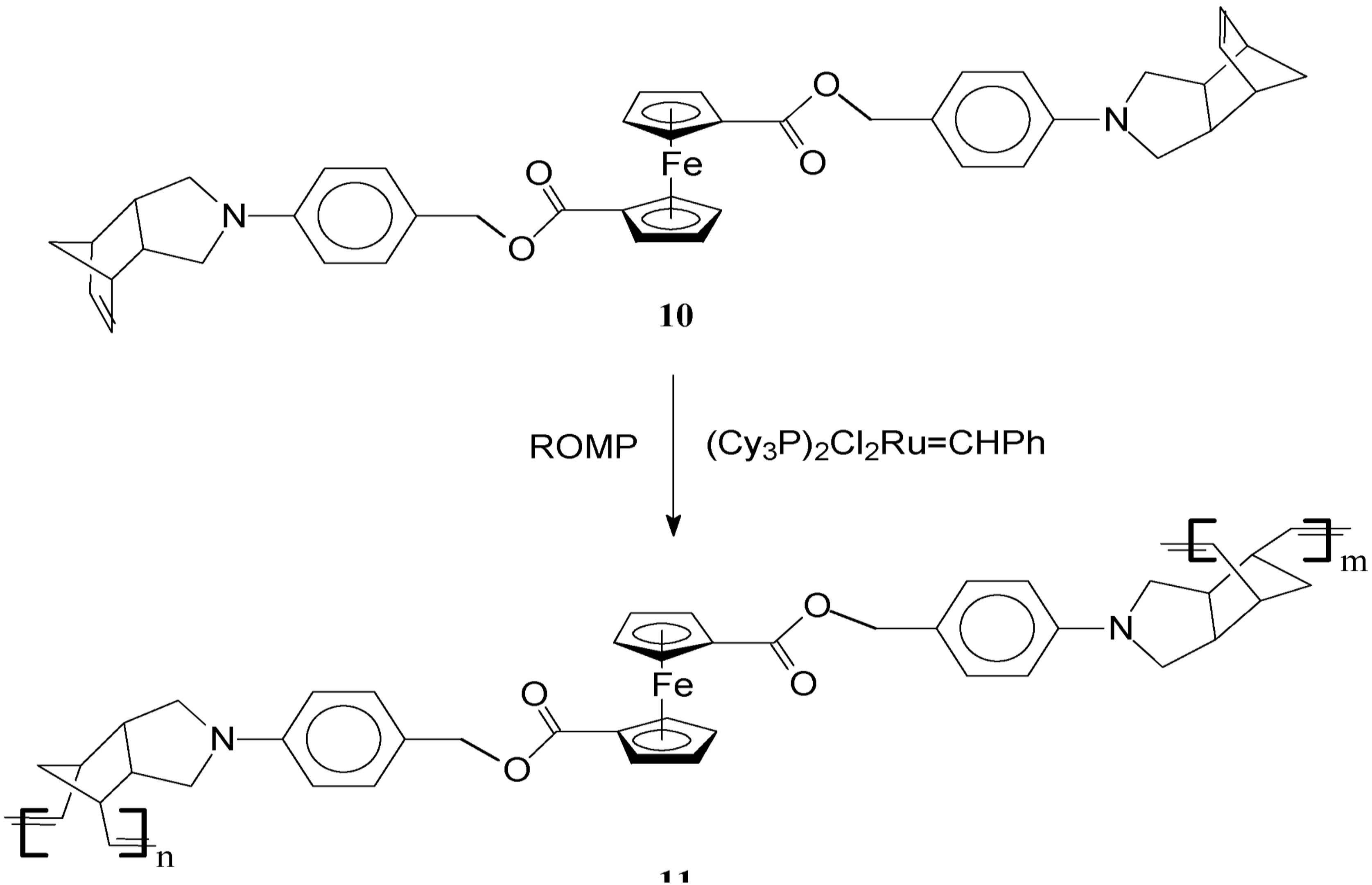
Can Computers Beat Humans at Design?

Wojciech Matusik
MIT

Design is Everywhere

Design is Everywhere



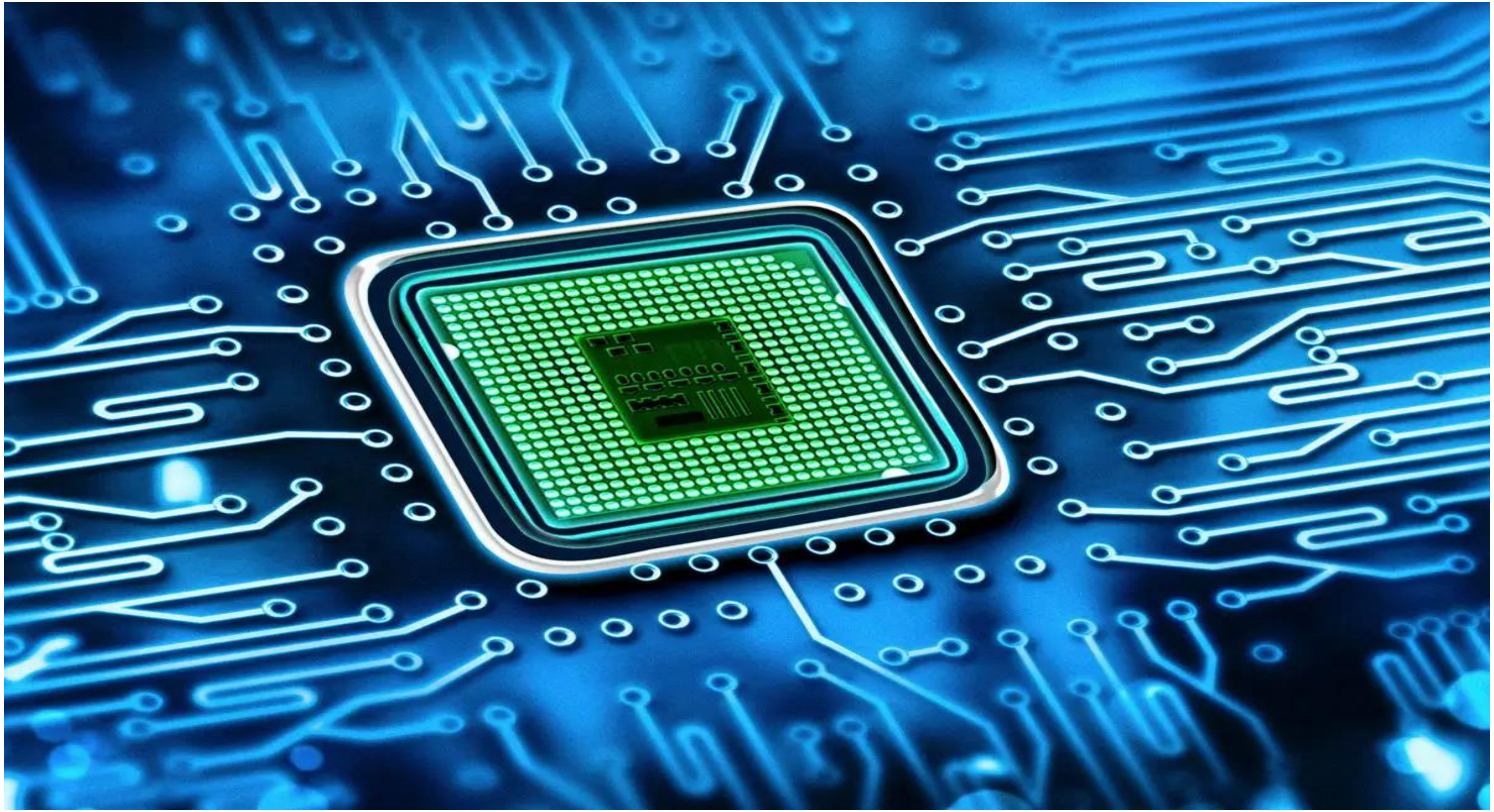
Design is Everywhere

Dijkstra's Algorithm Pseudocode in C++

```
function dijkstraalgorithm(G, S)
    for each node N in G
        dist[N] <- infinite
        prev[N] <- NULL
        If N != S, add N to Priority Queue Q
    dist[S] <- 0

    while Q IS NOT EMPTY
        U <- Extract MIN from Q
        for each unmarked neighbour N of U
            temporaryDist <- dist[U] + edgeWeight(U, N)
            if temporaryDist < dist[N]
                dist[N] <- temporaryDist
                prev[N] <- U
    return dist[], prev[]
```

Design is Everywhere

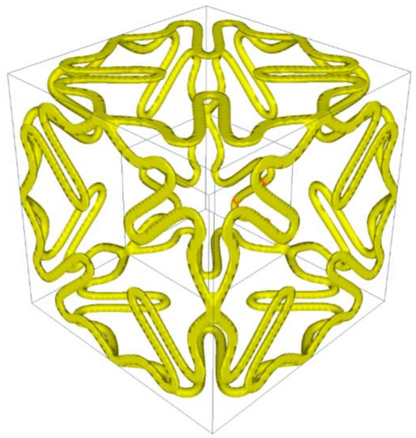


Printed, functional walker

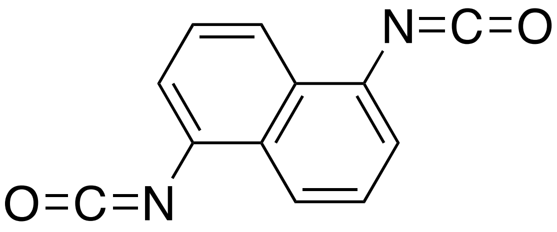
Approach

Study different design problems and generalize

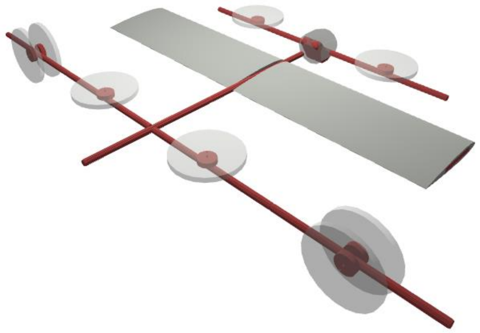
Metamaterials



Chemistry



Cyber-physical Systems



Key Questions for Computational Design

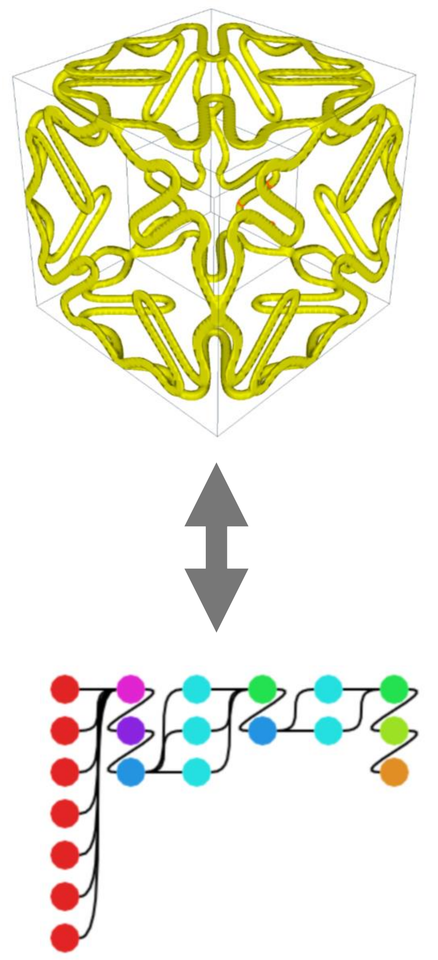
1. How to represent a design?
2. How to represent a design space?
3. How to learn a design space?
4. How to find designs with optimal performance?
5. How to bridge the gap between digital & real?

Key Questions for Computational Design

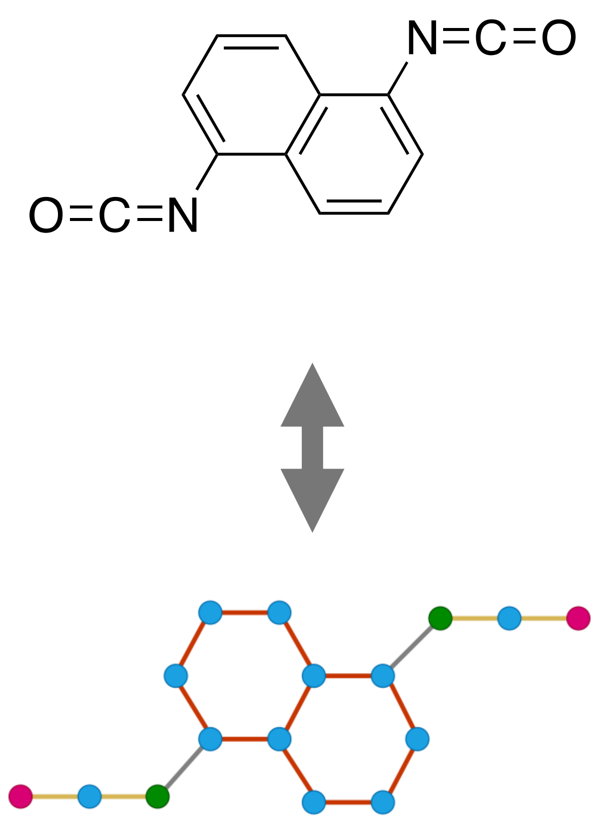
1. How to represent a design?
2. How to represent a design space?
3. How to learn a design space?
4. How to find designs with optimal performance?
5. How to bridge the gap between digital & real?

Design Can Be Represented As a Graph

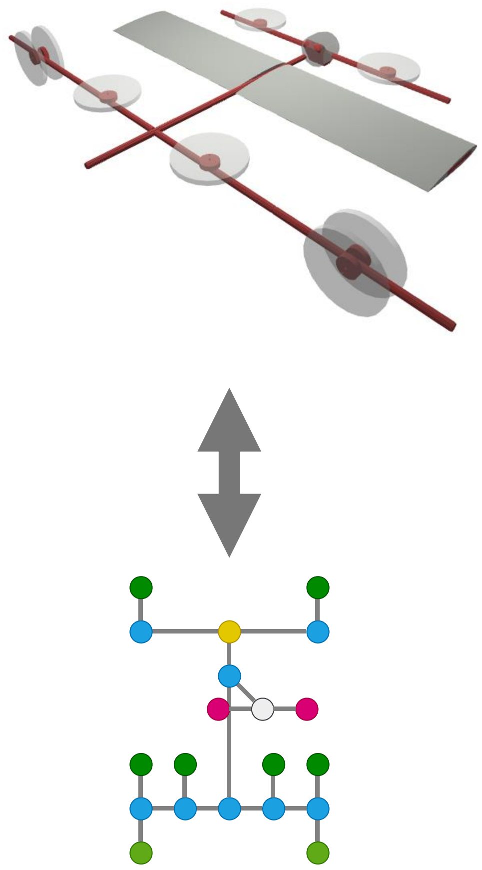
Metamaterials



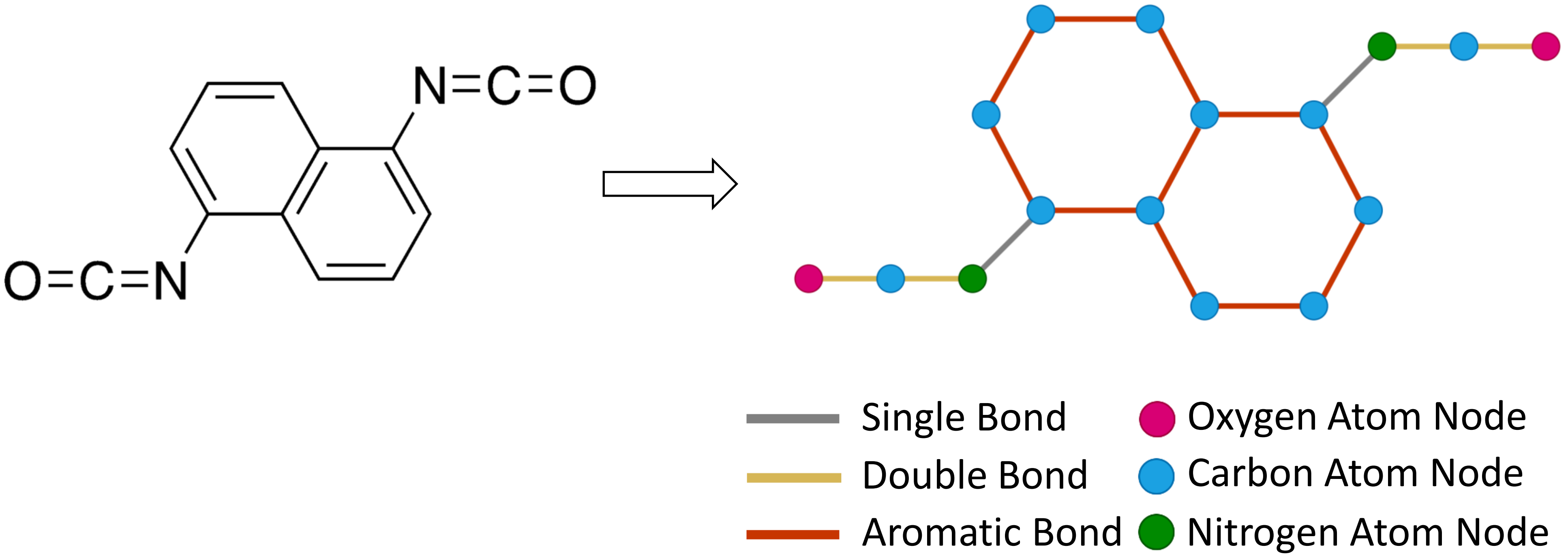
Chemistry



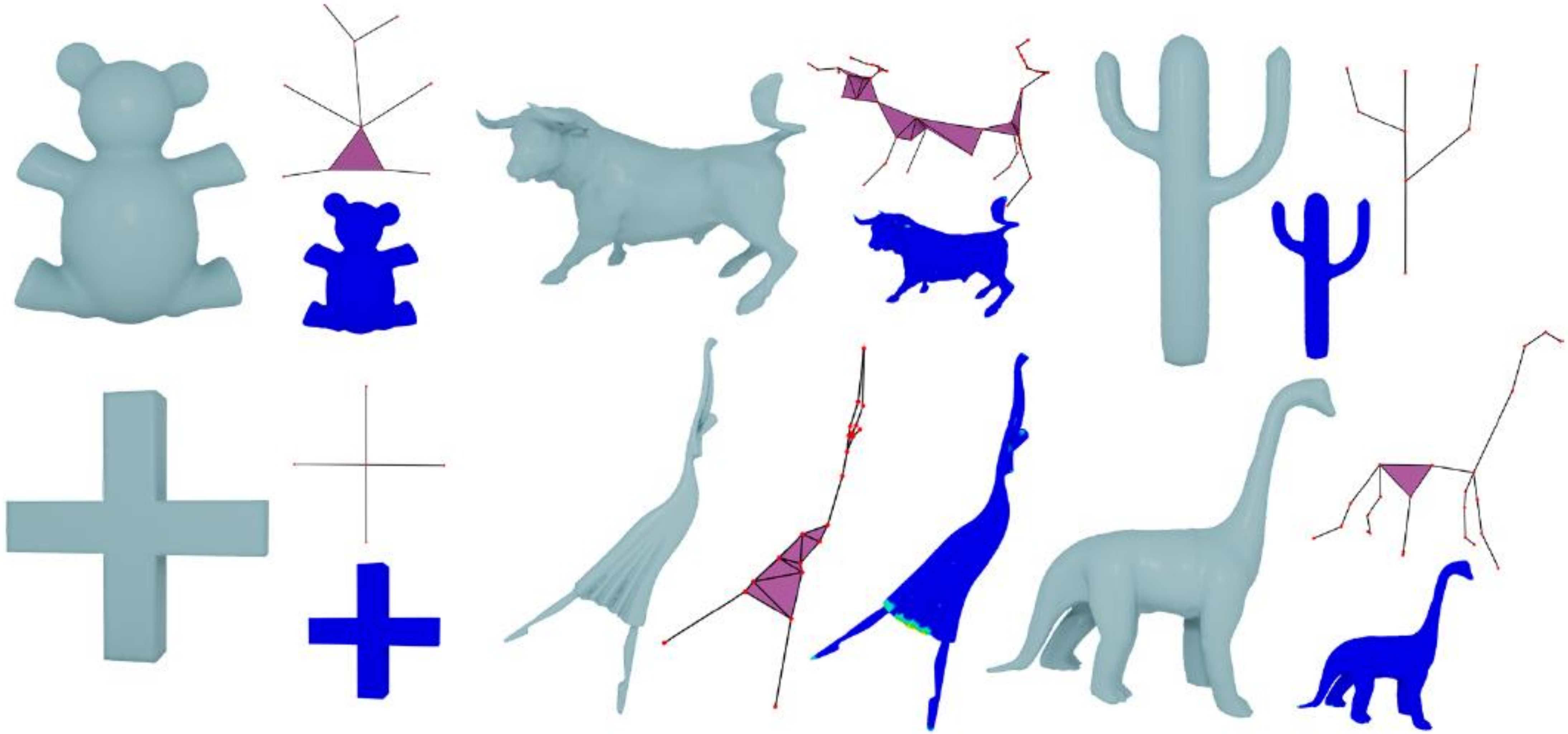
Cyber-physical Systems



Molecules As Graphs



Geometry As Graphs



Key Questions for Computational Design

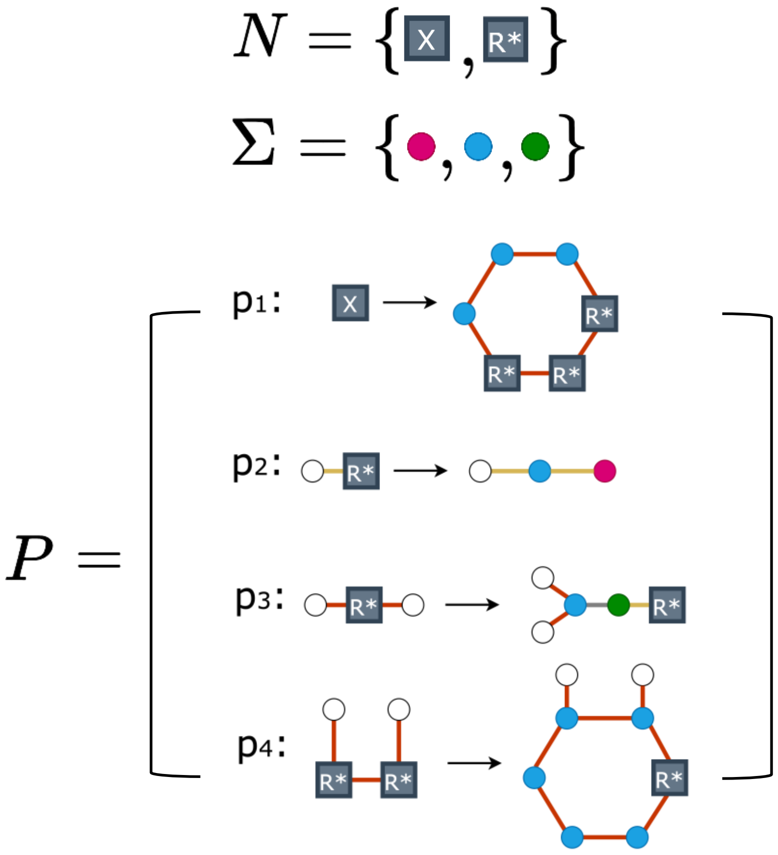
1. How to represent a design?
2. How to represent a design space?
3. How to learn a design space?
4. How to find designs with optimal performance?
5. How to bridge the gap between digital & real?

