Generative Adversarial Networks
Part II

Sicong Liu

Content

- Techniques try to make GAN more stable
- Discussion
- My toy GAN experiment

Generative Adversarial Nets

- Using Dropout in D(x)
- Using maxout nonlinear activation function.

Deep Convolutional Generative Adversial Networks (DCGAN)

Identified a family of architectures that result in stable training.

1. all convolutional net : using strided convolutions for upsampling (for Generator) and downsampling (for Discriminator) instead of using pooling.

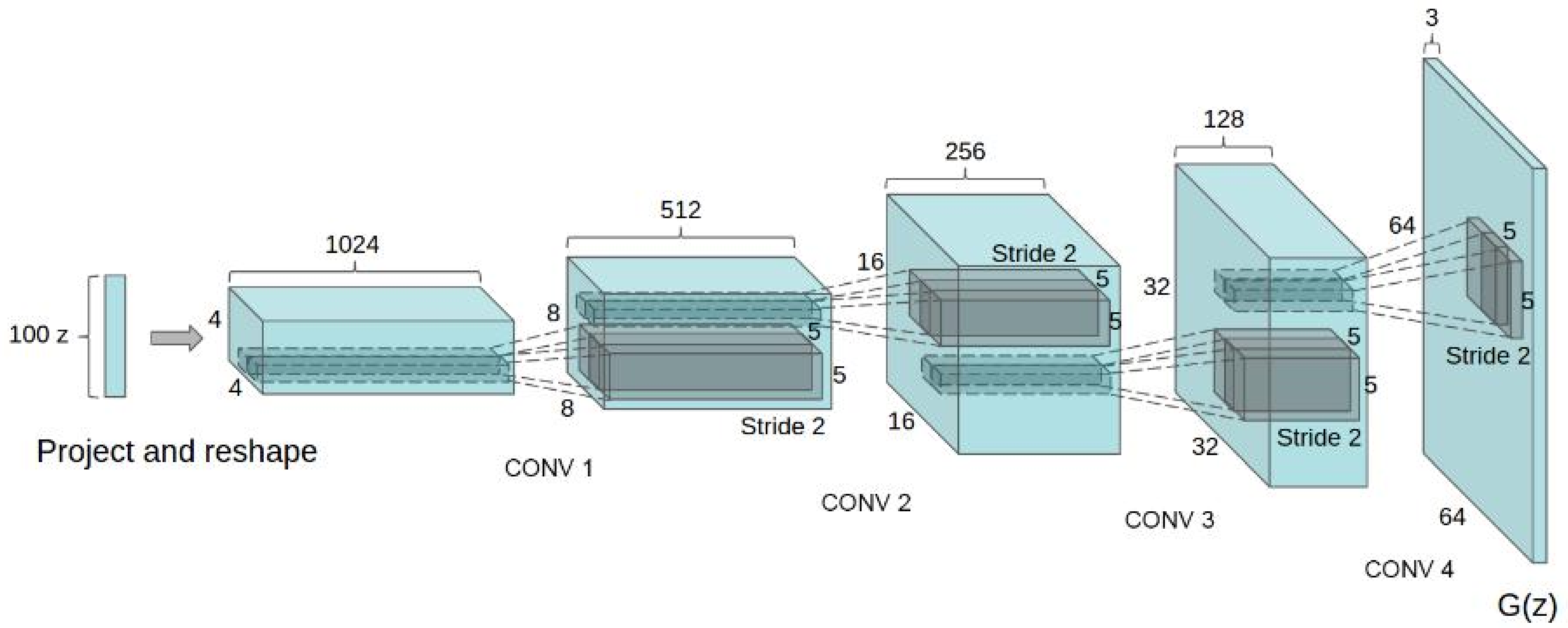
Deep Convolutional Generative Adversial Networks (DCGAN)

2. eliminating fully connected layers.

    2.1. global average pooling: increase model stability but hurt convergence speed. (not be used here)

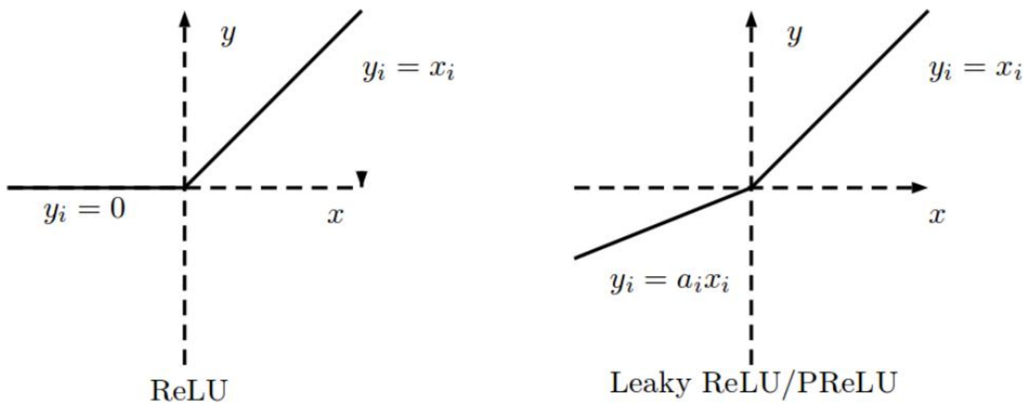
    2.2. remove fully connected hidden layers

