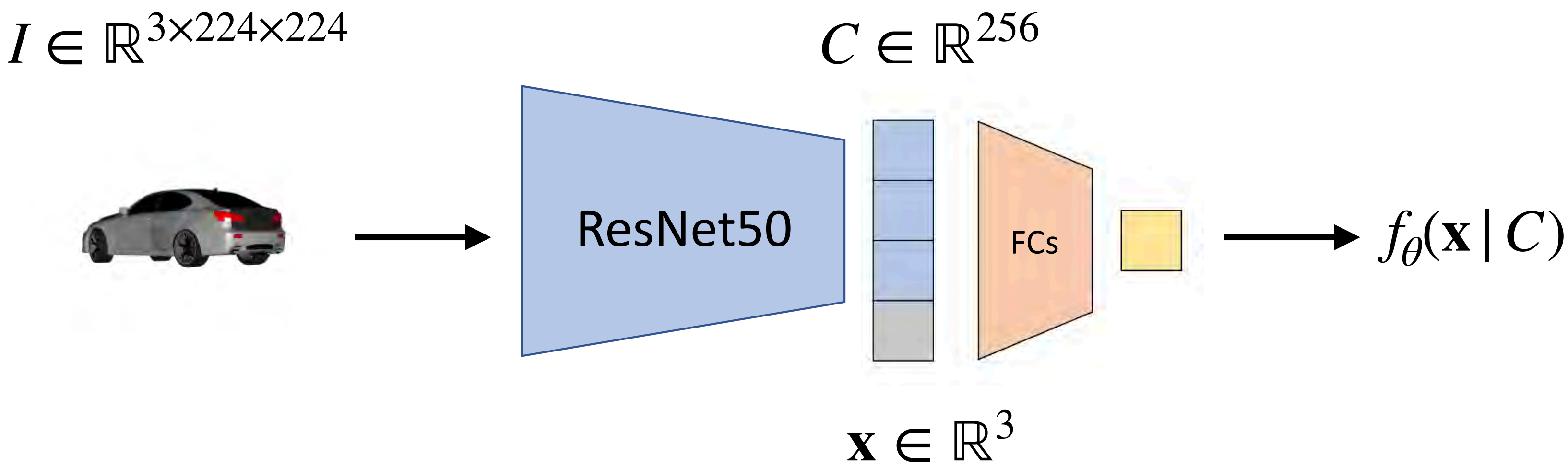
—> We can turn a spherical mesh into a toroidal one by minimizing a differentiable objection function.

From Genus 0 to Genus 1

[Remelli et al., NeurIPS'20]

Application: Single View Reconstruction

Network Specification



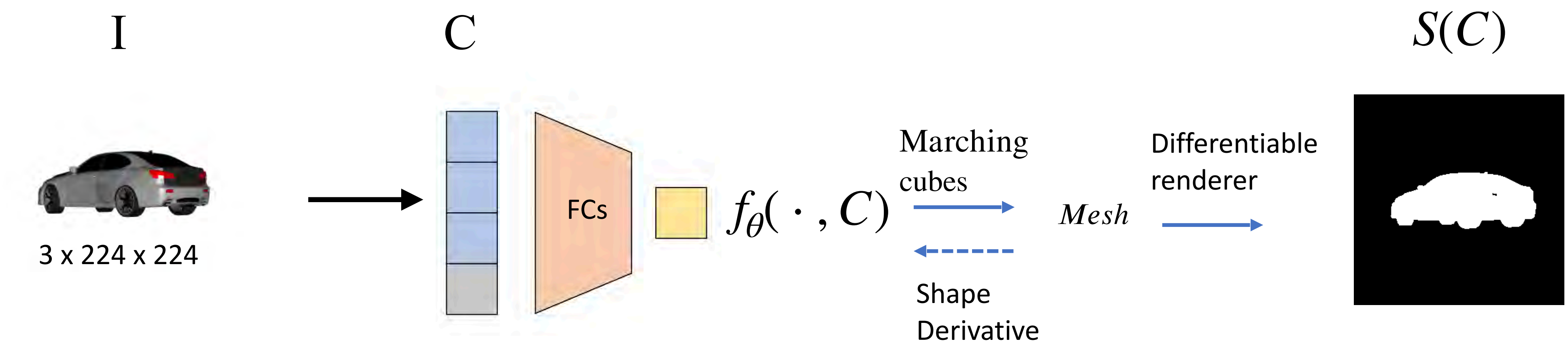
Number of encoder parameters: 24,032,576
Number of decoder parameters: 1,843,195

Trained by minimizing

$$\sum_{\mathbf{x}} | f_I^{gt}(\mathbf{x}) - f_\theta(\mathbf{x} \mid C(I)) |_1 + \lambda | C(I) |_2$$

with respect to θ.

From Discriminative to Generative



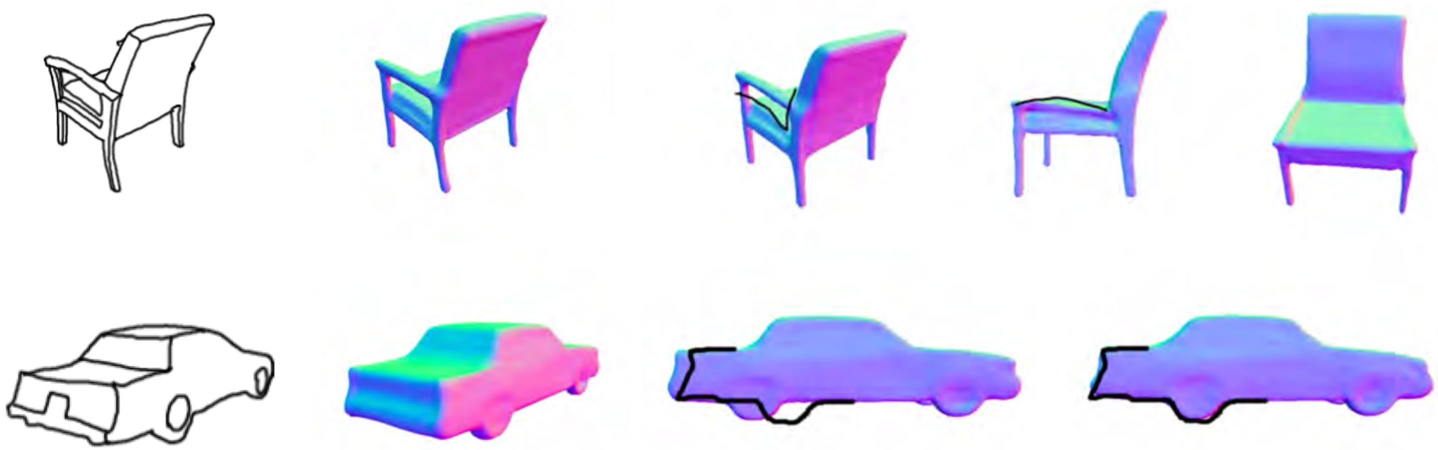
Refined by minimizing:

$$|S_I - S(C)|_1 + \lambda |C|_2$$

with respect to C.

From Silhouettes to 3D Shapes

3D Model from Image

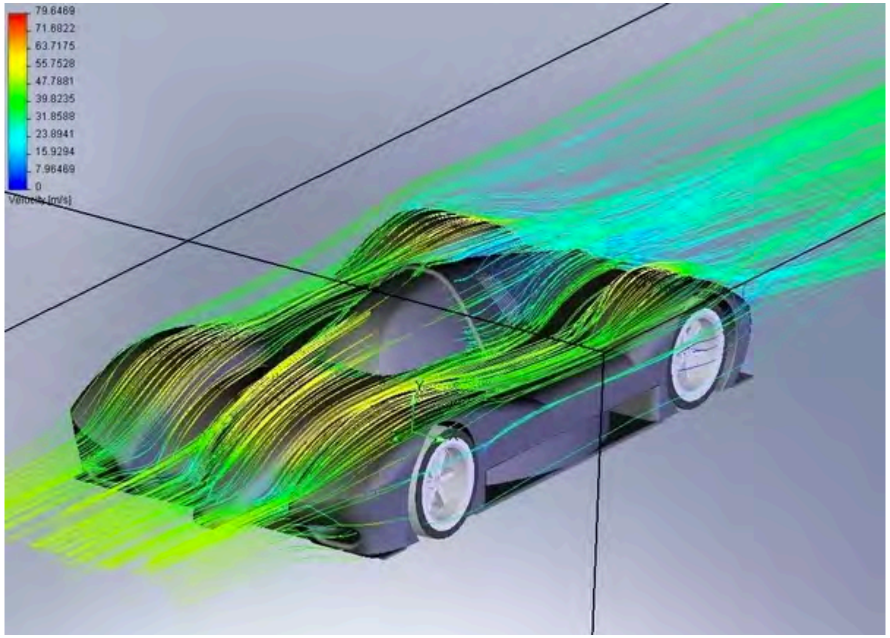


Editable 3D Model from Sketch

Application: Shape Optimization

3D Shape Design

- Design a shape.
- Simulate its performance.
- Redesign.