Graph Grammars As Design Spaces

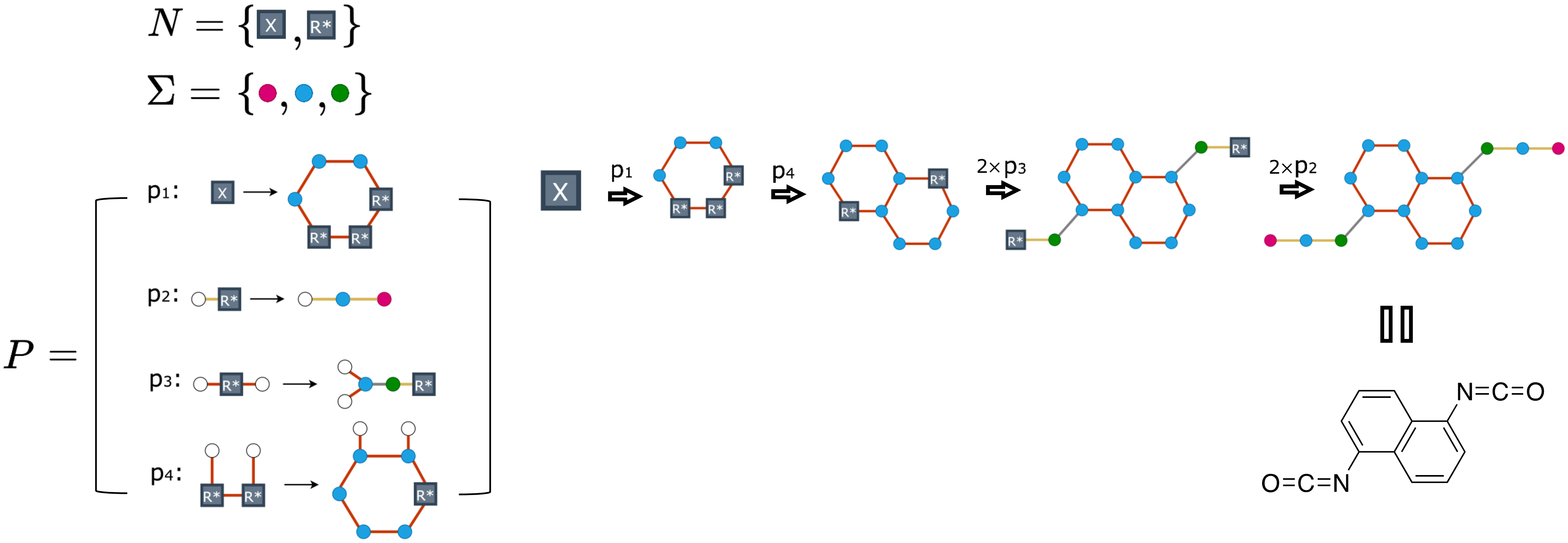
$$N = \{\boxed{\text{X}}, \boxed{\text{R*}}\}$$

$$\Sigma = \{\bullet, \bullet, \bullet\}$$

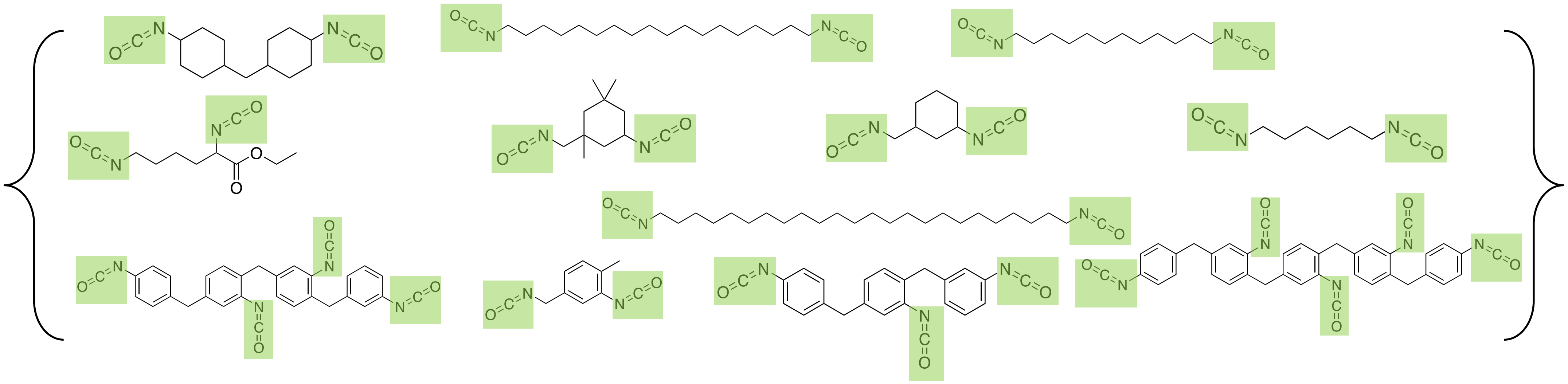
Graph Grammars As Design Spaces



Graph Grammars As Design Spaces

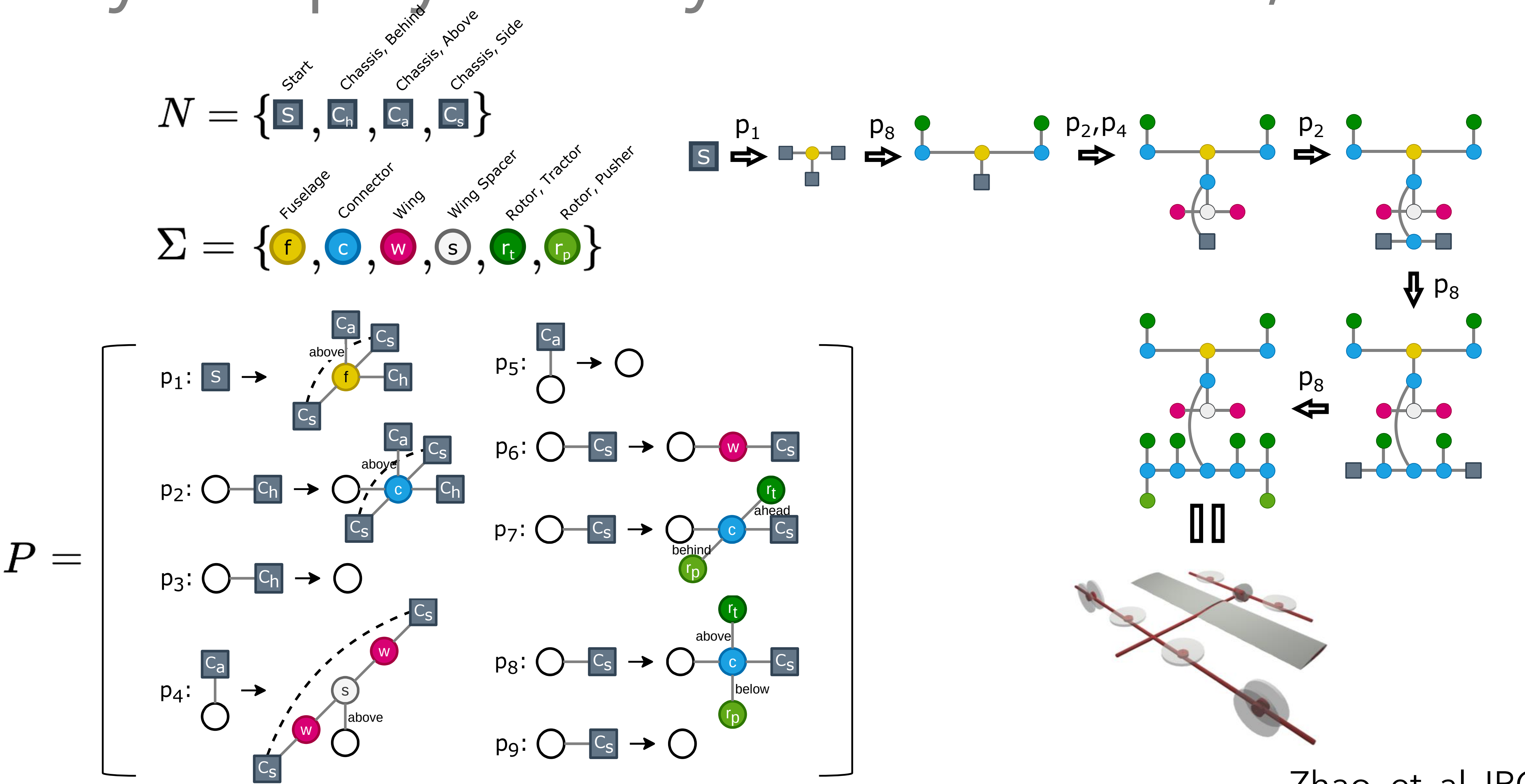


Molecules: Polygrammar



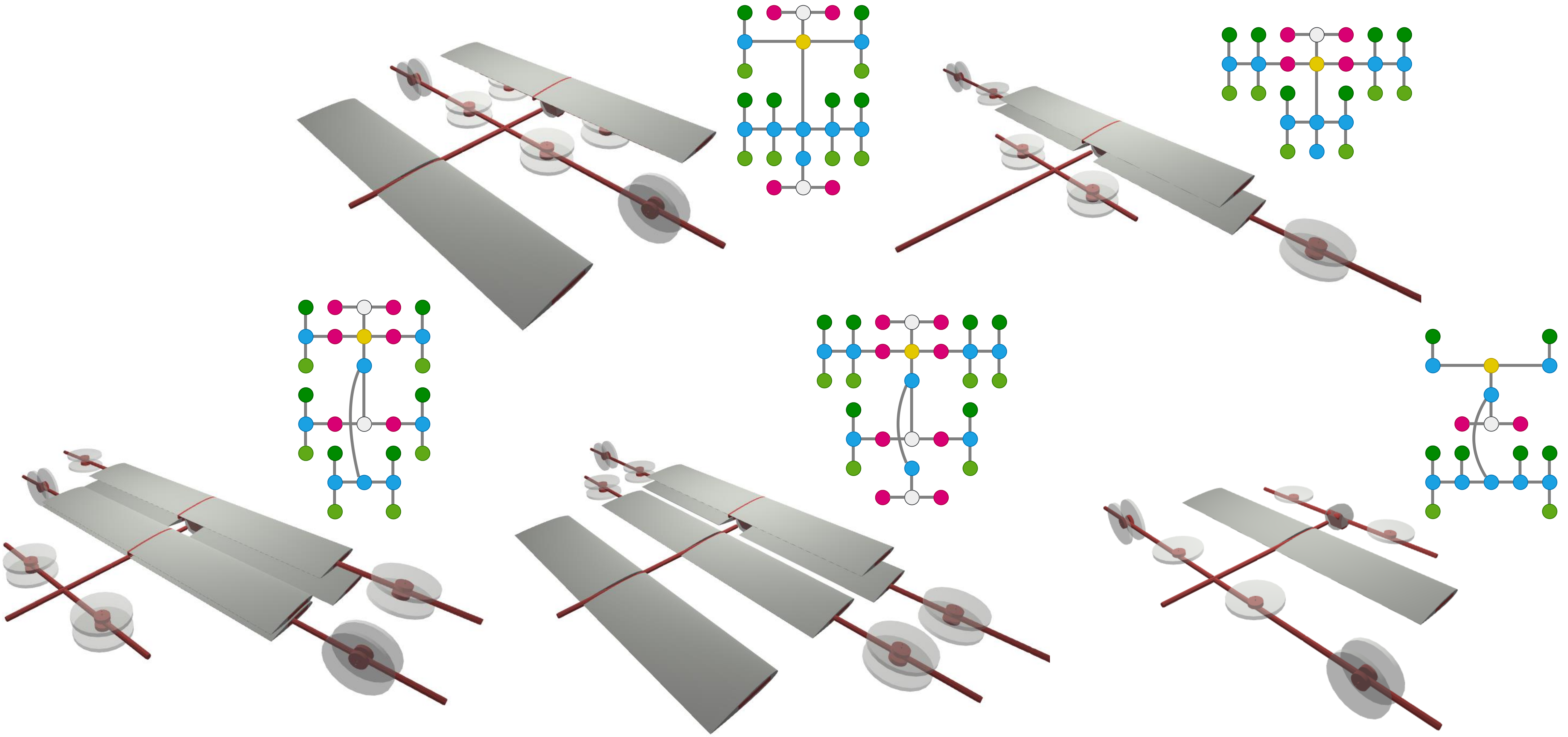
Guo et. al, Advanced Science, 2022

Cyber-physical Systems: Robots, Drones

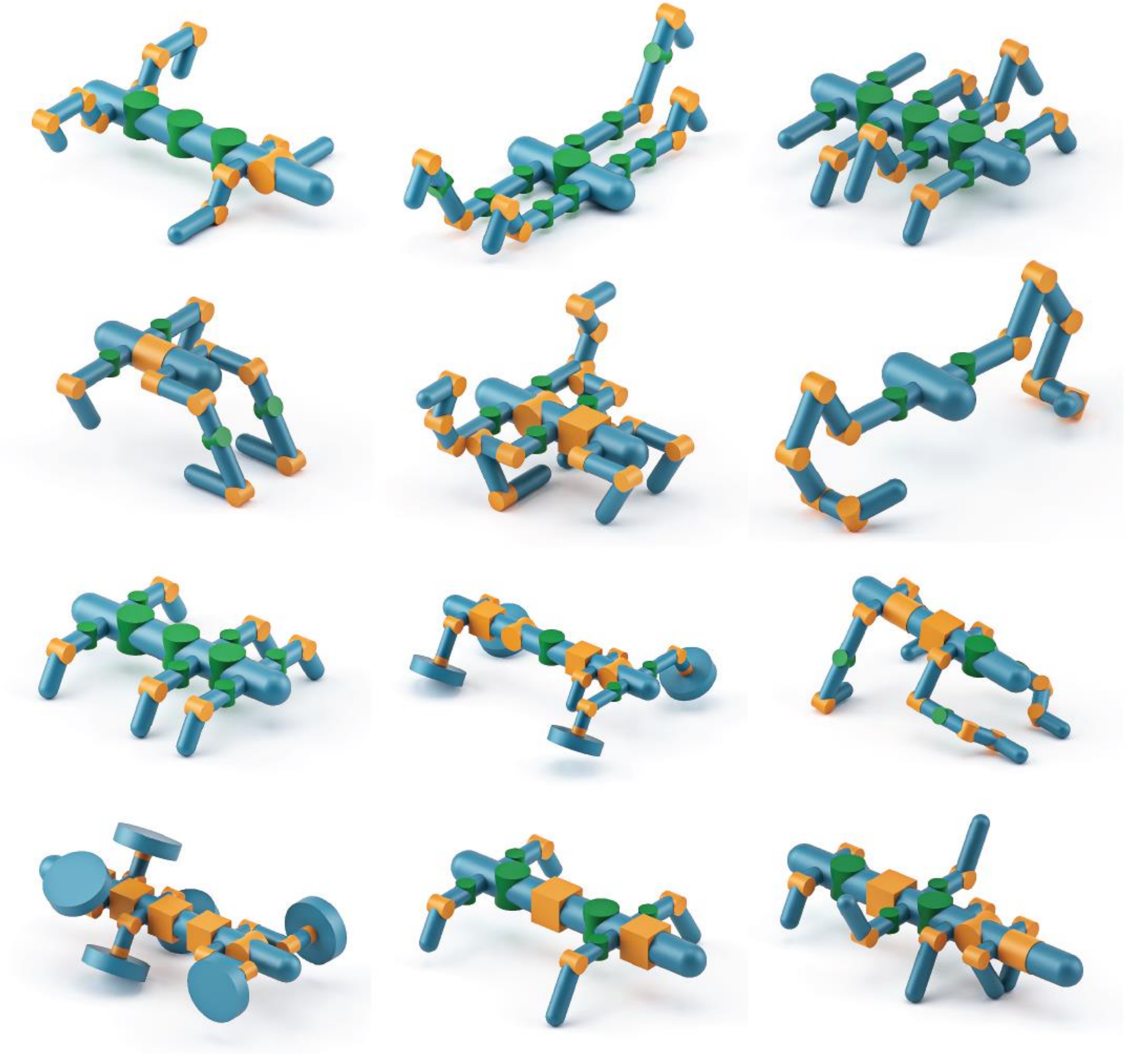


Zhao et. al, IROS, 2022

Cyber-physical Systems: Robots, Drones

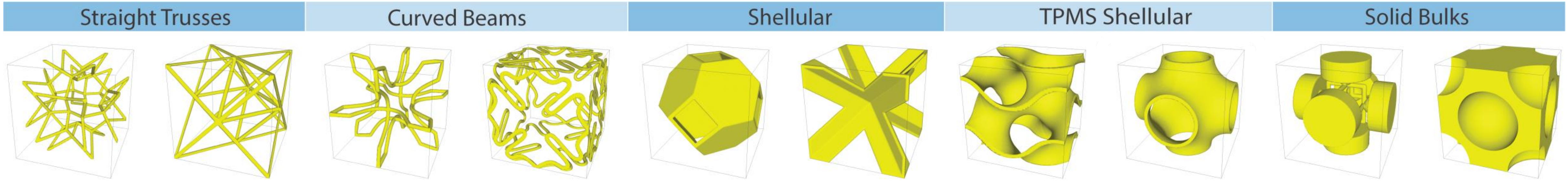


Cyber-physical Systems: Robots, Drones

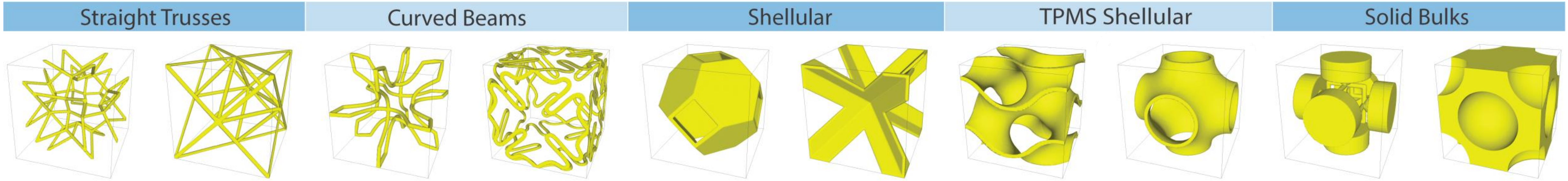


Zhao et. al, Siggraph, 2020

Architected Materials

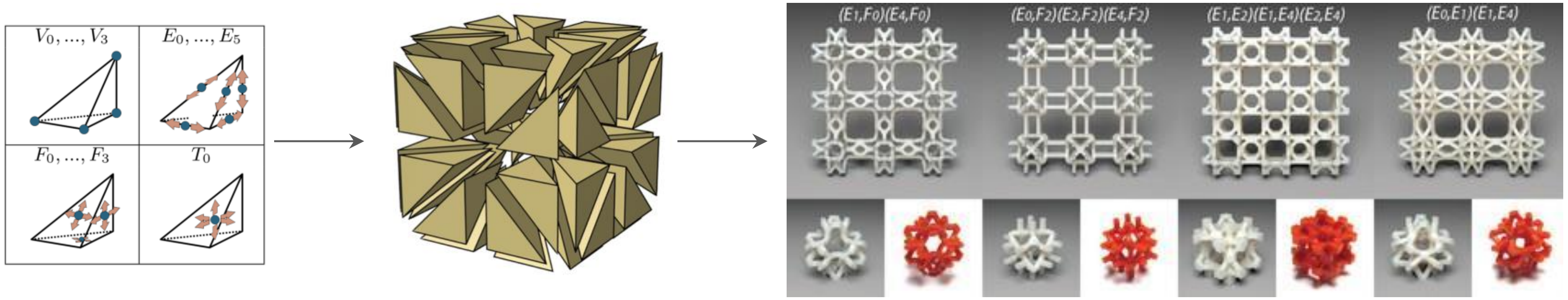


Architected Materials



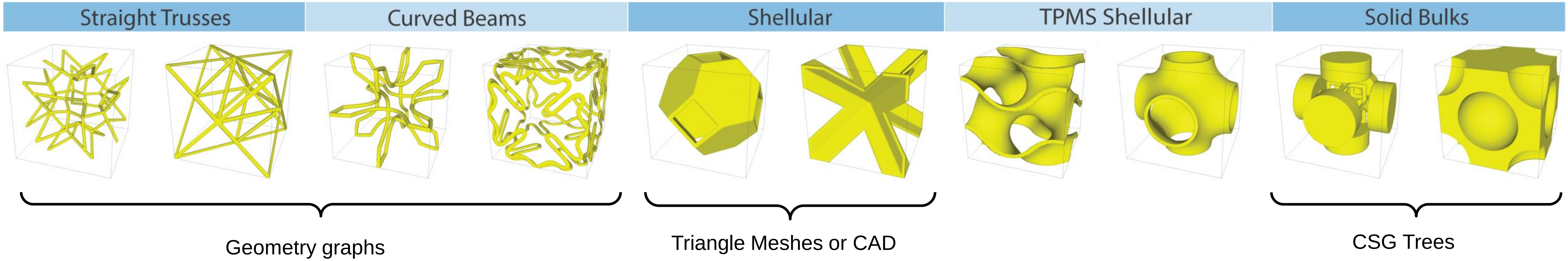
Geometry graphs

[Panetta et al. 2015]