Deep Convolutional Generative Adversial Networks (DCGAN)



Deep Convolutional Generative Adversial Networks (DCGAN)

3. Batch Normalization.
4. ReLU in Generator except for the output, which uses Tanh
5. LeakyReLU in Discriminator
6. Adam Optimizer



Improved Techniques for Training GANs

1. Feature matching
2. Minibatch discrimination
3. Historical averaging
4. One-sided label smoothing
5. Virtual batch normalazation

1.Feature matching

Motivation: prevents Generator overtraining on the current discriminator.

Letting f(x) denote activations on an intermidiate layer of the discriminator. The new object is:

$$||\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}\mathbf{f}(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\mathbf{f}(G(\boldsymbol{z}))||_2^2$$

2.Minibatch discrimination

Motivation: Deal with the collapse problem of the model.

Collapse: the generator always emits the same point. No mechanism to tell the outputs of generator to become more dissimilar to each other.

Strategy: allow the discriminator to look at multiple data examples in combination.

2.Minibatch discrimination

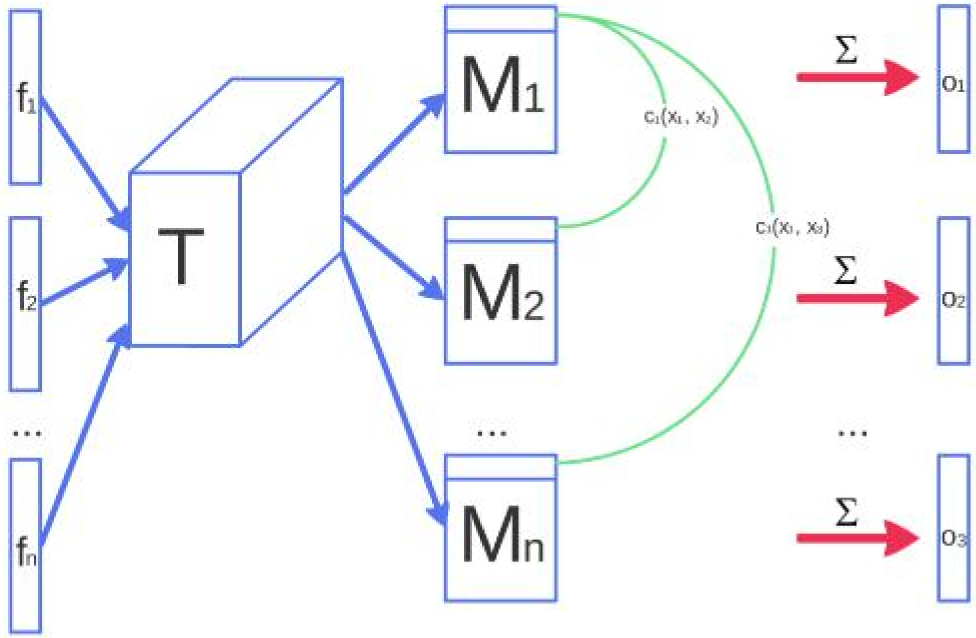
$$\mathbf{f}(\boldsymbol{x}_i) \in \mathbb{R}^A$$
$$T \in \mathbb{R}^{A\times B\times C}$$
$$M_i \in \mathbb{R}^{B\times C}$$

$$c_b(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-||M_{i,b} - M_{j,b}||_{L_1}) \in \mathbb{R}$$

$$o(\boldsymbol{x}_i)_b = \sum_{j=1}^{n} c_b(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathbb{R}$$

$$o(\boldsymbol{x}_i) = \Big[o(\boldsymbol{x}_i)_1, o(\boldsymbol{x}_i)_2, \ldots, o(\boldsymbol{x}_i)_B\Big] \in \mathbb{R}^B$$

$$o(\mathbf{X}) \in \mathbb{R}^{n\times B}$$



3.Historical averaging

A object for both for Generator and discriminator

Motivation: Loosely inspired by the fictious play algorithm (?) that can find equilibria in other kind of games.

$$||\boldsymbol{\theta} - \frac{1}{t}\sum_{i=1}^{t}\boldsymbol{\theta}[i]||^2$$

4. One-sided label smoothing

Label smoothing, a technique shown to reduce the vulnerability of neural networks to adversarial examples.

Replace positive classification targets with α and negative target with β. [Label smoothing]

a batch target [1, 0, 1, 0] will be replaced by [α, β, α, β].

