

Compositional Generative Inverse Design



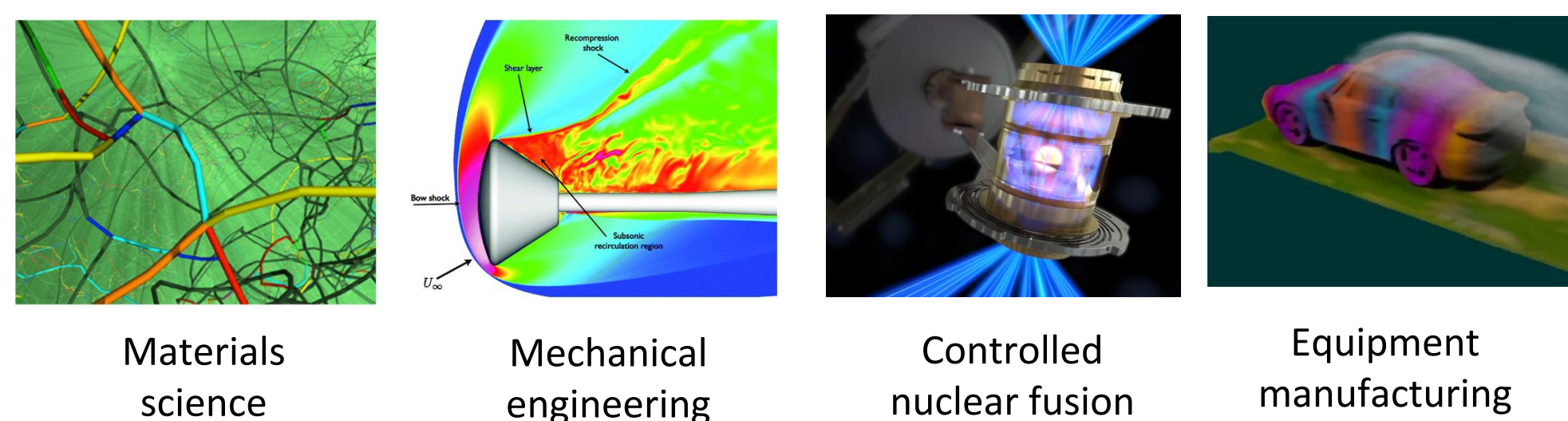
¹Tailin Wu*, ²Takashi Maruyama*, ¹Long Wei*, ¹Tao Zhang*, ³Yilun Du*, ⁴Gianluca Iaccarino, ⁴Jure Leskovec

¹Westlake University, ²NEC Laboratories Europe,

³Massachusetts Institute of Technology, ⁴Stanford University

Motivation

- Inverse design is an important task to find a set of high-dimensional design parameters for a dynamical system to optimize an objective under a set of constraints. It has wide applications in both science and engineering.
- Compositional inverse design is an inverse design task that finds a set of optimal parameters for a dynamical system that contains more parameters than seen datasets.



Key challenges:

- Complex design space and high computational cost
- How to characterize the complex dynamics and interactions among components within a physical process?
- How to generalize to more complex composition scenarios?

Traditional physical simulation methods

Pros: (1) first principle-based, (2) accurate, (3) error guaranteed.
Challenges: (1) high computational cost, (2) requirements for rich expert knowledge, (3) weak at dealing with high-dimensional design space

Recent deep learning-based methods

Pros: (1) alleviates much engineering efforts. (2) Offer speedup
Challenges: (1) still need huge computation via system evolution, (2) easy to fall into adversarial mode, (3) hard to generalize to more complex composition scenarios beyond training set

Summary of prior works:

Existing methods have significant limitations such as easily getting stuck in suboptimal solutions and inability to generalize.

Approach

Problem setting:

- Simultaneously design the state U and the parameter γ (e.g., boundary and initial conditions).
- Joint objective:

