# Functional Generative Design: An Evolutionary Approach to 3D-Printing

Cem C. Tutum, Supawit Chockchowwat, Etienne Vouga, Risto Miikkulainen

GECCO 2018
Kyoto, Japan

neural networks research group

The University of Texas at Austin
Computer Science

# Overview

- **Motivation**: Generating 3D Printed functional designs
- **Methodology**: 4 building blocks
  - Variational Autoencoder (VAE)
  - Noisy Kriging
  - Efficient Global Optimization (EGO)
  - Real parameter GA (rGA)
- **Experiments**:
  - Uniform vs. Random Sampling
- **Results and Discussion**:
  - Optimal design and a few others
  - Discovered 3D printing rule
- **Future Work and Conclusions**

# Motivation

How can we design functional parts for 3D printing?

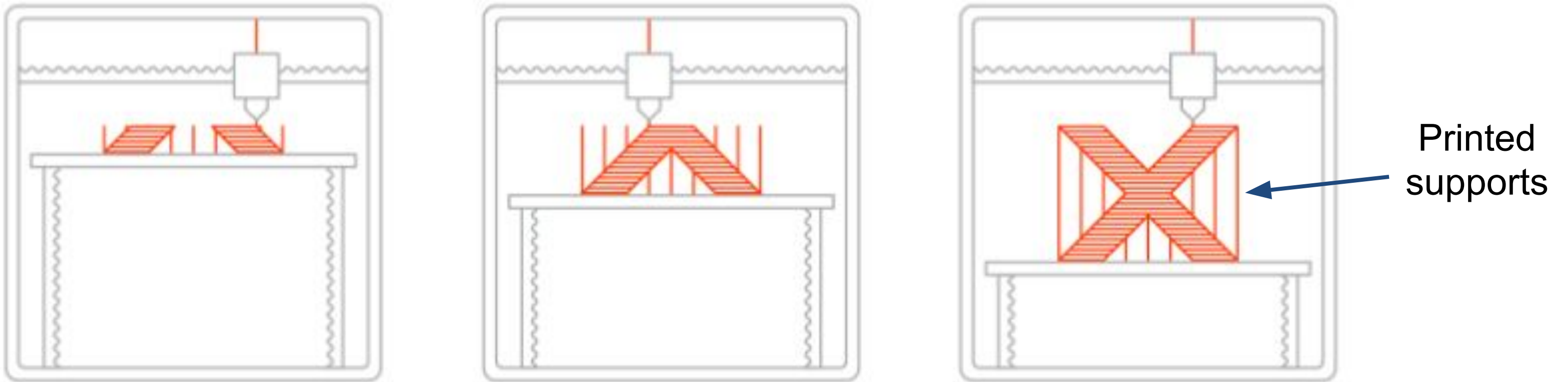
# Motivation

How can we design functional parts for 3D printing?

(kinematic) (FDM)

# Motivation

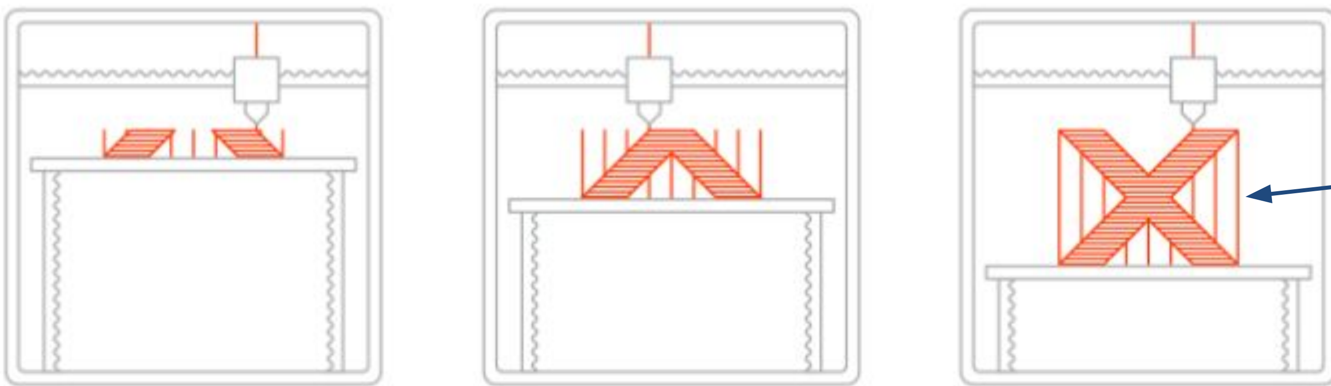
How can we design functional parts for 3D printing?




Fused Deposition Modeling Process

# Motivation

## How can we design functional parts for 3D printing?



Printed supports

Fused Deposition Modeling Process

- Takes time/effort to remove
- Produces waste material
- Leaves excess plastic behind
- Damage object during removal

requires post-processing

# Motivation

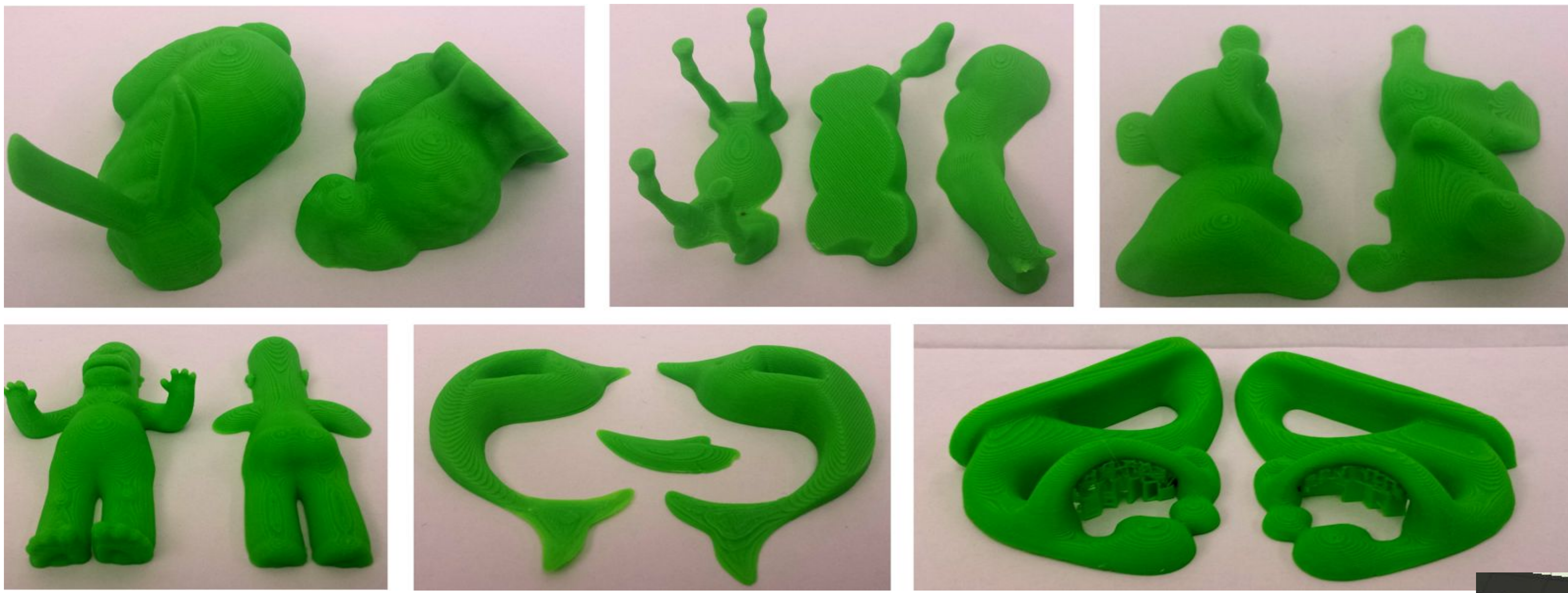
- Examples for 3D printed non-functional parts

# Motivation

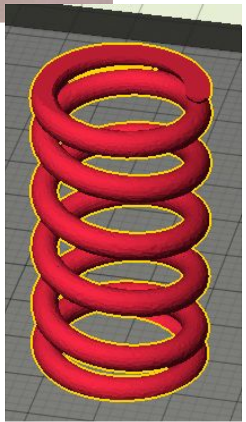
- One solution is Evolutionary Decomposition (E. Yu et.al., GECCO’17)

# Motivation

- One solution is Evolutionary Decomposition (E. Yu et.al., GECCO'17)



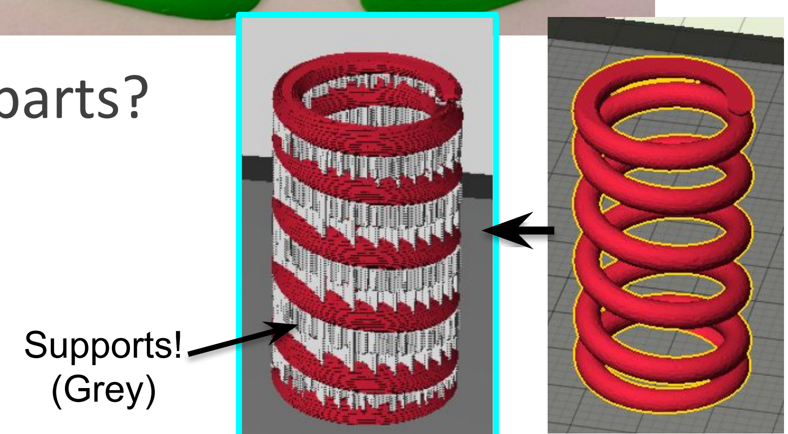
- **PROBLEM-1:** What about functional parts?



# Motivation

- One solution is Evolutionary Decomposition (E. Yu et.al., GECCO'17)

- **PROBLEM-1:** What about functional parts?

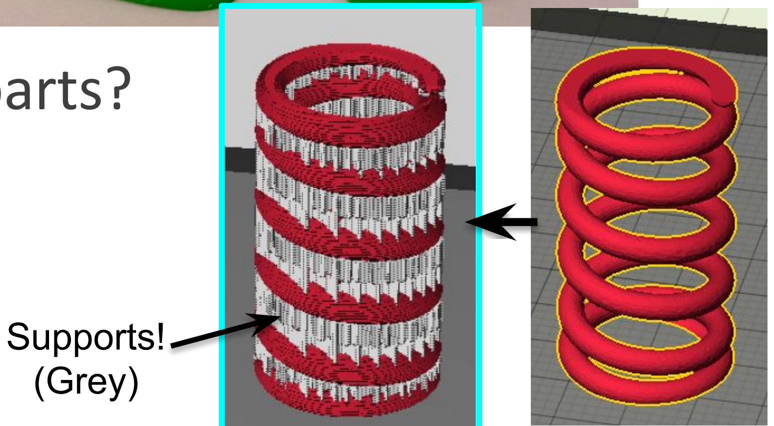


# Motivation

- One solution is Evolutionary Decomposition (E. Yu et.al., GECCO'17)

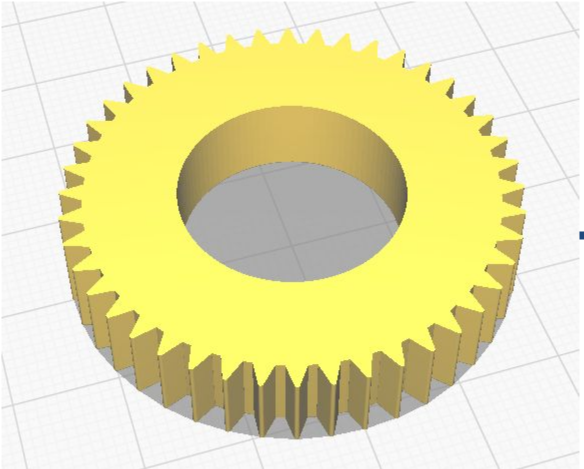
- **PROBLEM-1:** What about functional parts?

- We cannot decompose load-carrying parts!