4. One-sided label smoothing

The optimal discriminator becomes $\frac{\alpha p_{\text{data}}(\boldsymbol{x})+\beta p_{\text{model}}(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x})+p_{\text{model}}(\boldsymbol{x})}$

But, here, only set positive targets to be α, leaving negative labels set to 0. [One-sided label smoothing]

5. Virtual batch normalization

Motivation: Batch normalization causes the output for x highly dependent on several other x' in the same minibatch.

Virtual Batch normalization: x is normalized based on the statistics collected on a reference batch of examples that are chosen once and fixed at the start of trainning, and on x itself.

computationally expensive (2 mini-batch examples in forward propagtion). So only used on Generator.

Discussion

- Should we use dropout or global average pooling?
- What make GAN training unstable?
- How to evaluate GANs' performance?

How to evaluate GANs' performance?

- Human annotators
- the Inception Score

A pretrained Inception model (kind of CNN) $p(y|x)$

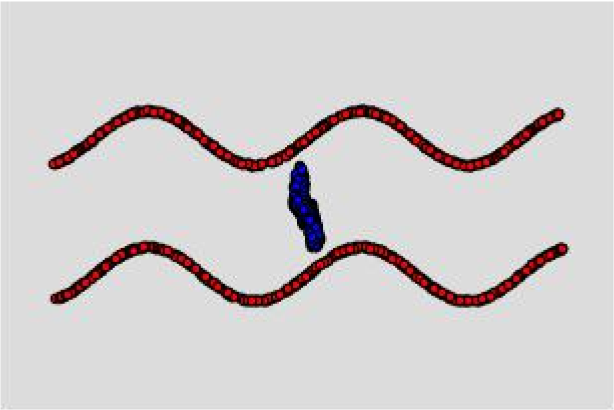
1. $p(y|x)$: entroy should be small
2. $\int p(y|x = G(z))d(z)$: entroy should be large

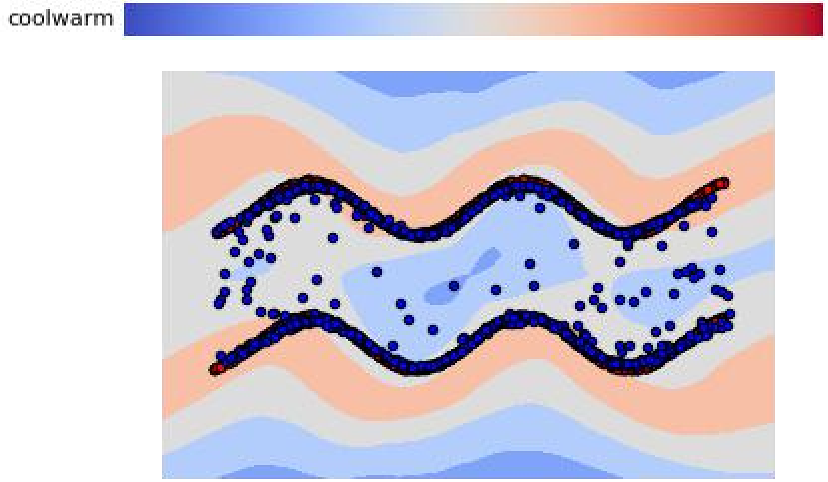
Proposed: $\exp(\mathbb{E}_{\boldsymbol{x}} \mathrm{KL}(p(y|\boldsymbol{x}) || p(y)))$

The Inception Score correlates very well with human judgment.