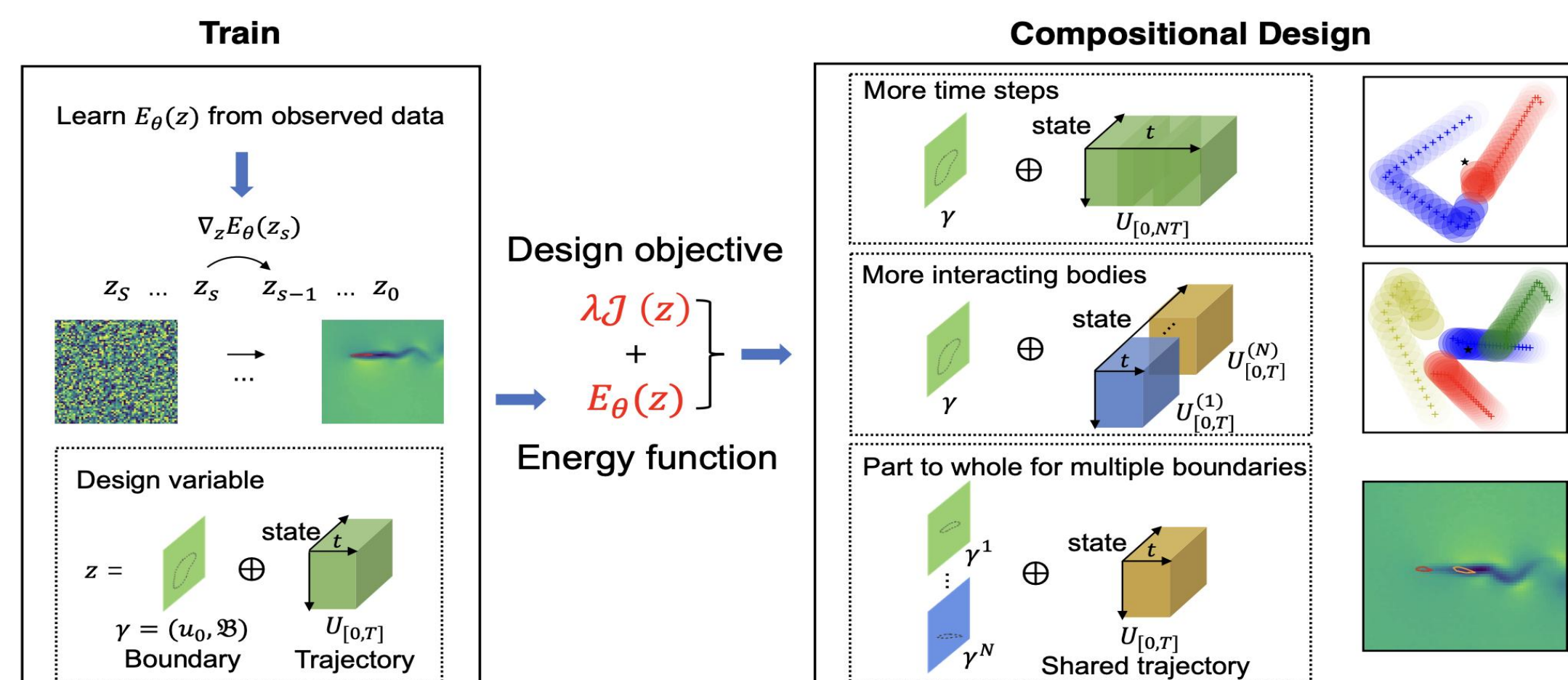
$$\hat{\gamma} = \arg \min_{\gamma, U_{[0,T]}} [E_{\theta}(U_{[0,T]}, \gamma) + \lambda \cdot \mathcal{J}(U_{[0,T]}, \gamma)]$$

E_{θ} : energy-based model; serves the purpose of a surrogate model in approximating simulator dynamics

\mathcal{J} : design objective

Keypoints of CinDM:

- Uses energy functions to implicitly govern dynamics and interactions among different components faithfully in the system
- Uses diffusion model to learn the energy functions of composition
- Introduces the design objective effectively during inference, while ensuring physical consistency



Implementation of CinDM

- Training: $L_{MSE} = \|\epsilon - \epsilon_{\theta}(\sqrt{1 - \beta_s} z + \sqrt{\beta_s} \epsilon, s)\|_2^2, \epsilon \sim N(0, I)$
- Compositional inference:

$$z_{s-1} = z_s - \eta \frac{1}{N} \sum_{i=1}^N (\epsilon_{\theta}^i(z_s^i, s) + \lambda \nabla_z \mathcal{J}(z_s)) + \xi, \quad \xi \sim N(0, \sigma_s^2 I)$$

- 1. Generalization to more time steps

$$E_{\theta}(U_{[0,T]}, \gamma) = \sum_{i=1}^N E_{\theta}(U_{[(i-1) \cdot t_g, i \cdot t_g + T^{tr}]}, \gamma)$$

- 2. Generalization to more interacting bodies

$$E_{\theta}(U_{[0,T]}, \gamma) = \sum_{i < j} E_{\theta}(U_{[0,T]}^{(i)}, U_{[0,T]}^{(j)}, \gamma)$$