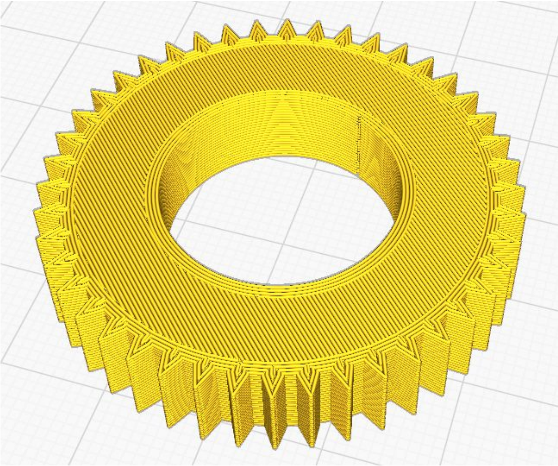


# 3D Printing Functional Parts

- **PROBLEM-2:** Resolution in 3D Printing



**Slicing**
**(Output: GCODE )**

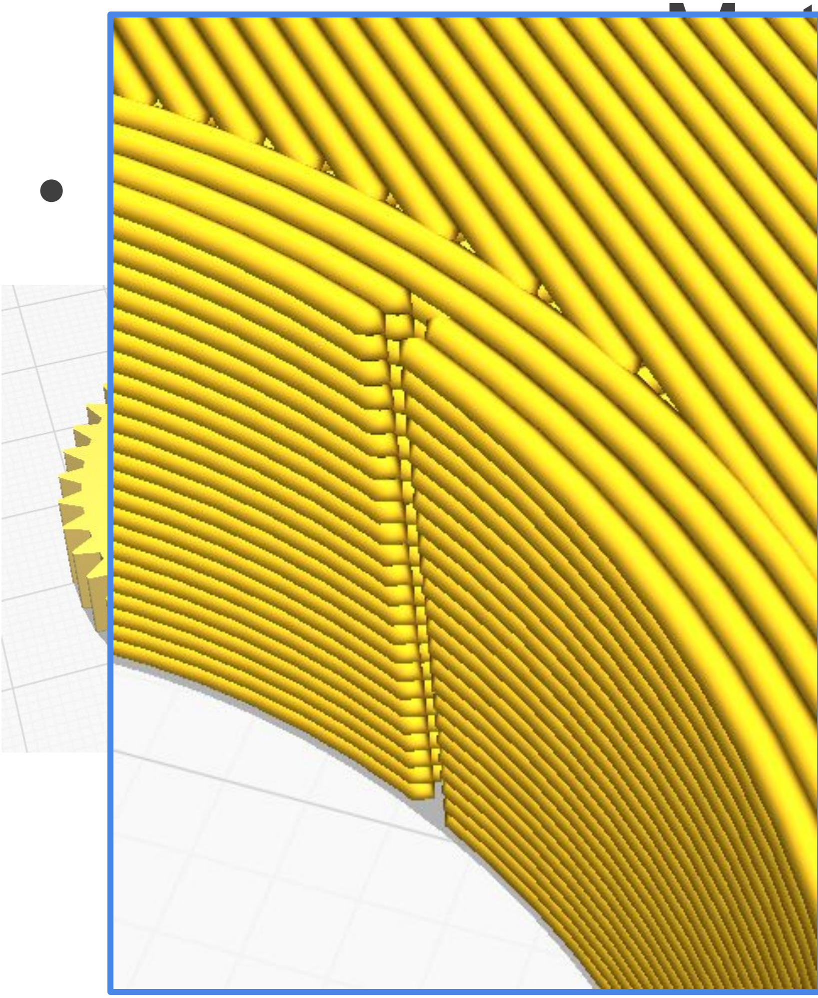


**3D gear design**
**(Continuum model)**

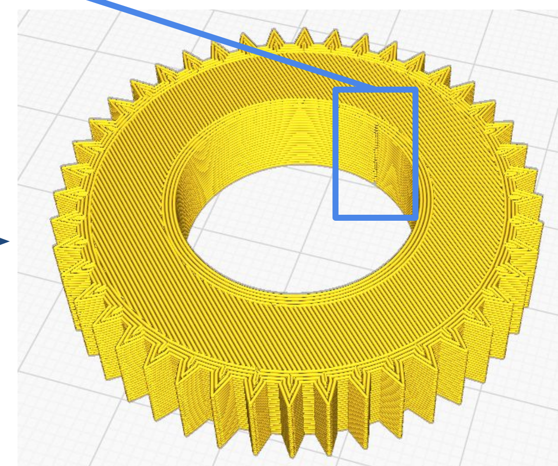
**3D Printing model**
**(Discrete model)**

# Motivation

- 3D Printing



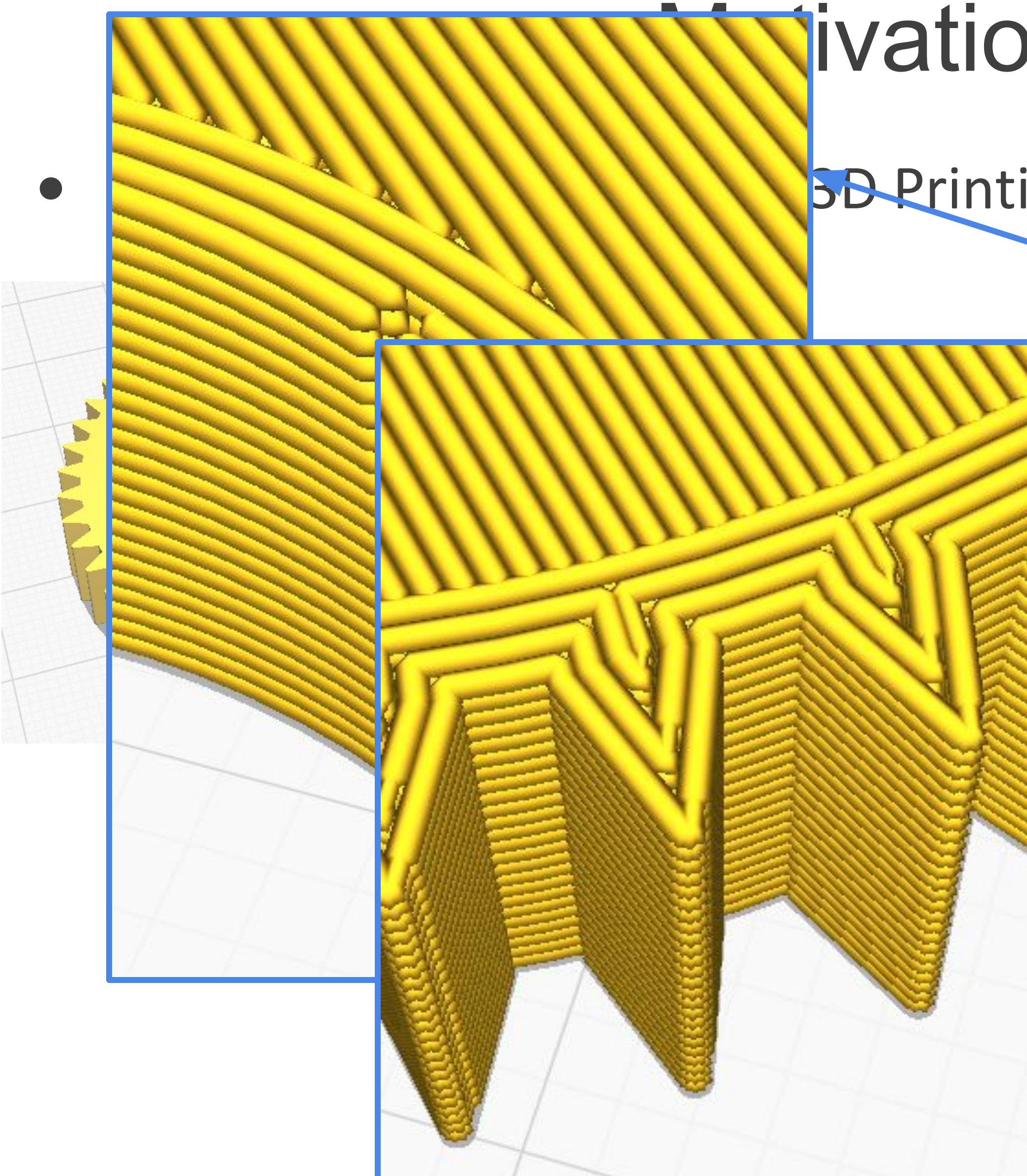
ng
CODE )

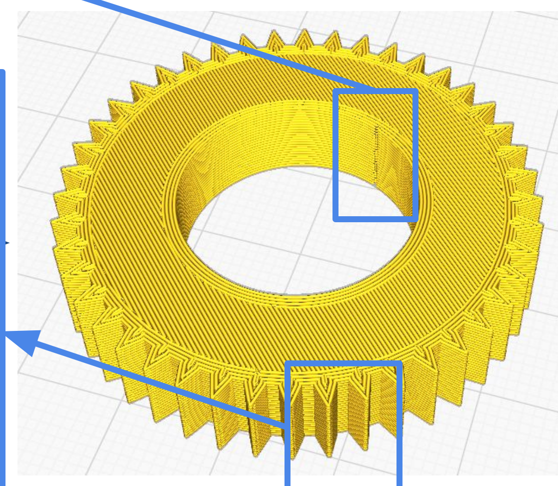


**3D Printing model**
**(Discrete model)**

# Motivation

- 3D Printing

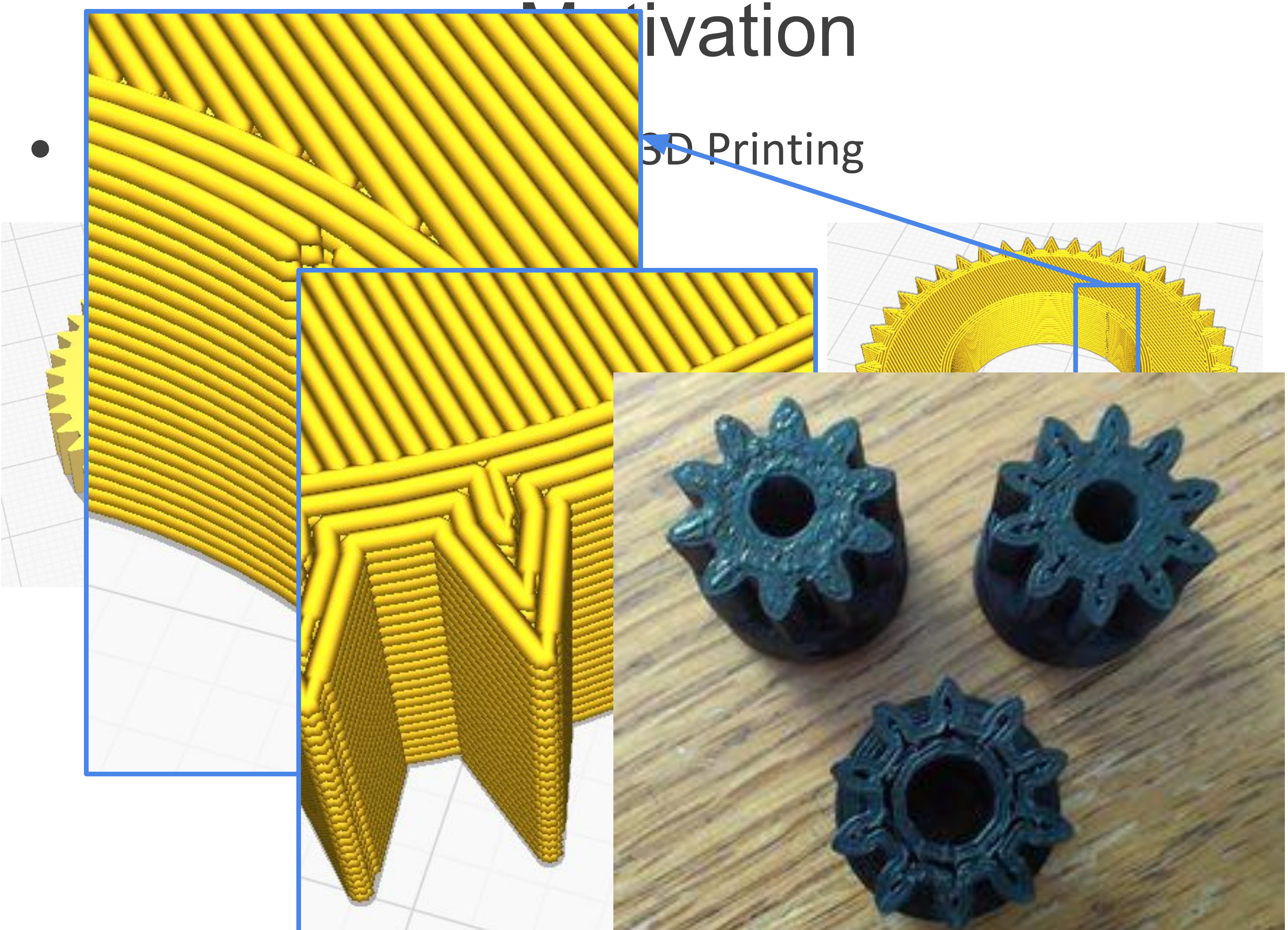




3D Printing model
(Discrete model)