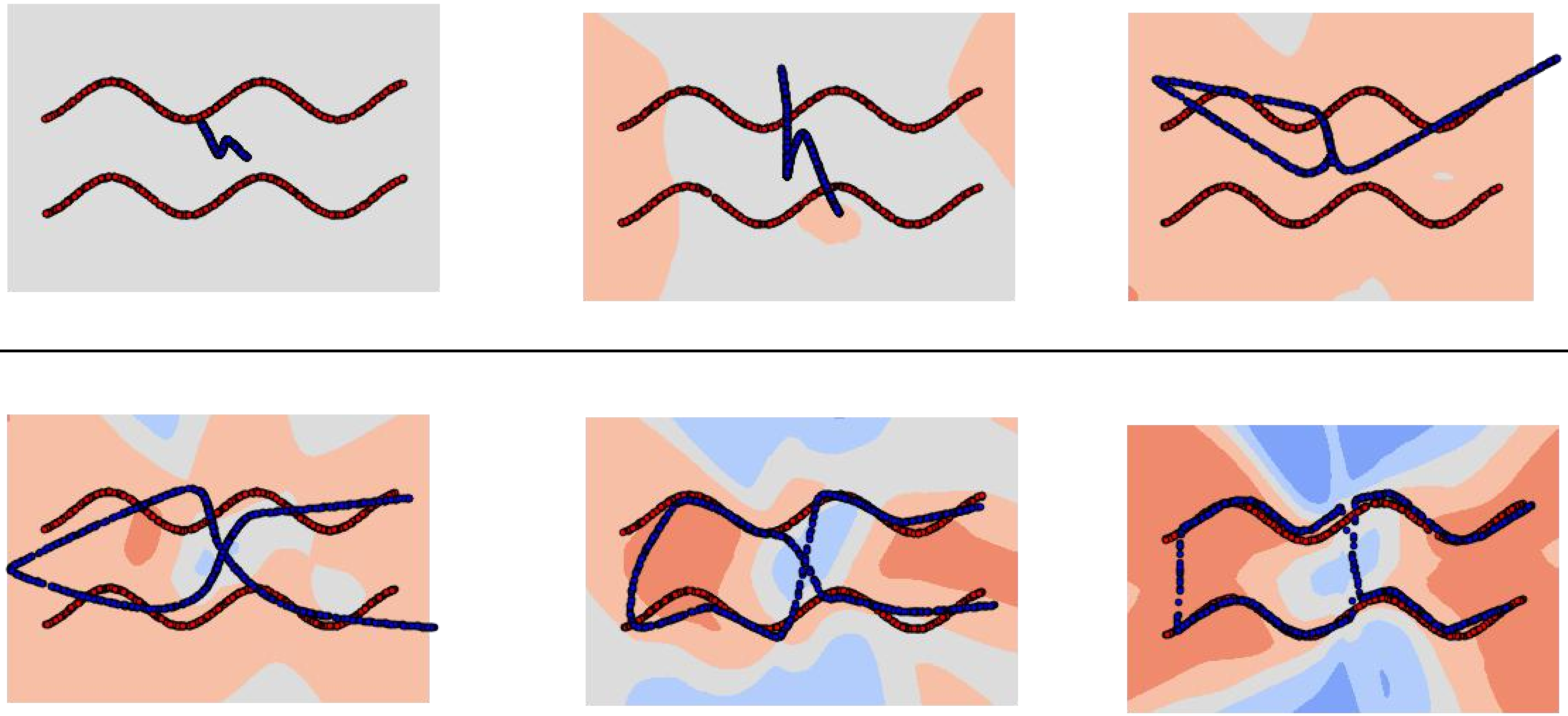
My toy GAN experiment

- real example (Using some y=sin(wx+b))
- fake example (by (x,y) = Generator(z))
- Genetator and Discriminator both fully connected neural networks.