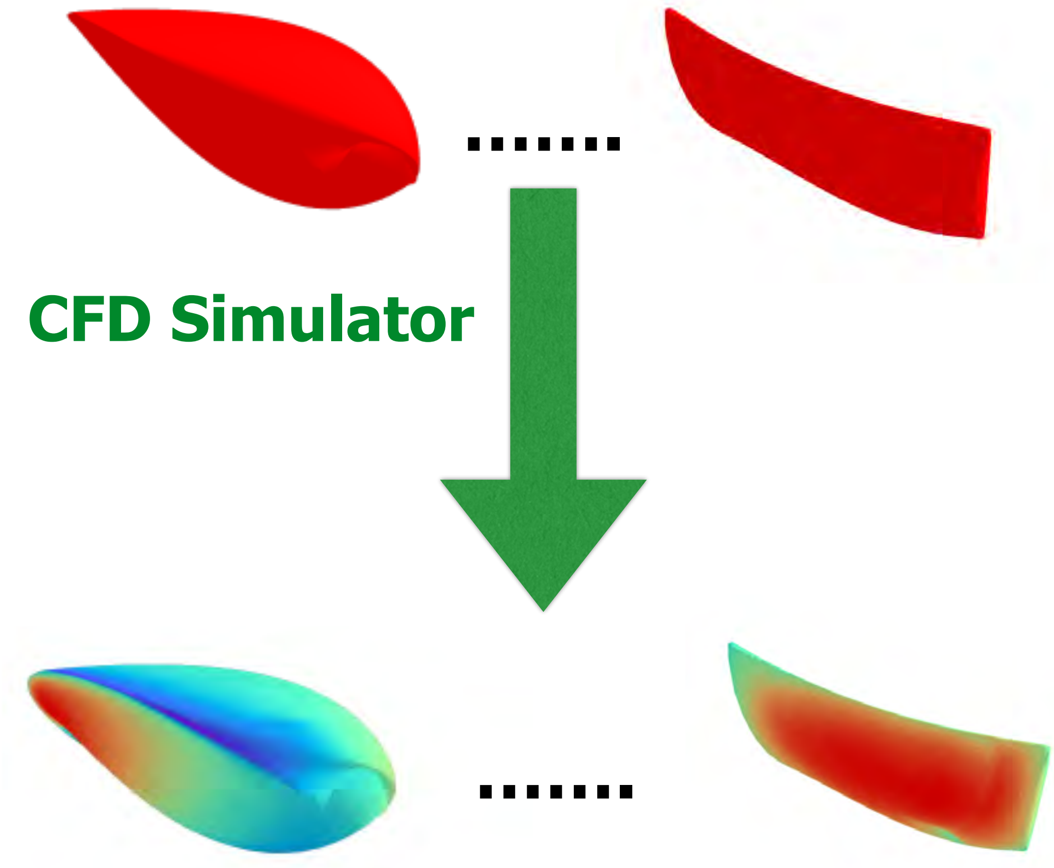
It works but:

It takes hours or days to produce a single simulation.

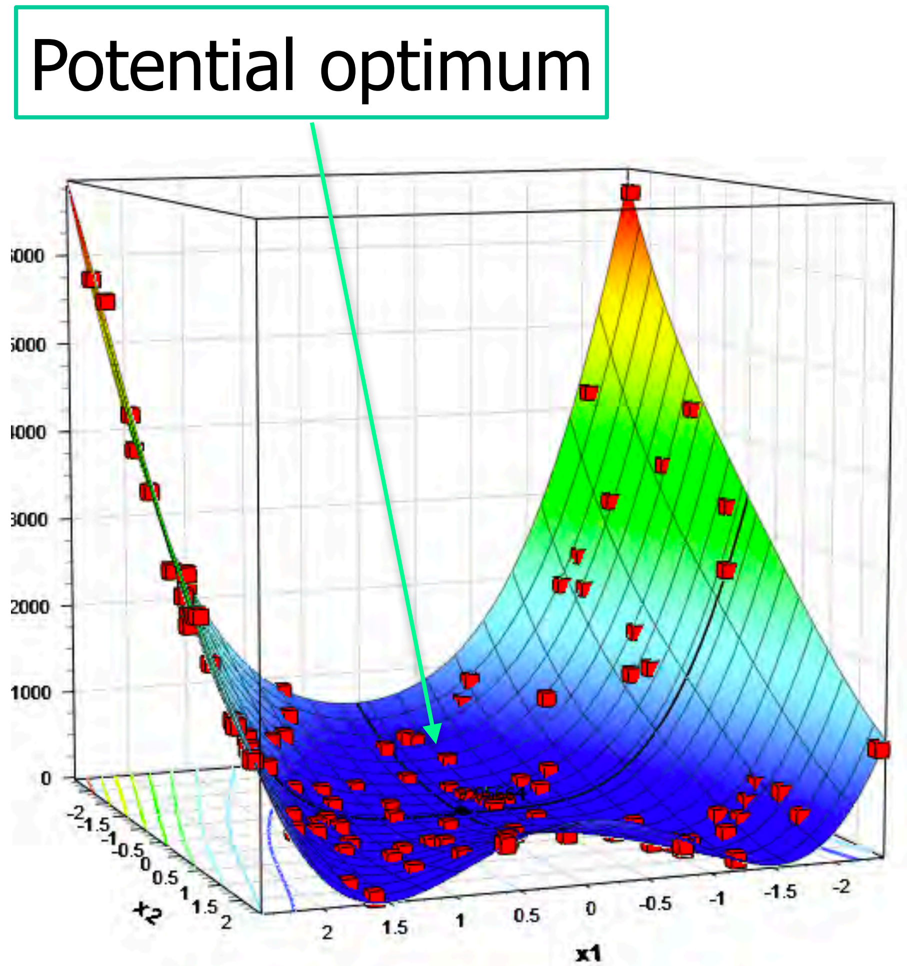
This constitutes a serious bottleneck in the exploration of the design space.

Designs are limited by humans' cognitive biases.