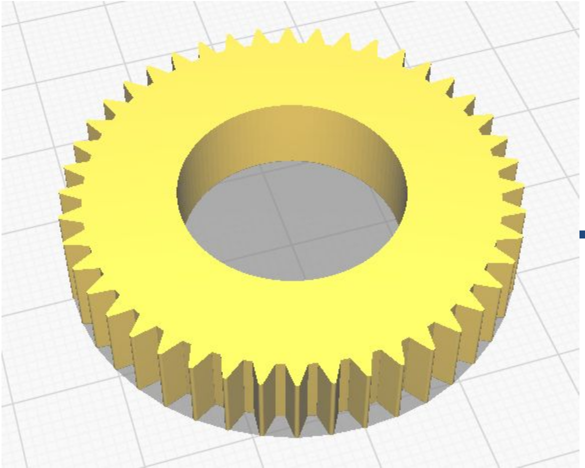
# Motivation

- 3D Printing



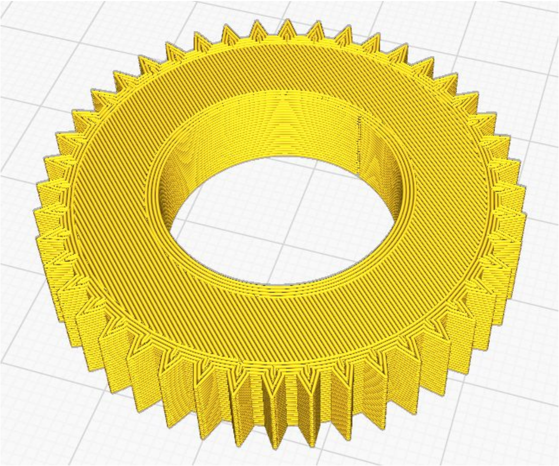
# 3D Printing Functional Parts

- **PROBLEM-2:** Resolution in 3D Printing



3D gear design
(Continuum model)

Slicing
(Output: GCODE )



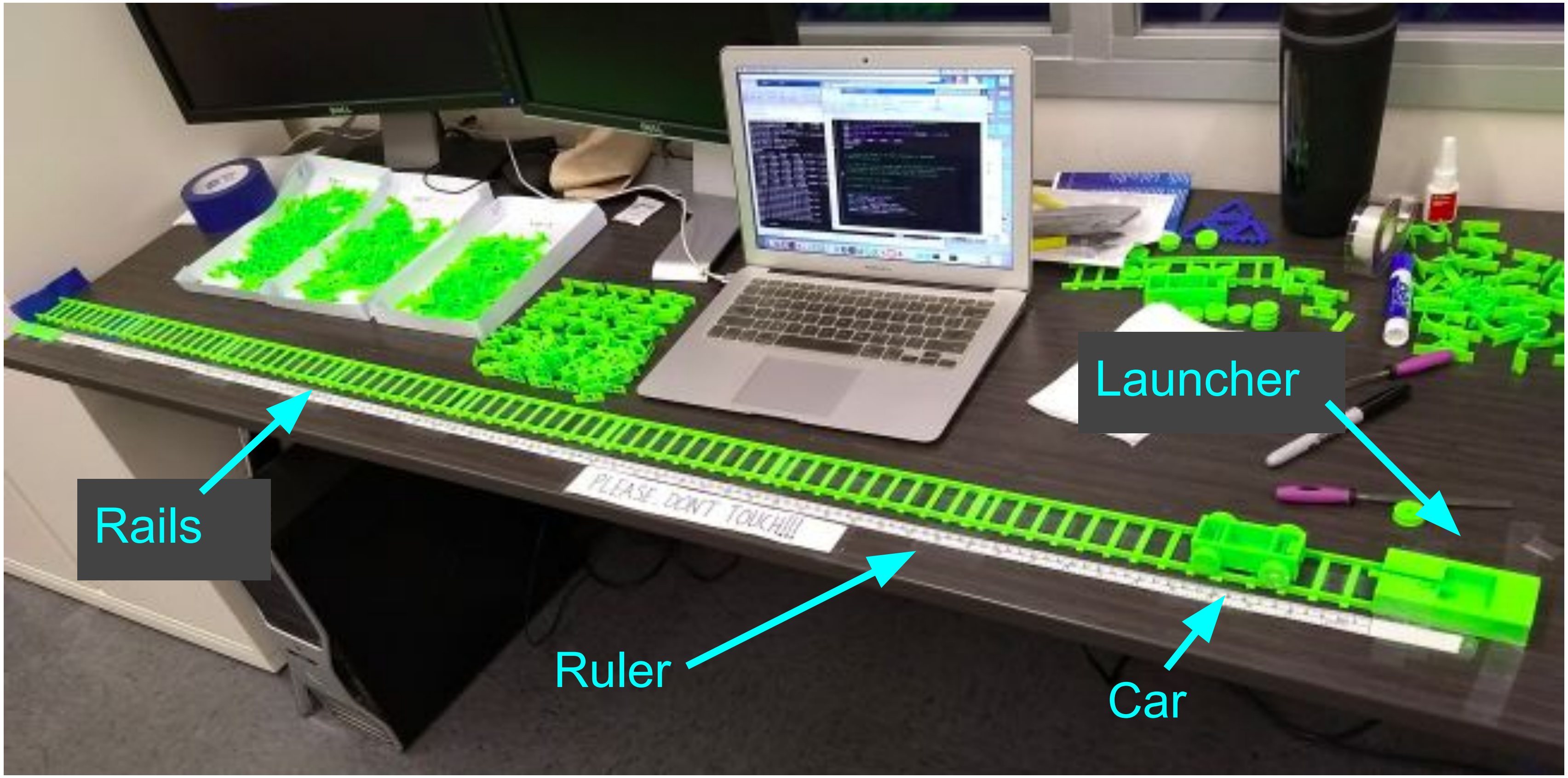
3D Printing model
(Discrete model)

- Effects of 3D printing process?
  - 3D Scanning cannot capture the details,
  - Rendering GCODE model
  - Deformation simulation

~2-3 weeks!