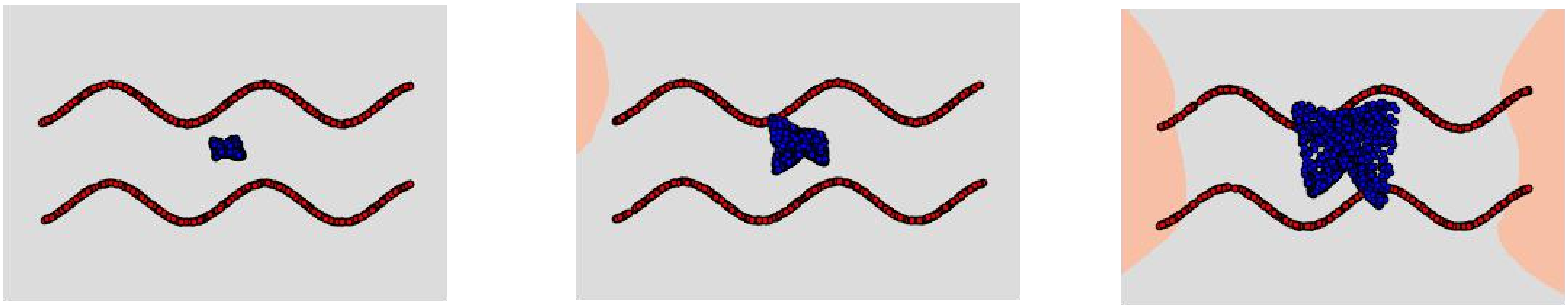


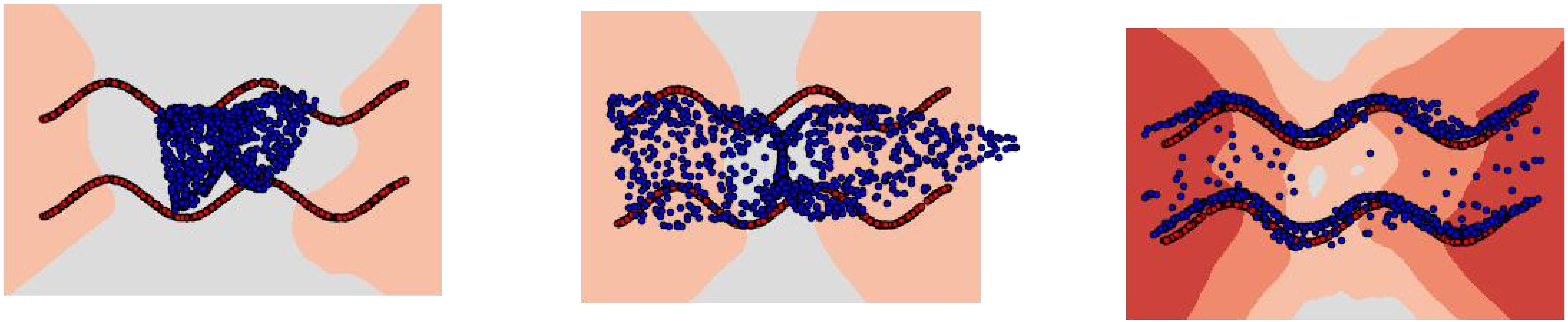


dimension of latent variable z

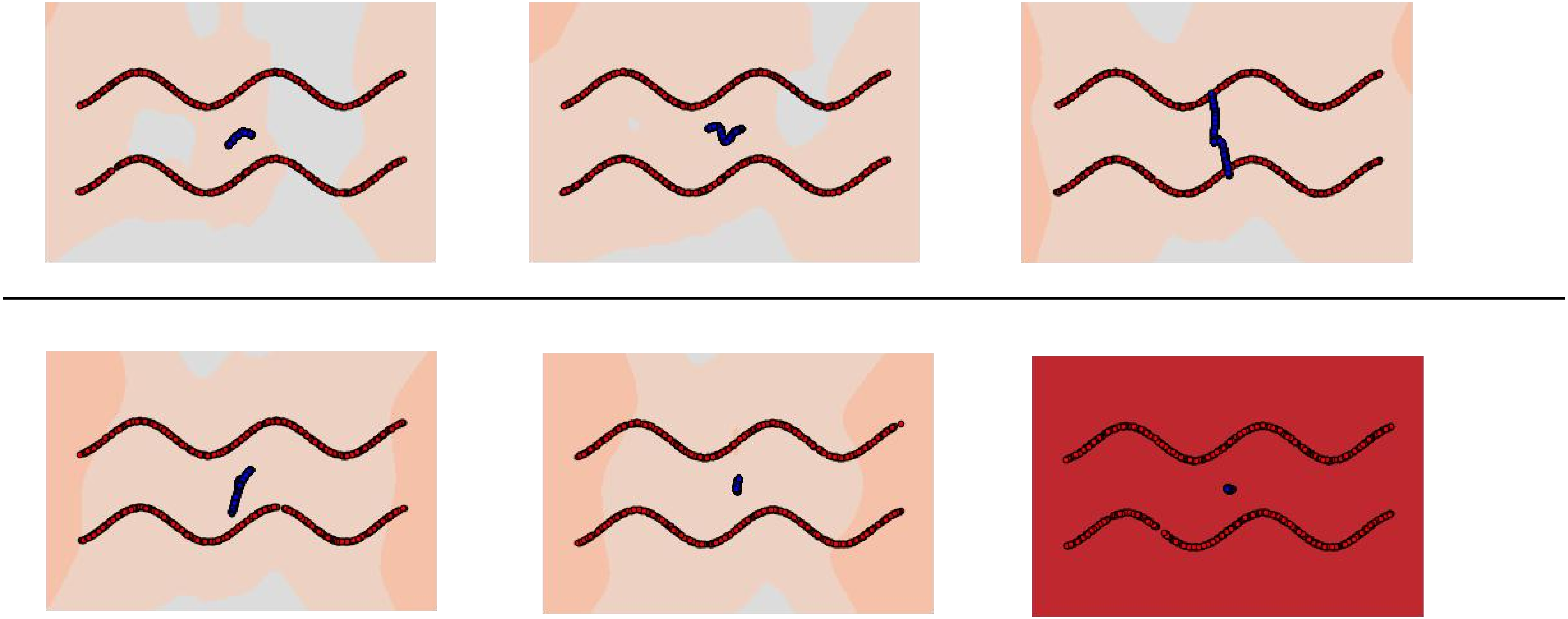