Kriging

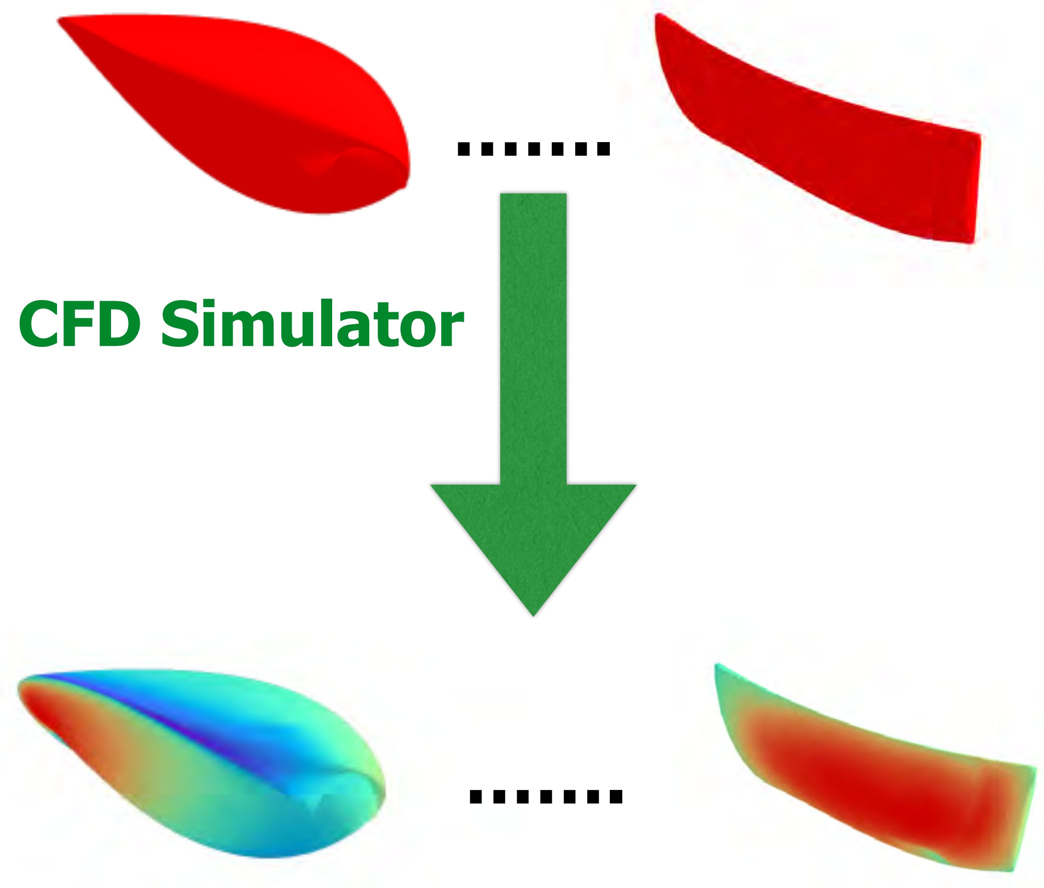


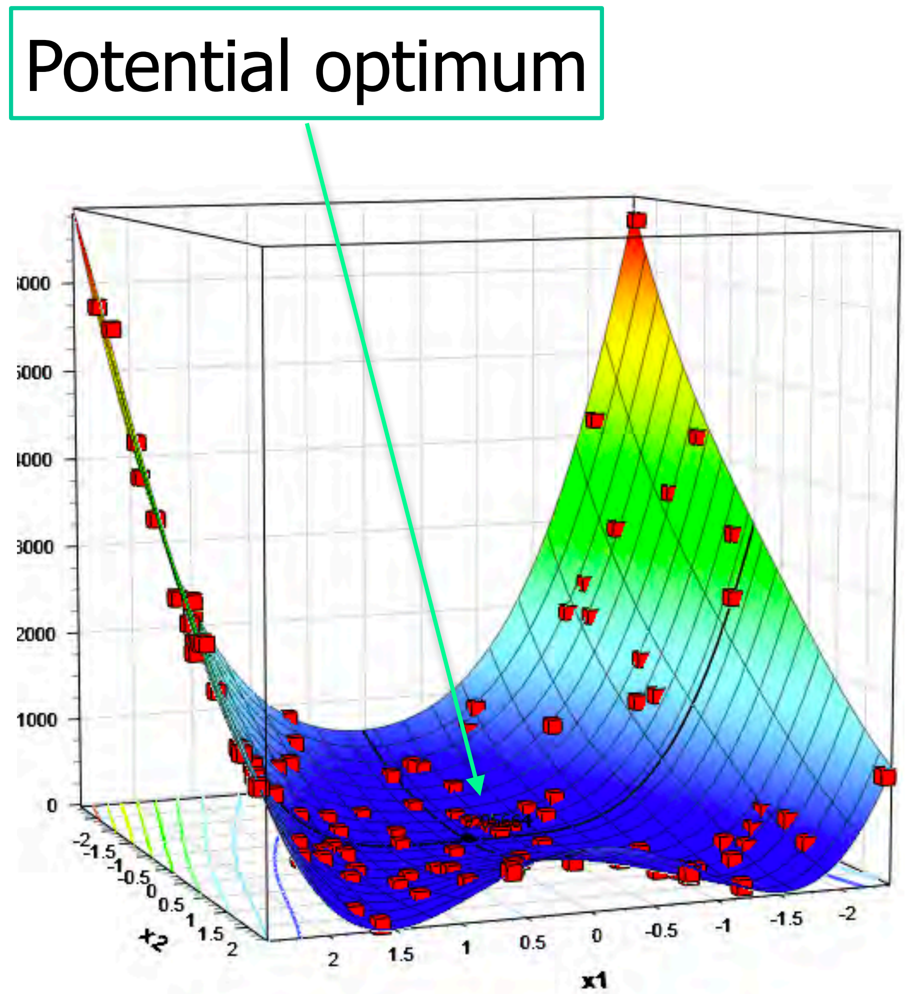
- Drag
- Pressure Coefficients
- Boundary Layer Velocities
- …



The response surface is approximated by a GP, which only works well when the model **has few parameters.**

Deep Surrogate Method



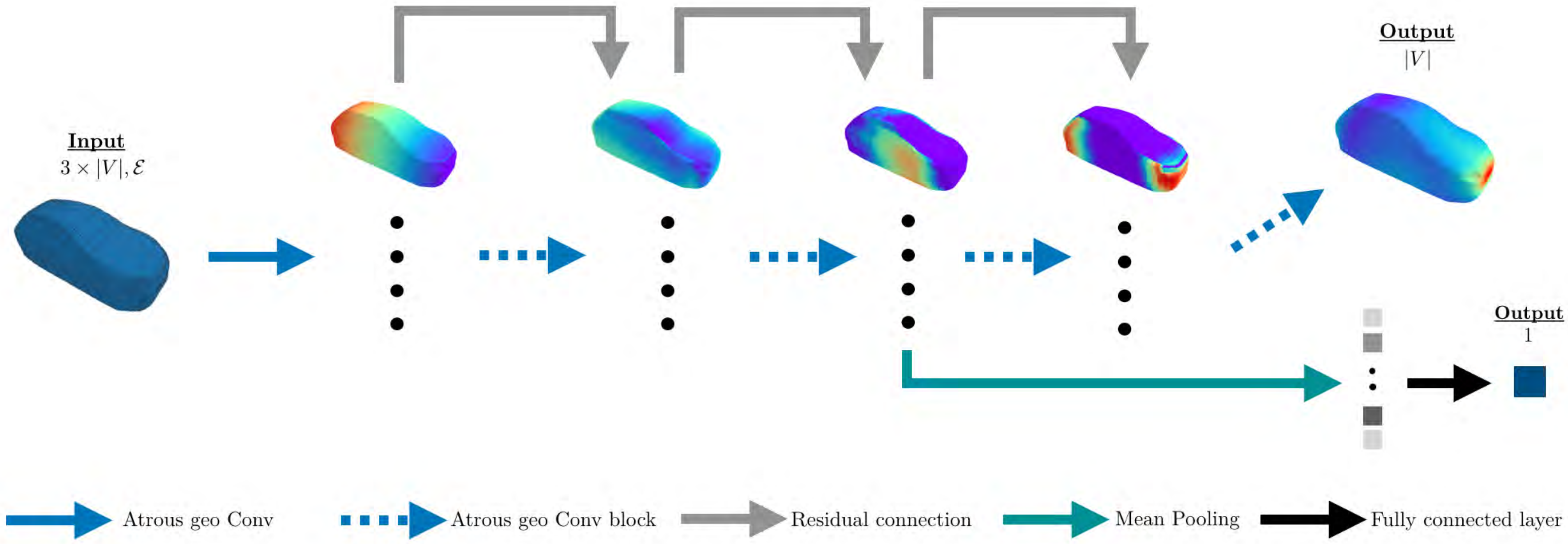


- Drag
- Pressure Coefficients
- Boundary Layer Velocities
- …

The response surface can be approximated by a **GCNN** instead of a GP.

—> The model can have any number of parameters.

GCNN

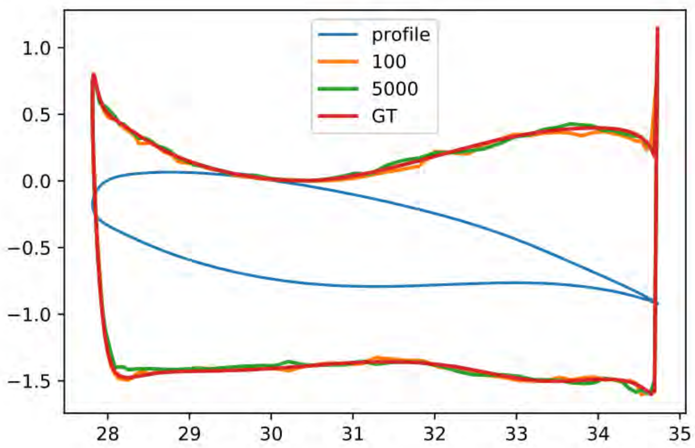


Operates directly on the mesh vertices.

Lift Prediction

Full Simulation (1 h)

GCNN Prediction (30 ms)