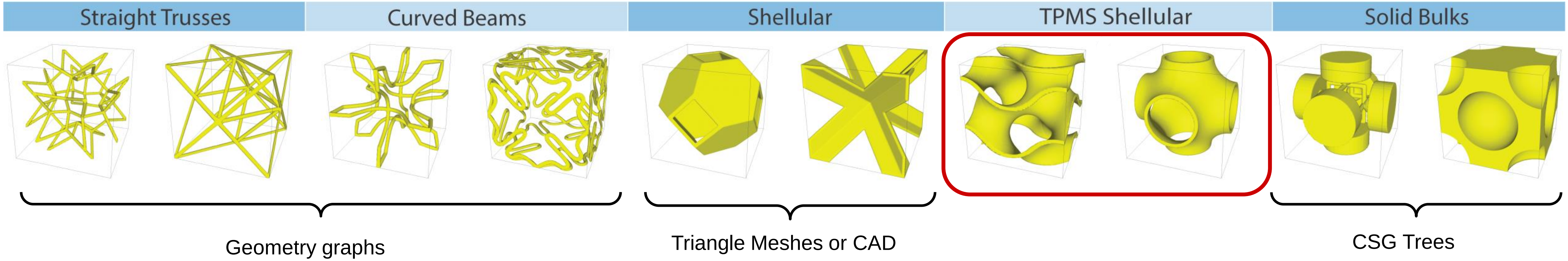


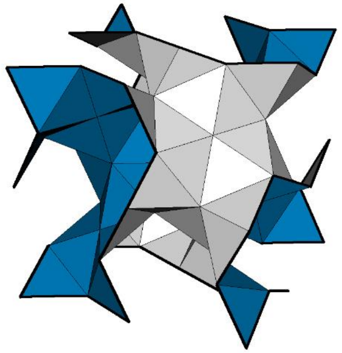
15 Nodes,
Choose 3 edges

Architected Materials

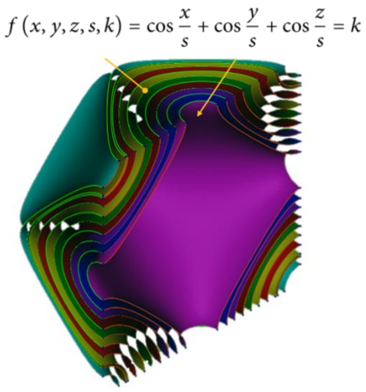


Architected Materials

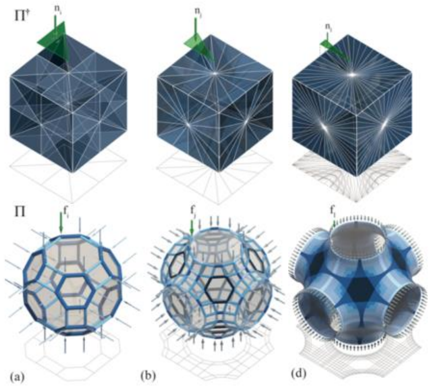




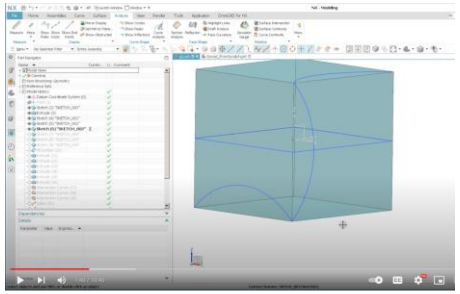
Discrete Approximations



Trigonometric Approximations

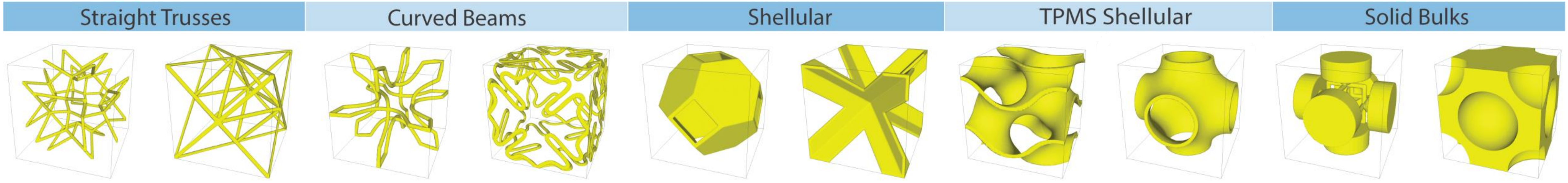


Static Graph Subdivision



Manual CAD Specification

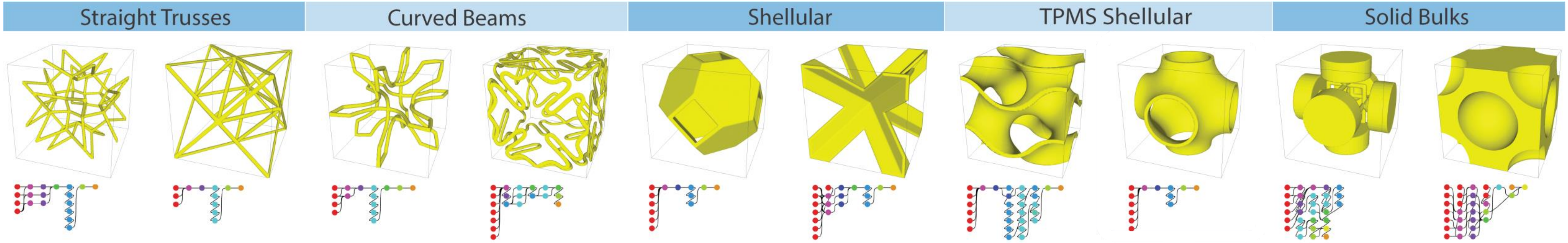
Architected Materials



Unified Representation

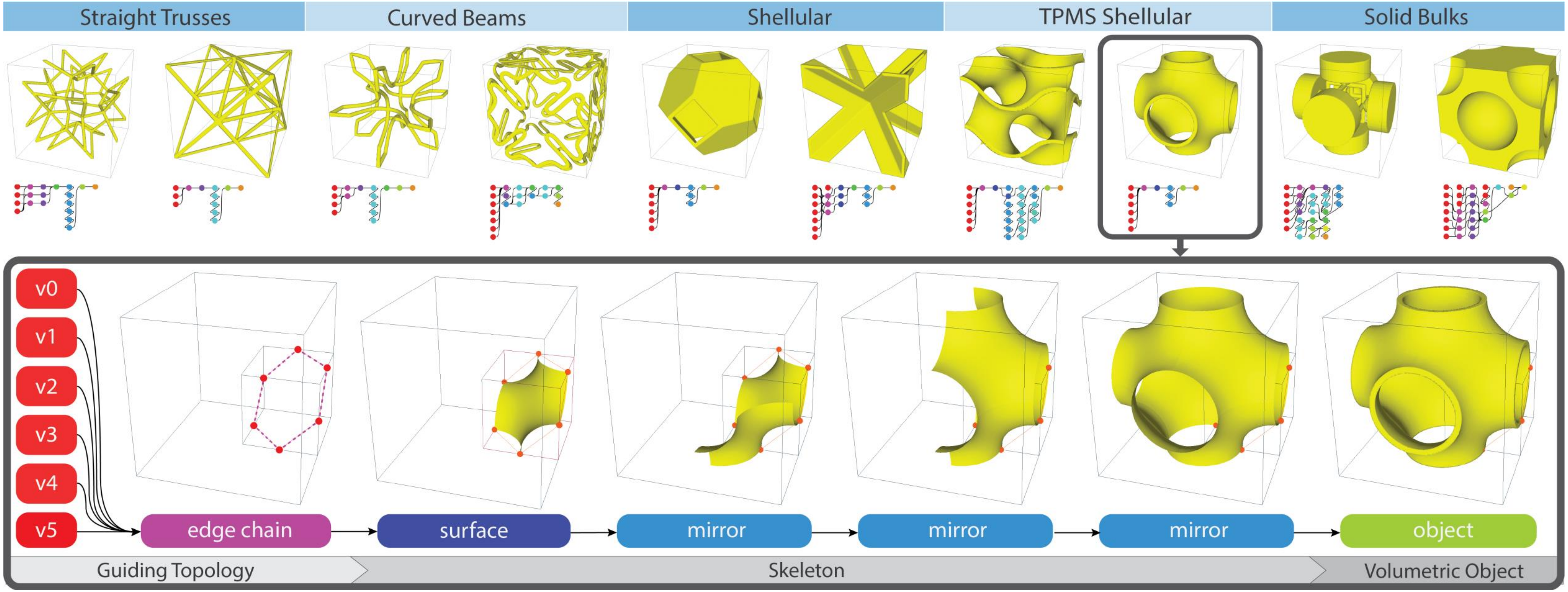
Makatura et. al, TOG, 2023

Architected Materials



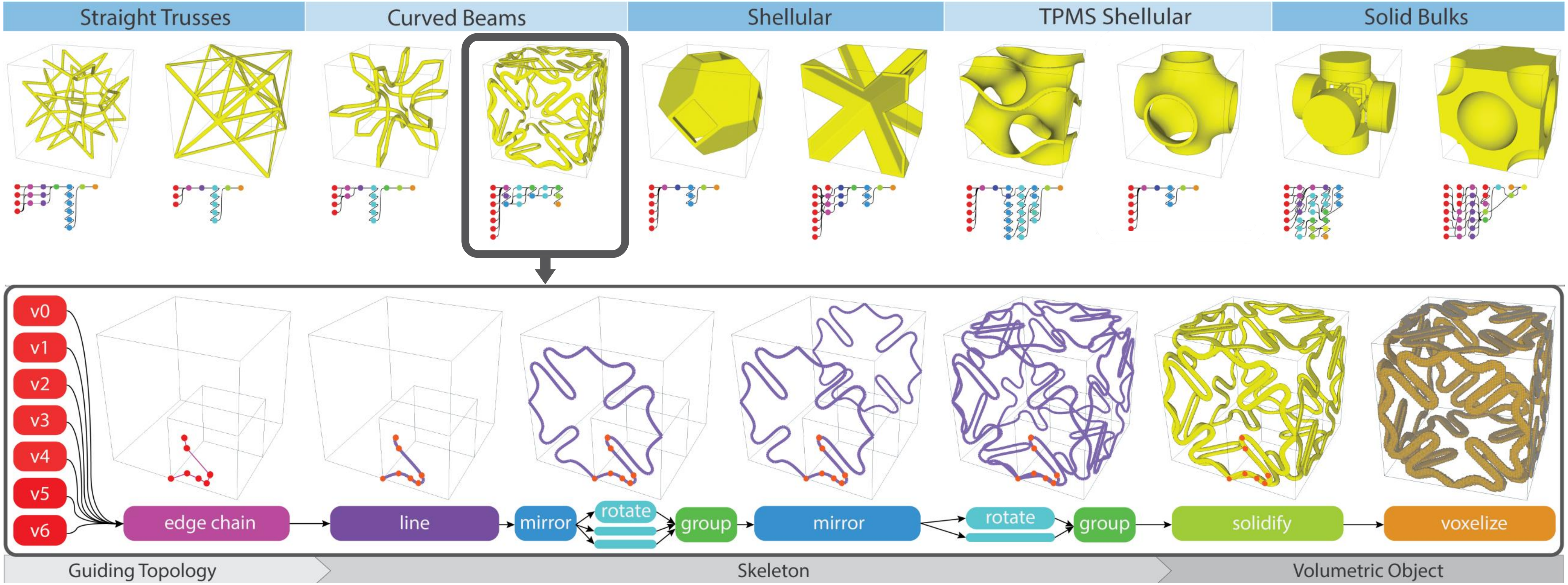
Makatura et. al, TOG, 2023

Architected Materials



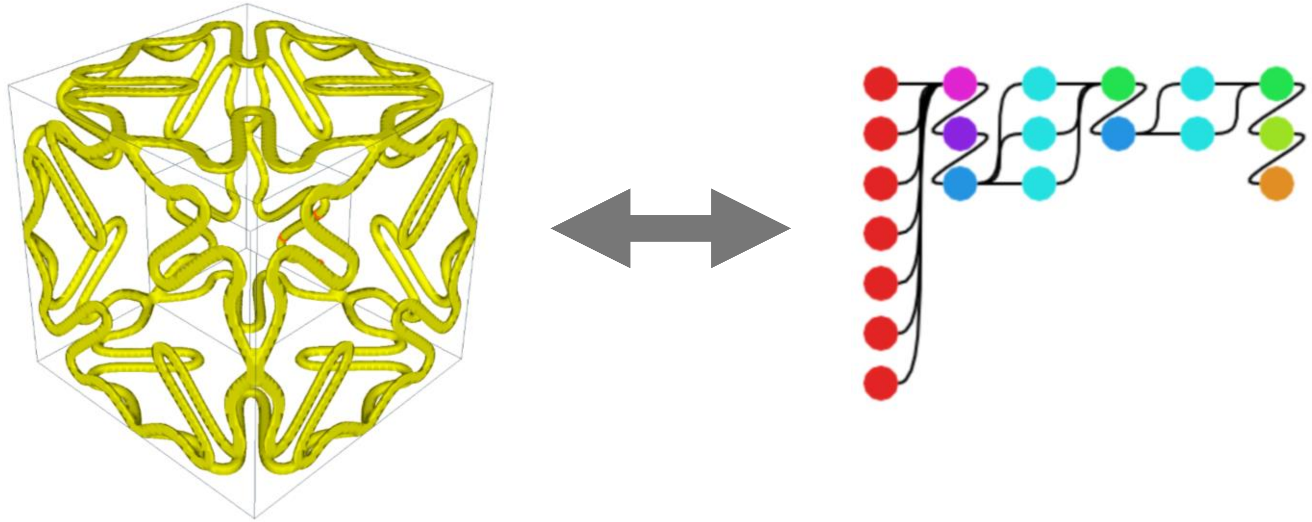
Makatura et. al, TOG, 2023

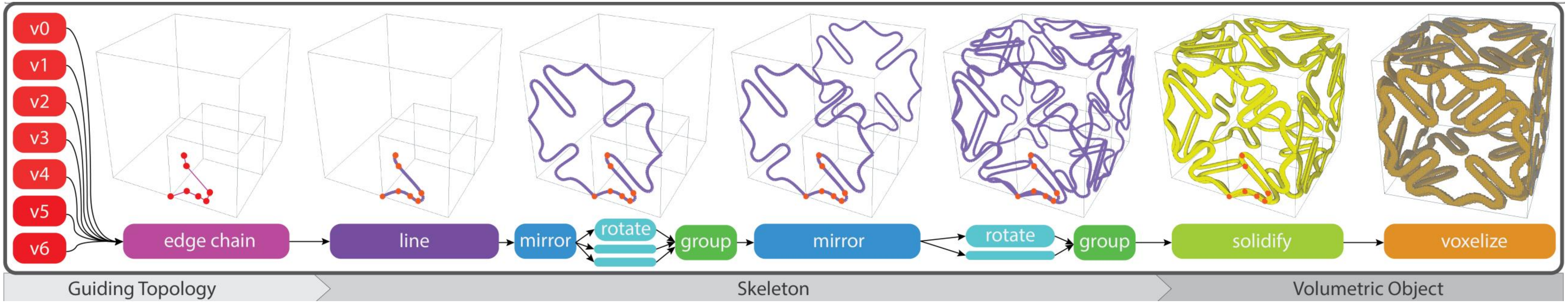
Architected Materials



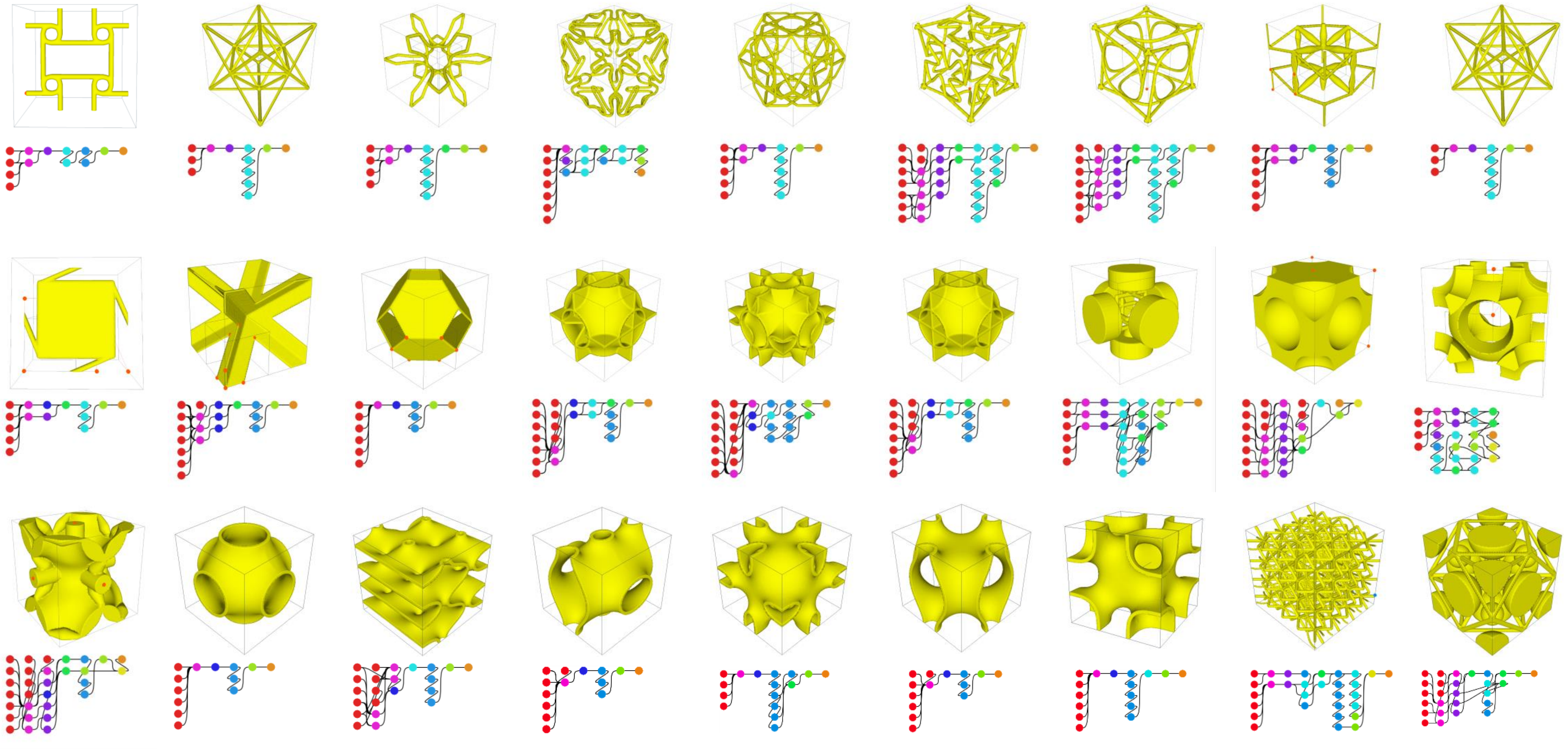
Makatura et. al, TOG, 2023

Architected Materials



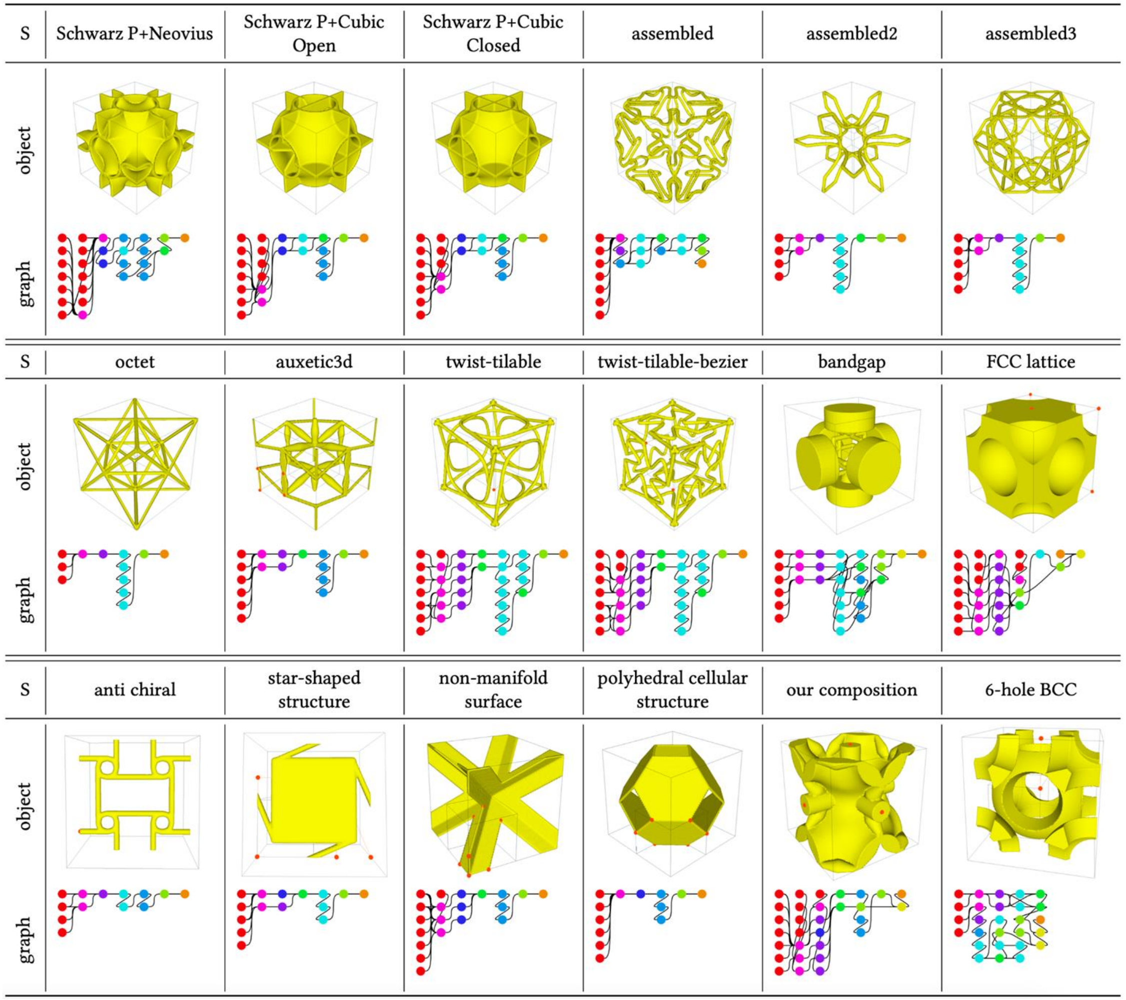


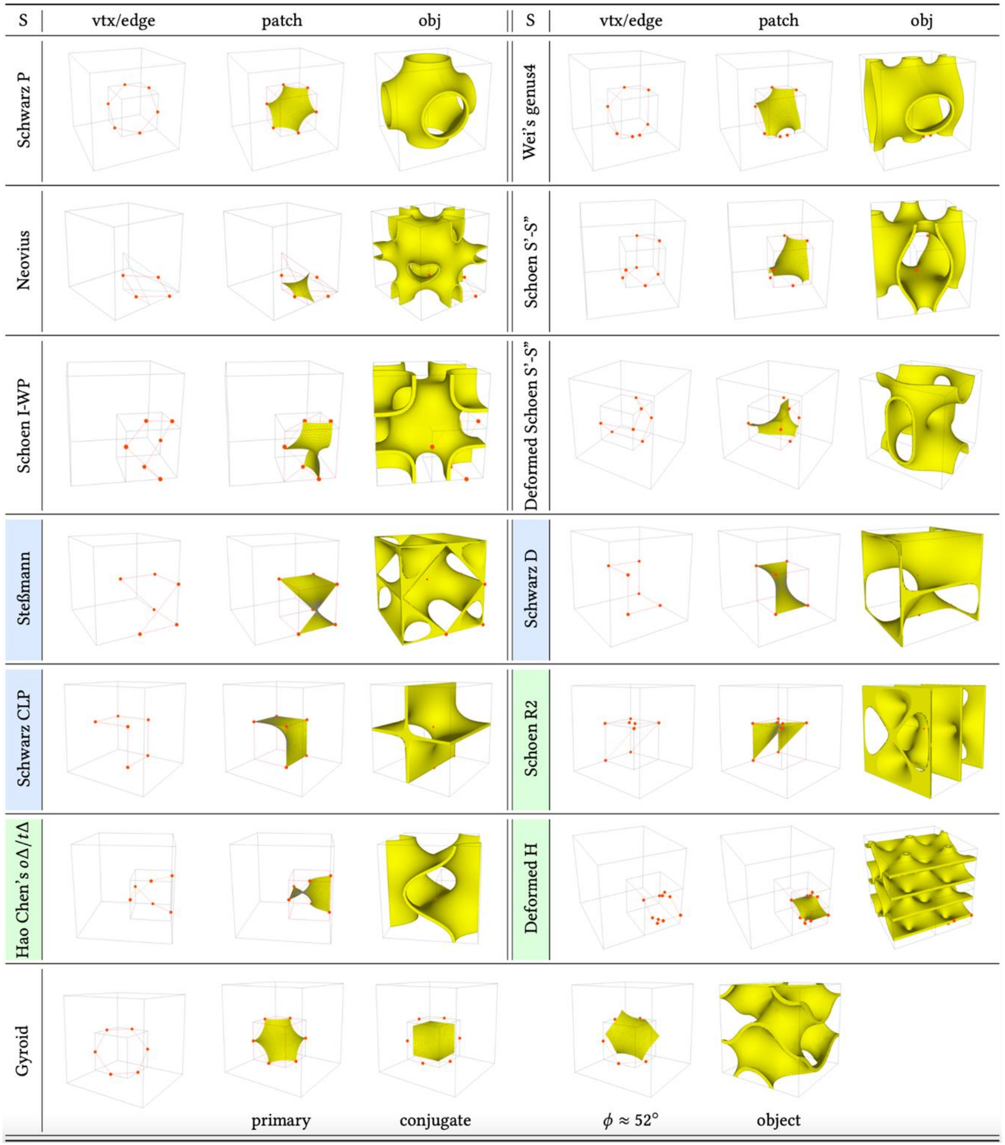
Representing Known Structures



Makatura et. al, TOG, 2023

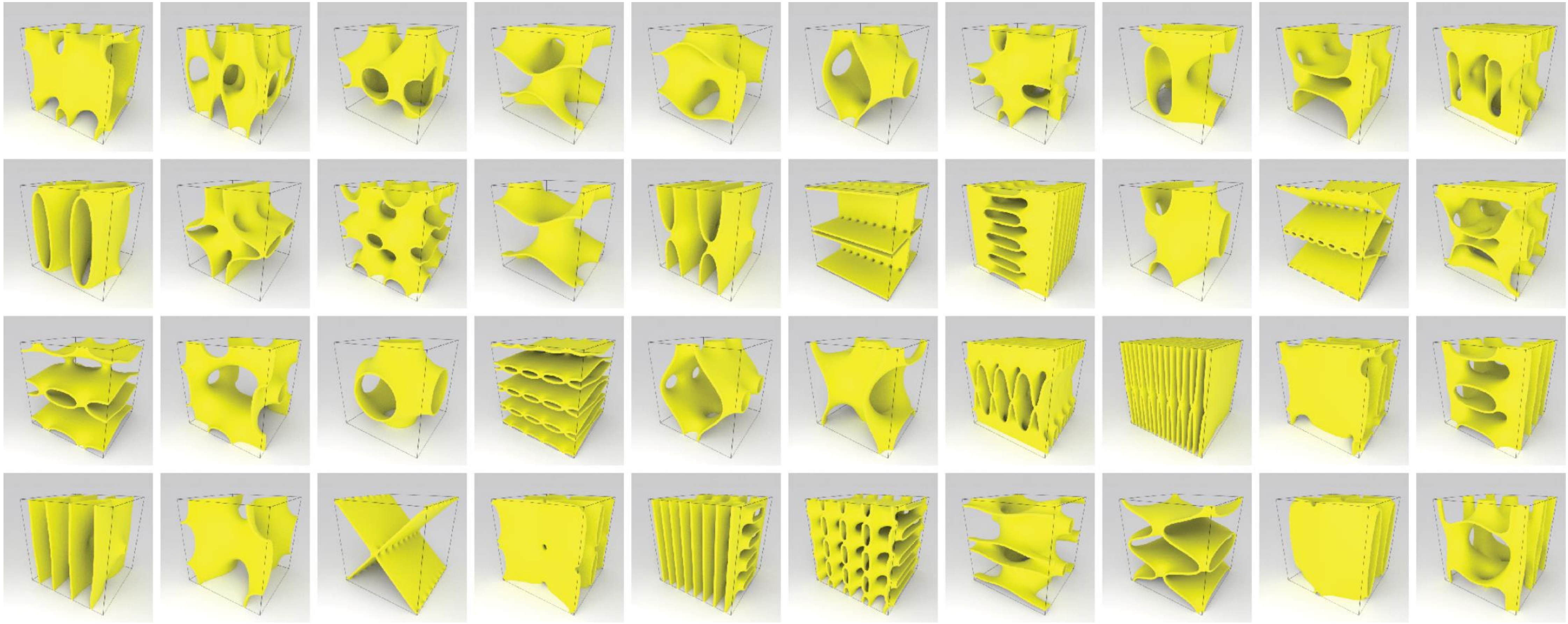
Established Structures





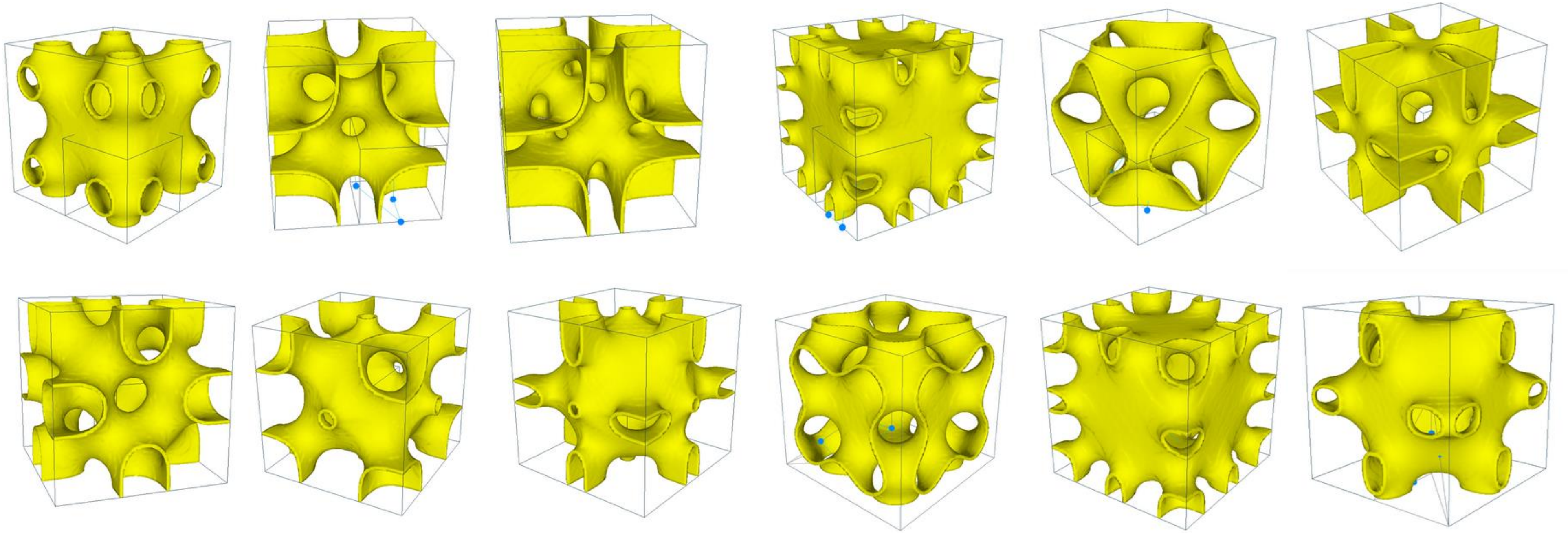
Makatura et. al, TOG, 2023

Randomly Generated Structures



Makatura et. al, TOG, 2023

Discovery of New TPMS



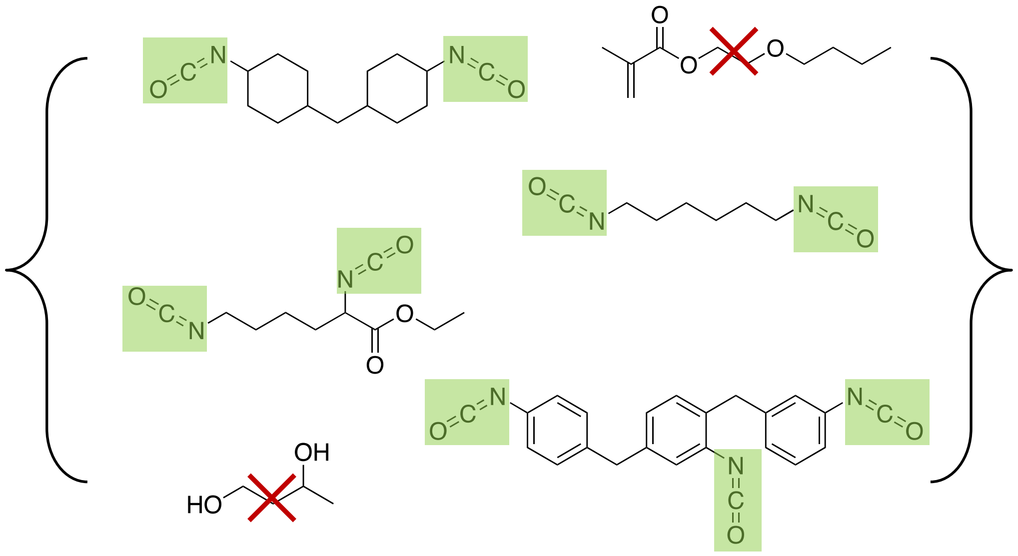
Hundreds of new TPMS structures

Key Questions for Computational Design

1. How to represent a design?
2. How to represent a design space?
3. How to learn a design space?
4. How to find designs with optimal performance?
5. How to bridge the gap between digital & real?