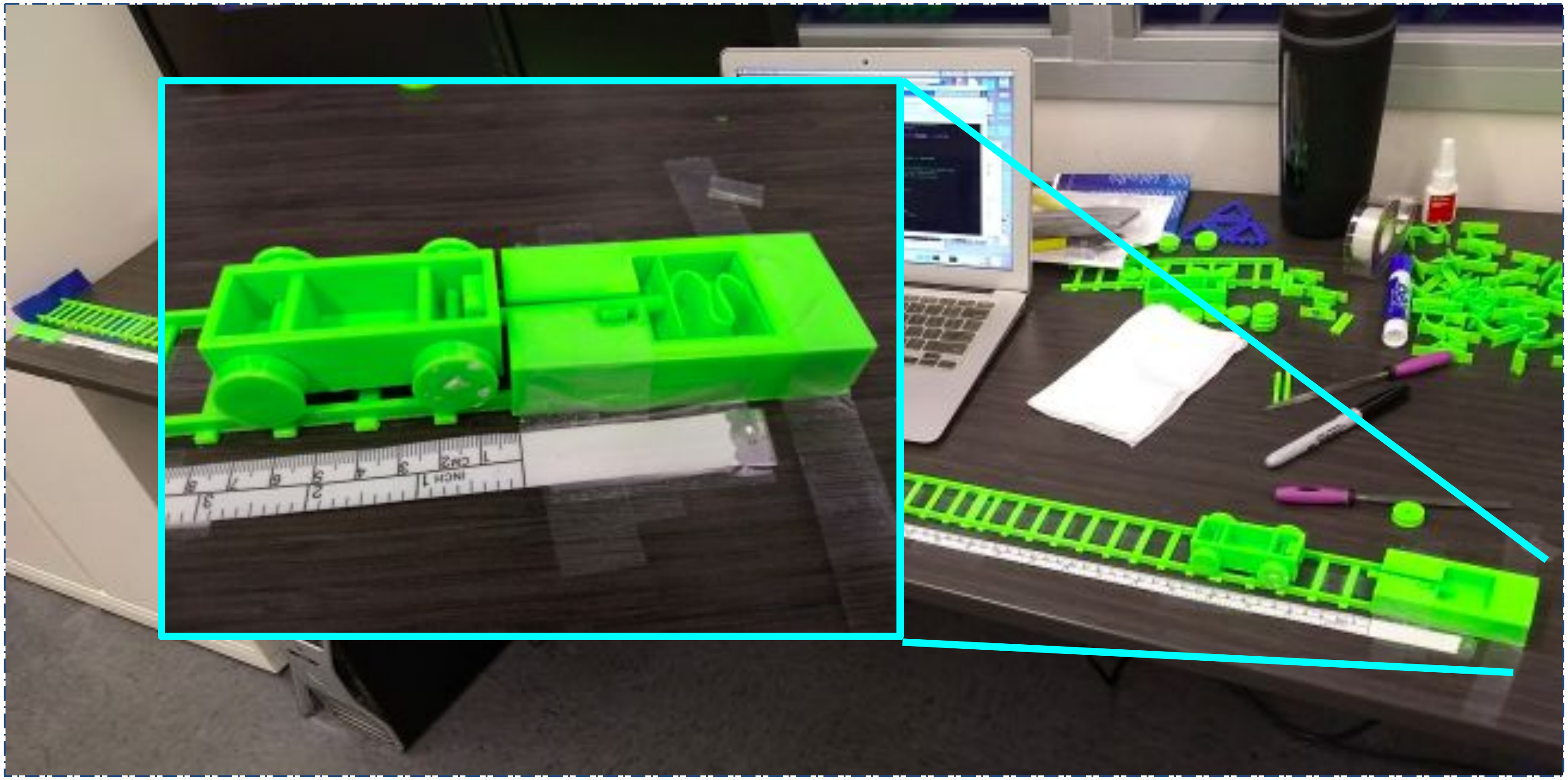
# Motivation

- Car-Launcher Mechanism — as a proof-of-concept



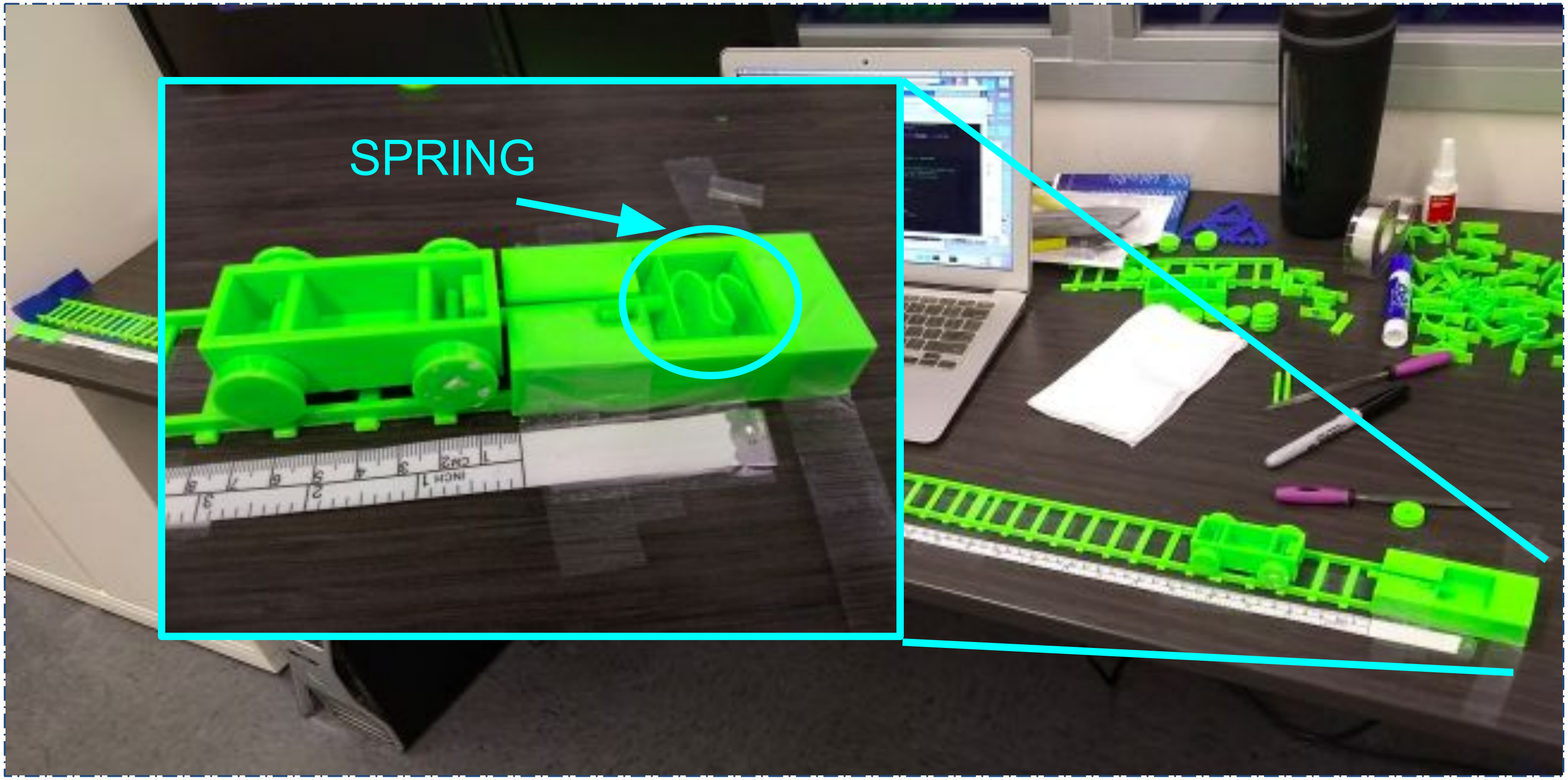
# Motivation

- Car-Launcher Mechanism — as a proof-of-concept



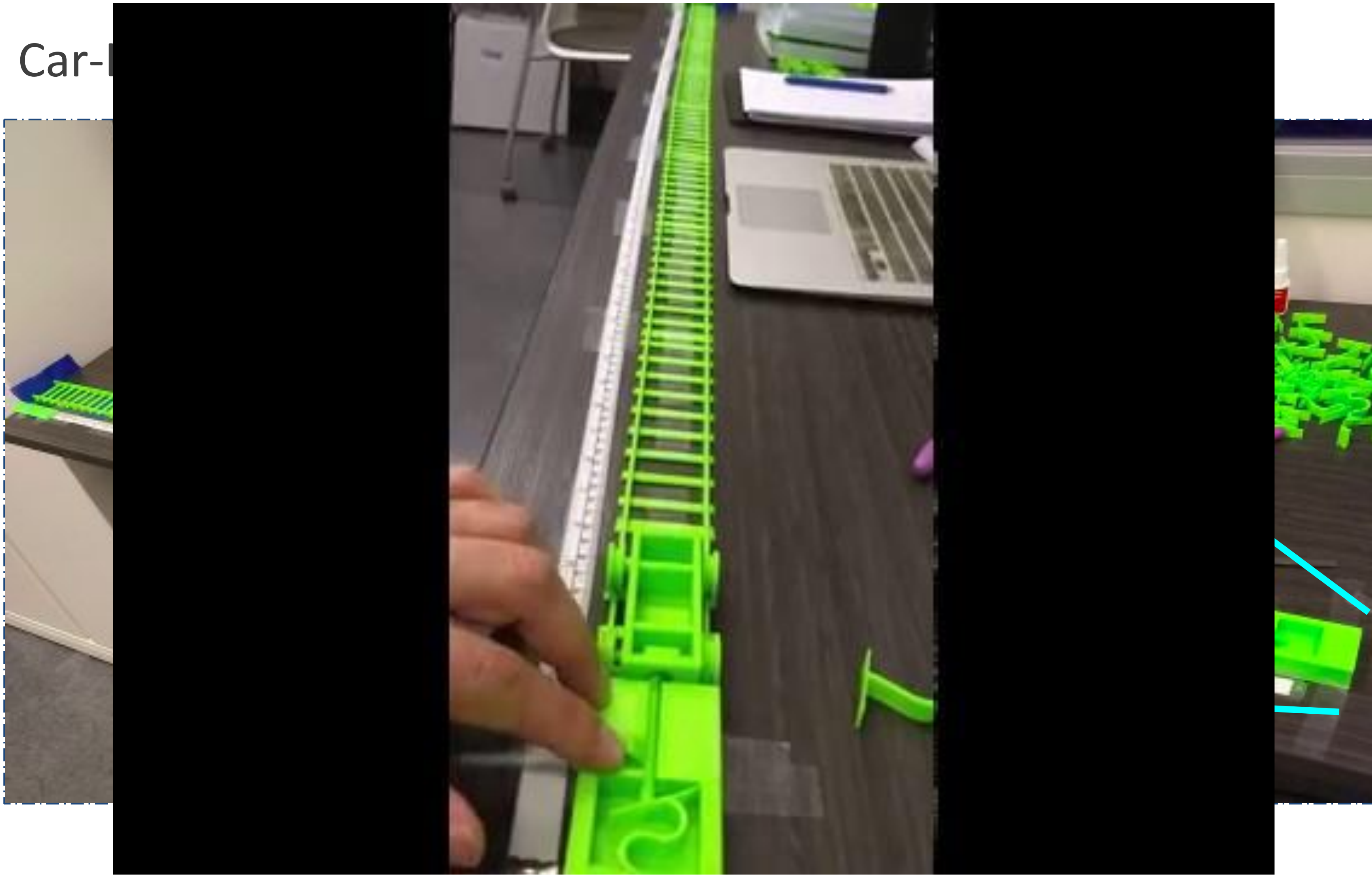
# Motivation

- Car-Launcher Mechanism — as a proof-of-concept



# Motivation

- Car-



# Available Methods

- Several popular design methods:
  - Shape Optimization
    - requires parameterization of the model
  - Topology Optimization
    - compute heavy for nonlinear material & dynamic problems
- Generative (Deep Learning) Methods:
  - Variational Autoencoders (VAEs)
  - Generative Adversarial Networks (GANs)
    - Mostly applied to images or 3D shapes (visual aspects)
    - Why not use their flexibility and creativity for functional design purposes?

# Methodology

- Building blocks:
  - ➢ Variational Autoencoders (VAE)
  - ➢ Noisy (Regressing) Kriging
  - ➢ Efficient Global Optimization (EGO)
  - ➢ Real-parameter Genetic Algorithm (rGA)
- Integration Flowchart