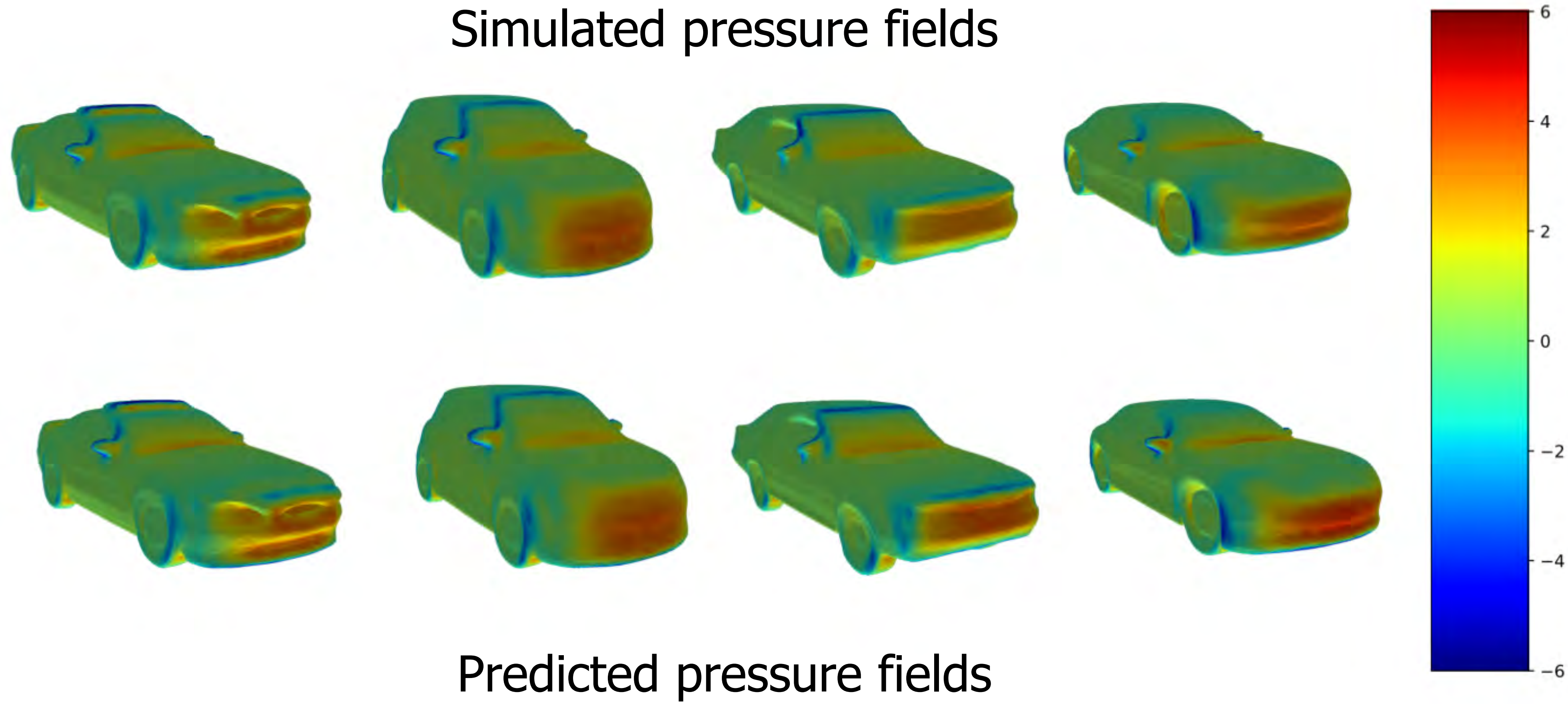


Physics Type	External Aerodynamics
Dataset size	~1000 shapes
R2-accuracy	95 %

EPFL

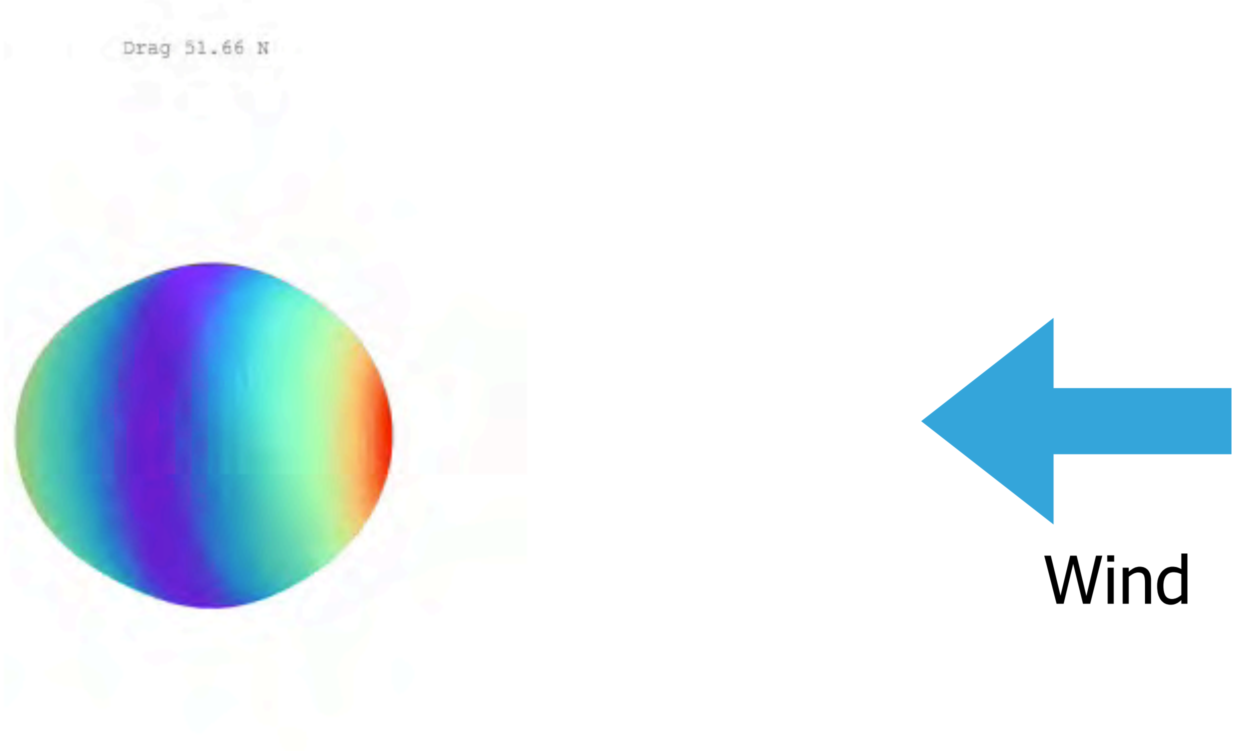
CVLab

Drag Prediction



- The predicted results are very close to the simulated ones.
- The aerodynamic drag $\mathscr{D}$ can be estimated from these predictions.
- $\mathscr{D}$ is a differentiable function of the surface mesh vertices.

Minimizing Drag Under Constraints

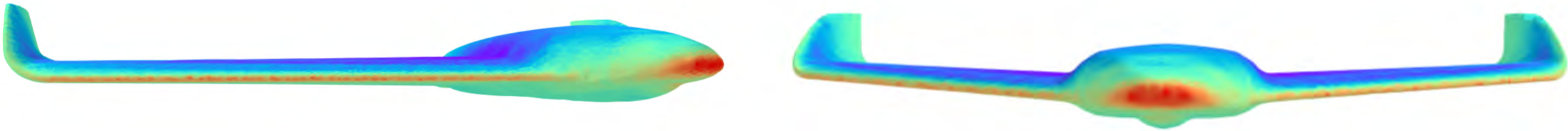


Minimizing drag while enclosing a sphere.

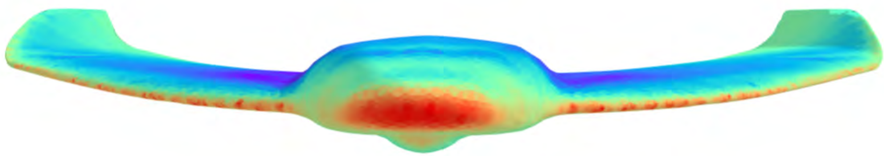
UAV Design

https://www.sensefly.com/

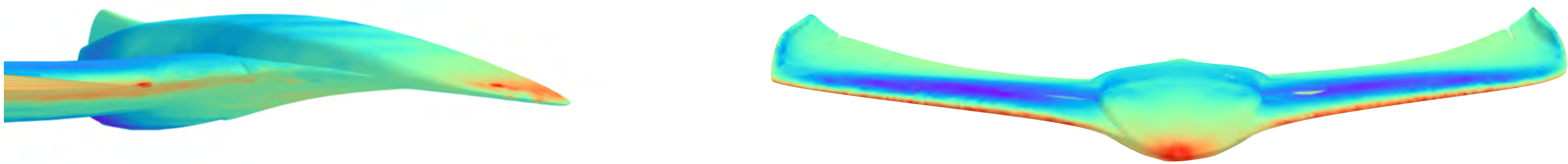
From UAV To Lifting Body



Sensefly drone (L/D 11.9)

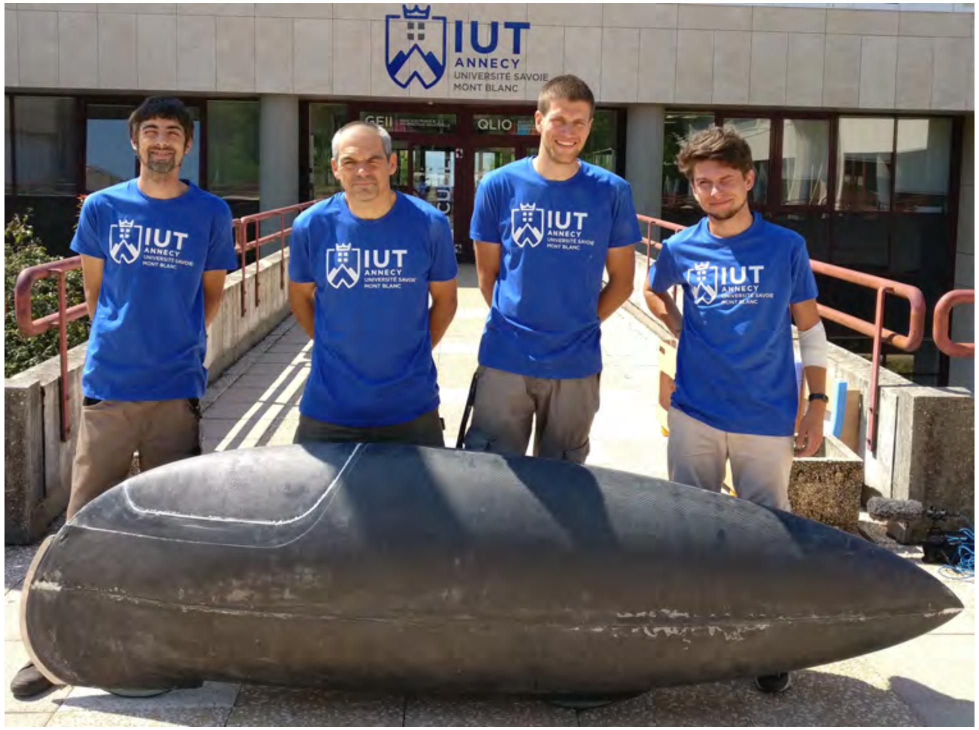


Optimize the wings (L/D 13.7)