Small Experimental Datasets

- Existing dataset for polyurethanes: Only 20 samples   [Menon et al. 2019]

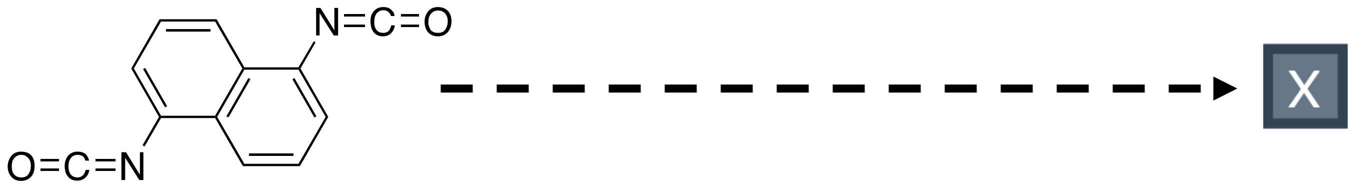


Isocyanates

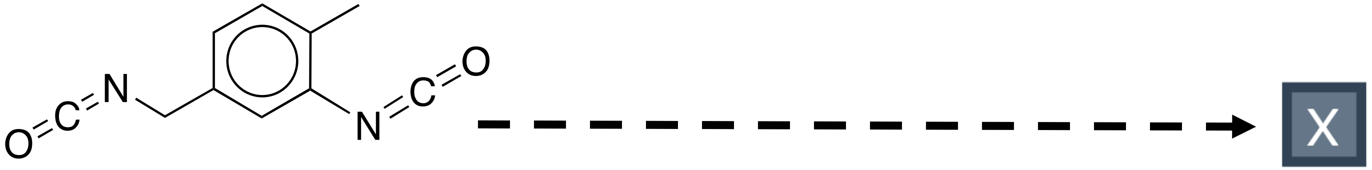
✕ Train/Finetune DL networks

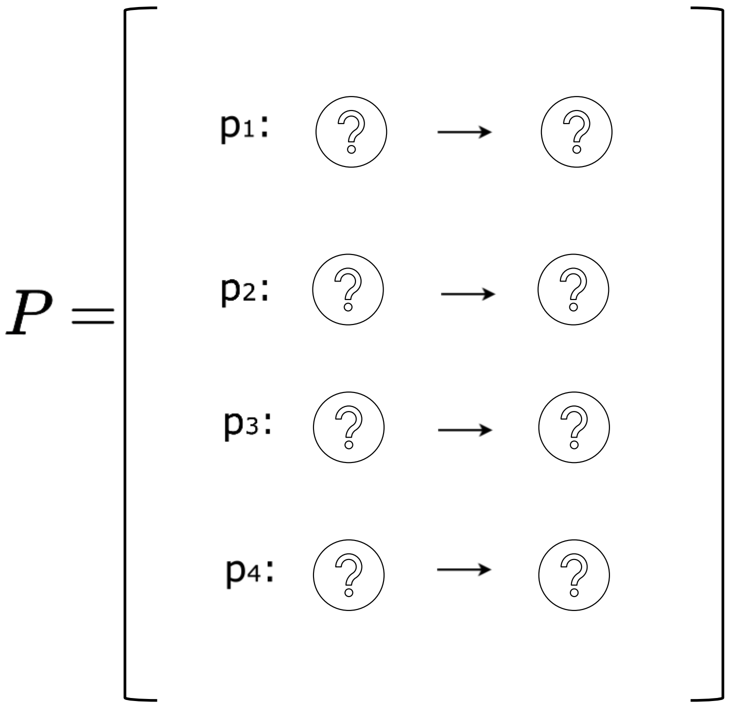
Learning graph grammars from examples

- We use bottom-up search to automatically generate the graph grammar



Progressively collapse edges of all input graphs till they become single nodes

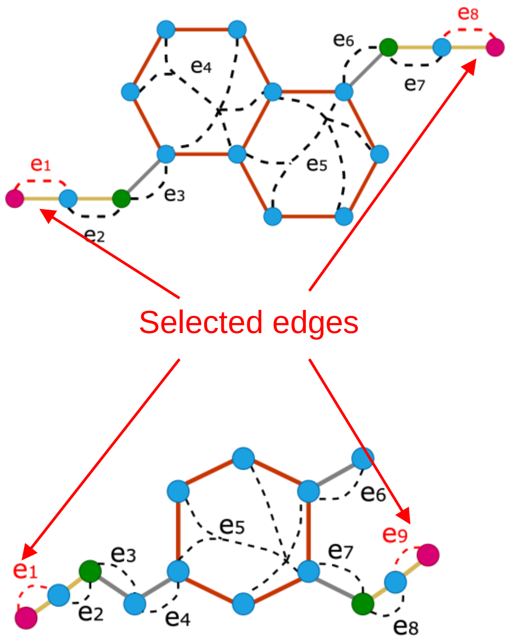




Guo et. al, ICLR, 2022

Learning graph grammars from examples

- We use bottom-up search to automatically generate the graph grammar

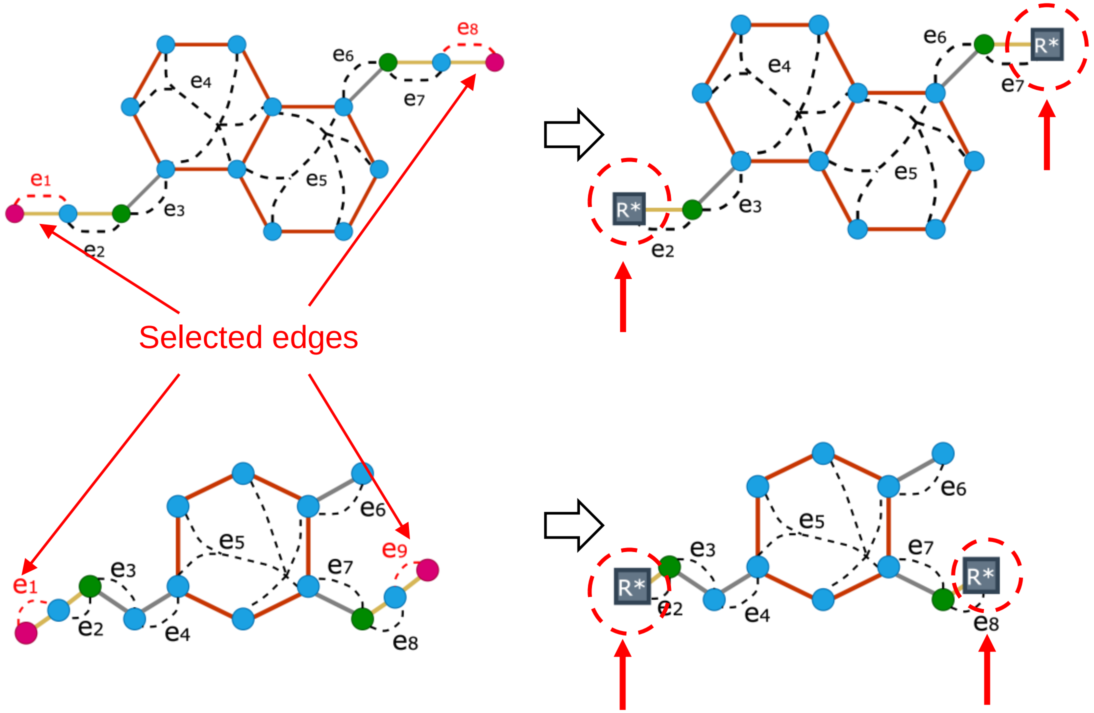


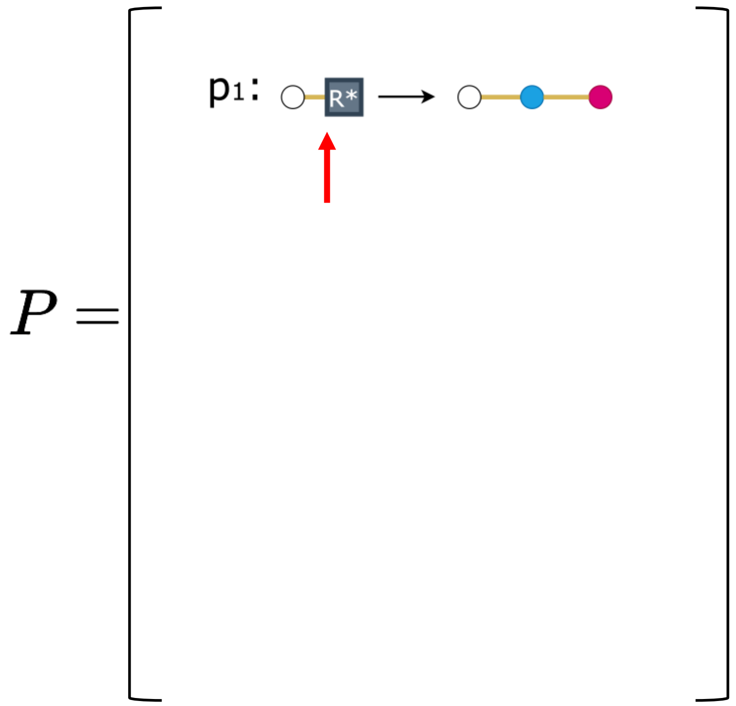
$$P = \begin{bmatrix} \\ \\ \\ \end{bmatrix}$$

Guo et. al, ICLR, 2022

Learning graph grammars from examples

- We use bottom-up search to automatically generate the graph grammar

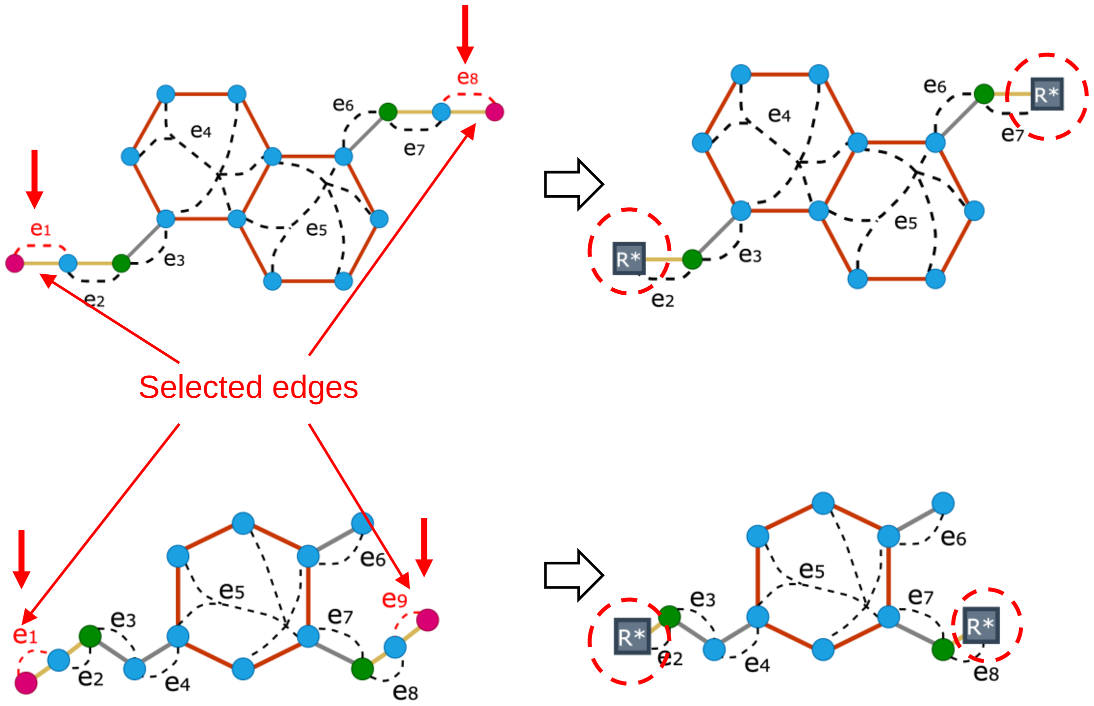


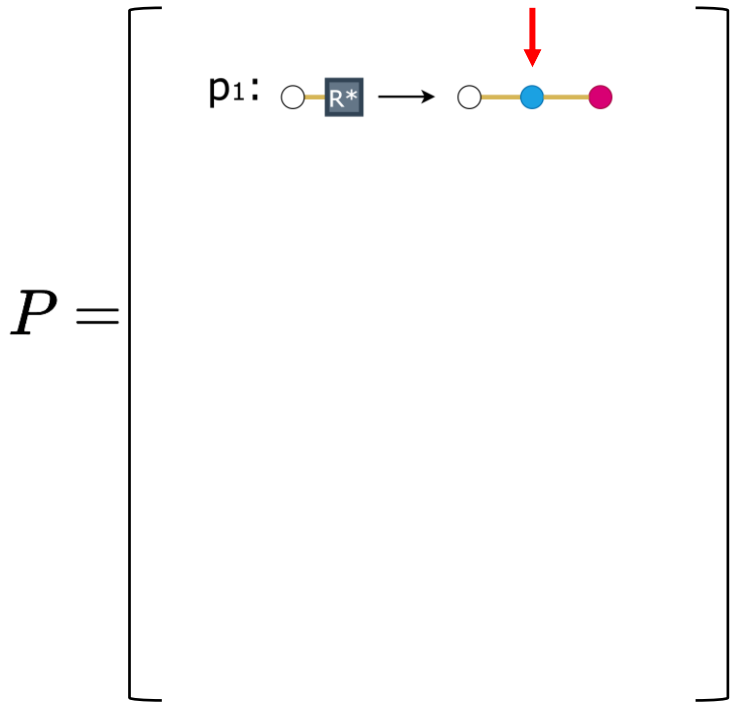


Guo et. al, ICLR, 2022

Learning graph grammars from examples

- We use bottom-up search to automatically generate the graph grammar

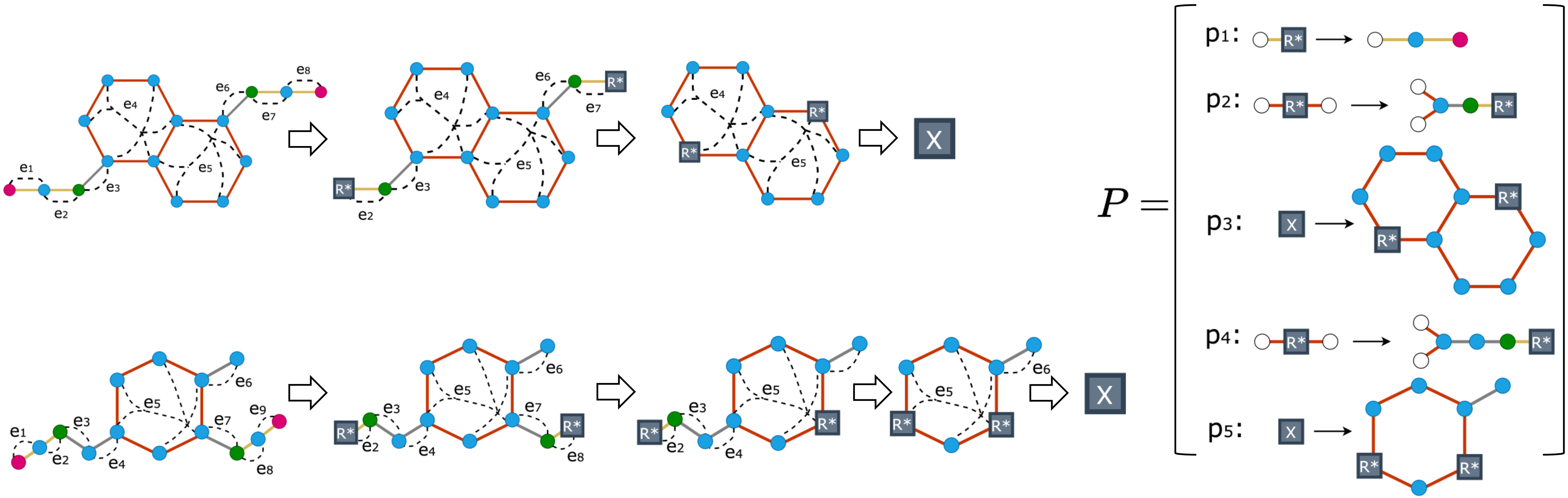




Guo et. al, ICLR, 2022

Learning graph grammars from examples

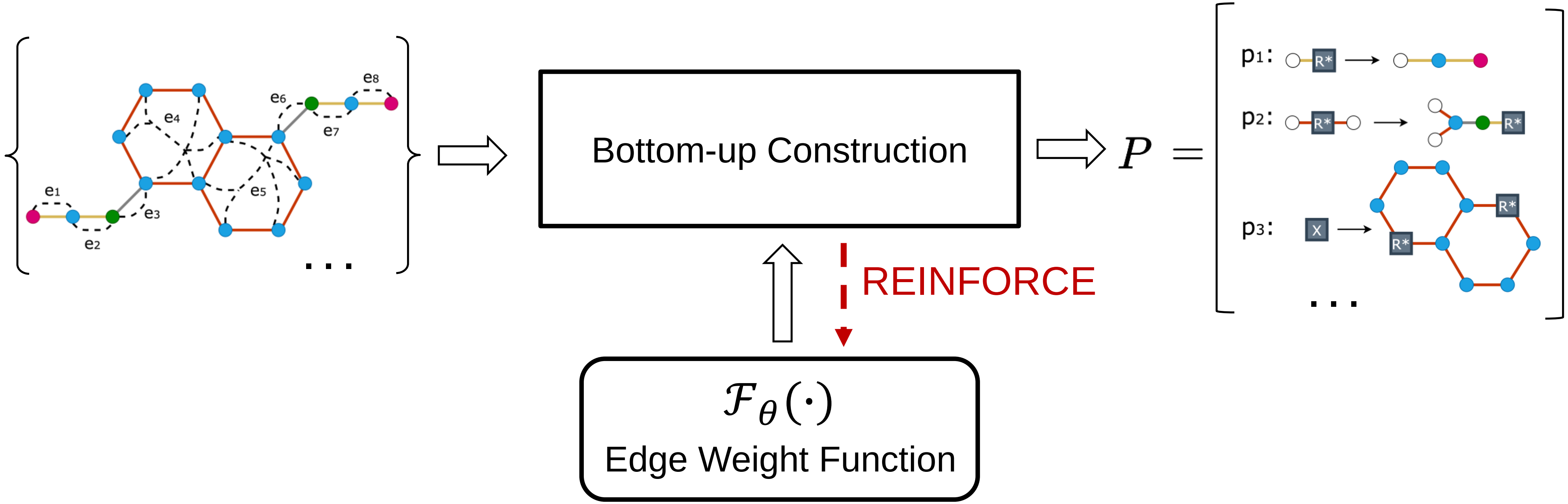
- We use bottom-up search to automatically generate the graph grammar



Guo et. al, ICLR, 2022

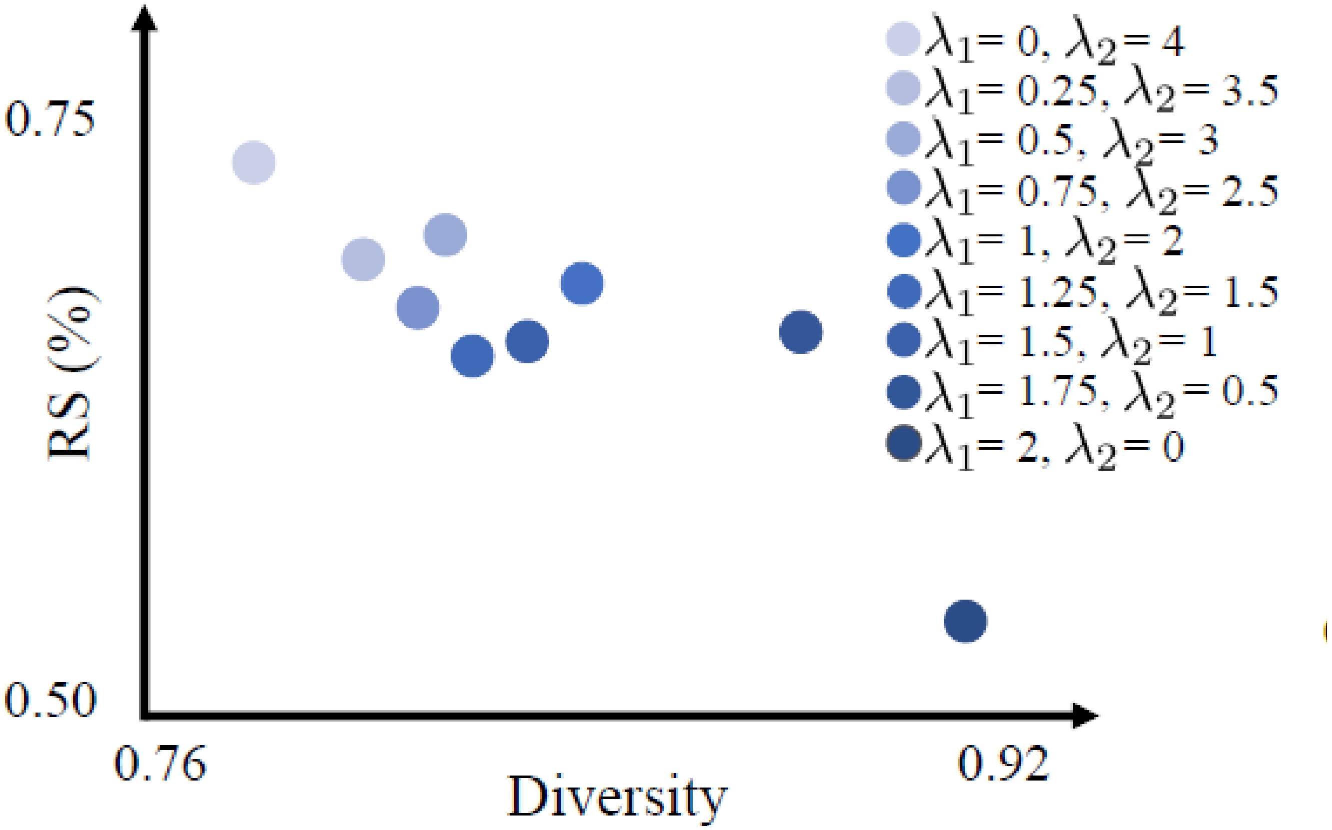
Learning graph grammar as inference

- $\max_\theta(diversity(G(\mathcal{F}_\theta(e)) + \lambda\, validity(G(\mathcal{F}_\theta(e)))$



Guo et. al, ICLR, 2022

Trade-off between diversity vs. validity



Guo et. al, ICLR, 2022

Results on Class-specific Polymer Data

Deep learning-based methods

Grammar-based methods

Method	Valid	Unique	Div.	Chamfer	RS	Memb.
Train data	100%	100%	0.61	0.00	100%	100%
GraphNVP	<u>0.16%</u>	---	---	---	0.00%	0.00%
JT-VAE	**100%**	5.8%	0.72	0.85	5.50%	66.5%
HierVAE	**100%**	<u>99.6%</u>	0.83	0.76	1.85%	0.05%
MHG	**100%**	75.9%	**0.88**	0.83	2.97%	12.1%
STONED	**100%**	**100%**	0.85	<u>0.86</u>	<u>5.63%</u>	<u>79.8%</u>
DEG	**100%**	**100%**	<u>0.86</u>	**0.87**	**27.2%**	**96.3%**

Guo et. al, ICLR, 2022

Results on Class-specific Polymer Data

Percentage of molecules belonging to the concerned class

Method	Valid	Unique	Div.	Chamfer	RS	Memb.
Train data	100%	100%	0.61	0.00	100%	100%
GraphNVP	<u>0.16%</u>	---	---	---	0.00%	0.00%
JT-VAE	**100%**	5.8%	0.72	0.85	5.50%	66.5%
HierVAE	**100%**	<u>99.6%</u>	0.83	0.76	1.85%	0.05%
MHG	**100%**	75.9%	**0.88**	0.83	2.97%	12.1%
STONED	**100%**	**100%**	0.85	<u>0.86</u>	<u>5.63%</u>	<u>79.8%</u>
DEG	**100%**	**100%**	<u>0.86</u>	**0.87**	**27.2%**	**96.3%**

Deep learning-based methods

Grammar-based methods

Percentage of **synthesizable** molecules

Guo et. al, ICLR, 2022

Results on Large Polymer Dataset

Method	Distribution Statistics (↓)				Sample Quality (↑)			
	logP	**SA**	**QED**	**MW**	**Valid**	**Unique**	**Div.**	**Chamfer**
Train data	0.12	0.02	0.002	2.98	100%	100%	0.83	0.00
SMILESVAE	9.63	2.99	0.19	751.6	0.01%	---	---	---
GraphNVP	2.94	0.65	0.03	435.6	0.23%	---	---	---
JT-VAE	2.93	0.32	0.10	210.1	**100%**	83.9%	<u>0.88</u>	0.50
HierVAE	**0.50**	**0.08**	**0.02**	**42.45**	**100%**	<u>99.9%</u>	0.82	0.32
MHG	9.20	1.91	0.10	380.3	**100%**	**100%**	**0.91**	0.56
STONED	2.43	0.81	0.07	179.9	99.9%	**100%**	0.83	0.45
DEG (0.15%, fitting)	<u>1.80</u>	0.25	**0.02**	<u>69.0</u>	**100%**	**100%**	0.82	0.60
DEG (0.15%)	5.52	0.51	0.20	334.2	**100%**	**100%**	0.86	<u>0.62</u>

Deep learning-based methods

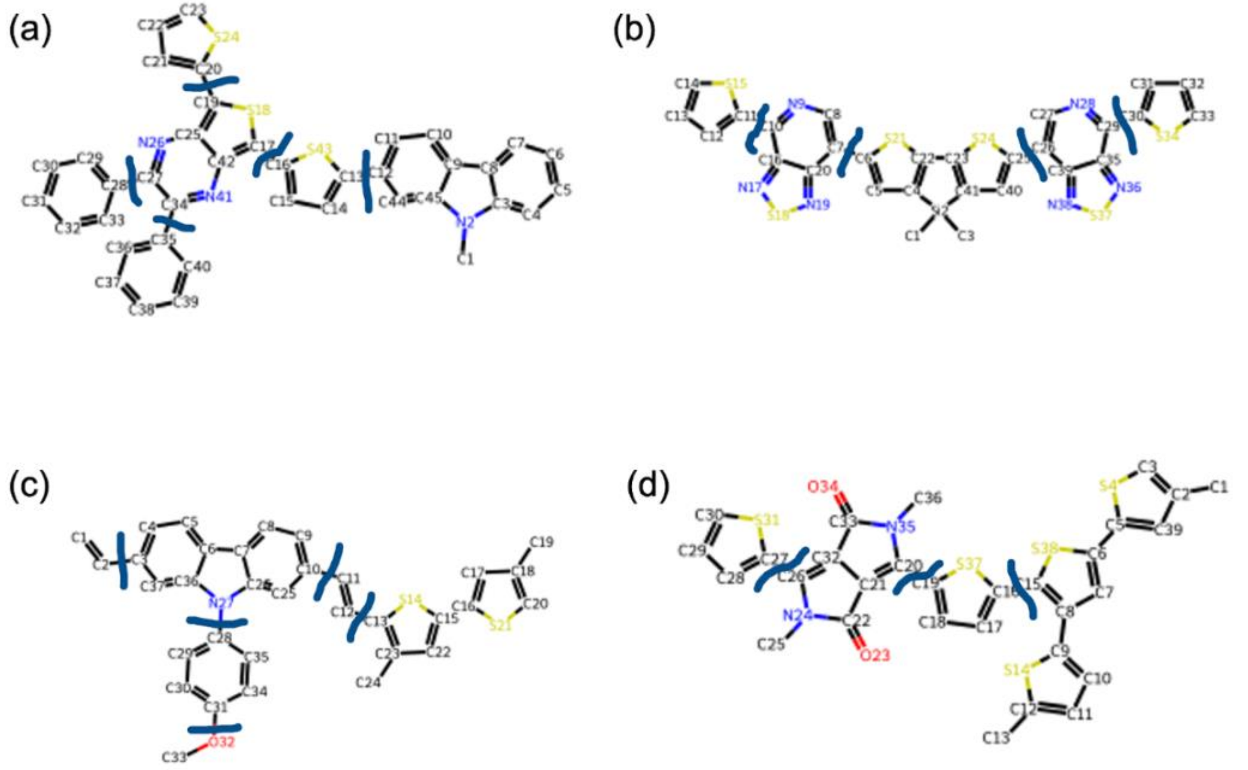
Grammar-based methods

Only trained on **117** samples of original 81k dataset

Guo et. al, ICLR, 2022

Key Insights

- Symbolic representations
- Automatic checkers/oracles
  - For example, retrosynthesis
- Expert annotations



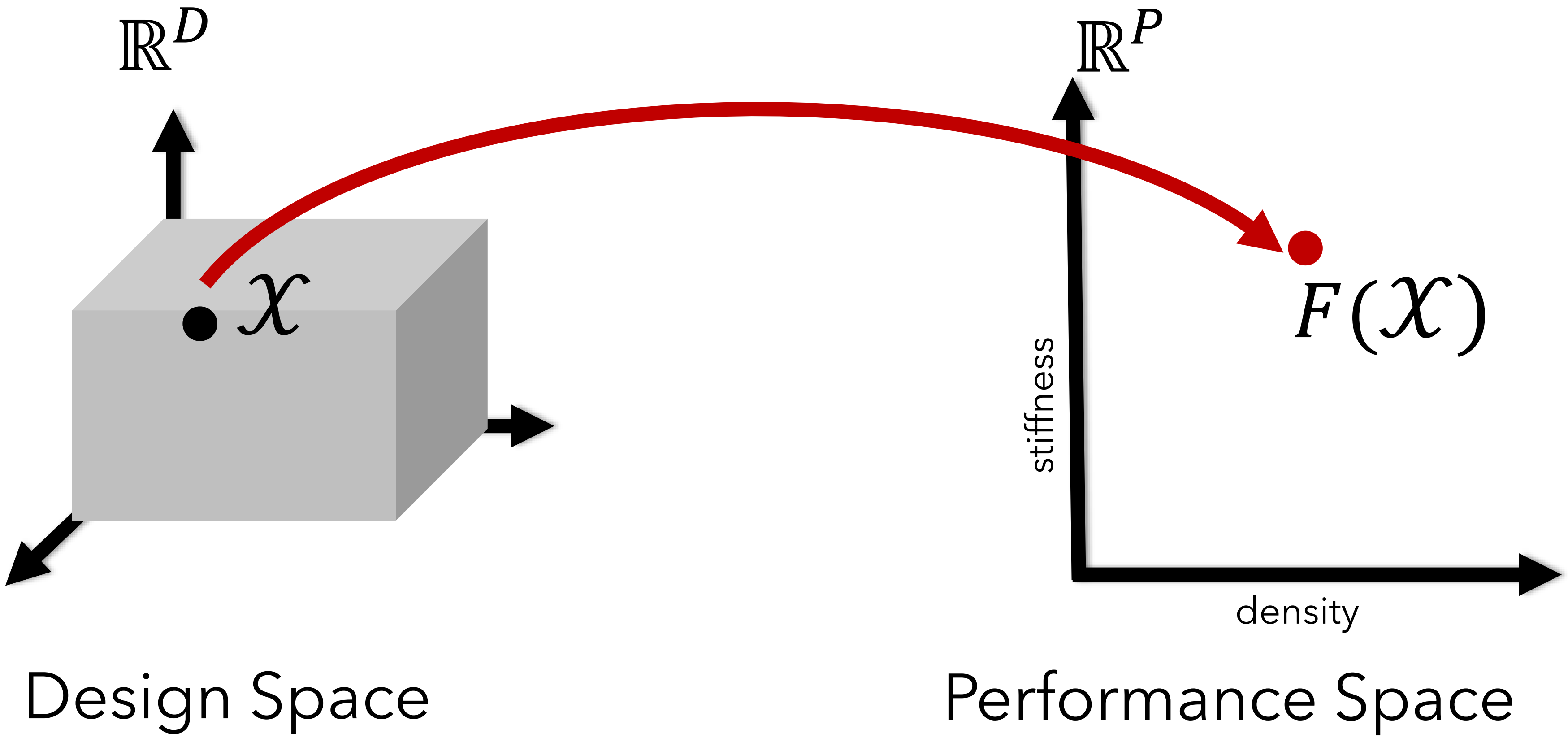
Sun et. al, ICML, 2024

Key Questions for Computational Design

1. How to represent a design?
2. How to represent a design space?
3. How to learn a design space?
4. How to find designs with optimal performance?
5. How to bridge the gap between digital & real?