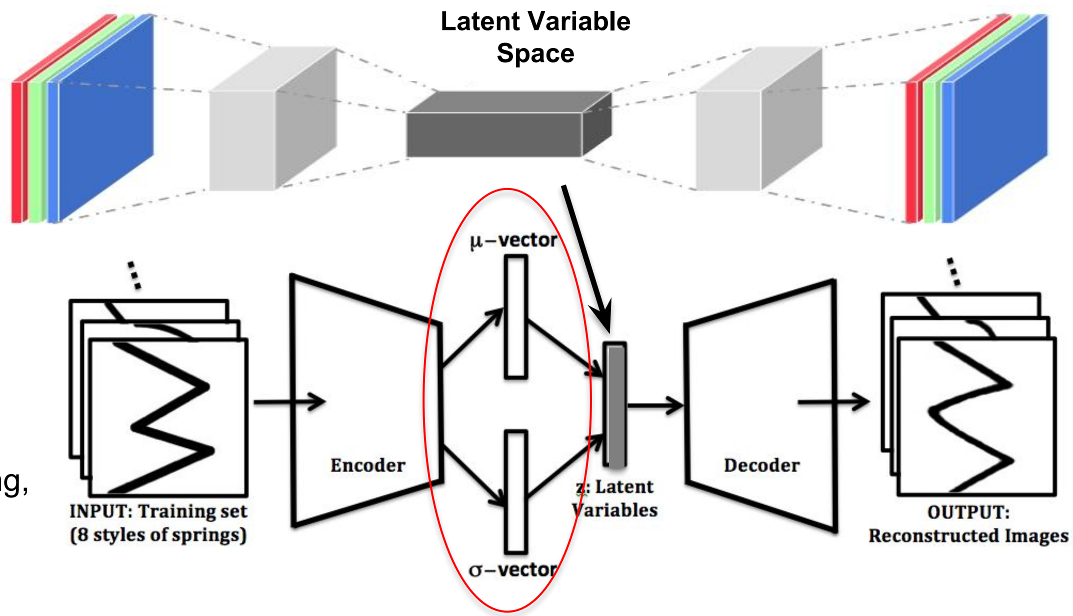
# VAE

- Autoencoder → efficient representation learning

AE →

VAE →

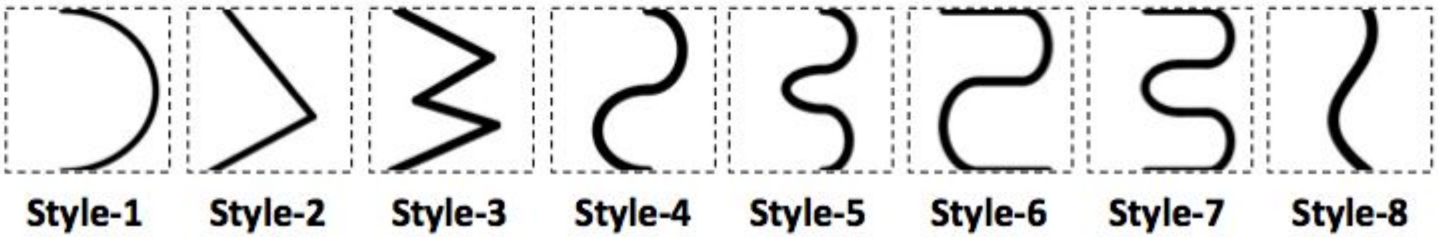
Kingma and Welling, 2014

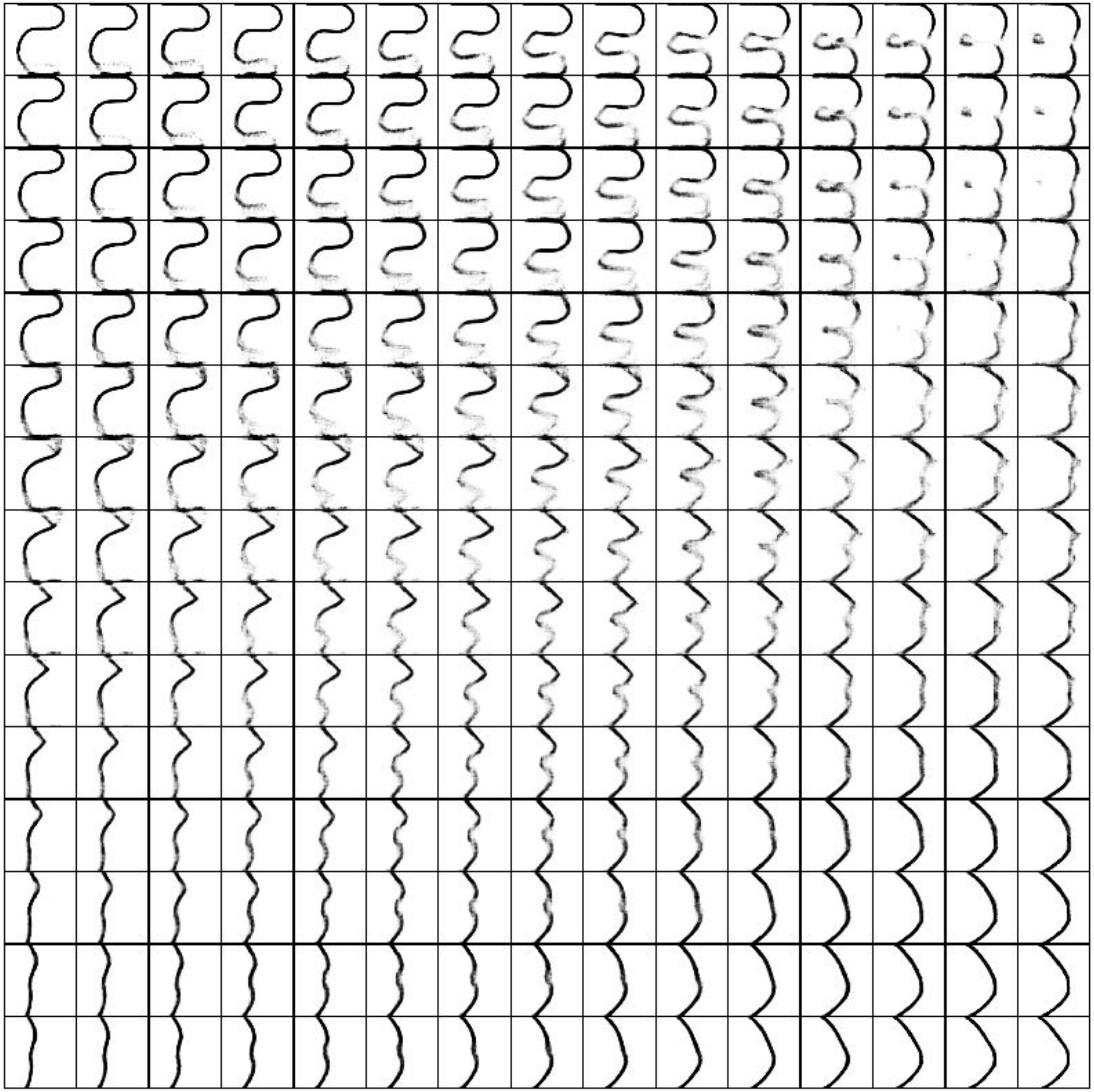



Training Loss =
Recons.Error + KL-divergence

Forces LVs to follow a unit Gaussian distribution

- 8 styles of springs – intuitively chosen

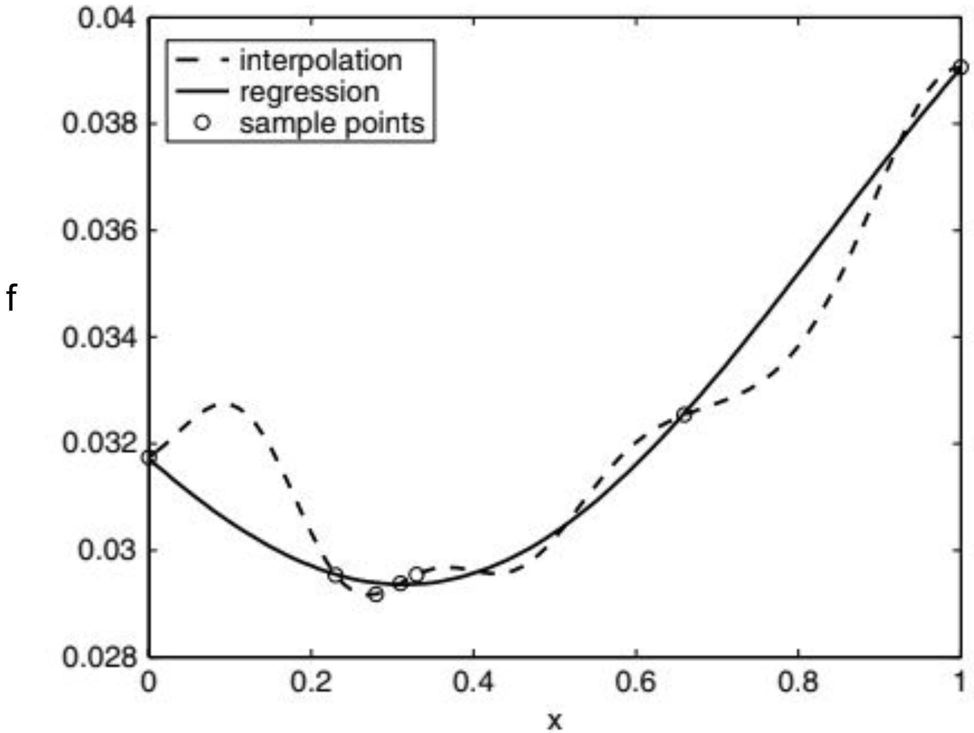




**Interpolation in LV-space**

# Noisy (Regressing) Kriging

- Kriging $\longrightarrow$ surrogate (e.g., function approximator, response surface model)
- Interpolation vs. <u>Regression</u>




Correlation between two points:

$$cor\left[y(\mathbf{x}^{(i)}), y(\mathbf{x}^{(j)})\right] = \prod_{k=1}^{d} exp\left(-\theta_k \left|\mathbf{x}_k^{(i)} - \mathbf{x}_k^{(j)}\right|^2\right)$$

Prediction at new point x*:

$$\hat{y}(\mathbf{x}^*) = \hat{\mu} + \mathbf{r}^T(\mathbf{R} + \lambda\mathbf{I})^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu})$$

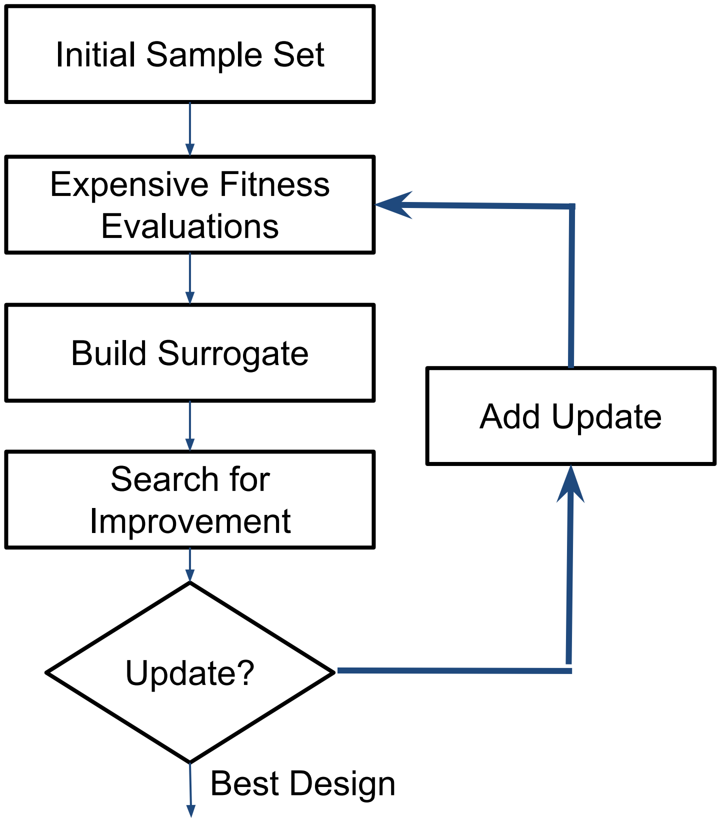
$$\hat{s}^2(\mathbf{x}^*) = \hat{\sigma}^2\left[1 + \lambda - \mathbf{r}^T(\mathbf{R} + \lambda\mathbf{I})^{-1}\mathbf{r} + \frac{1 - \mathbf{1}^T(\mathbf{R} + \lambda\mathbf{I})^{-1}\mathbf{r}}{\mathbf{1}^T(\mathbf{R} + \lambda\mathbf{I})^{-1}\mathbf{1}}\right]$$

Predicted (re-interpolation) error (for EGO):

$$\hat{s_{ri}}^2(\mathbf{x}^*) = \hat{\sigma_{ri}}^2\left[1 - \mathbf{r}^T\mathbf{R}^{-1}\mathbf{r} + \frac{1 - \mathbf{1}^T\mathbf{R}^{-1}\mathbf{r}}{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}}\right]$$

# EGO

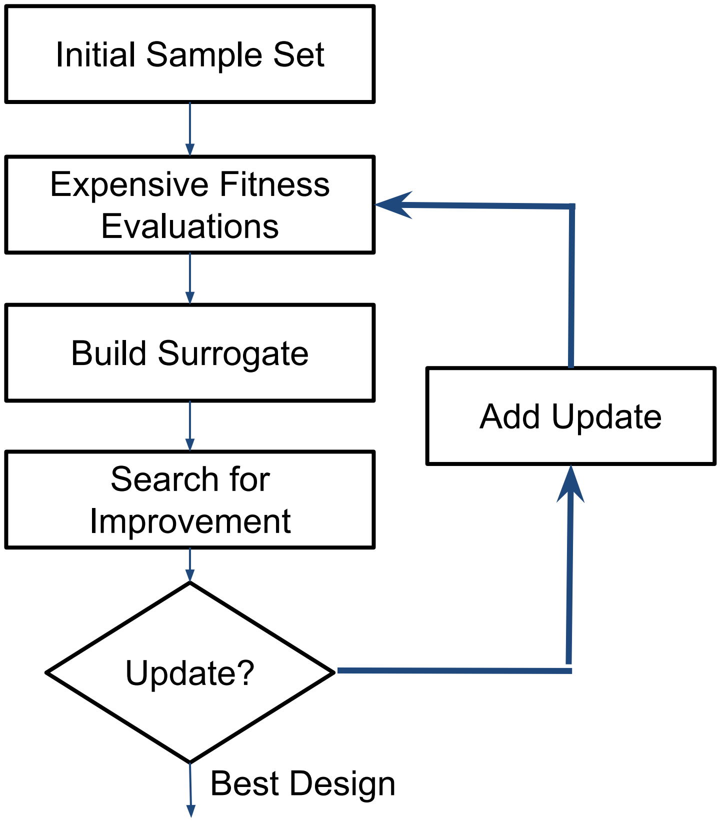
- EGO $\longrightarrow$ surrogate (or model)-based optimizer
- Iteratively updating the surrogate model with promising infill points which maximize Expected Improvement crit.