dimension of latent variable z

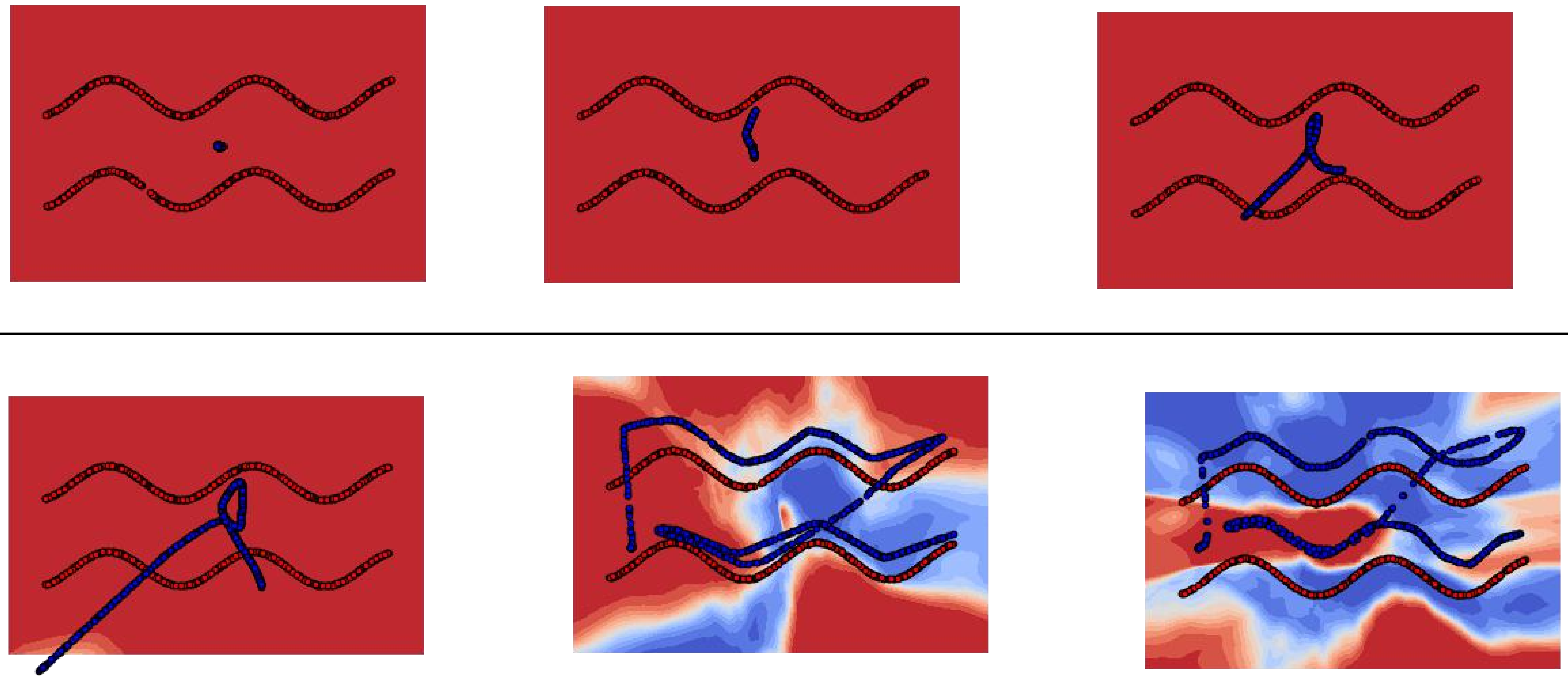




Collapse



Prison Break

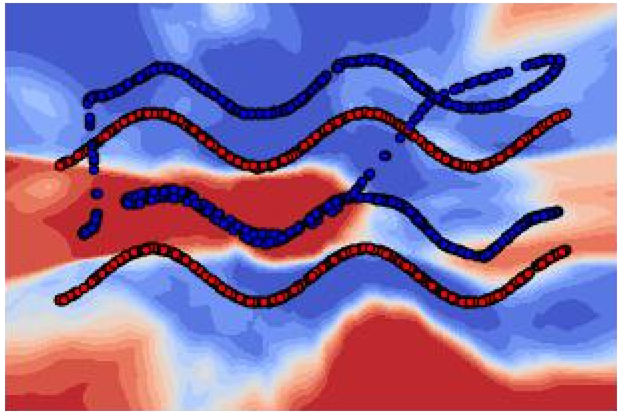


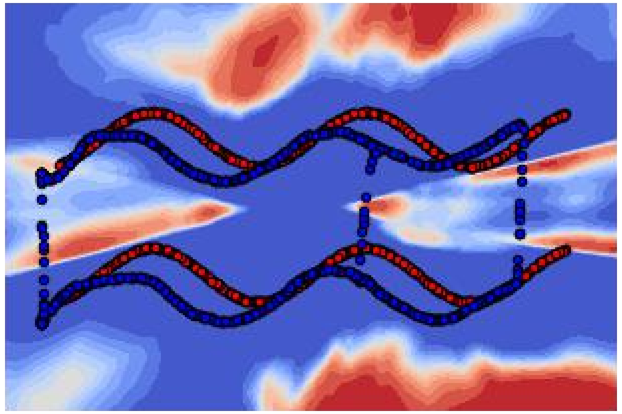