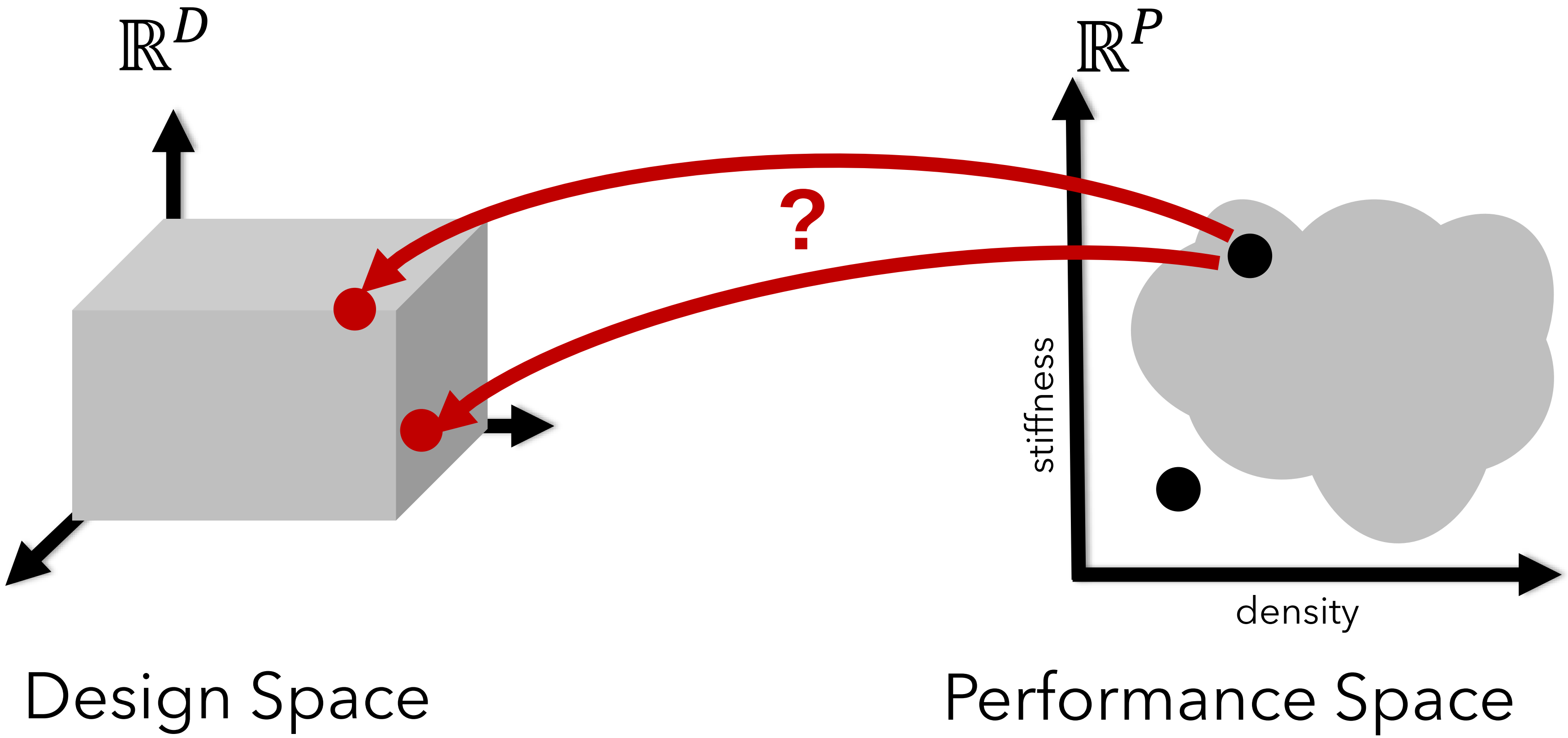
From Design To Performance

- Numerical simulations (or real experiments) map a point in design space to a point in performance space



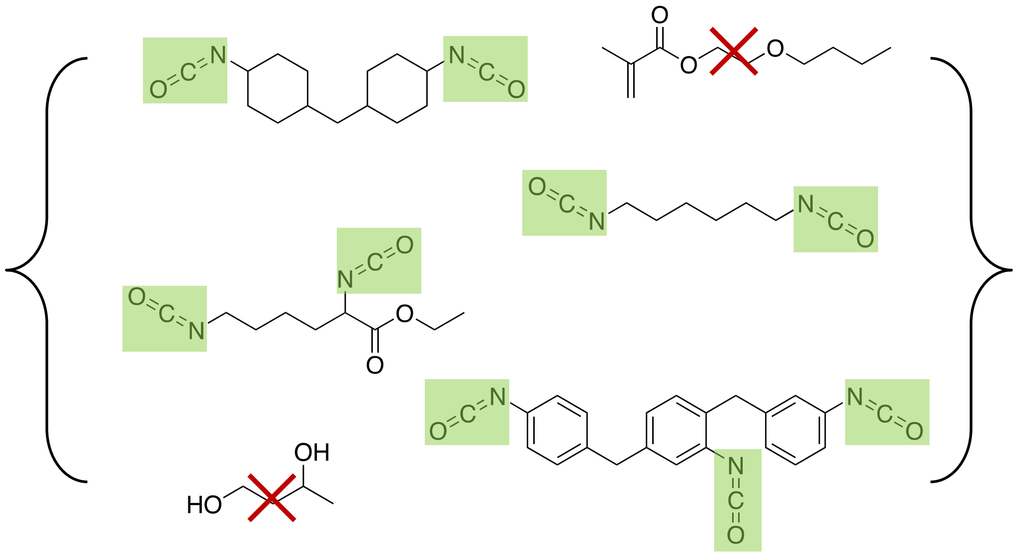
Inverse Design

- Inverse problem is much more difficult



Case 1: Small Experimental Datasets

- We would like to find a molecule with desired material properties



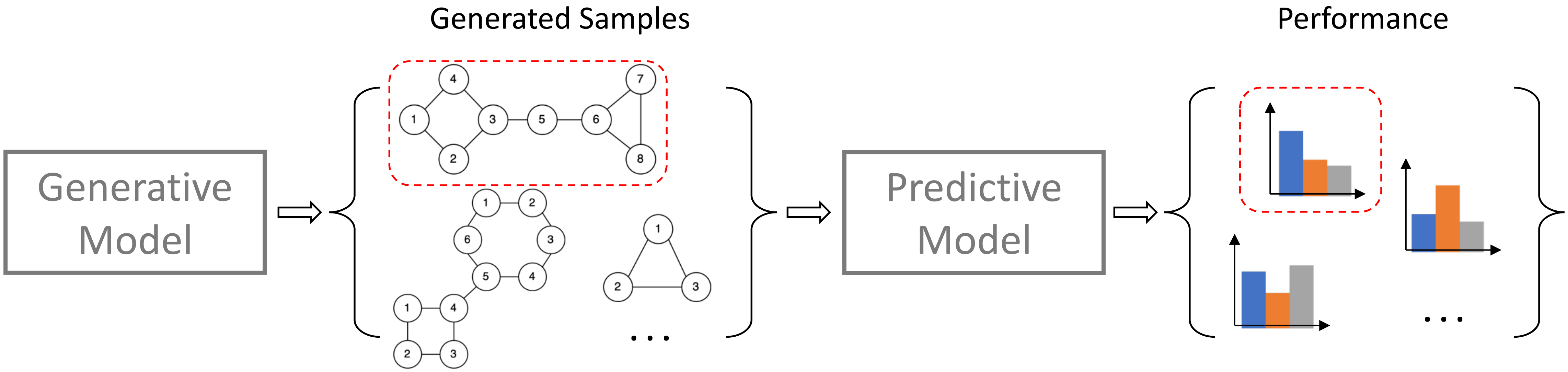
Isocyanates

- Existing dataset for polyurethanes:

  Only 20 samples [Menon et al. 2019]

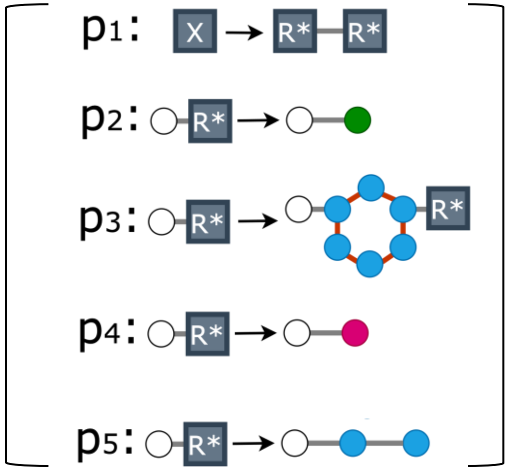
✗ Train/Finetune DL networks

Finding new molecules & their properties

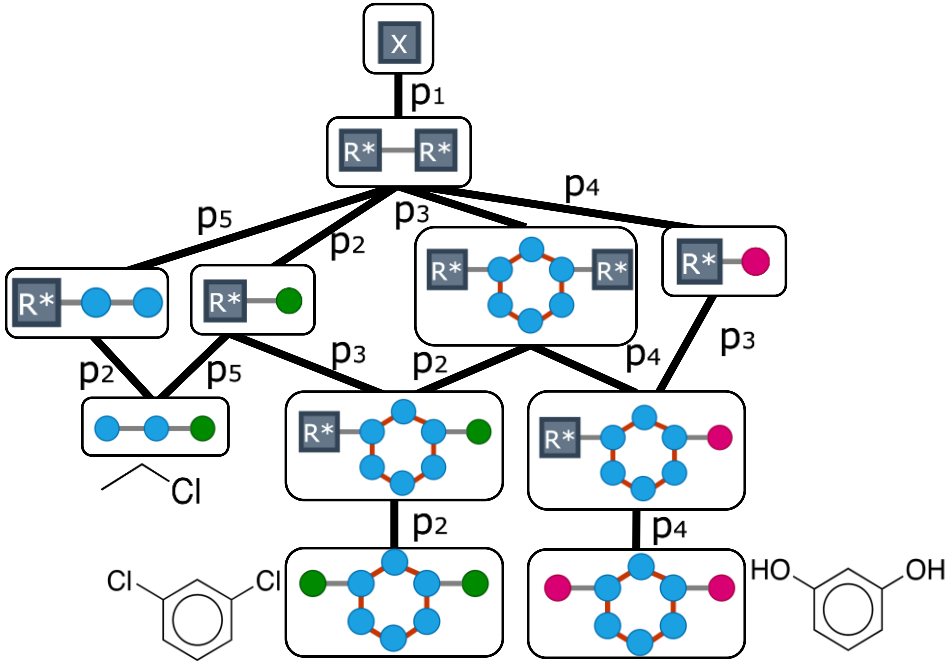


Guo et. al, ICML, 2023

Grammar Induces Manifold Geometry



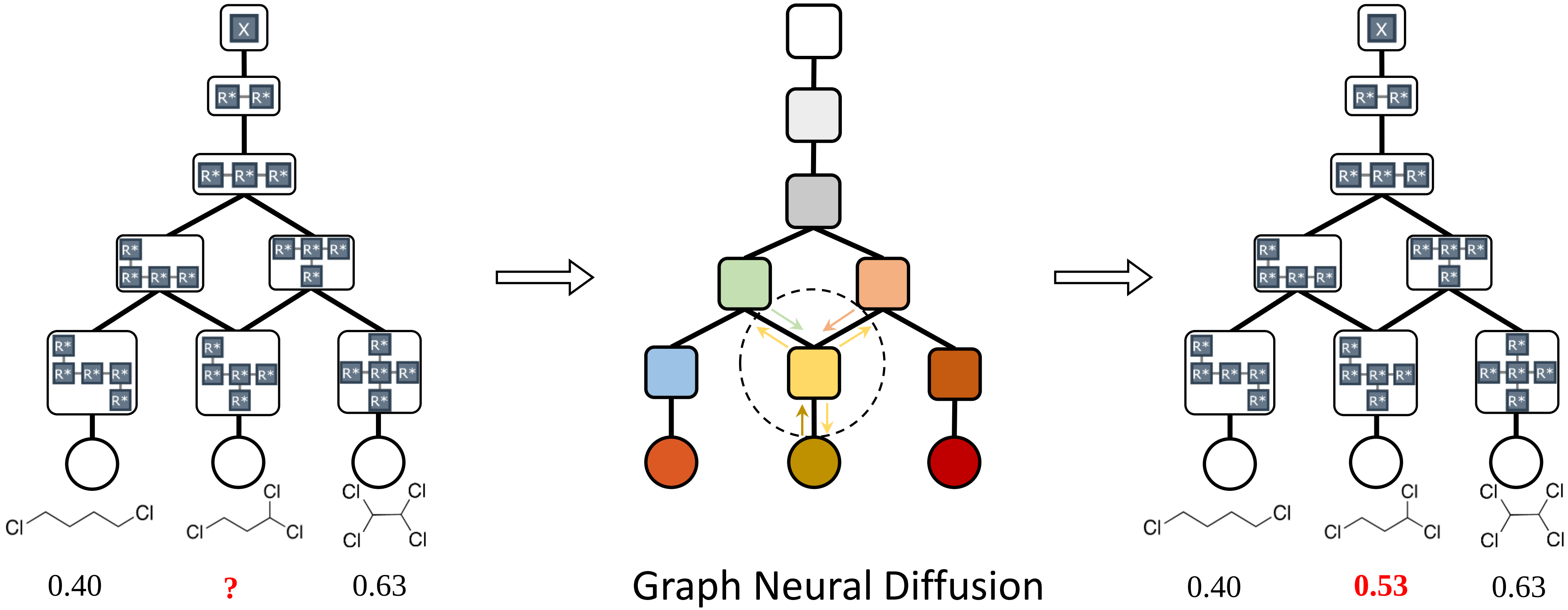
Graph Grammar



Grammar-induced Geometry

Guo et. al, ICML, 2023

Property Predictor



Graph Neural Diffusion

[Chamberlain et al. 2021]

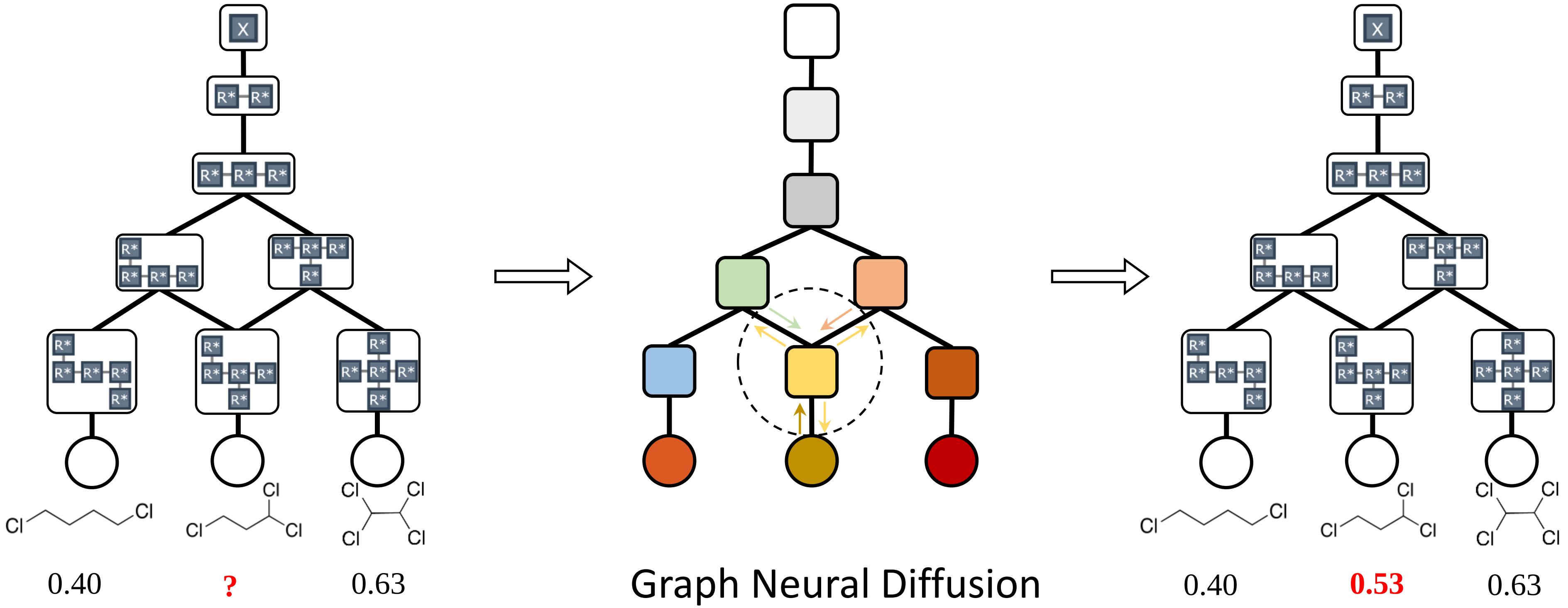
Guo et. al, ICML, 2023

Joint optimization of grammar & predictor

Graph neural diffusion

$$\min_{\theta,(\varphi,\psi,\alpha)} l(\mathbf{u}_T, \hat{\mathbf{u}}) = \min_{\theta} \min_{(\varphi,\psi,\alpha)} l(\mathbf{u}_T, \hat{\mathbf{u}})$$

Molecular grammar learning

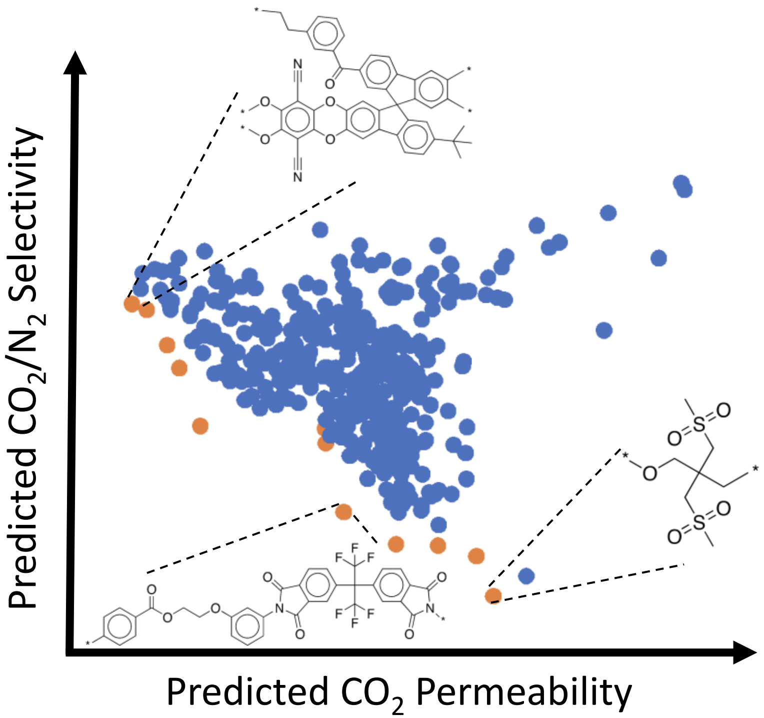


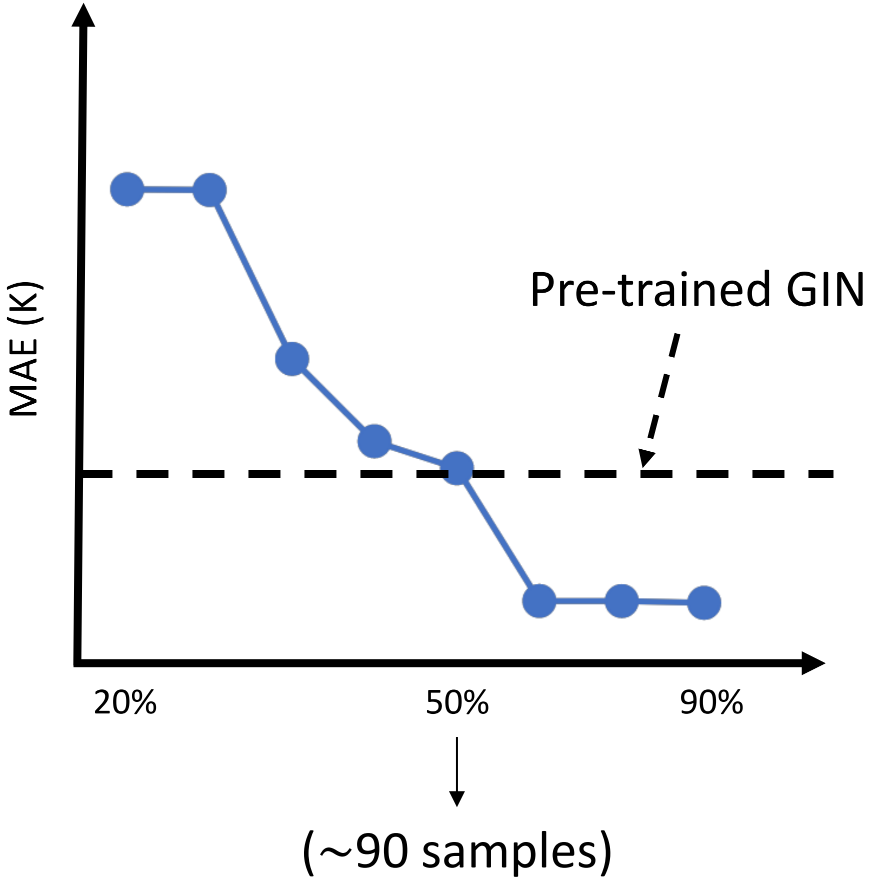
Graph Neural Diffusion

[Chamberlain et al. 2021]