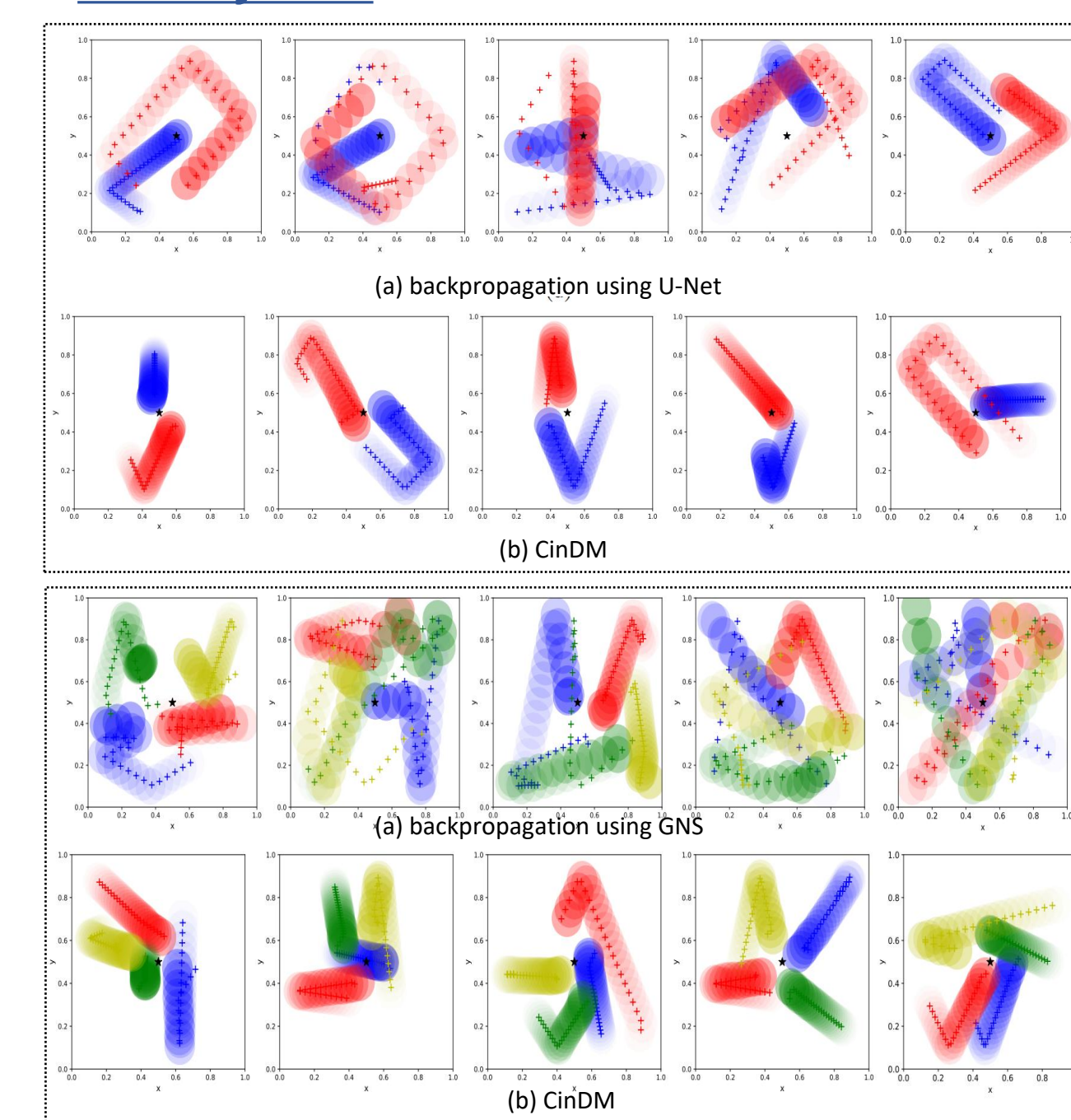
- 3. Generalization from part to whole for boundaries.

$$E_{\theta}(U_{[0,T]}, \gamma) = \sum_{i=1}^N E_{\theta_i}(U_{[0,T]}, \gamma^i)$$

Results

- Our method outperforms state-of-the-art neural inverse design method by an average of 41.5% in prediction MAE and 14.3% in design objective for the N-body dataset
- Our method discovers formation flying to minimize drag in the multi-airfoil design task.