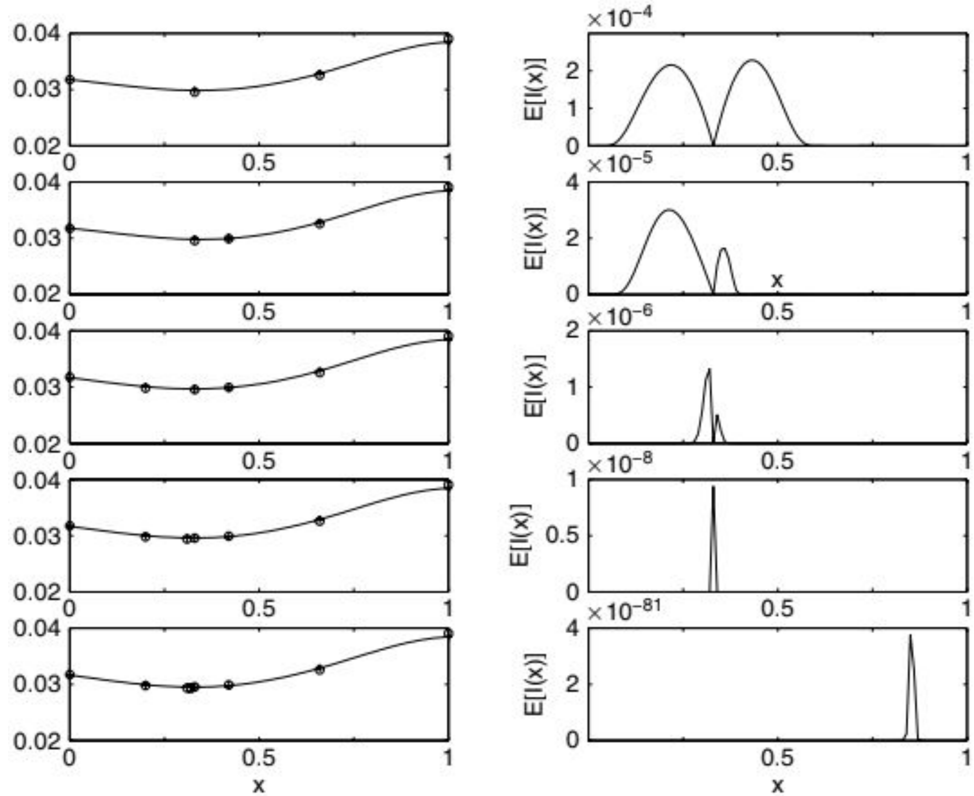



$$E[I(\mathbf{x})] = (y_{best} - \hat{y}(\mathbf{x}))\Phi\left(\frac{y_{best} - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right) + \hat{s}(\mathbf{x})\phi\left(\frac{y_{best} - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right)$$

# EGO

- EGO → surrogate (or model)-based optimizer
- Iteratively updating the surrogate model with promising infill points which maximize <u>Expected Improvement</u> crit.

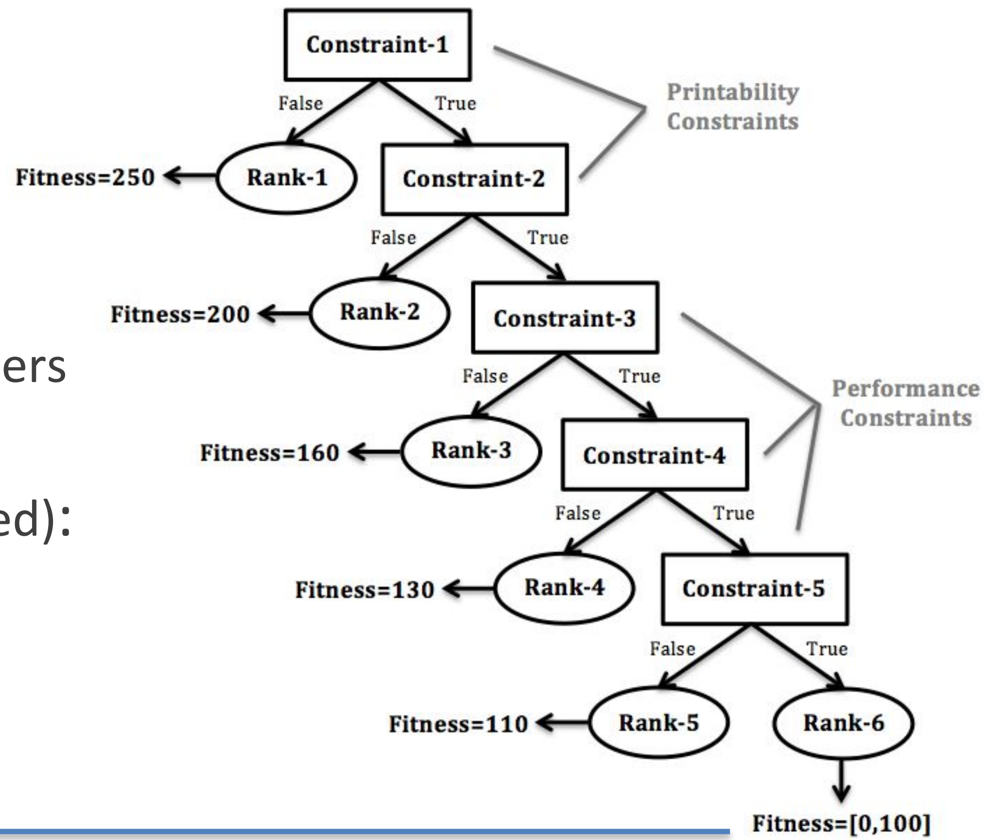


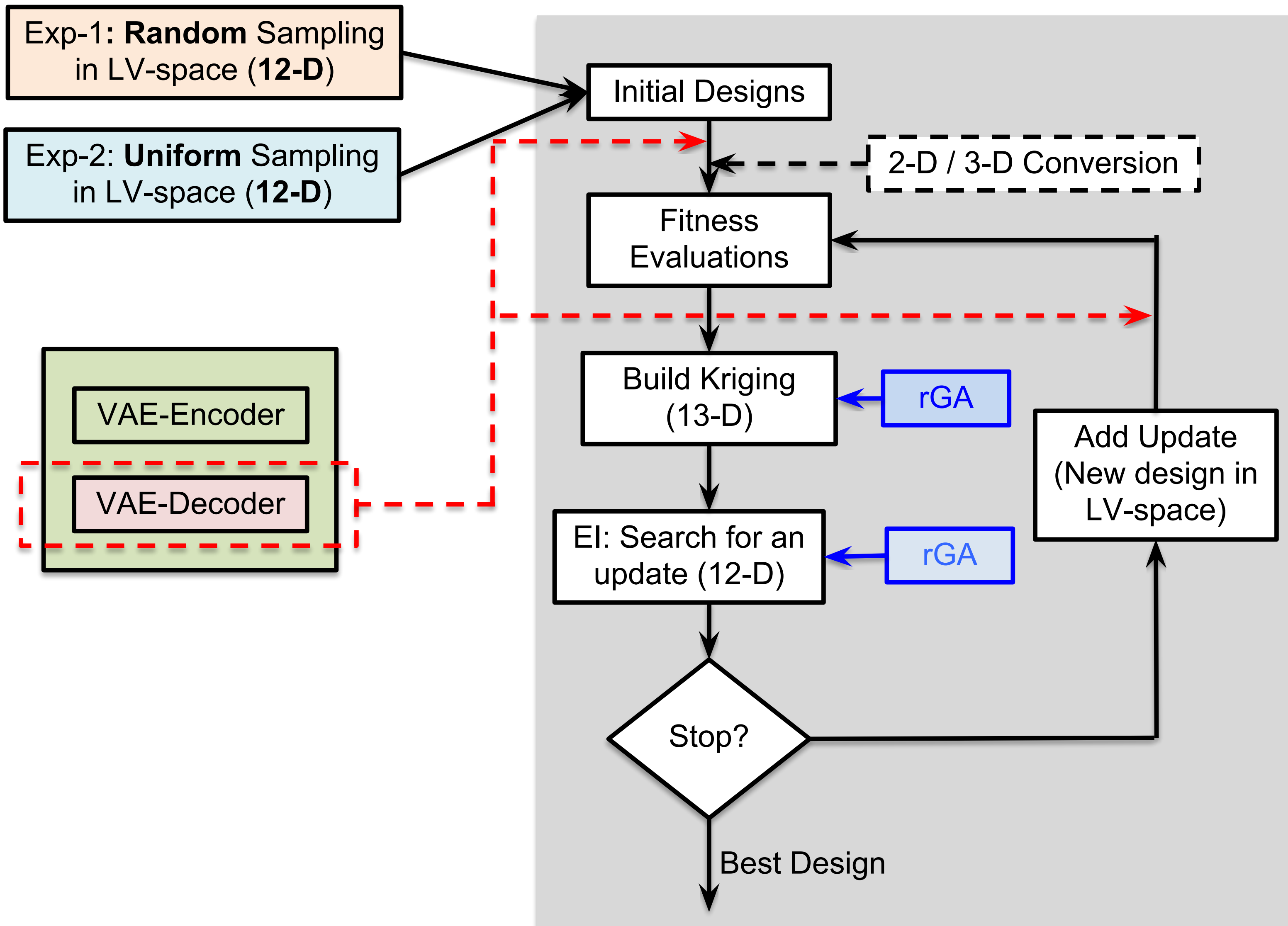
# rGA

- rGA → real parameter Genetic Algorithm
- Operators:
  - Tournament Selection,
  - α-Blend Xover,
  - Gaussian Mutation
- Its roles:
  - Tuning Kriging hyperparameters
  - Maximizing EI in EGO
- Fitness evaluation (Normalized):

$$f(x) = \sum_{i=1}^{n_{exp}} MSE_i = \frac{1}{10}\sum_{i=1}^{10} |d_i - 75|^2,$$