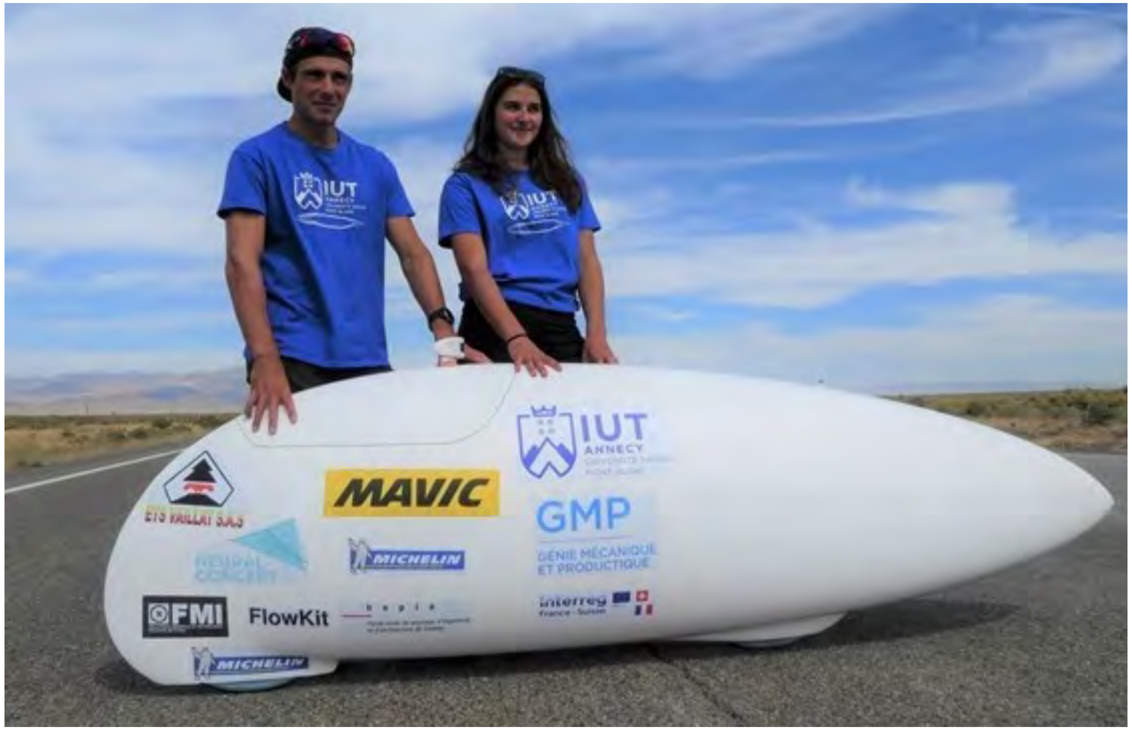


Optimize the fuselage as well

Bicycle Shell



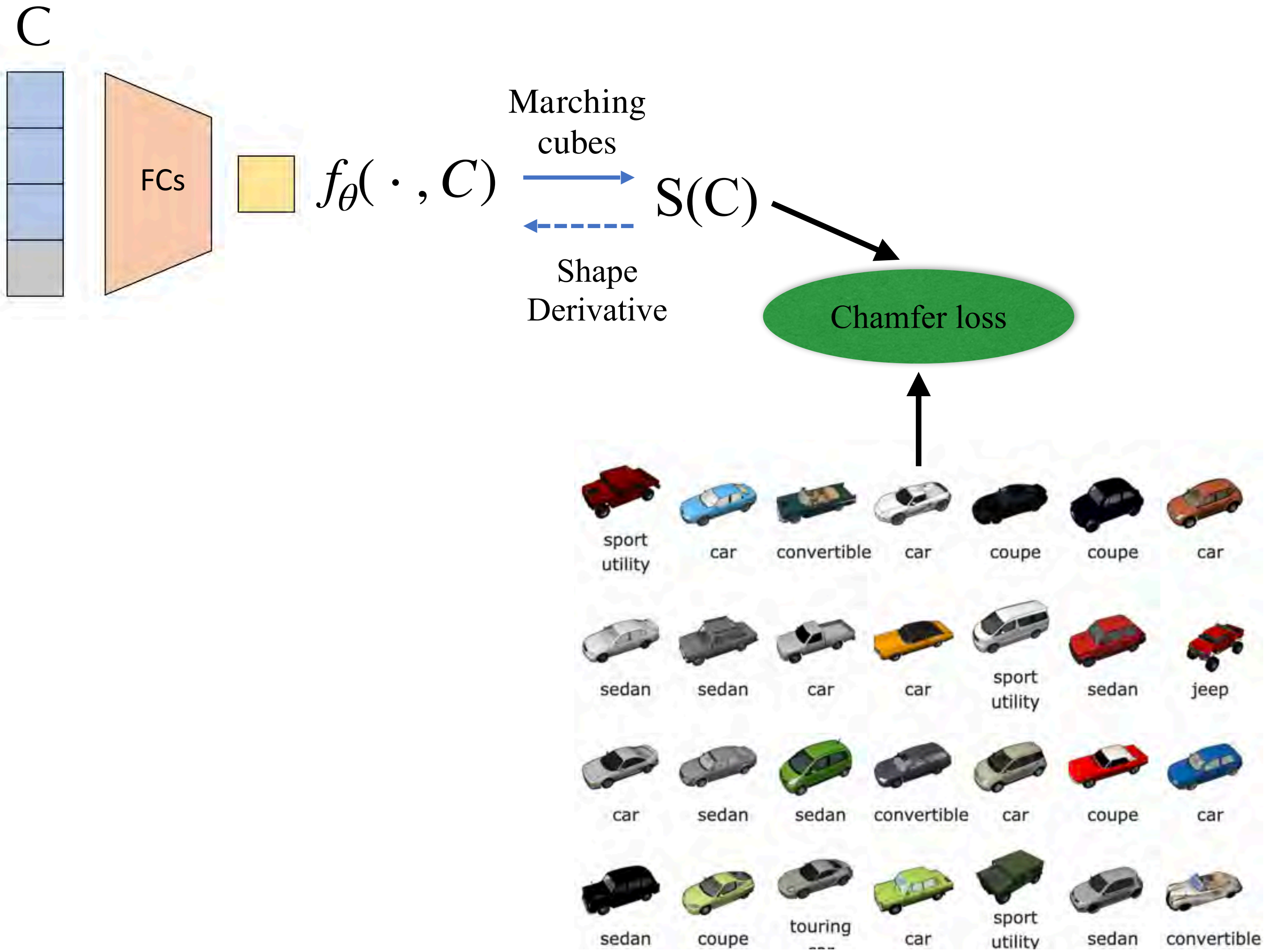
Altair 6, IUT Annecy, 2018



World Human Powered Speed Challenge
Battle Mountain Nevada, 2019

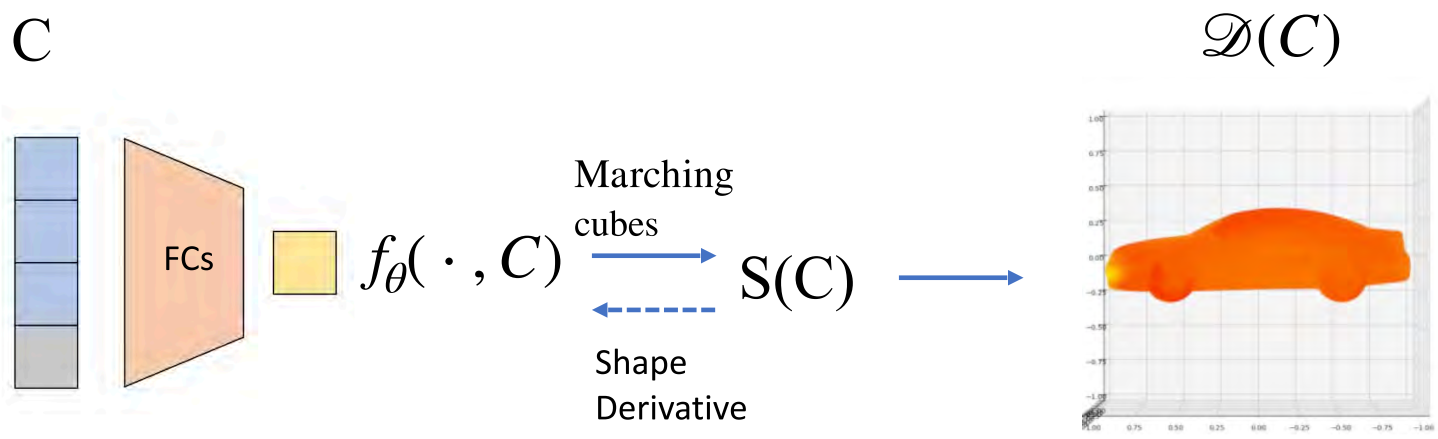
Women world record: 126,48 km/h
Men student world record: 136.74 km/h

Introducing Priors



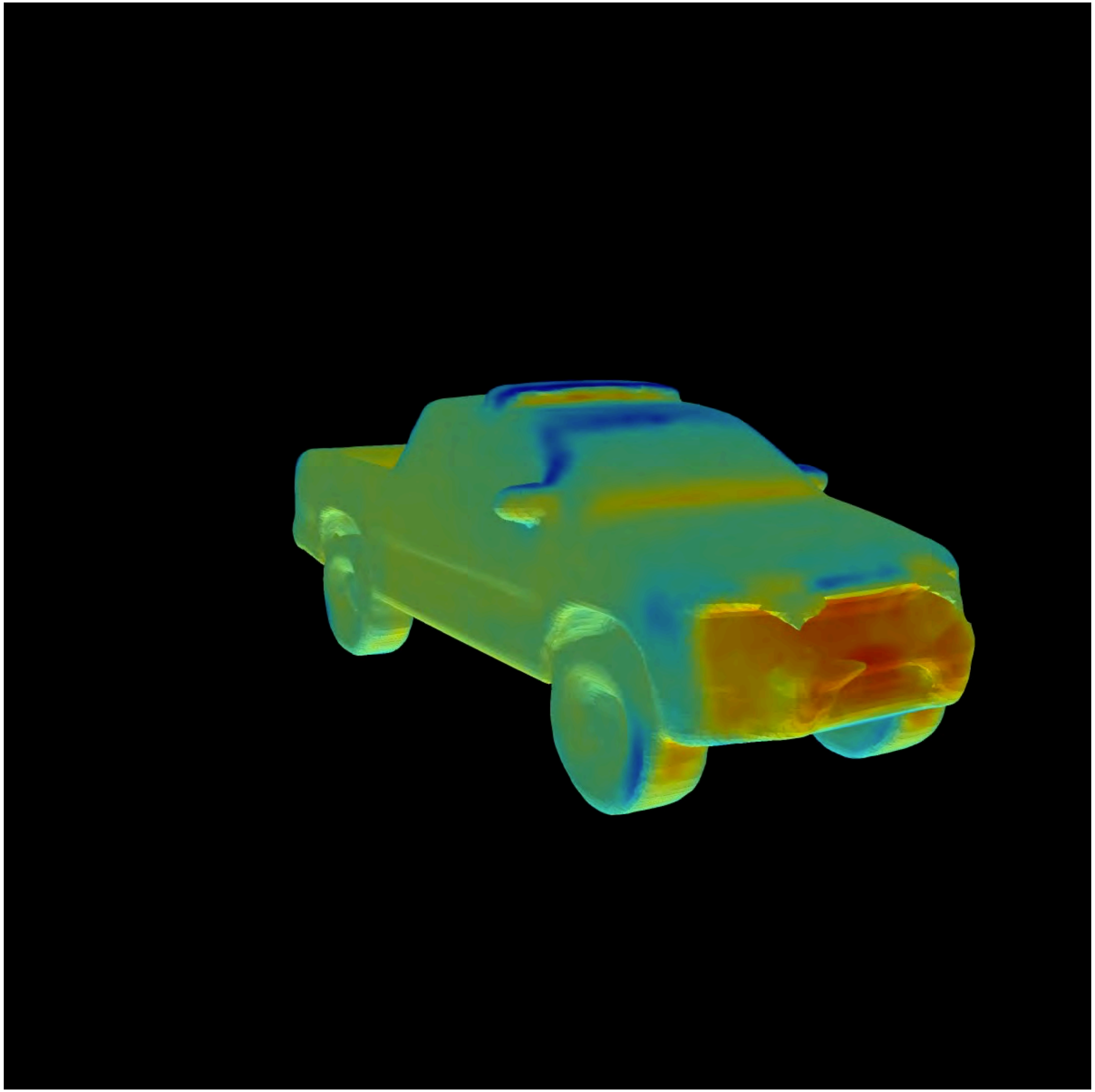
Train an auto-decoder using ShapeNet cars.