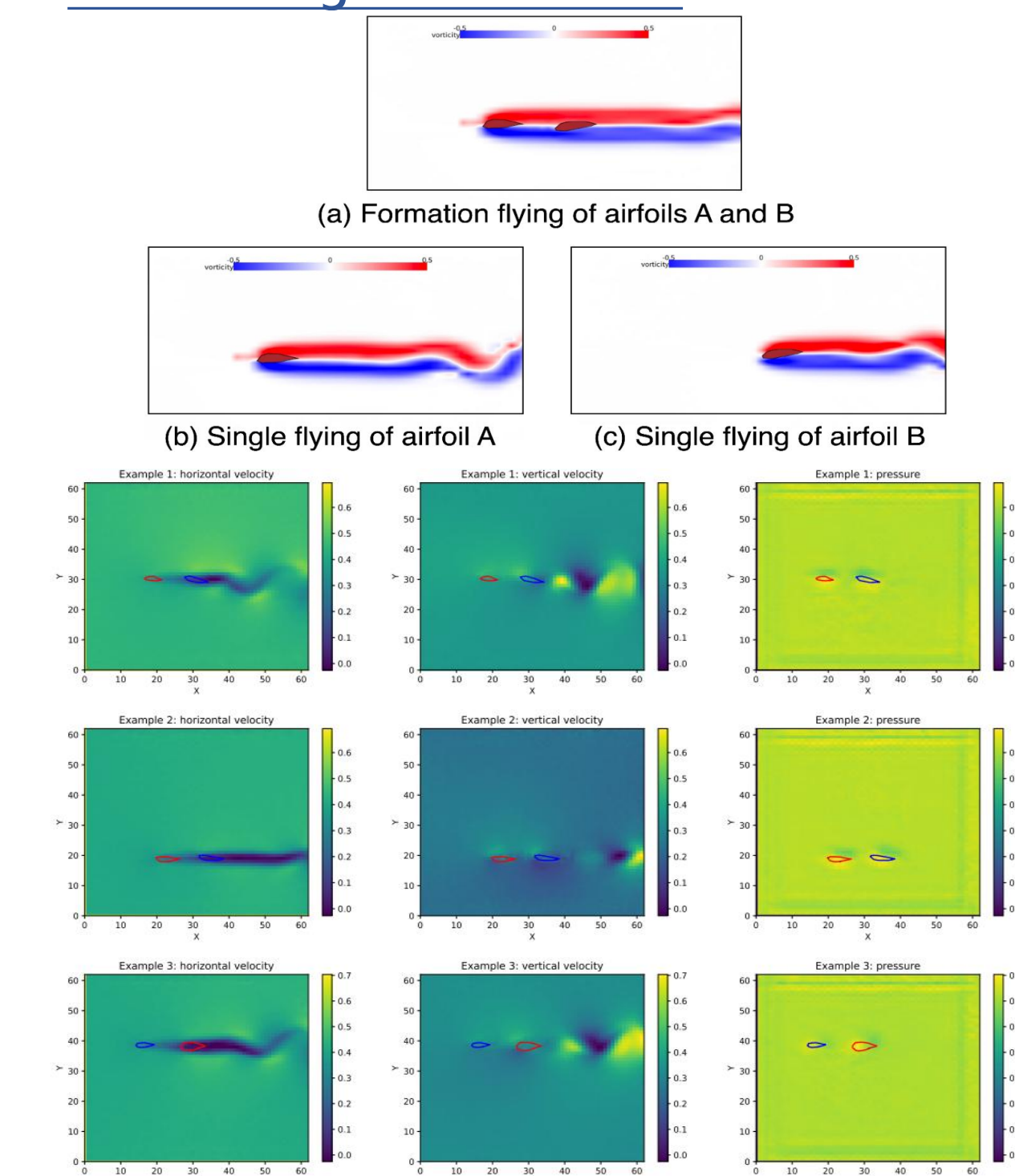
N-body task



Compositional inverse design in time: 2-body 54 steps: results of the best baseline(backpropagation using U-Net) and CinDM

Compositional inverse design generalizing to more objects: 4-body 44 steps: results of the best baseline (backpropagation using GNS) and CinDM

2D airfoils generation task



Discovered formation flying. Our model's designed boundary forms a "leader" and "follower" formation (a), reducing the drag by 53.6% and increasing the lift-to-drag ratio by 66.1% compared to each airfoil flying separately (b)(c).

Compositional design results of our method in 2D airfoil generation. Each row represents an example. We show velocities in horizontal and vertical directions and pressure in the initial time step, inside which we plot the generated airfoil boundaries.



QR code for paper