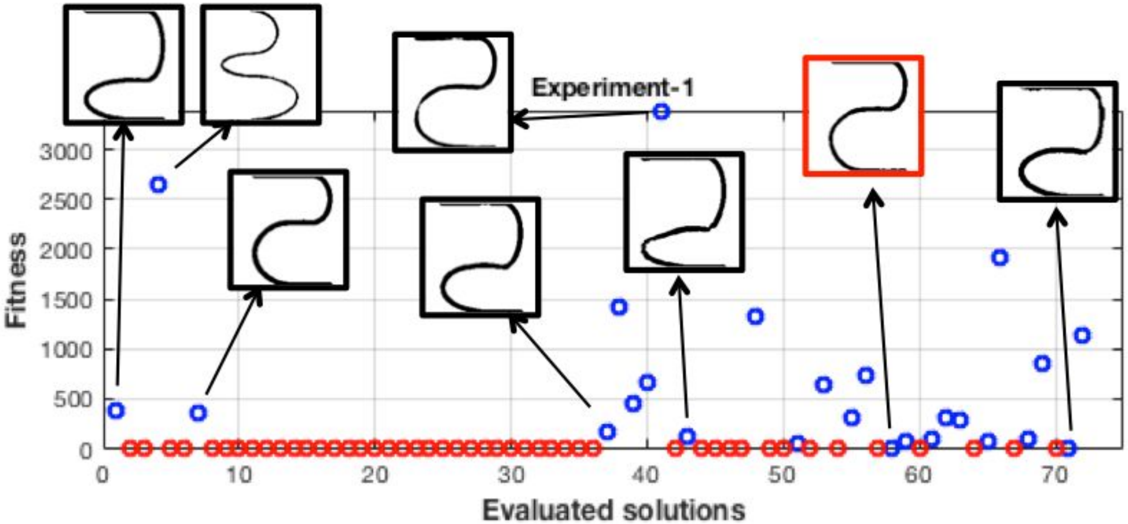
# Integrated Method

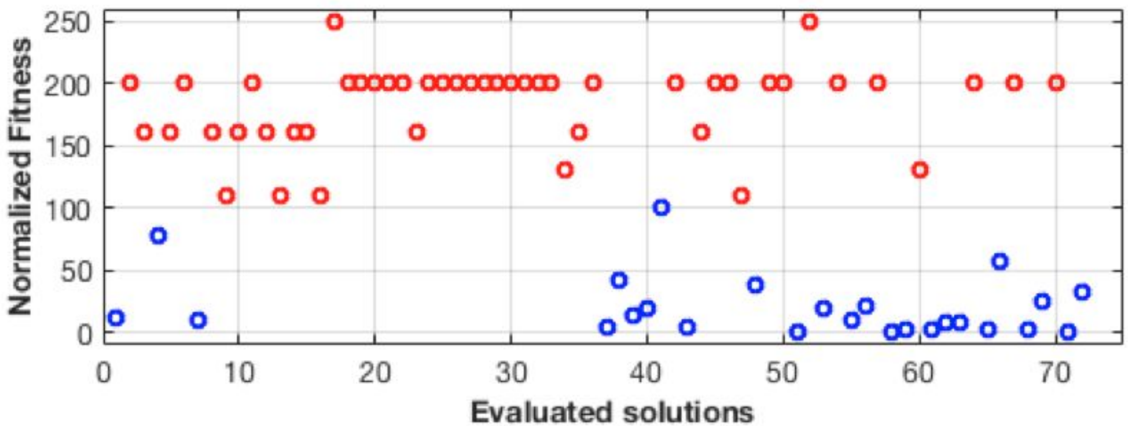


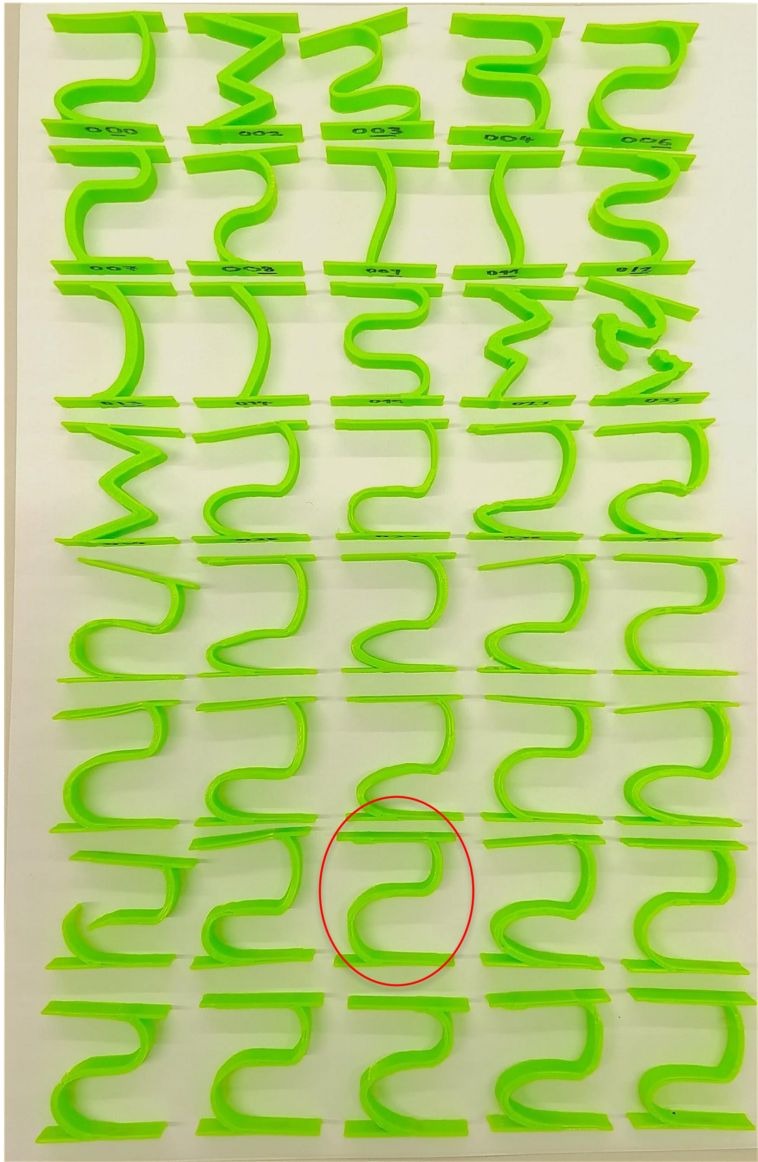
# Experiments

- Total of 72 fitness evaluations
- 36 initial samples = 16 + 20
- Two experiments for choosing those 16 designs
  - Exp-1: Random Sampling
  - Exp-2: Uniform Sampling
- 20 designs by Normal distribution
- Rest (36) are incrementally chosen by EGO