Drag Minimization

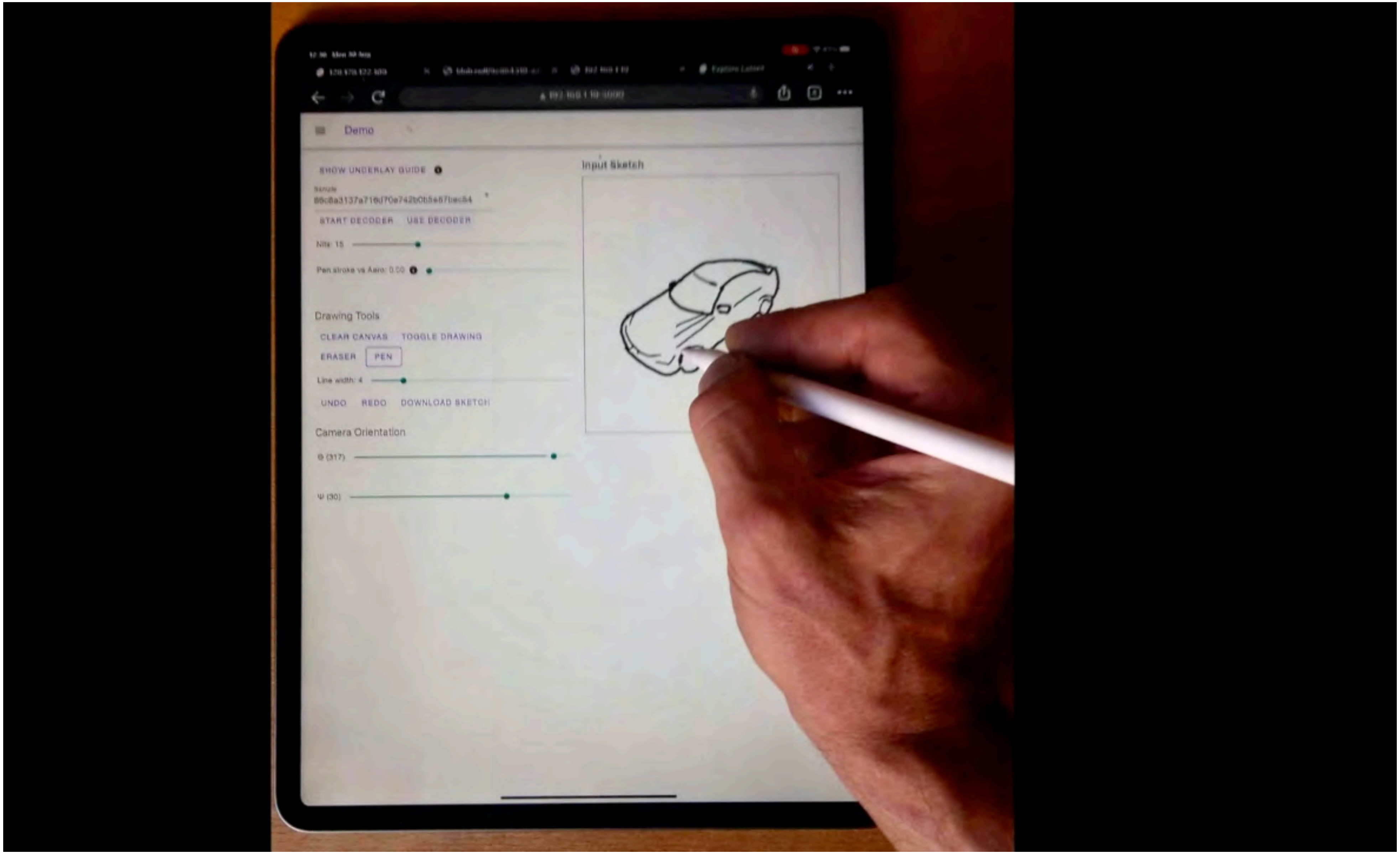


Minimize $\mathscr{D}(C)$ with respect to C under constraint.