Guo et. al, ICML, 2023

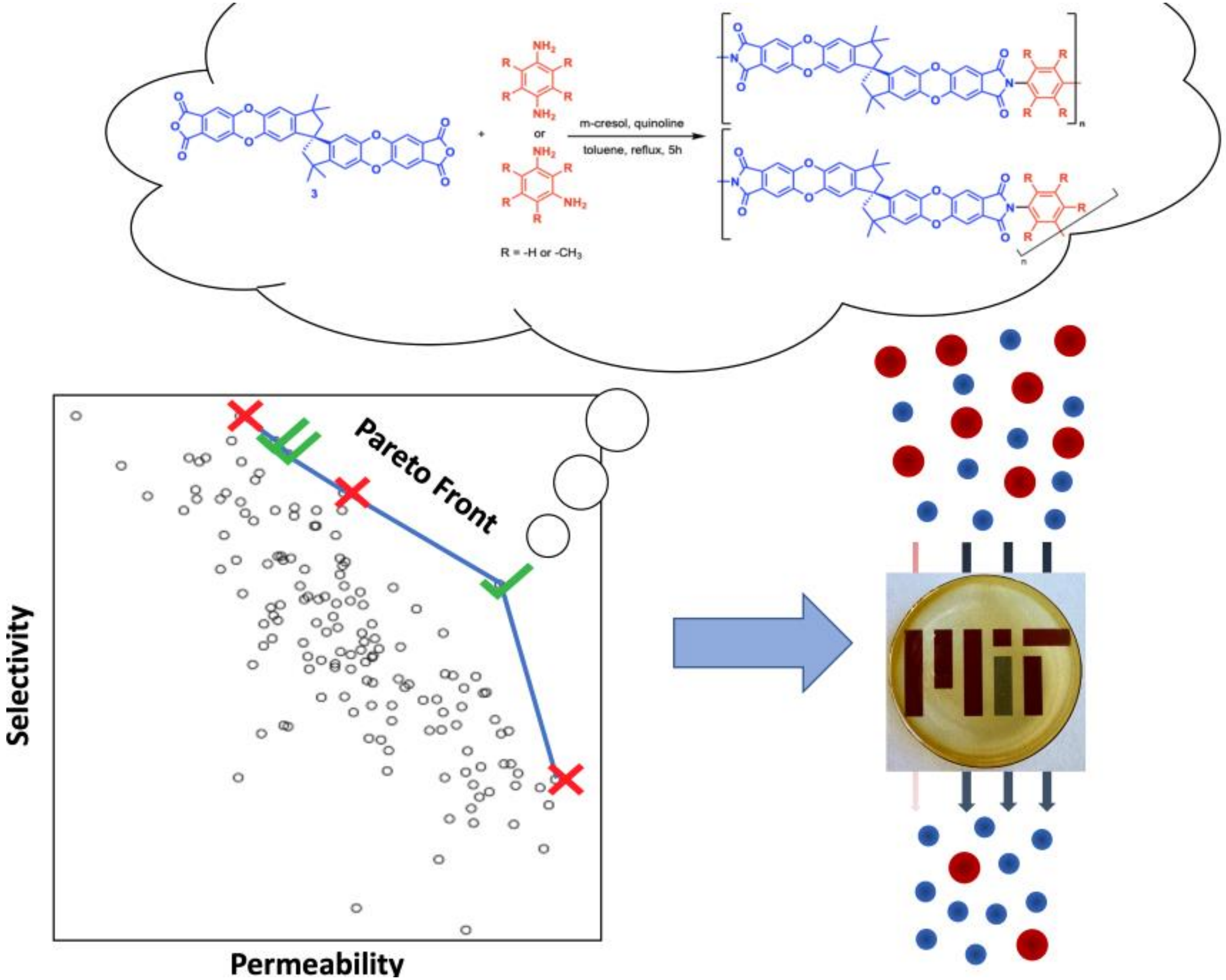
Property Prediction Results





Guo et. al, ICML, 2023

Design of Novel Molecules

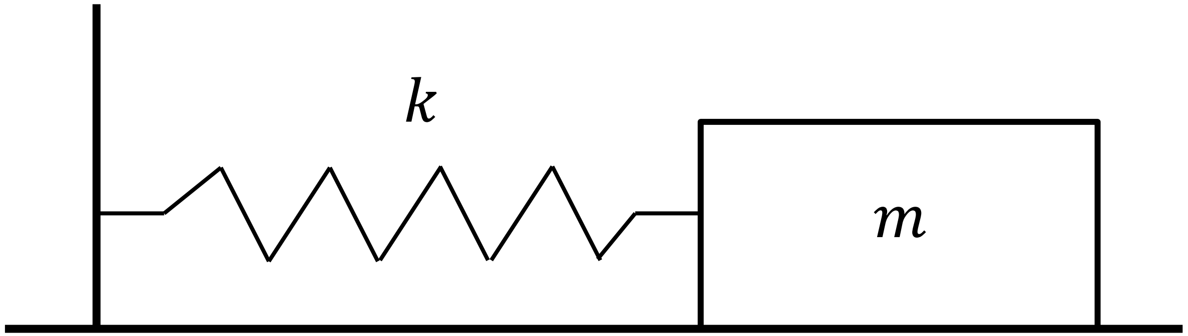


Case 2: Simulation is Possible

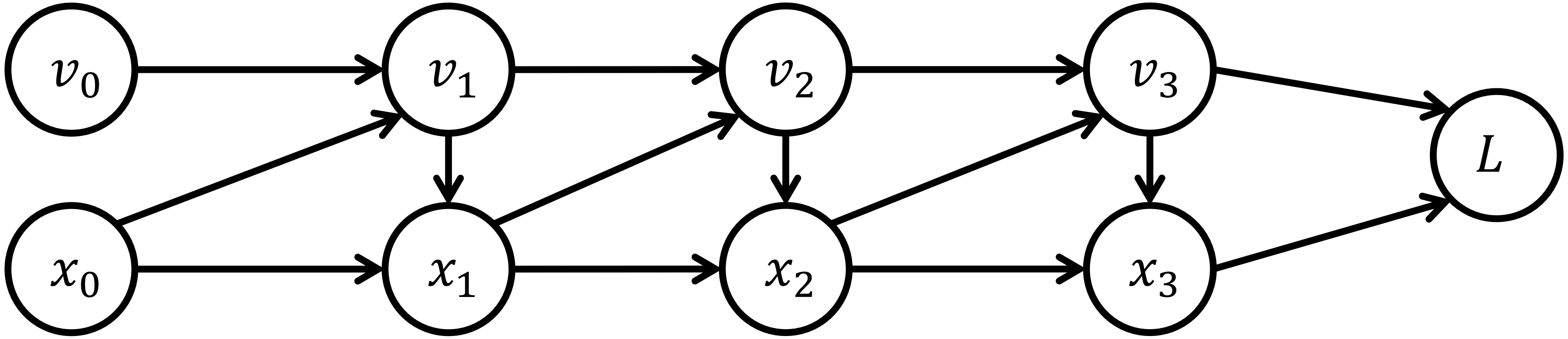
Differentiable Simulation Can Help

Forward pass

$$v_{i+1} = v_i - hk\frac{x_i}{m}$$
$$x_{i+1} = x_i + hv_{i+1}$$

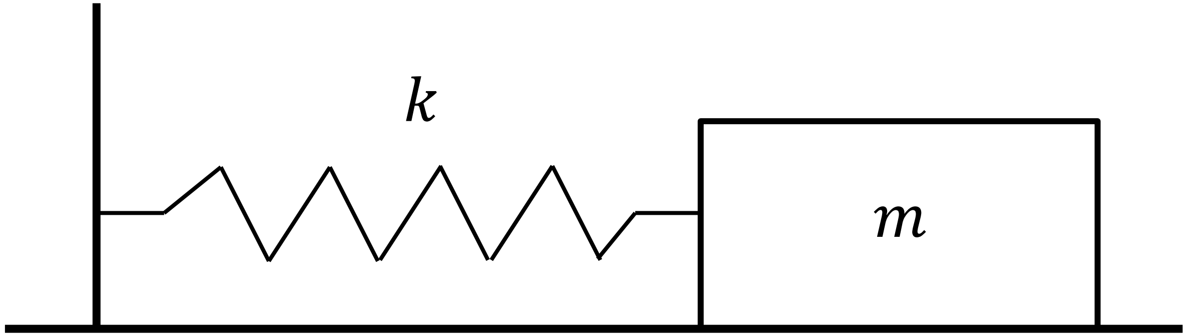


The computational graph



Differentiable Simulation Can Help

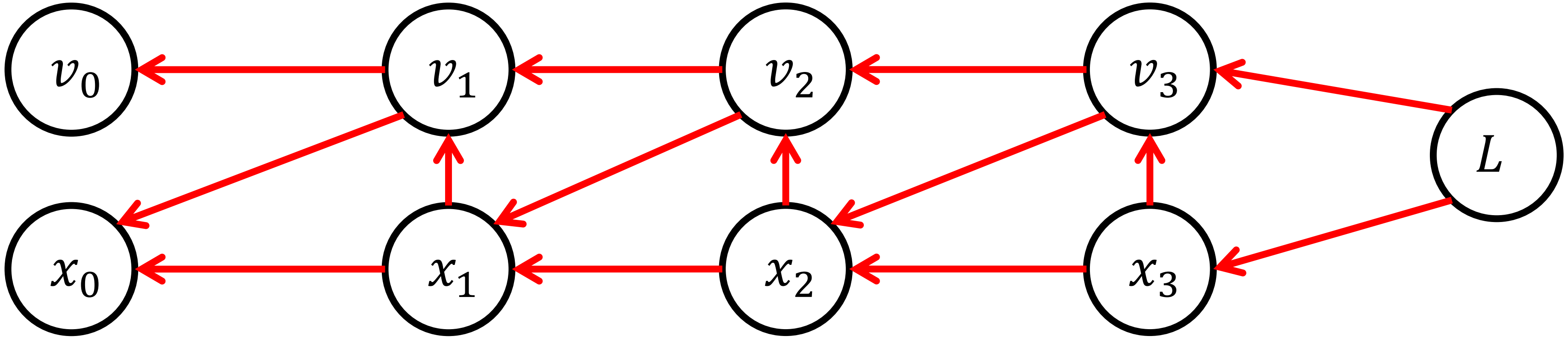
Backward pass



$$v_{i+1} = v_i - hk\frac{x_i}{m}$$
$$x_{i+1} = x_i + hv_{i+1}$$

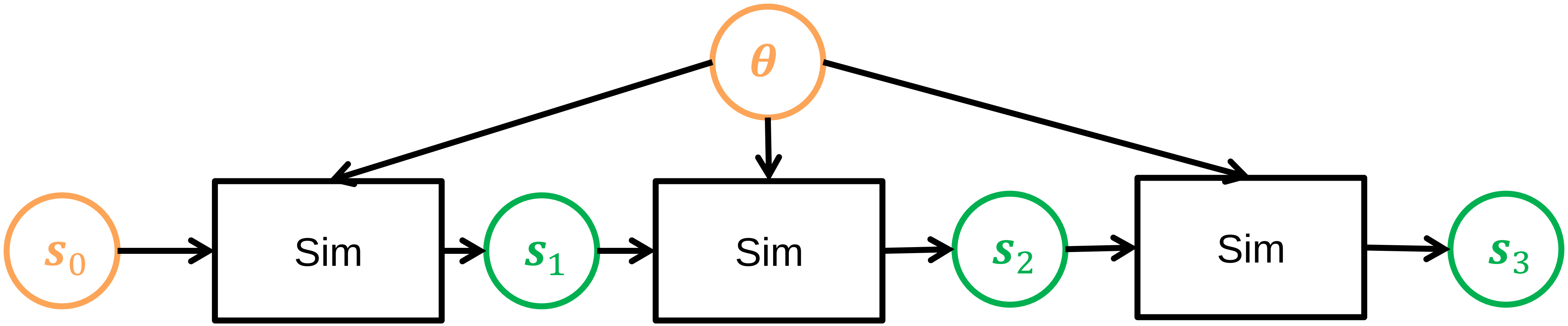
$$\frac{\partial L}{\partial v_i} = \frac{\partial L}{\partial v_{i+1}} \qquad \frac{\partial L}{\partial x_i} = -hk\frac{\partial L}{\partial v_{i+1}}$$
$$\frac{\partial L}{\partial x_i} = \frac{\partial L}{\partial x_{i+1}} \qquad \frac{\partial L}{\partial v_{i+1}} = h\frac{\partial L}{\partial x_{i+1}}$$

Backpropagation

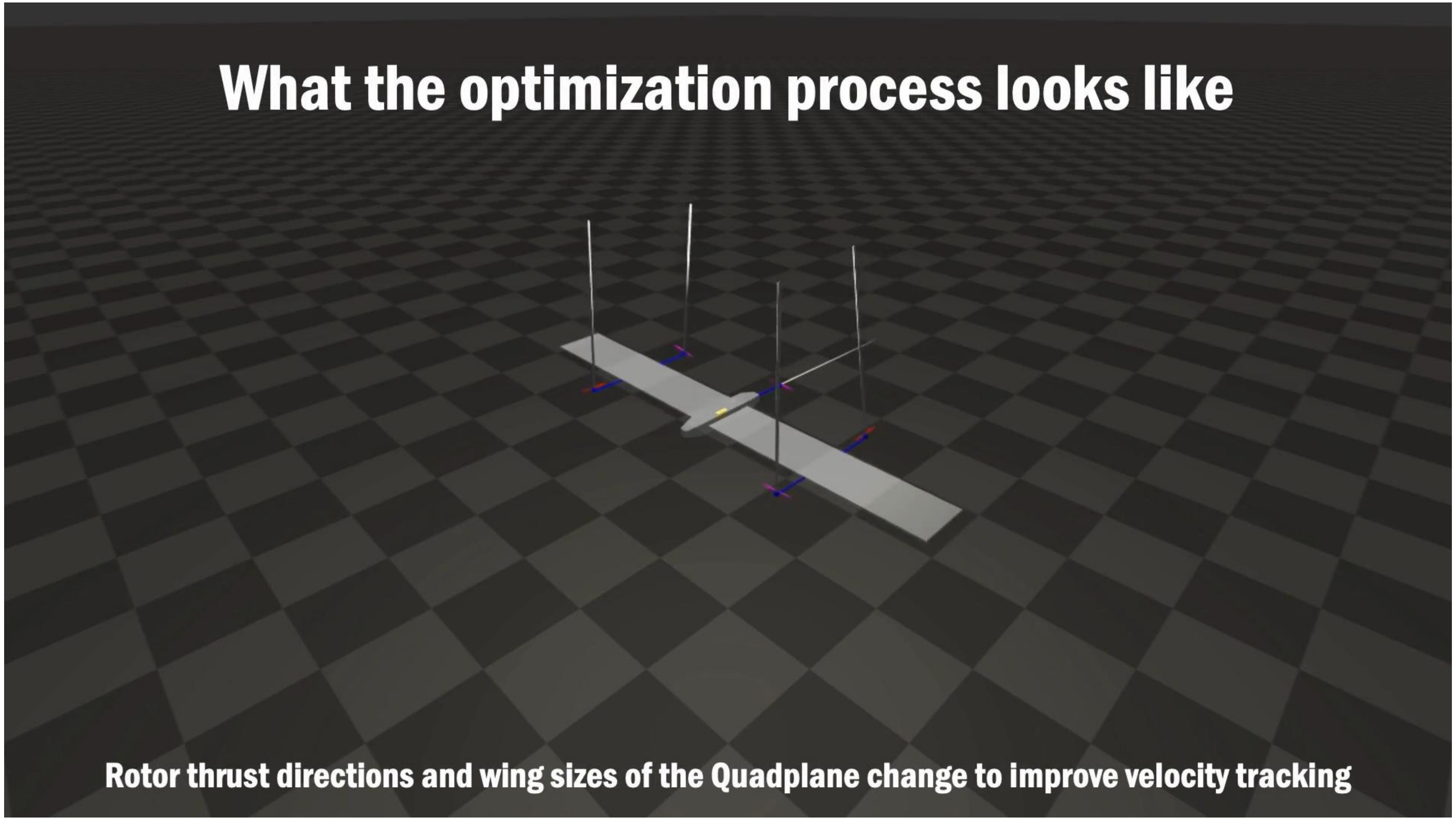


Differentiable Simulation Can Help

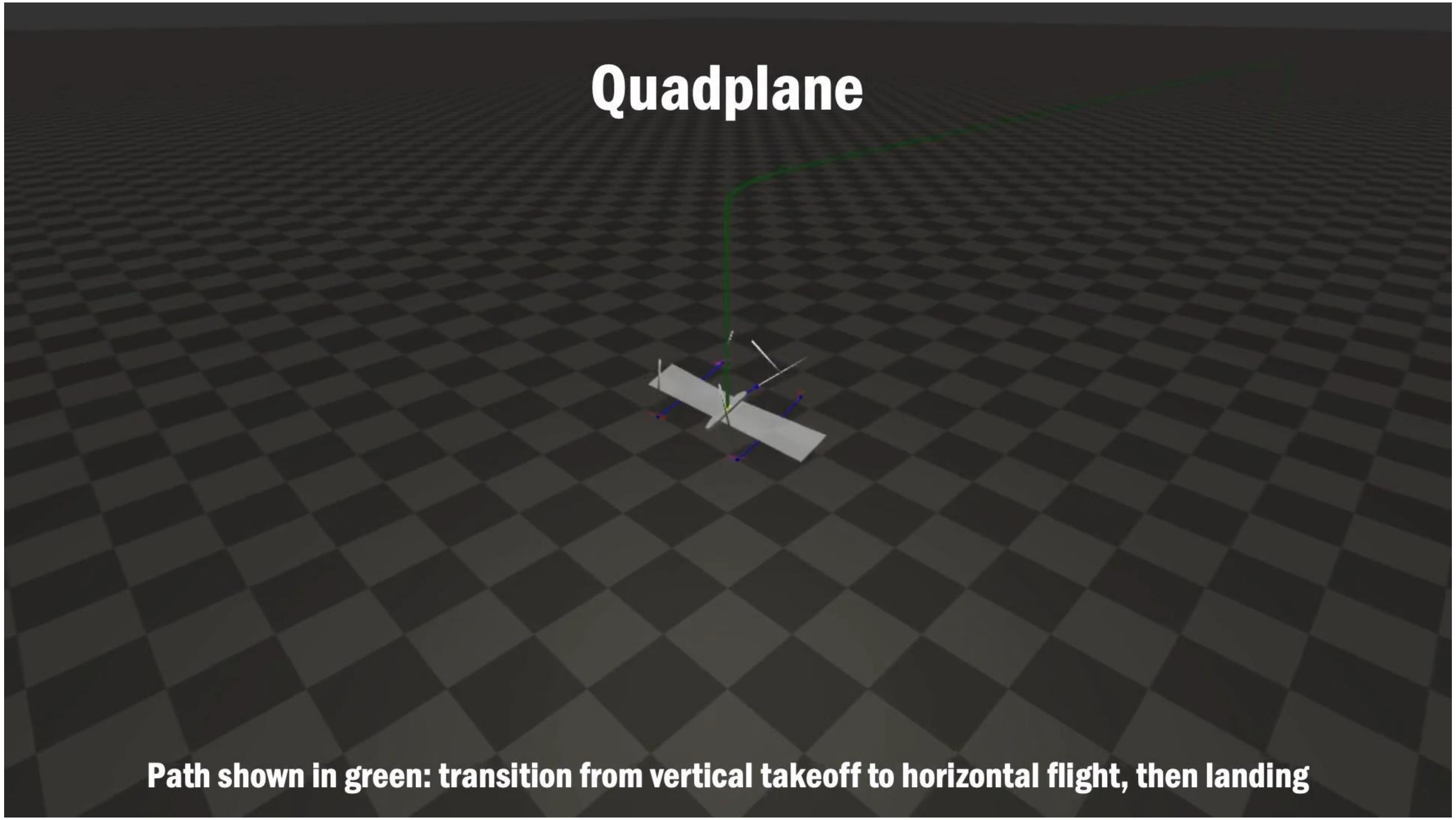
- System identification (optimizing θ)
- Initial condition optimization (optimizing s_0)



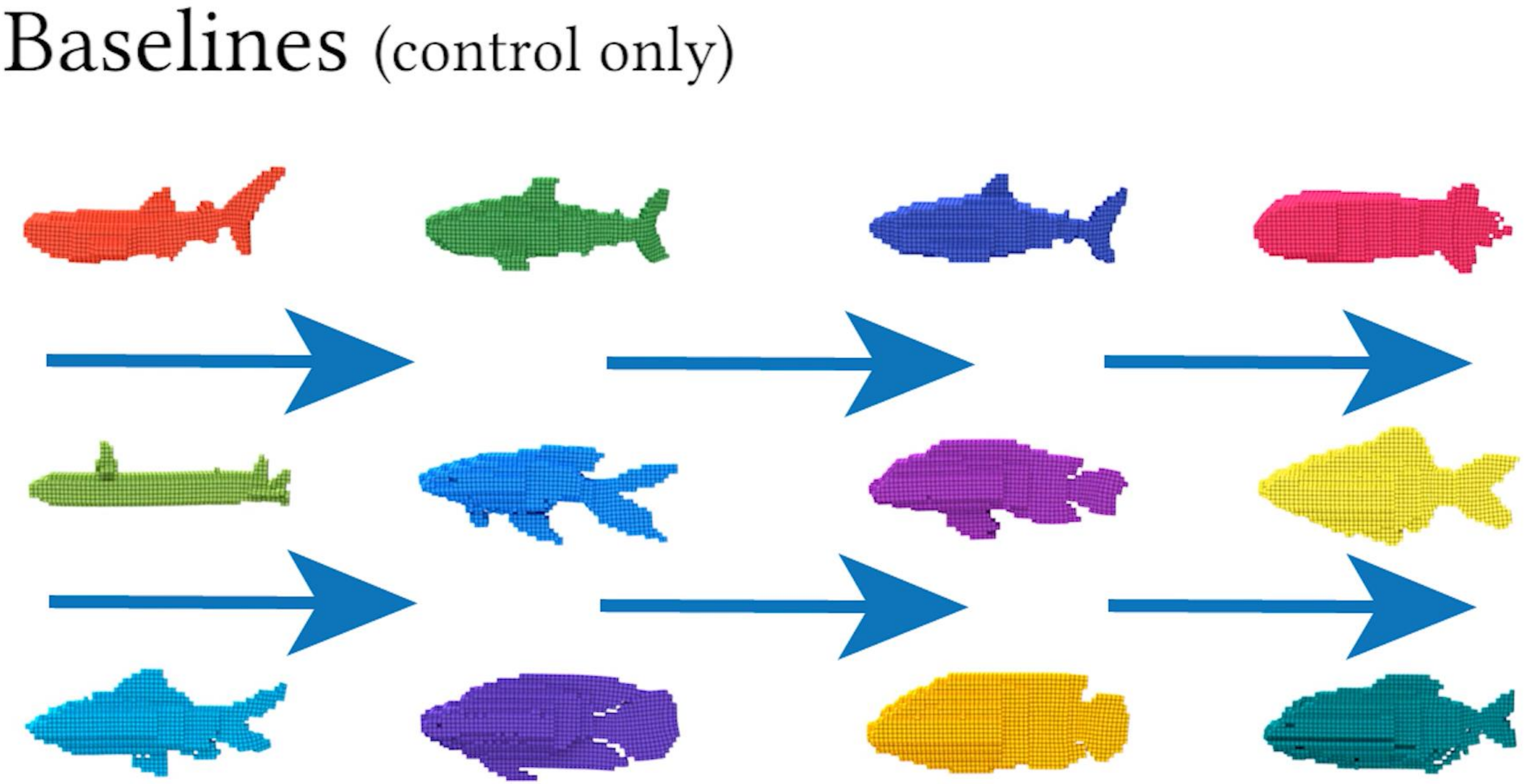
Optimization of Hybrid UAVs



Optimization of Hybrid UAVs

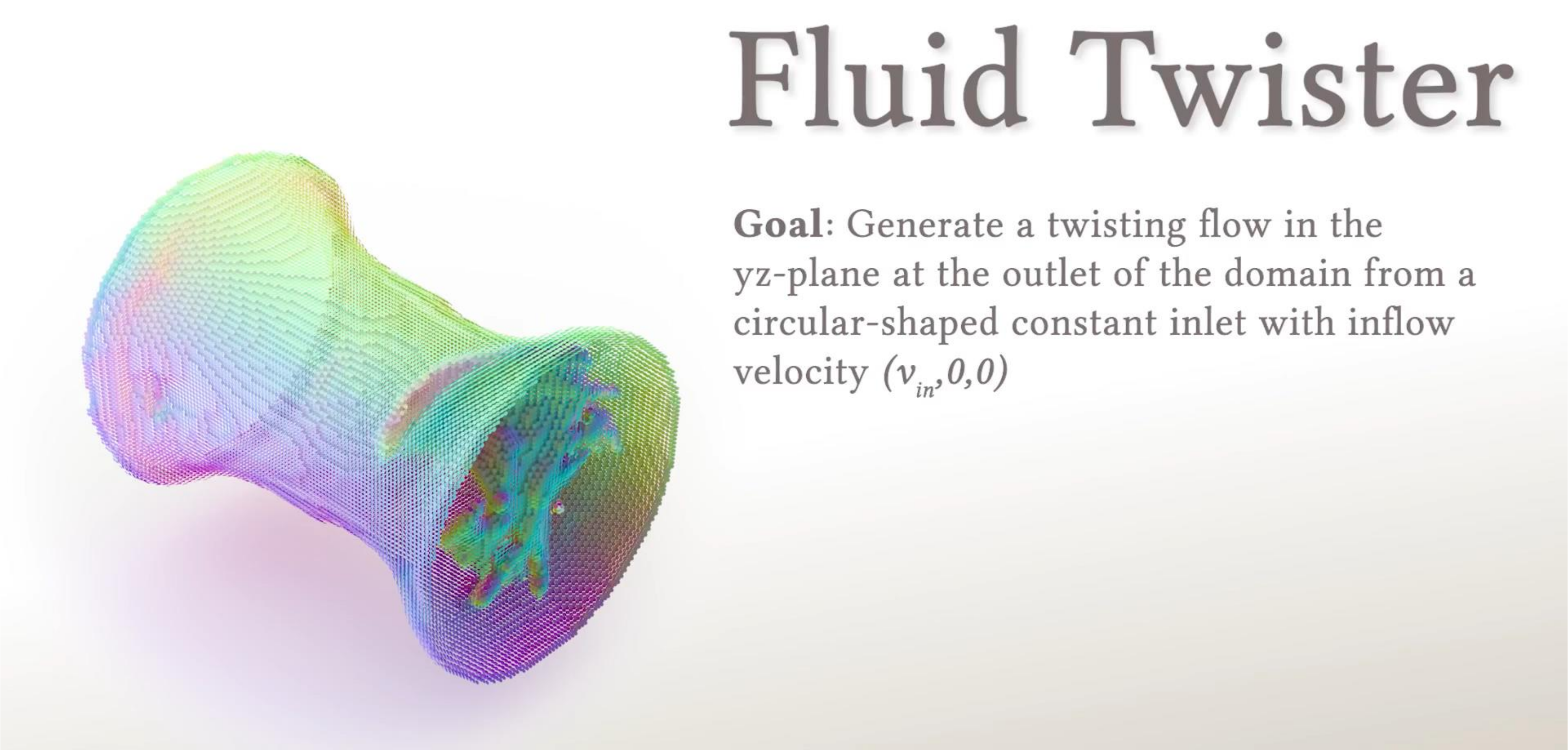


Optimization of Soft Fish



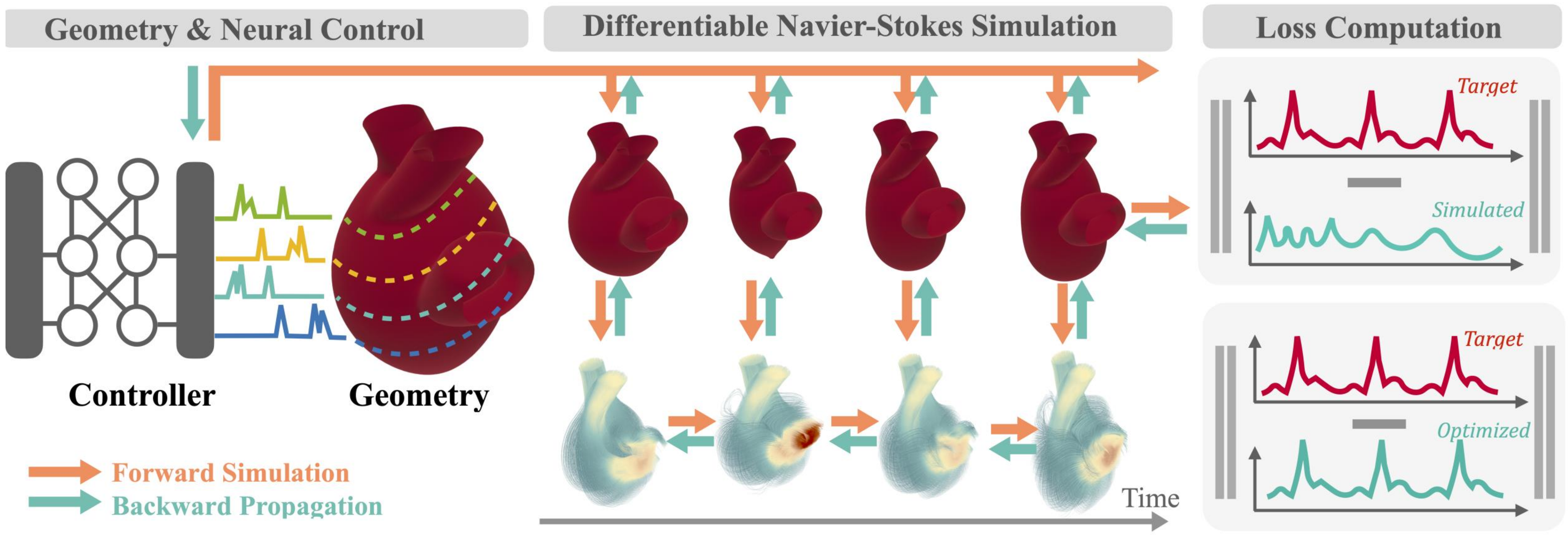
Ma et. al, Siggraph, 2022

Optimization of Fluidic Systems



Ma et. al, Siggraph Asia, 2022

Optimization of Fluidic Systems



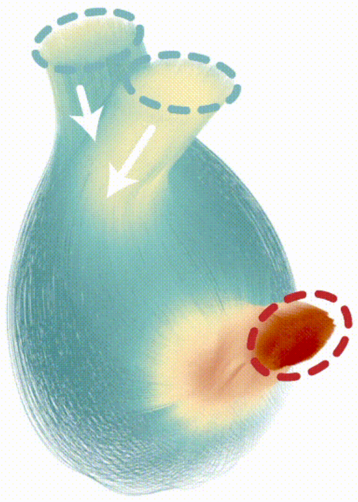
Li et. al, NeurIPS, 2024

Optimization of Fluidic Systems

Neural Heart

Goal: Optimize a closed-loop controller parameterized by a two-layer MLP to control the muscle excitation signal at four cross-sections of an aritificial heart to match a target outlet flow profile