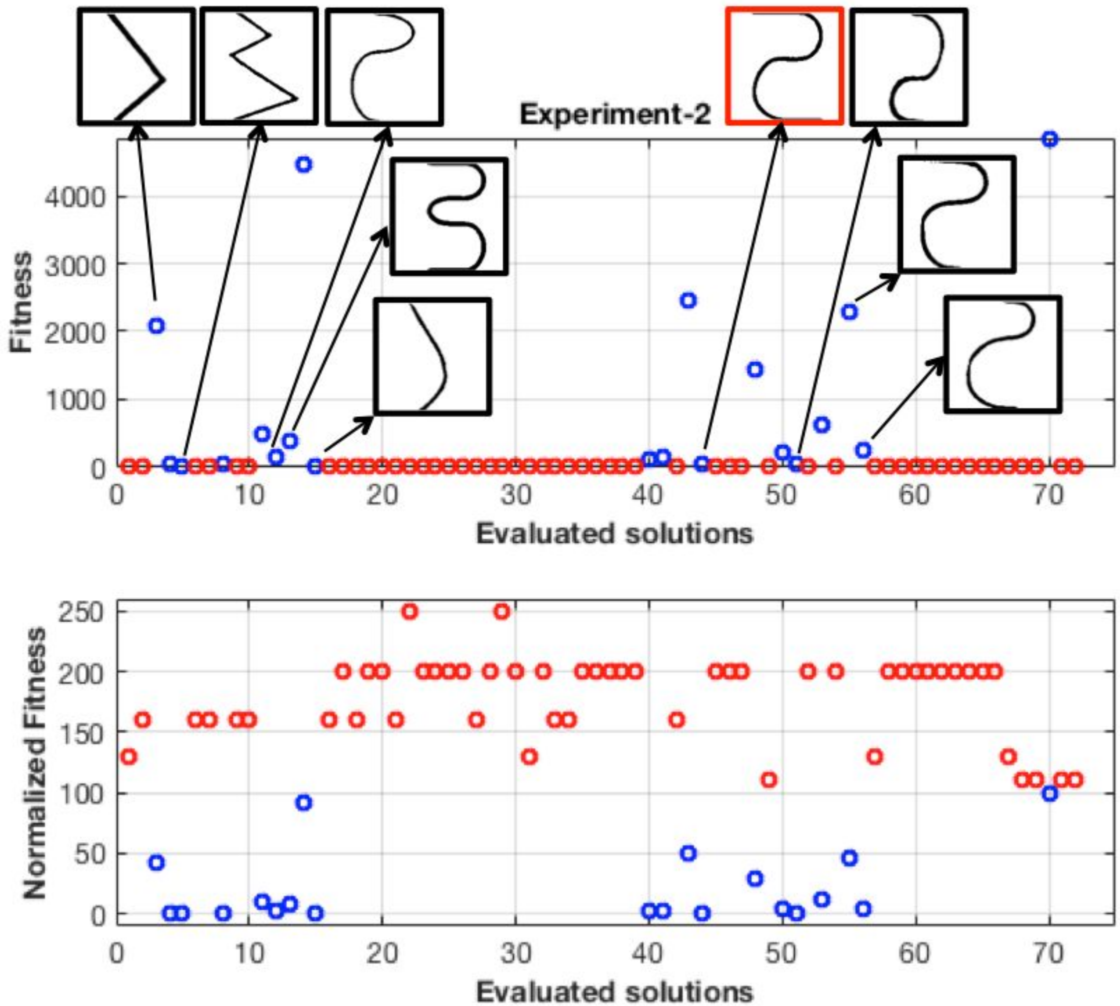
# Exp-1 Results

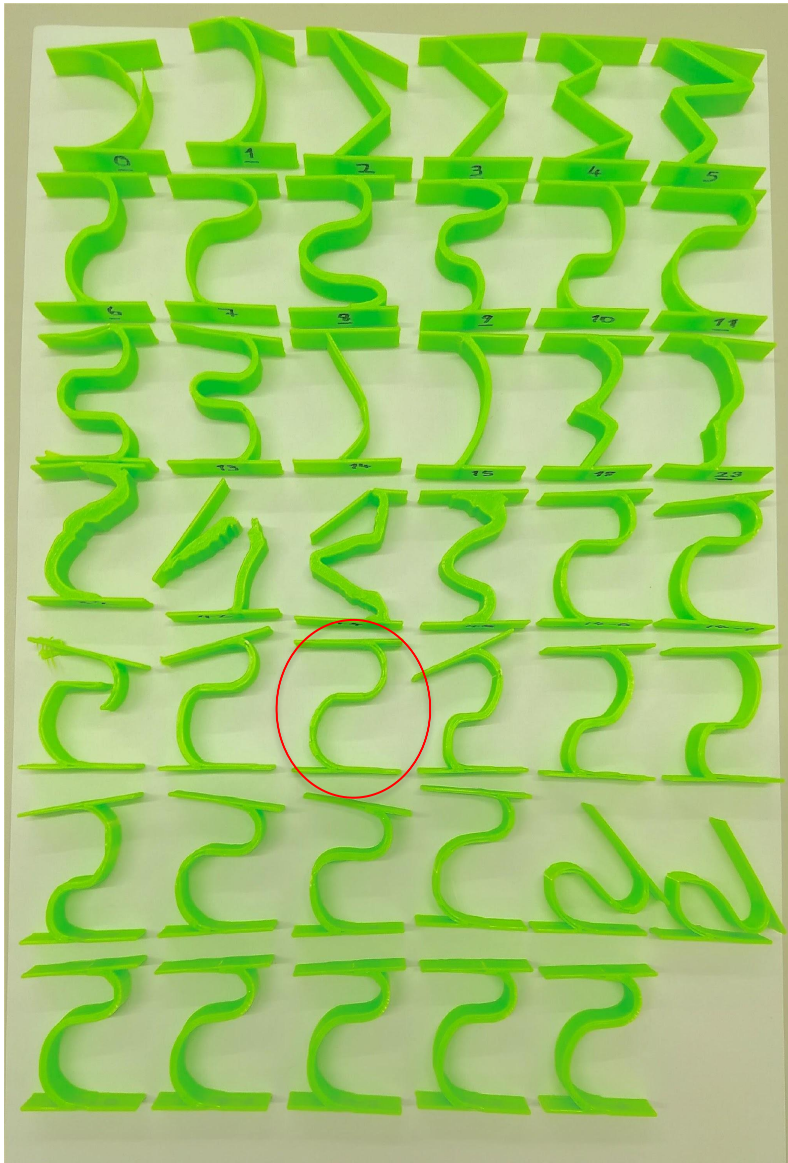


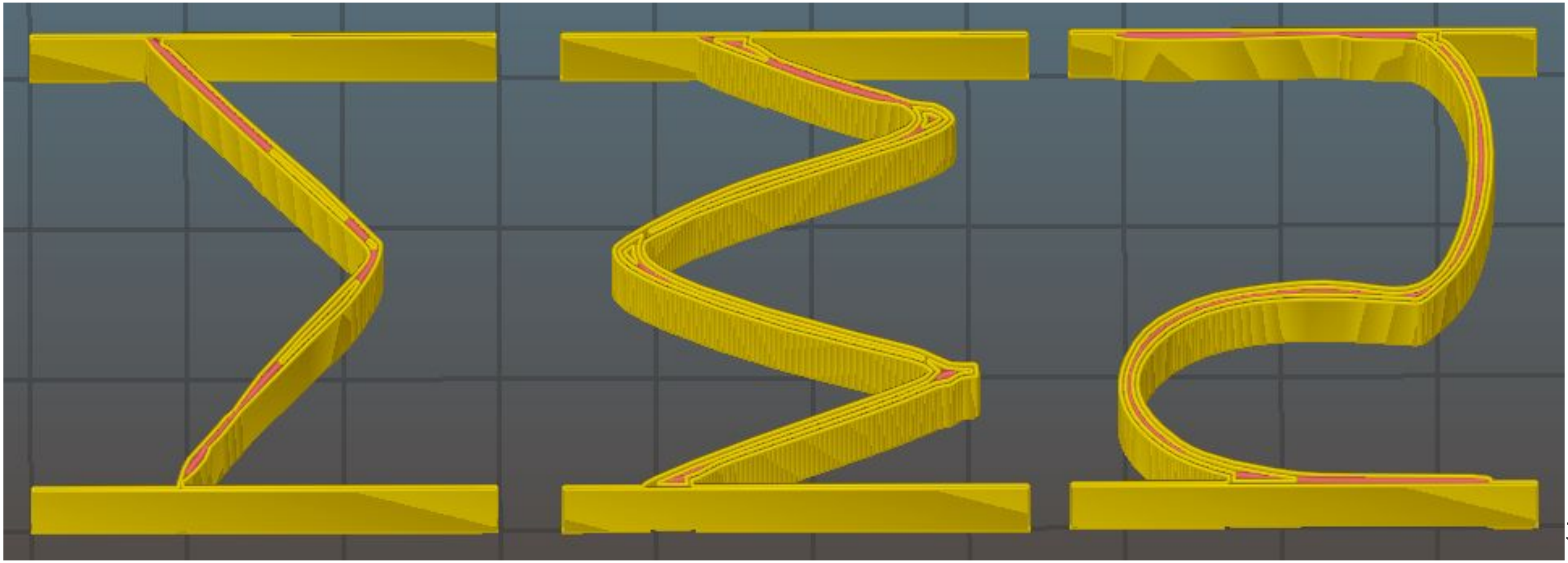
# Exp-2 Results



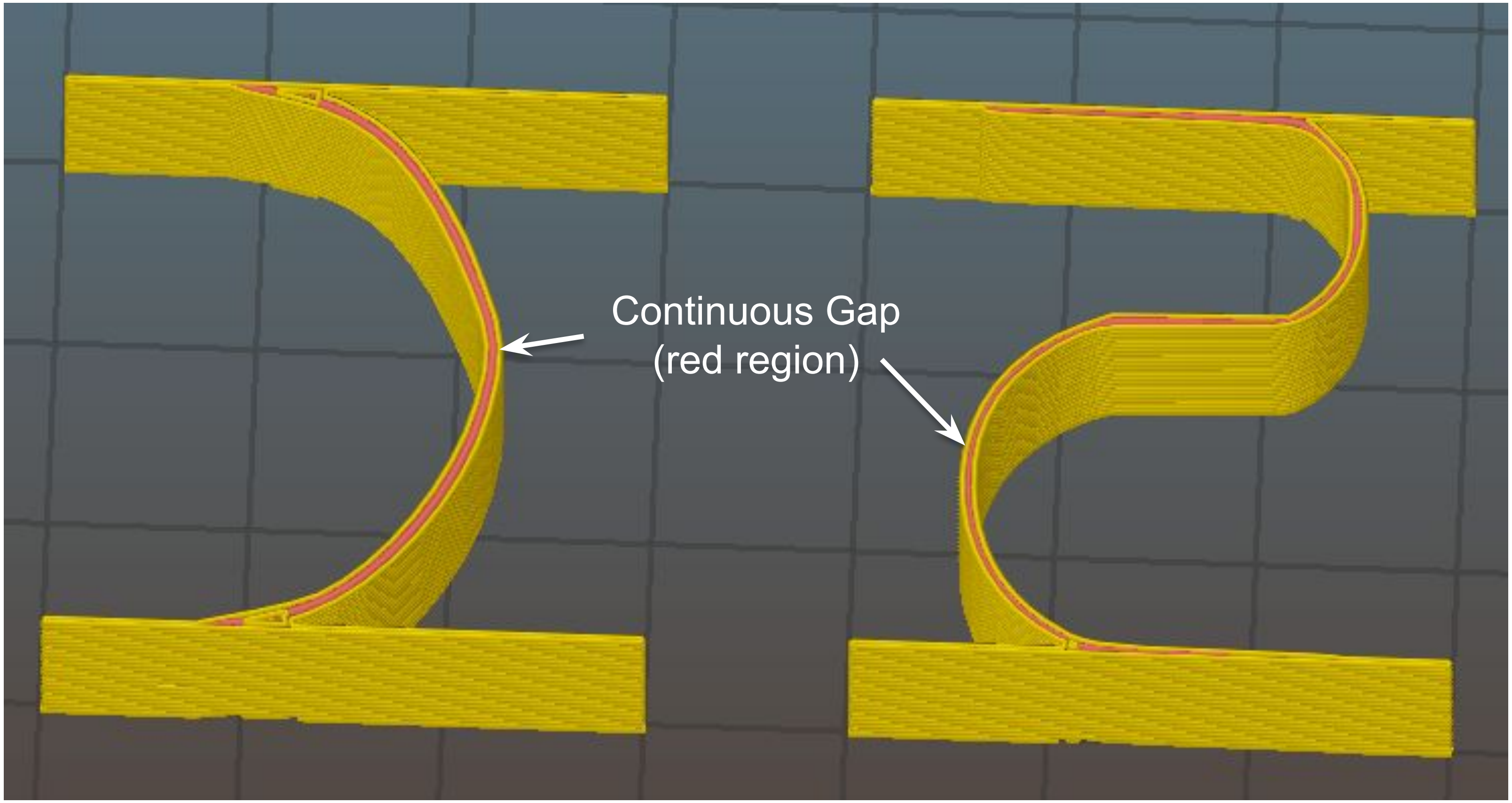
# Discussion

- Exp-1 ⟶ Exploitation
- Exp-2 ⟶ Exploration
- Similar optimal designs
- Gaps cause fracture or act like a hinge!



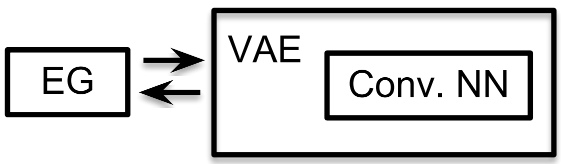
# Discussion

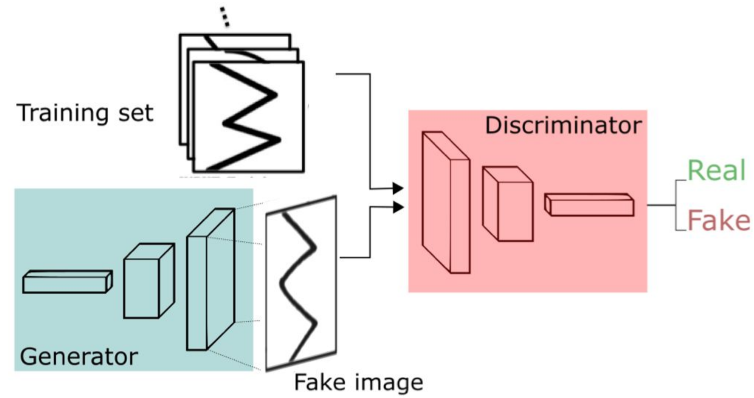
- Gaps are not always bad: Double-line print!



# Future Work

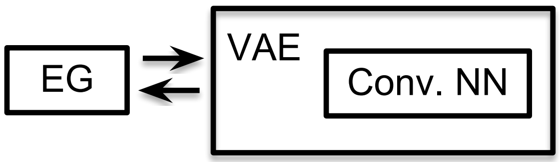
- Hyperparameter optimization for VAE, or GAN

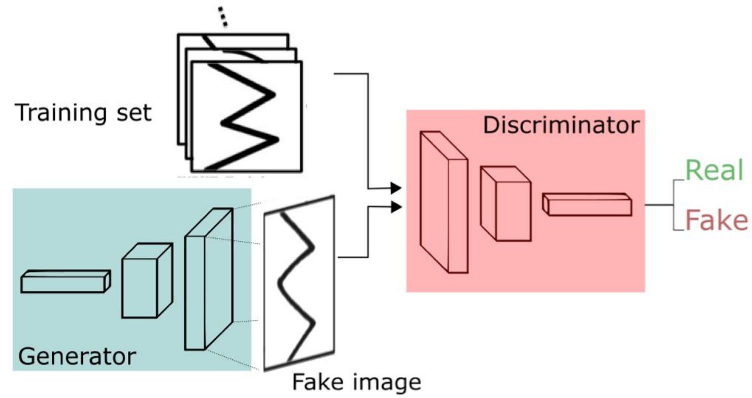


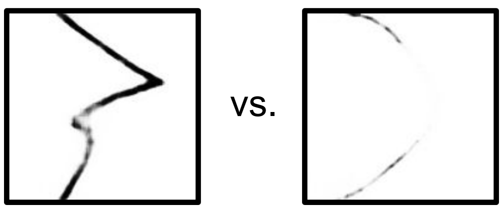
# Future Work

- Hyperparameter optimization for VAE, or GAN



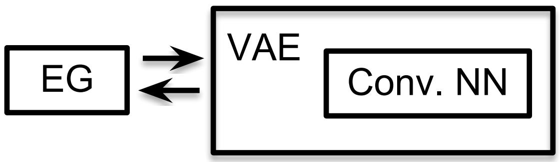



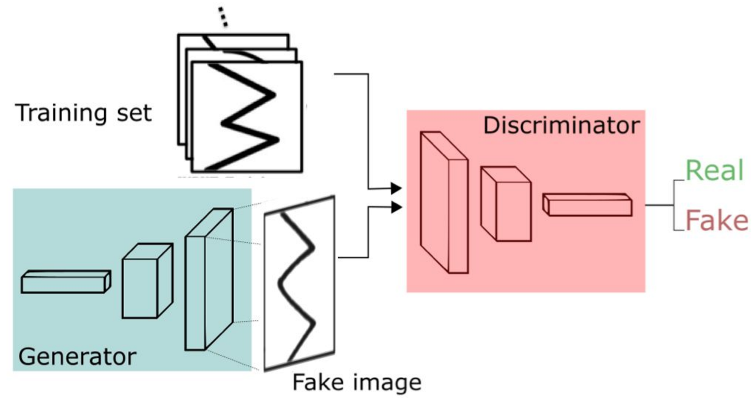

- Quantification of infeasibility



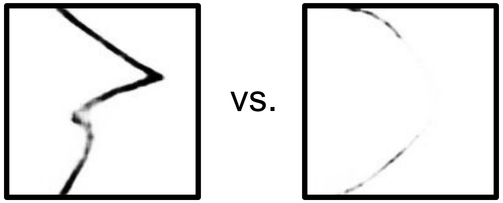
# Future Work

- Hyperparameter optimization for VAE, or GAN



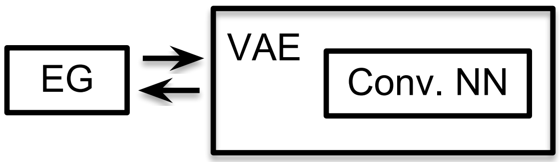

- Quantification of infeasibility

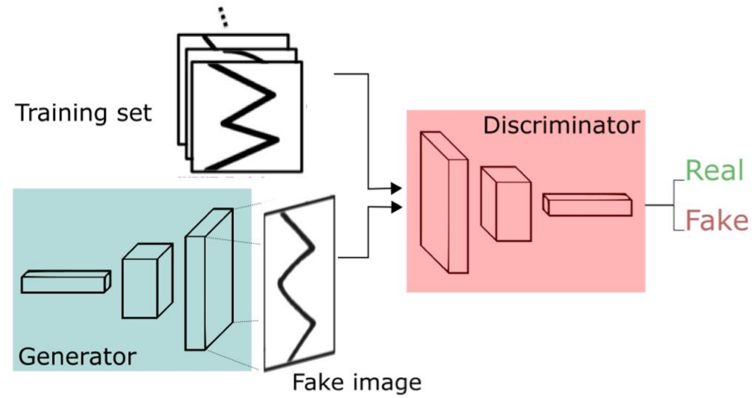



- Repair hallucinations

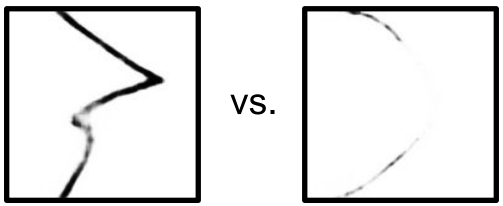
# Future Work

- Hyperparameter optimization for VAE, or GAN




- Quantification of infeasibility




- Repair hallucinations

- More design applications

# Conclusions

- Methodology: Known ingredients, new recipe
- Design + Manufacturing → Functional Performance
- Successful Generative method for:
  - Complex design problem
  - Limited #fitness evaluations
  - Losses
    - Reconstruction (encoder-decoder)
    - Production (Slicing & 3D Printing process)
  - Many missing values
  - Noisy landscape
- Clever use of gaps

THANK YOU!

Any Questions?