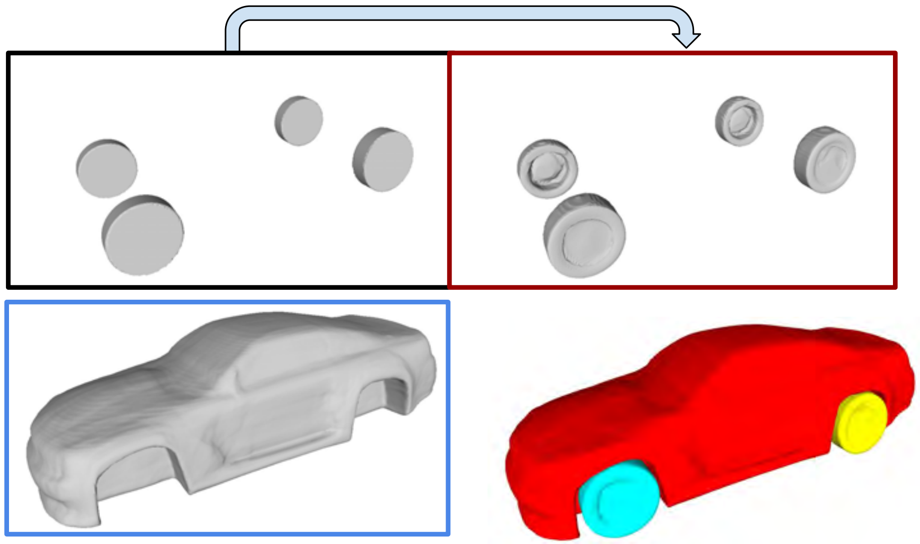
From Pickup-Truck to Sports Car



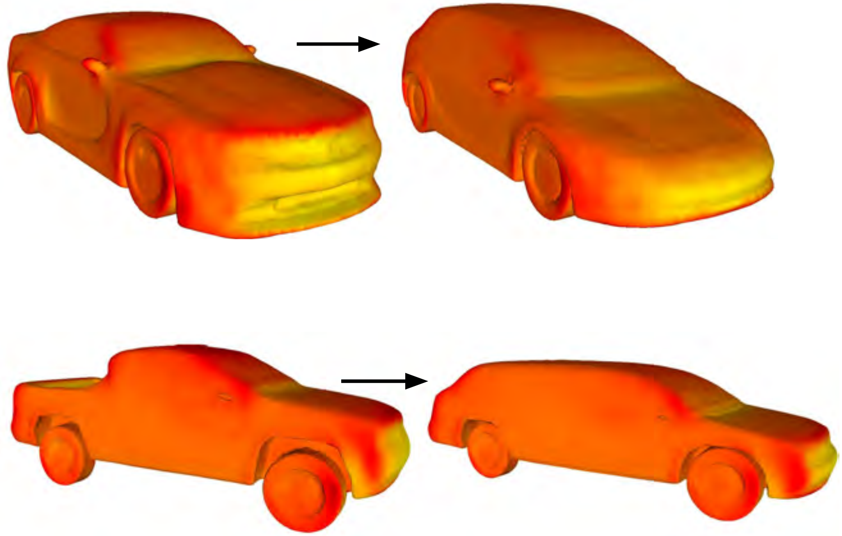
Interactive Design



Hybrid Shape Representation



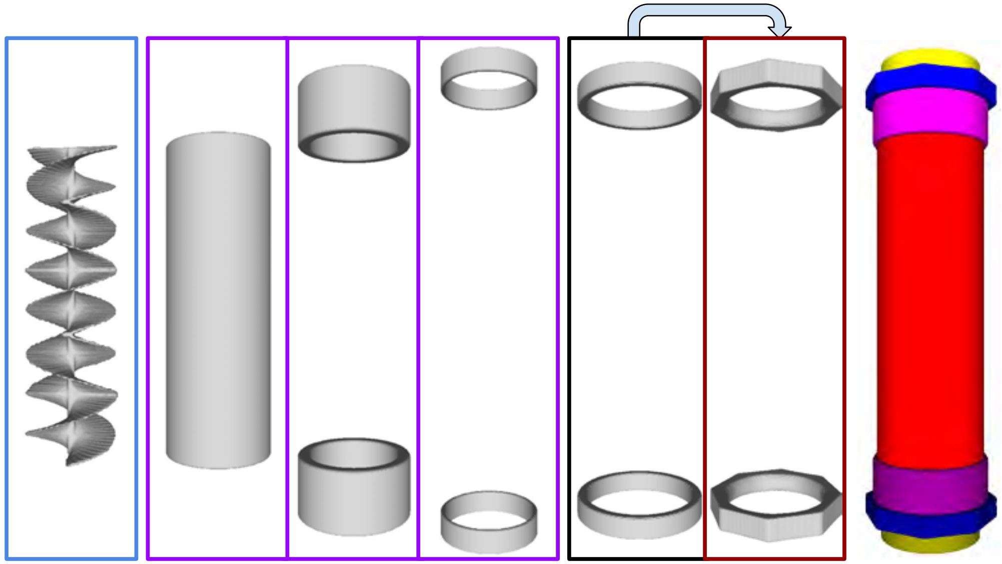
Different types of primitives



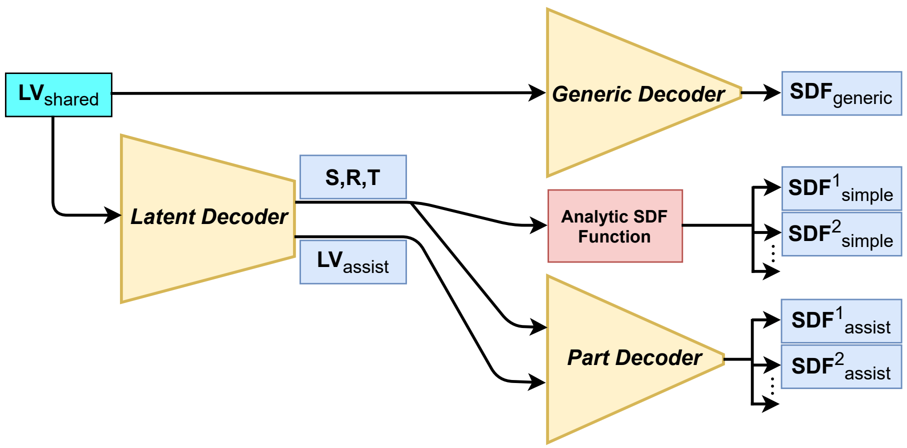
Optimization results

—> Individual parts adapt to each other.

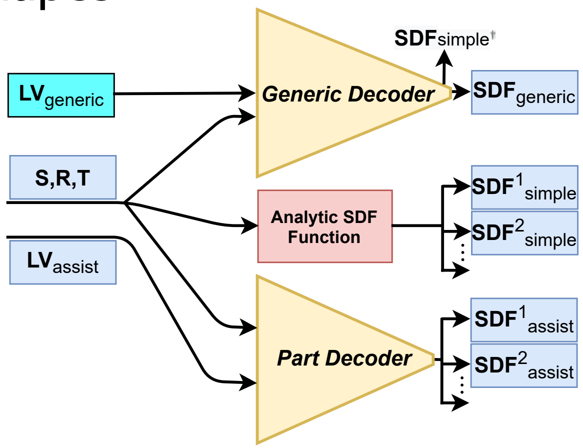
From Latent Vector to Primitives



We use SDFs to represent:
- Simple geometric primitives, such as spheres and cylinders.
- Primitives that bear a close resemblance to the simple ones but can deviate from them.
- Free form primitives that have arbitrarily complex shapes.

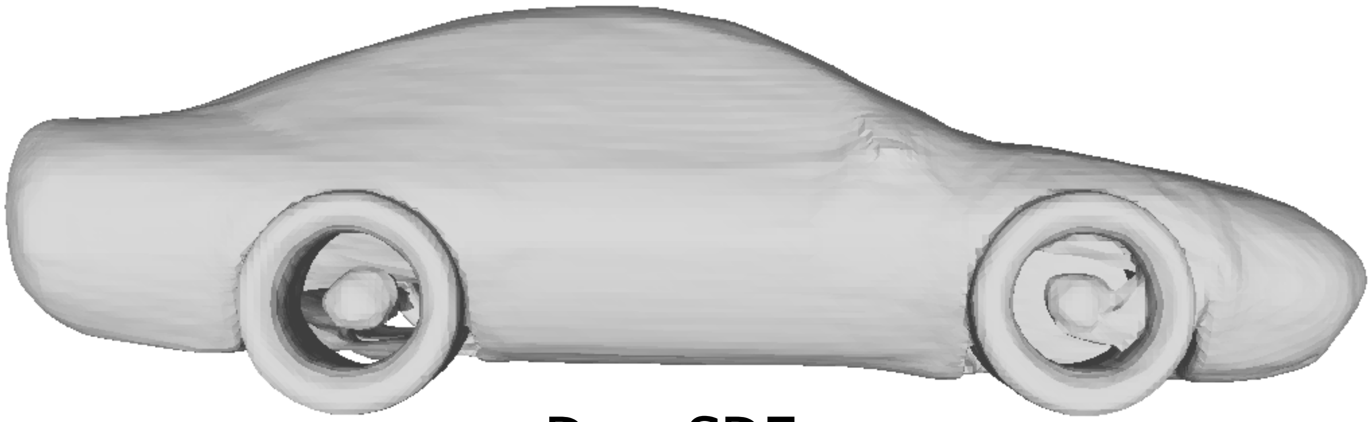


Shared Latent Vector

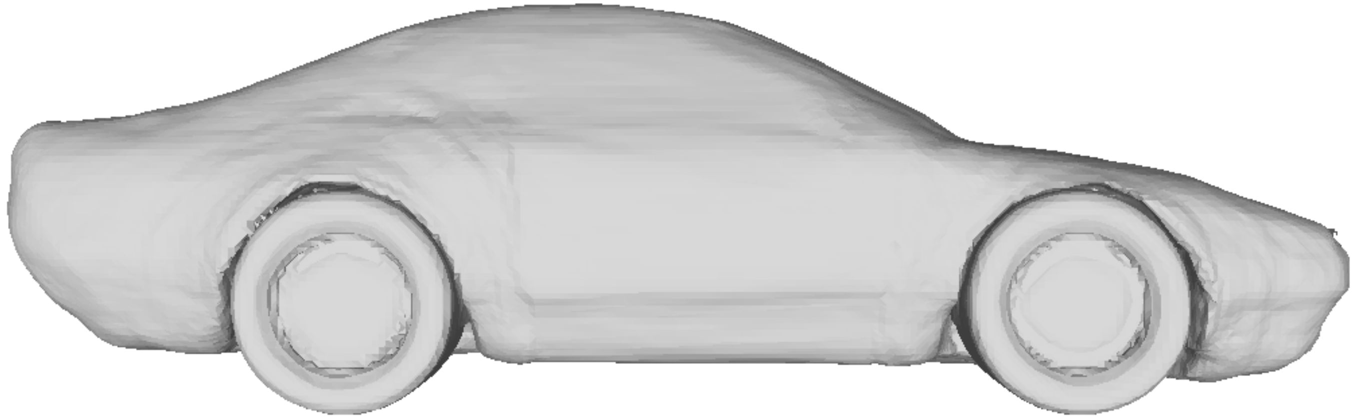


Disentangled Latent Vector

Car Wheels



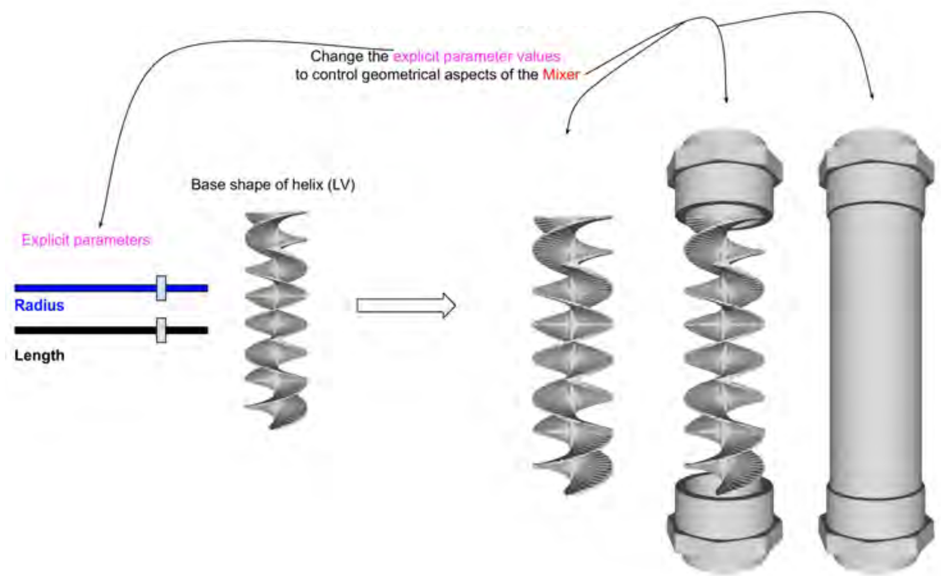
DeepSDF

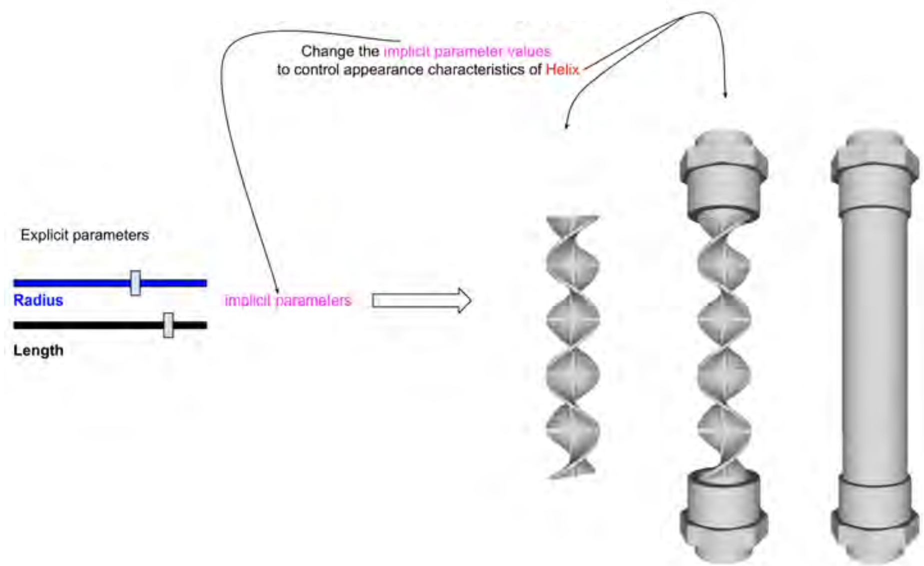


HybridSDF

The wheels are better separated from the car body.