Domain: 48x48x48



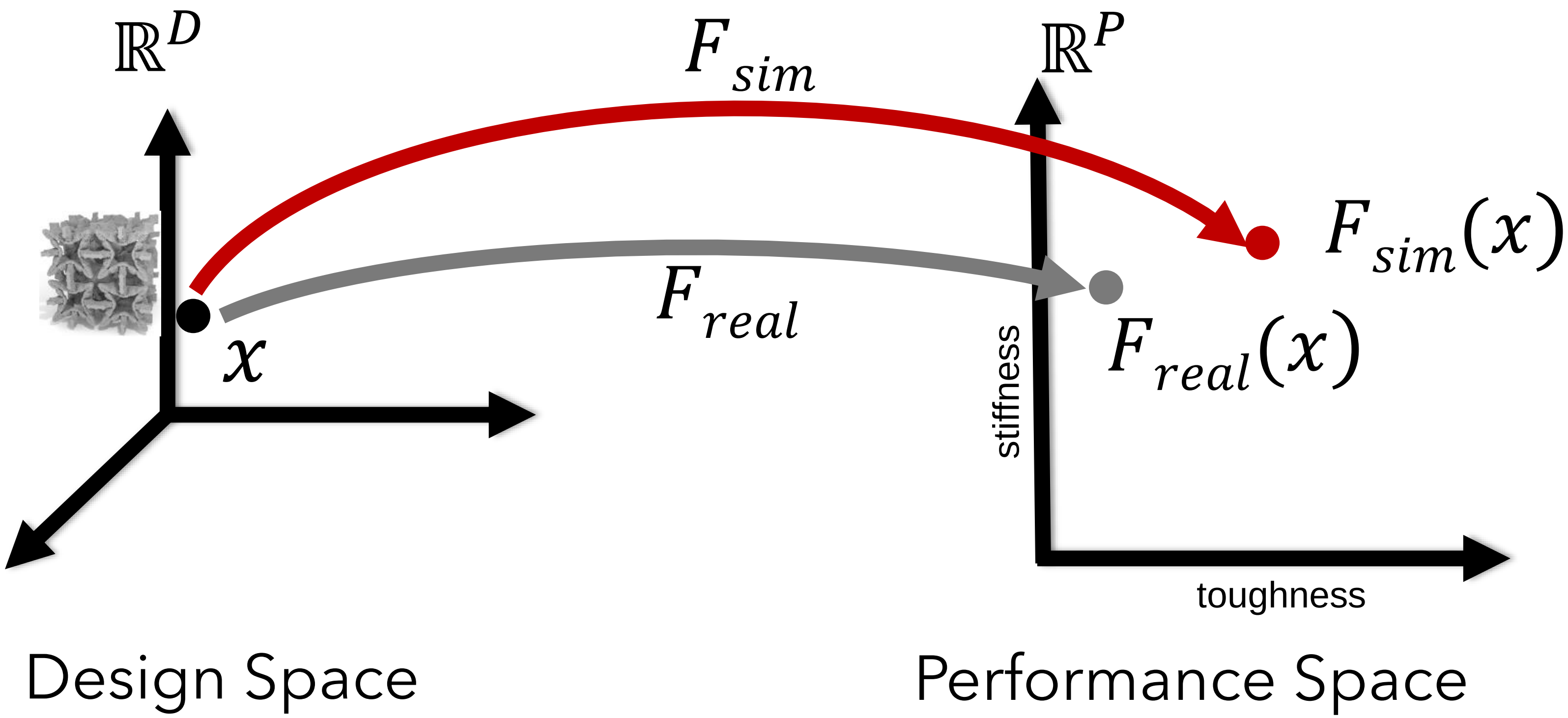
Li et. al, NeurIPS, 2024

Key Questions for Computational Design

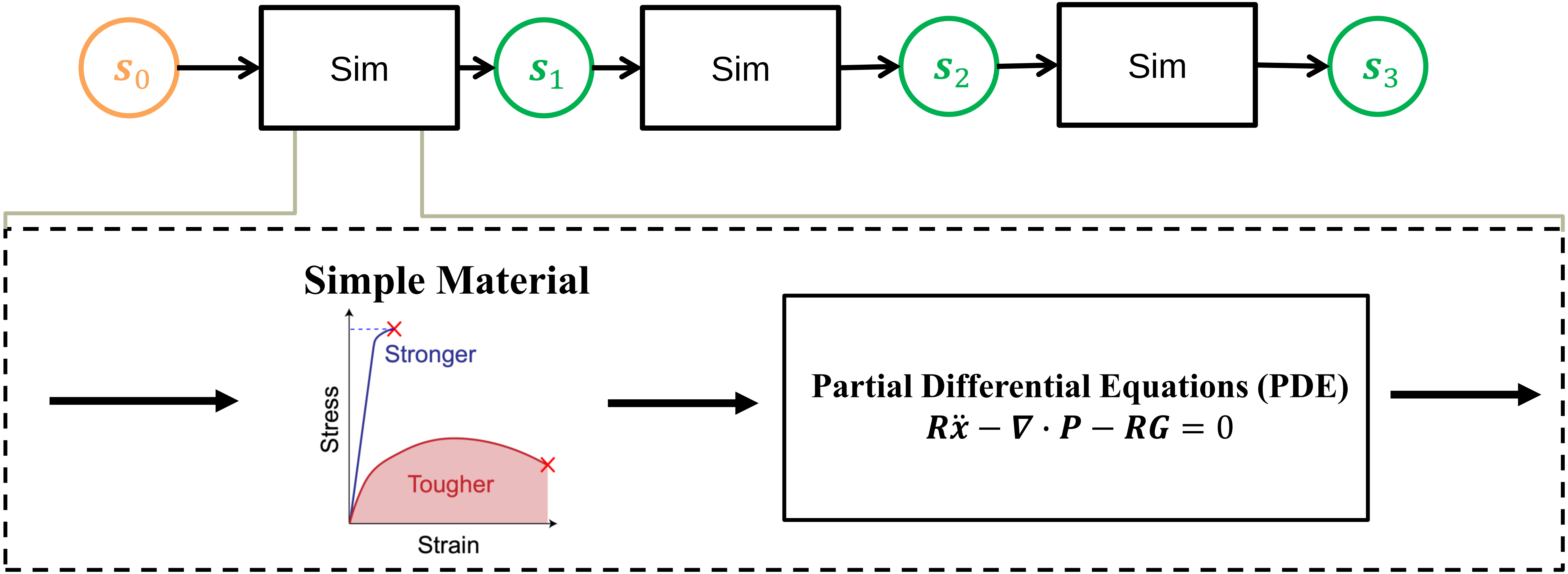
1. How to represent a design?
2. How to represent a design space?
3. How to learn a design space?
4. How to find designs with optimal performance?
5. How to bridge the gap between digital & real?

Simulation is Often Unreliable

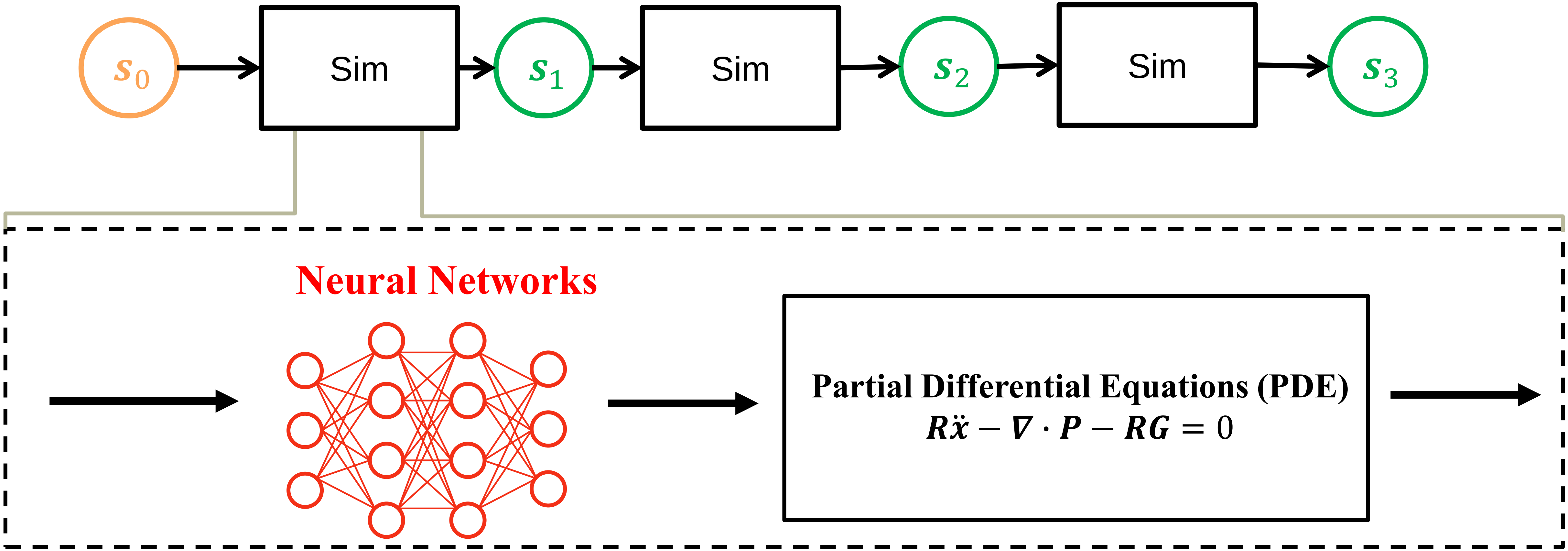
- Simulation does not match real experiments



Classical Physics-based Simulation



A Hybrid Neural-PDE Approach



Ma et. al, ICML, 2023

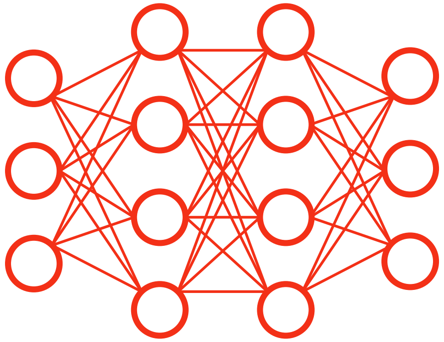
Closing the Sim-to-real Gap

Fits the real-world data better than classic models

Loss

Backpropagation

Neural Networks



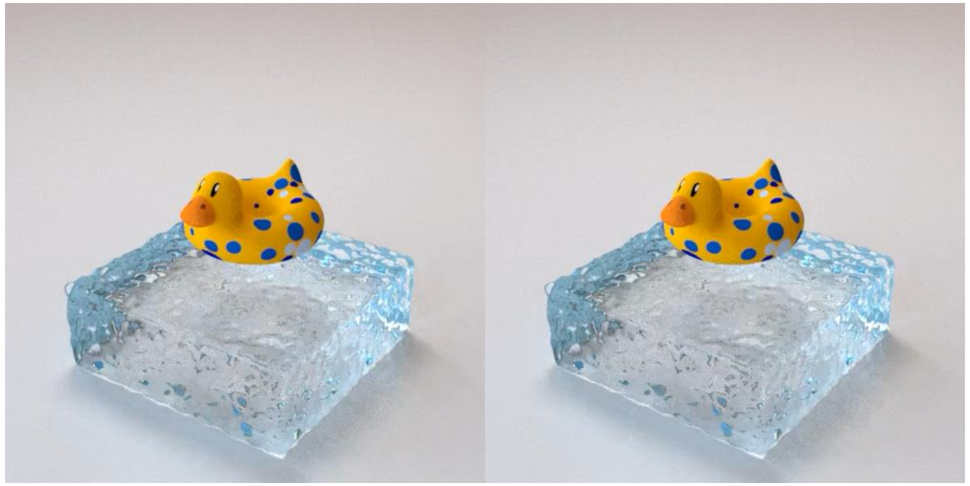
Partial Differential Equations (PDE)

$$\boldsymbol{R}\ddot{\boldsymbol{x}} - \boldsymbol{\nabla} \cdot \boldsymbol{P} - \boldsymbol{R}\boldsymbol{G} = 0$$

Ma et. al, ICML, 2023

A Hybrid Neural-PDE Approach

Ground truth Simulation



Ground truth Simulation

Data efficiency: one-shot generalization over **geometries**, boundary conditions, temporal range, and **multi-physics.**

Ma et. al, ICML, 2023

Comparison to Data Driven Methods

Training data | Ground truth | Ours | GNS

Generalization: **over 100X times more accurate** than end-to-end ML approaches that do not keep the PDEs, e.g., graph neural network (GNN) simulation

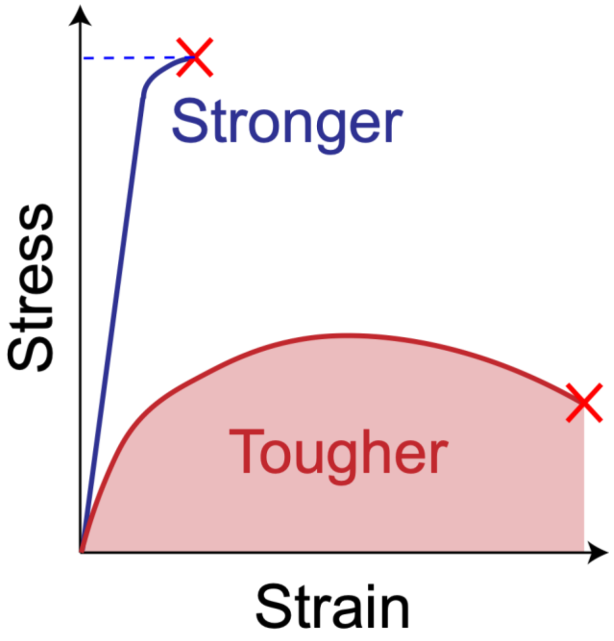
Ma et. al, ICML, 2023

Example: tough & strong composites

Strength: the ability to recover from an applied load.

Toughness: the ability to resist cracks.

Engineering applications require materials to be simultaneously strong and tough.



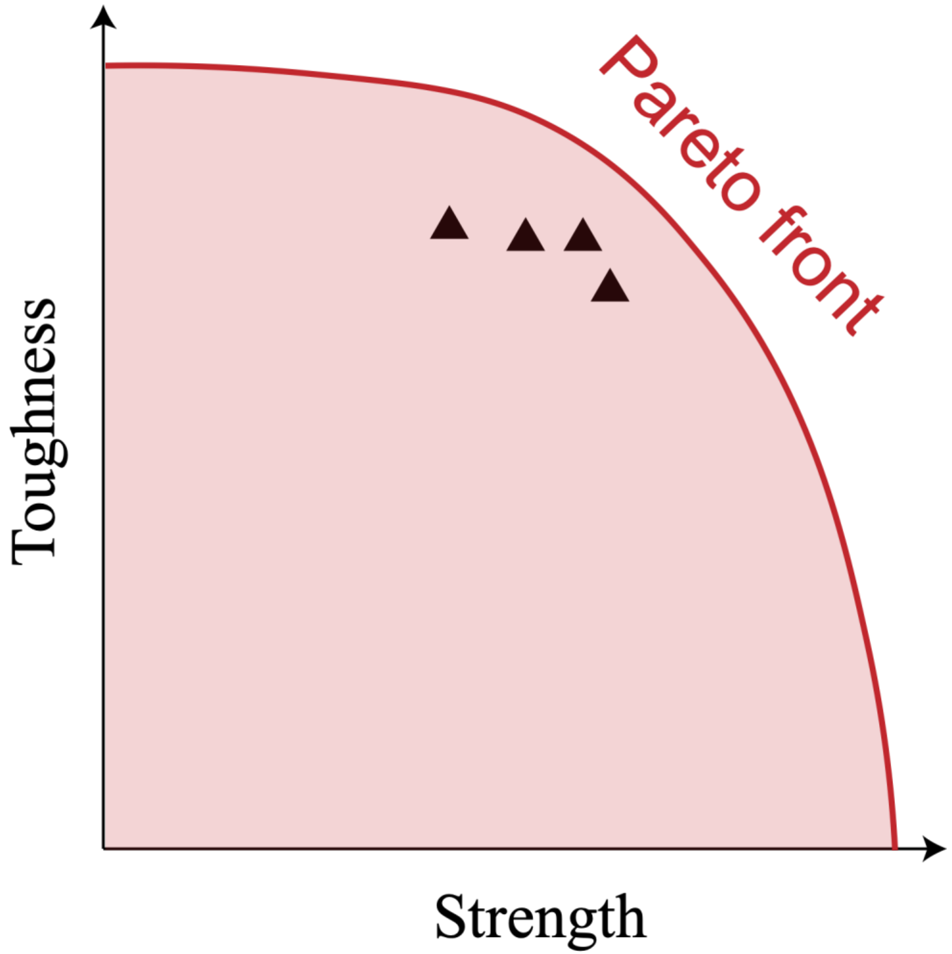
Strength and Toughness are often <u>mutually exclusive</u>.

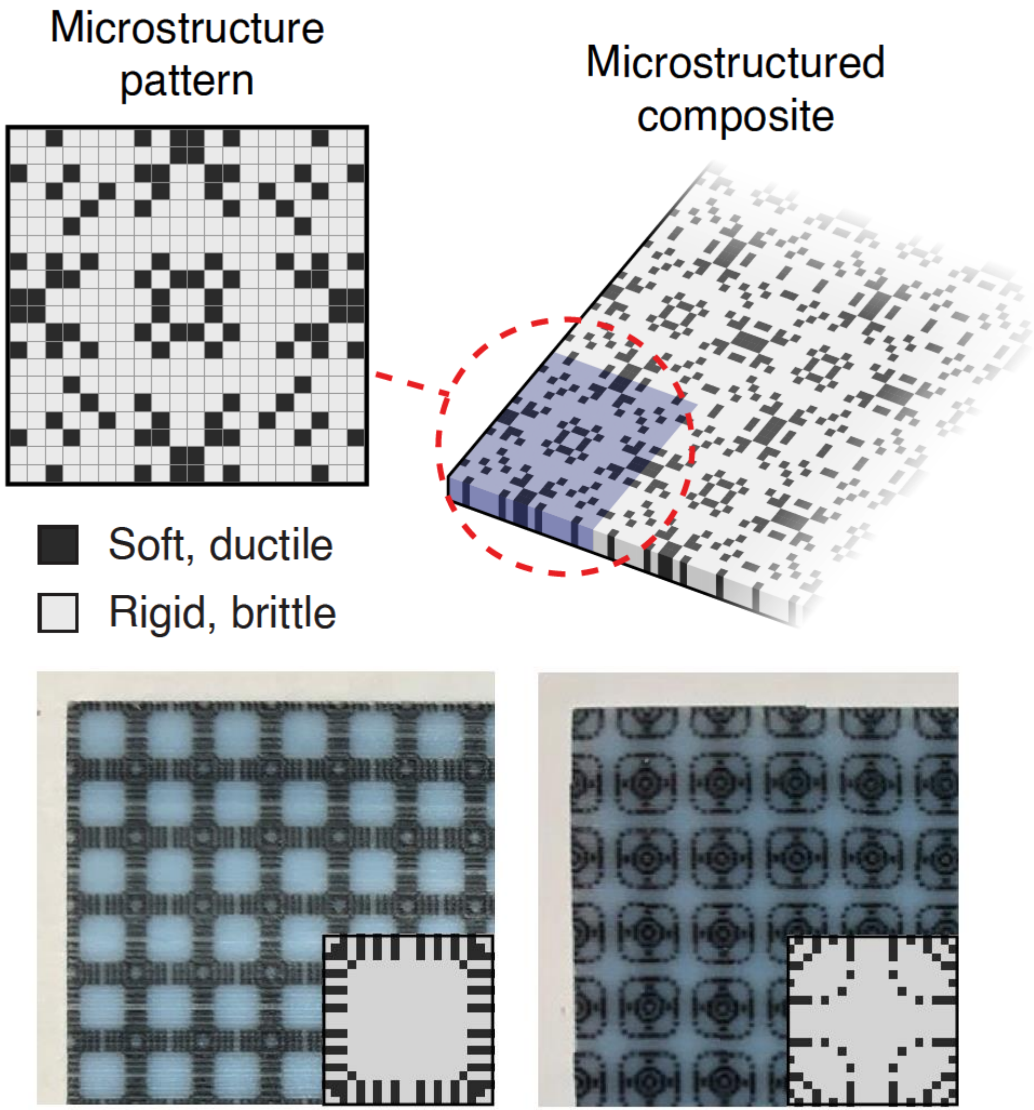
Because to be tough, a material has to be ductile enough to tolerate long cracks and absorb more energy during fracture.

How to design materials that are simultaneously strong and tough?

Example: tough & strong composites

A full picture: Pareto Front

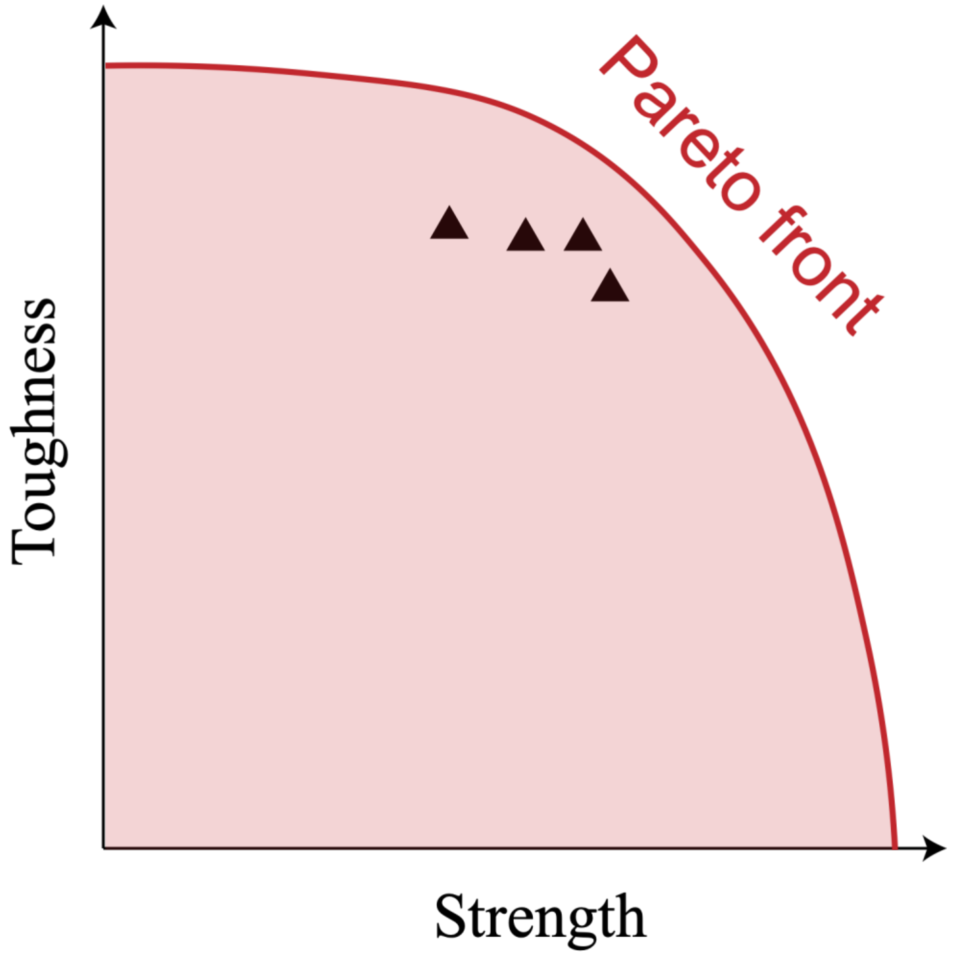


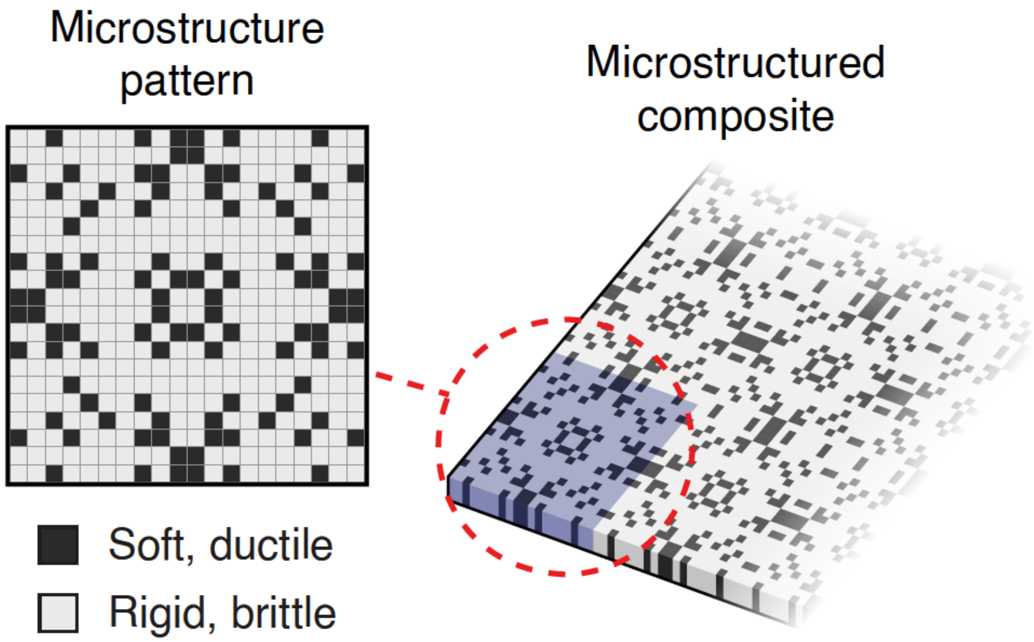


3D printed by OBJET

Example: tough & strong composites

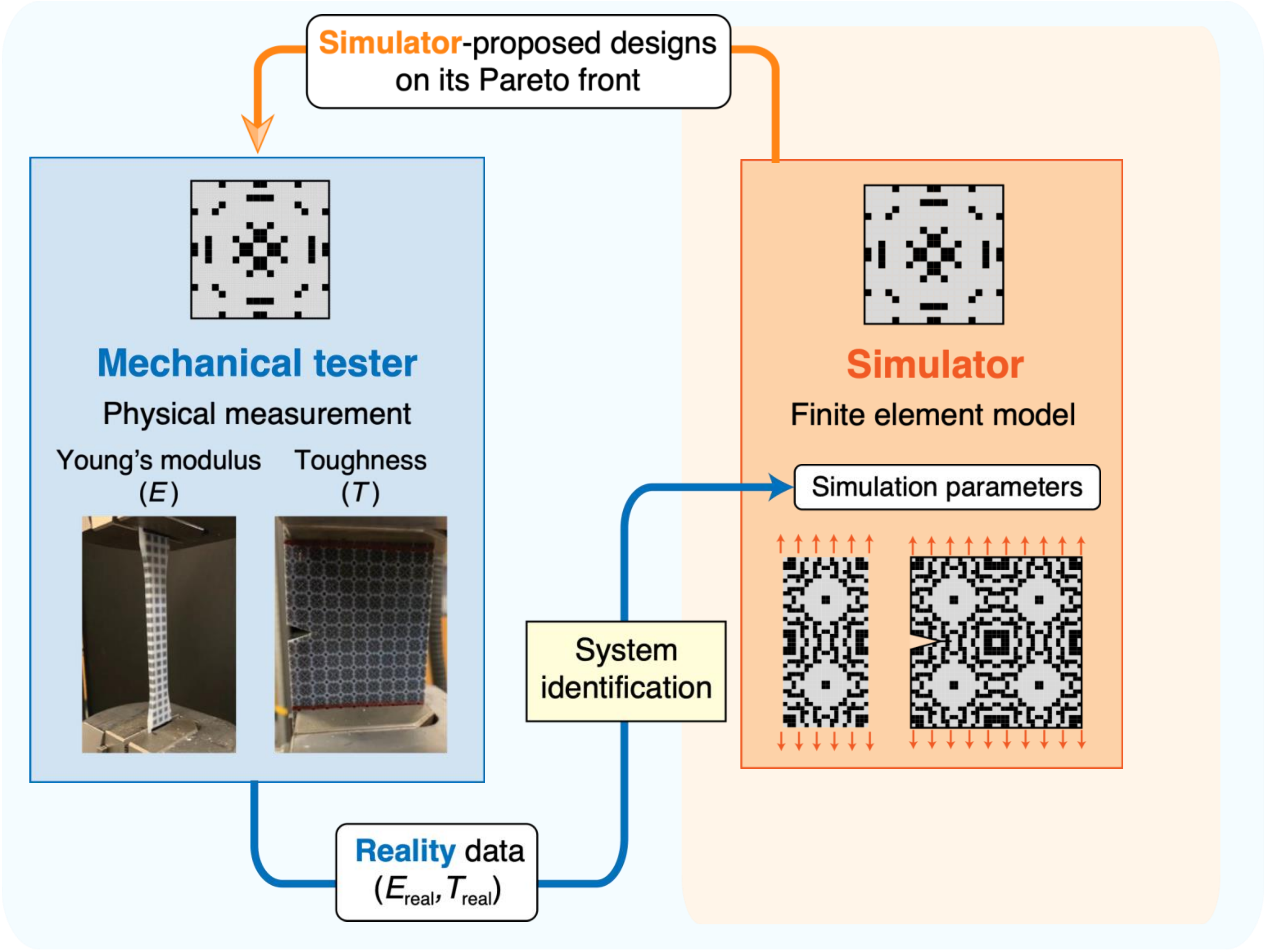
A full picture: Pareto Front



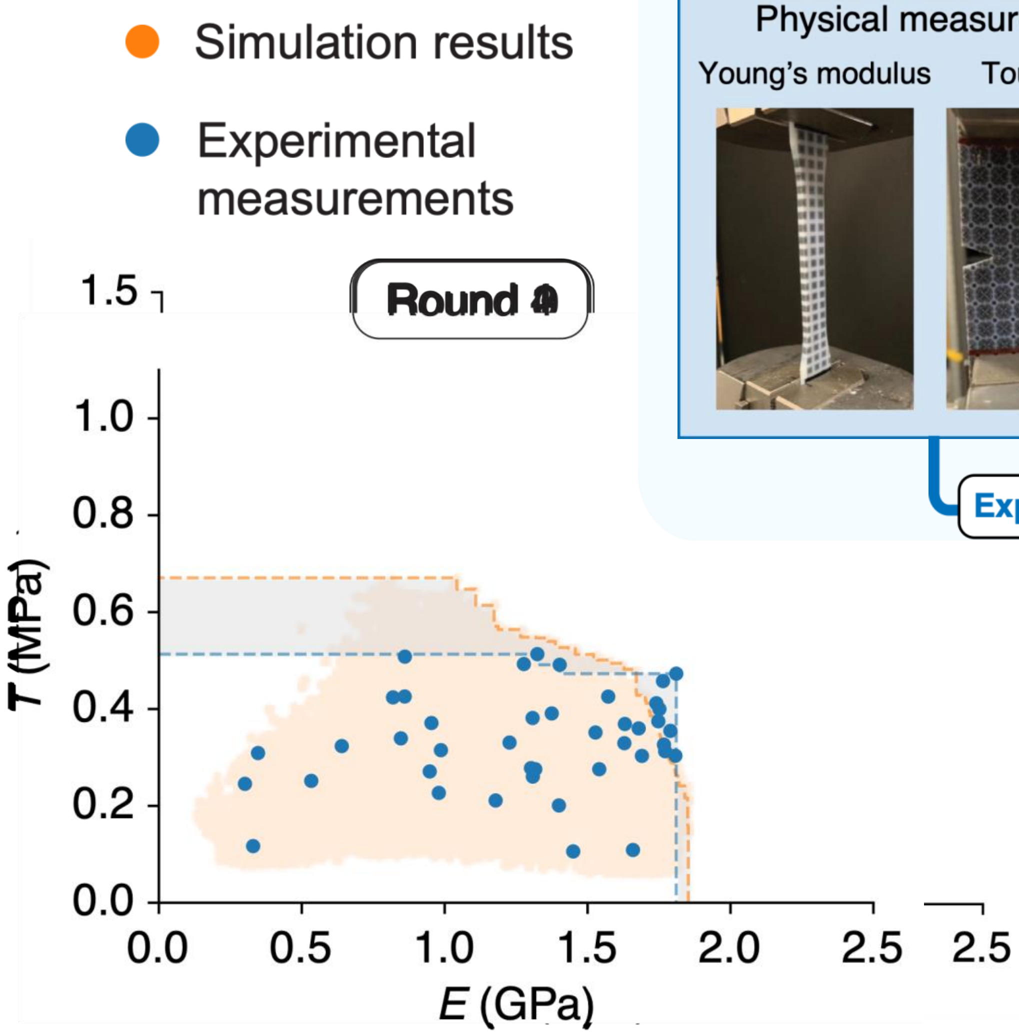


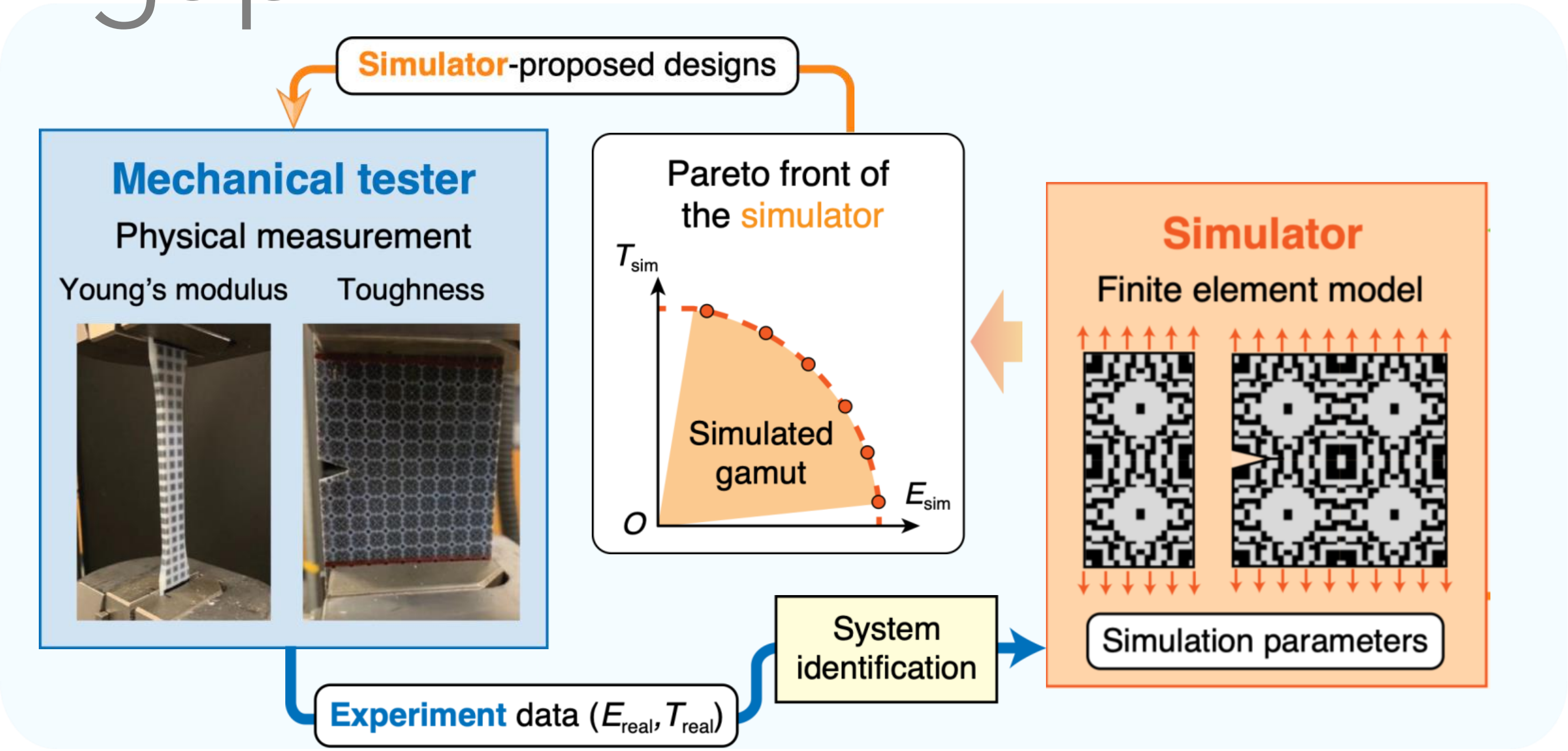
Challenge: simulation-to-reality gap.
Reason: highly nonlinear fracture dynamics.

A competitive game



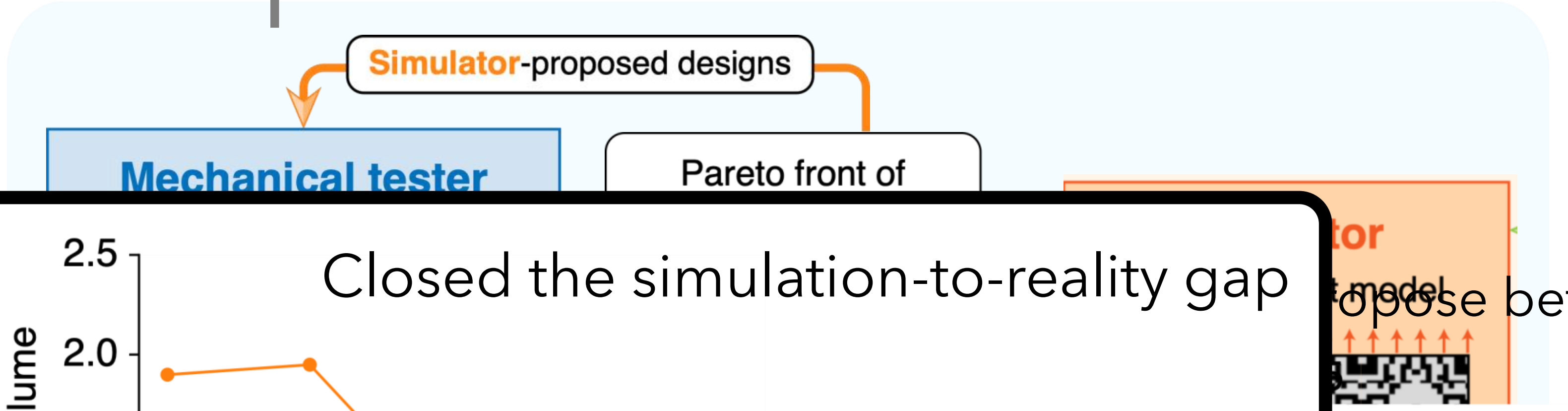
Sim-to-real gap



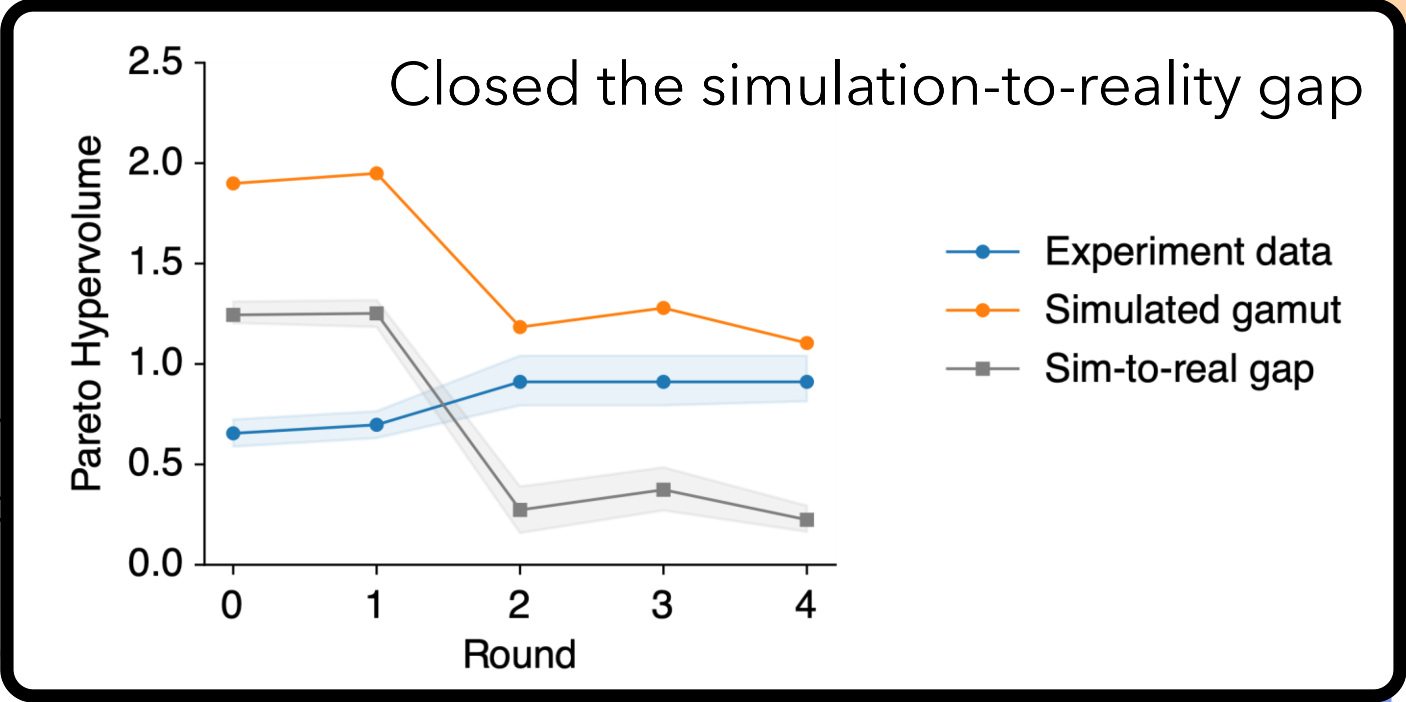


https://arxiv.org/abs/2302.01078

Sim-to-real Gap

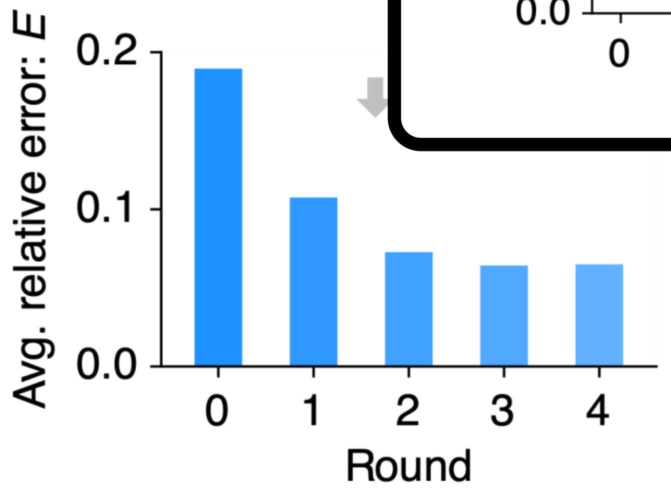


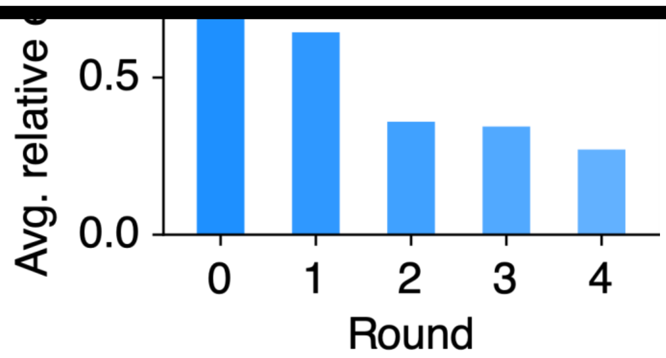
opose better designs

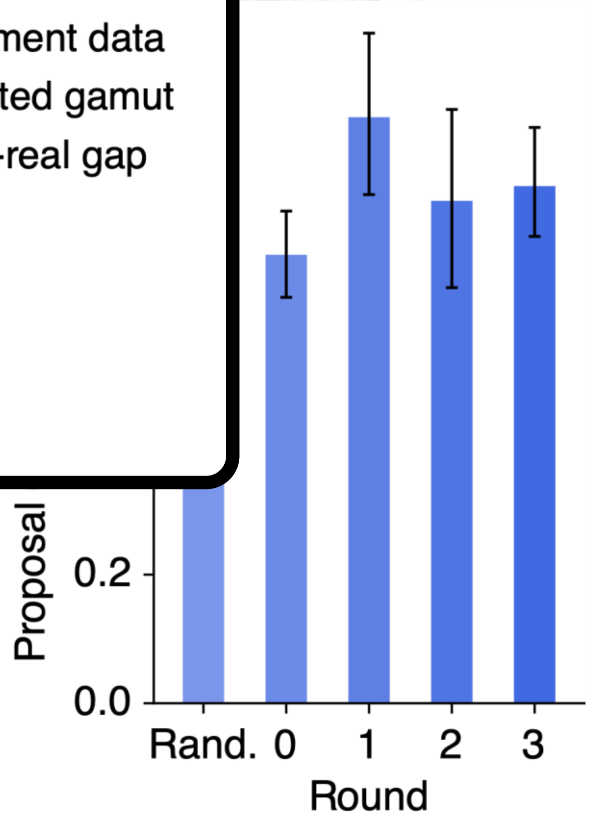


As the algorithm iter

1. Simulator bec

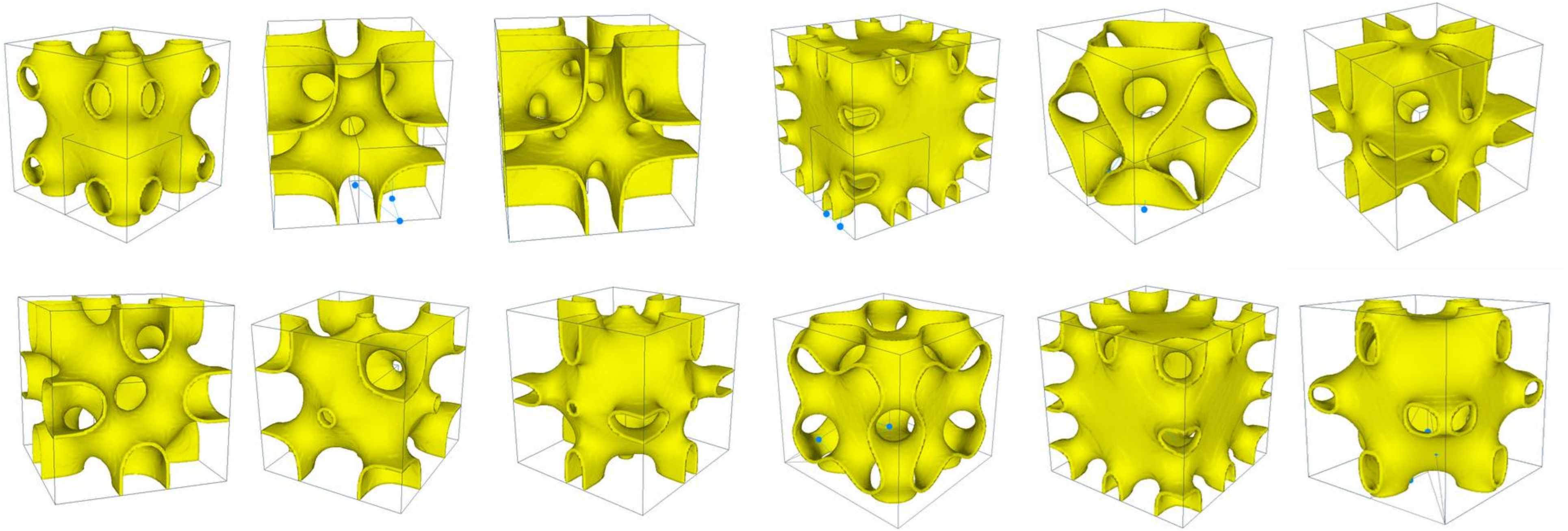






Can computers beat humans at design?

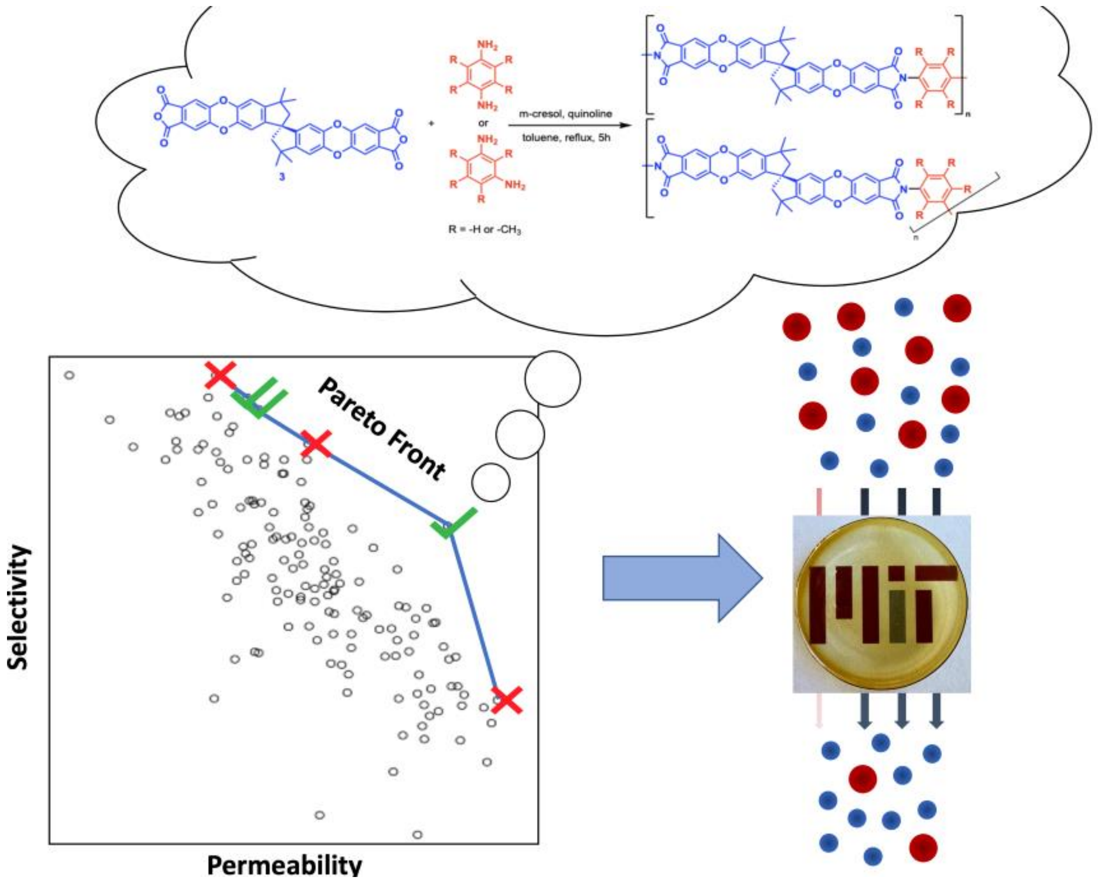
Representations are Key



Hundreds of new TPMS structures

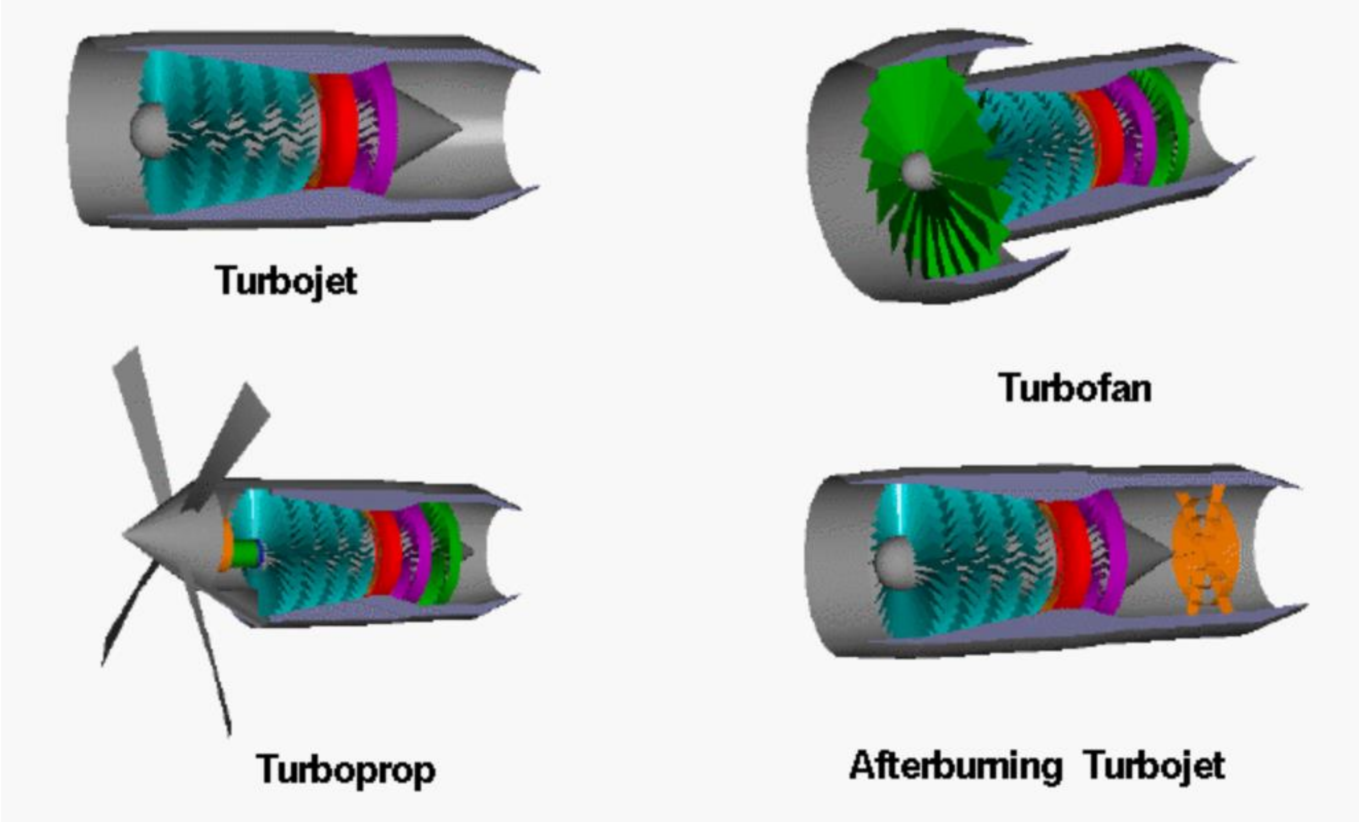
Design of Novel Molecules

- We can find new designs with optimal performance trade-offs



Design of Novel Molecules

- We can find new designs with optimal performance trade-offs
- The grand challenge is extrapolation beyond current data.



Will computers and humans produce better designs?

Lessons Learned

- Representations are key
- Scientists need workflows for small experimental datasets
- New workflows will learn/create specialized design spaces
- Workflows need to incorporate knowledge from experts & check validity
- Workflows will couple generative and predictive models to find optimal design
- Predictive models will combine neural/classical models
- We will be able to find new designs with optimal performance trade-offs
- But the real grand challenge is extrapolation beyond current data.

Acknowledgements

Allan Zhao, MIT
Beichen Li, MIT
Liane Makatura, MIT
Liang Shi, MIT
Timothy Erps, MIT
Michael Foshey, MIT
Minghao Guo, MIT
Pingchuan Ma, MIT
Yifei Li, MIT
Sebastien Wah, MIT
John Zhang, MIT
Bolei Deng, MIT
Bohan Wang, MIT

Daniela Rus, MIT
Wei Wang, MIT
Josie Hughes, EPFL
Juan Salazar, MIT
Mina Konakovic Lukovic, MIT
Sangeeta Srinivasan, Wisconsin
Eftychios Sifakis, Wisconsin
Robert Katzschman, ETH
Yu Zhang , ETH
Elvis Nava , ETH
Philip Arm , ETH
Mike Yan Michelis , ETH
Benjamin F. Grewe , ETH

Jie Chen, IBM
Payel Das, IBM
Veronika Thost, IBM
Wan Shou, U. Arkansas
Jie Xu, NVIDIA
Andy Spielberg, Harvard
Bo Zhu, Dartmouth
Kui Wu, Tencent
Tao Du, Tsinghua
Tae Hyun Oh, Postech
Bernd Bickel, IST
Chris Wojtan, IST
Yi-Lu Chen, IST

Thank you

Email:
wojciech@mit.edu