Shape Manipulation

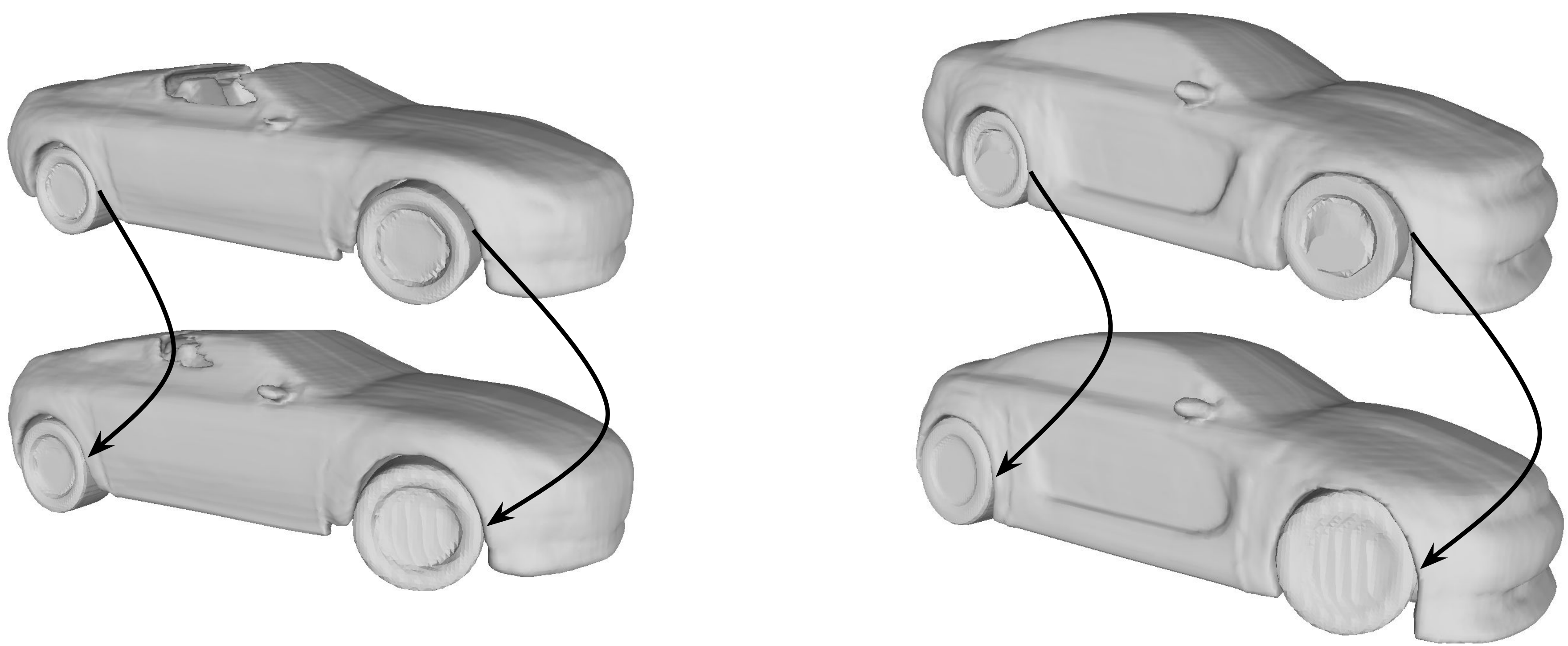




Changing the explicit parameters

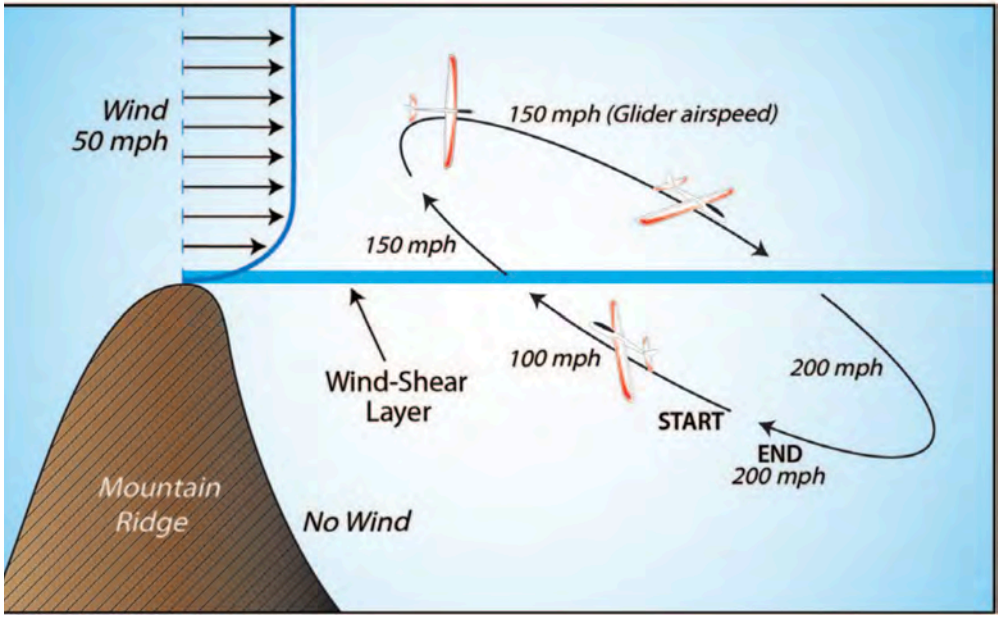
Changing the implicit parameters

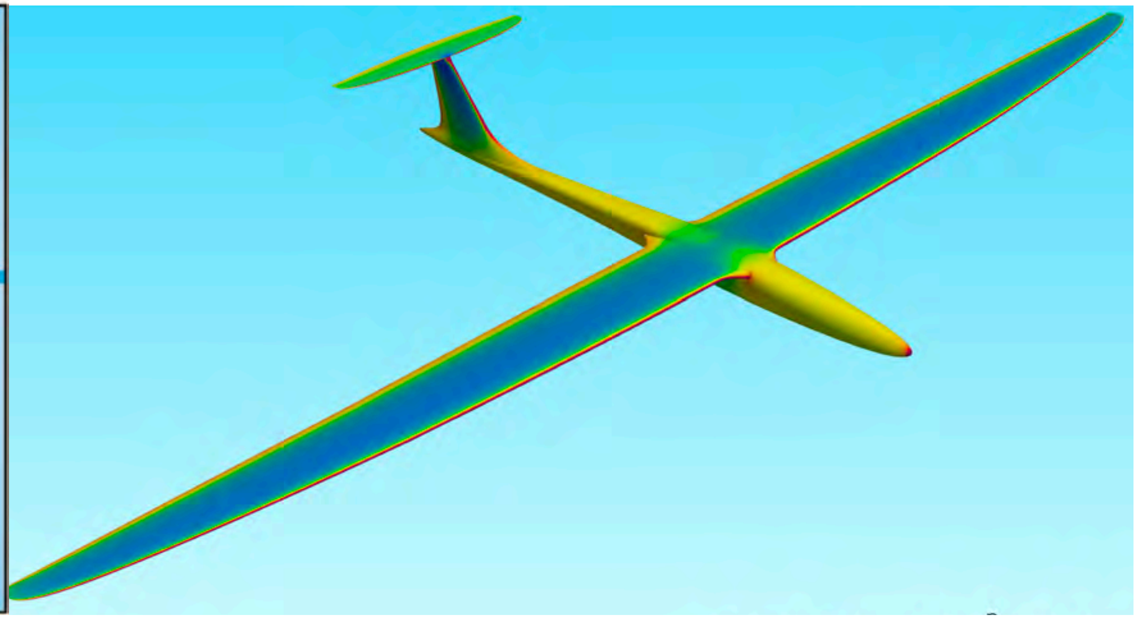
Interactive Shape Manipulation



Changing the wheels

Dynamic Soaring





- We plan to design for ease of control.
- We will use dynamic soaring to prove the concept.

Conclusion

- Combining explicit and implicit representations early makes it possible to exploit the strength of both representations.

- Deep Signed Distance Functions can be used to implement 3D surface meshes that can change their topology while preserving end-to-end differentiability.

—> This opens the door for new applications in fields as diverse as Computer Assisted Design and Medical Imaging.

Many Thanks To

• Timur Bagautdinov	(NeuralConcept)
• Pierre Baque	(NeuralConcept)
• Benoît Guillard	(EPFL)
• Graham Knott	(EPFL)
• Artem Lukoianov	(NeuralConcept)
• Edoardo Remelli	(EPFL)
• Stephan Richter	(Intel)
• Udaranga Wickramasinghe	(EPFL)
• Pierre Yvernay	(EPFL)