Loss

Origin: D_loss = D_real_loss + D_fake_loss

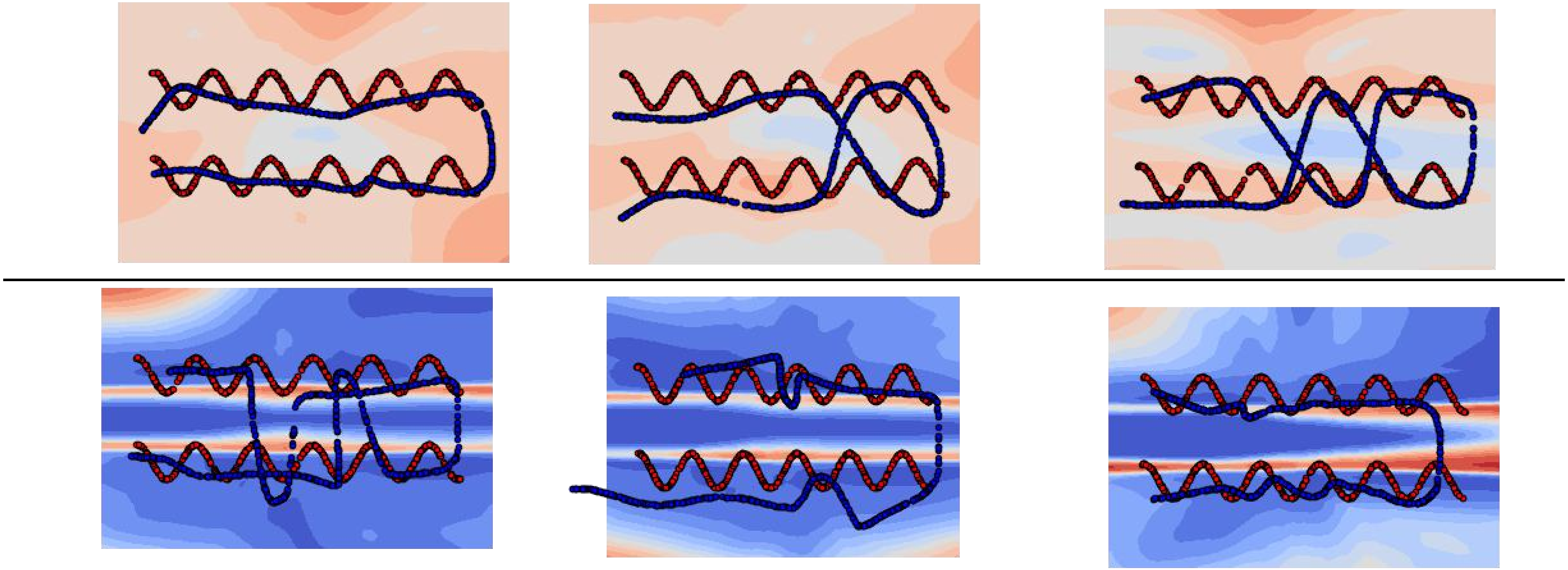
Changed: D_loss = 1.1*D_real_loss + 0.9*D_fake_loss

Result:

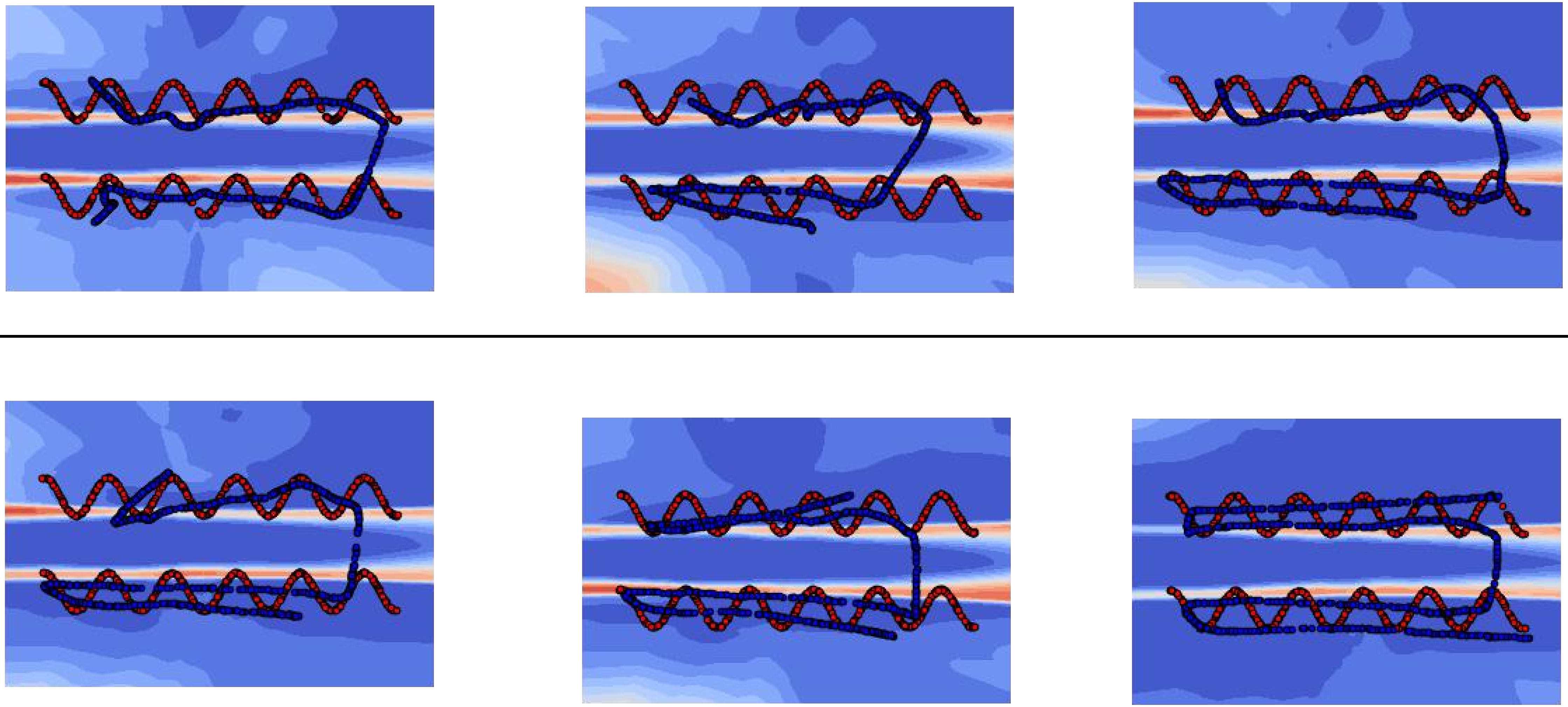




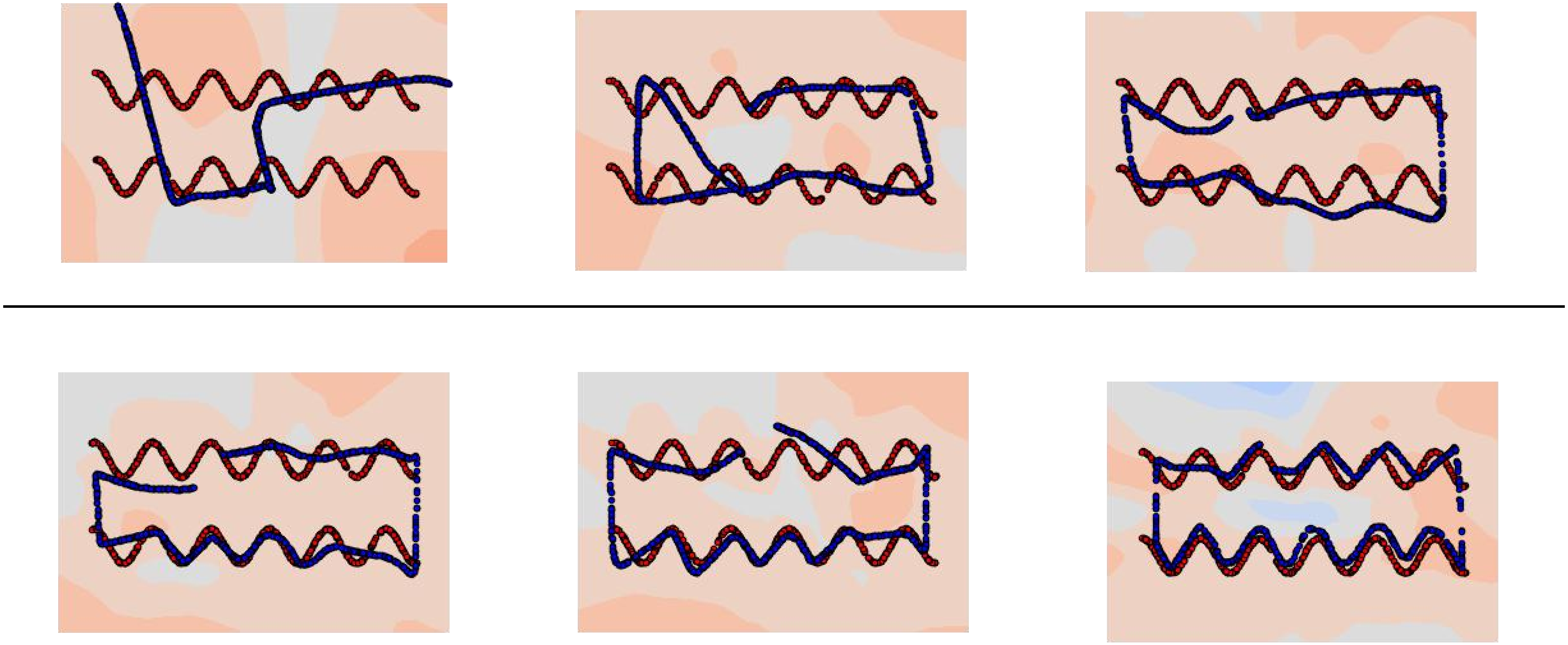
Interesting behavior



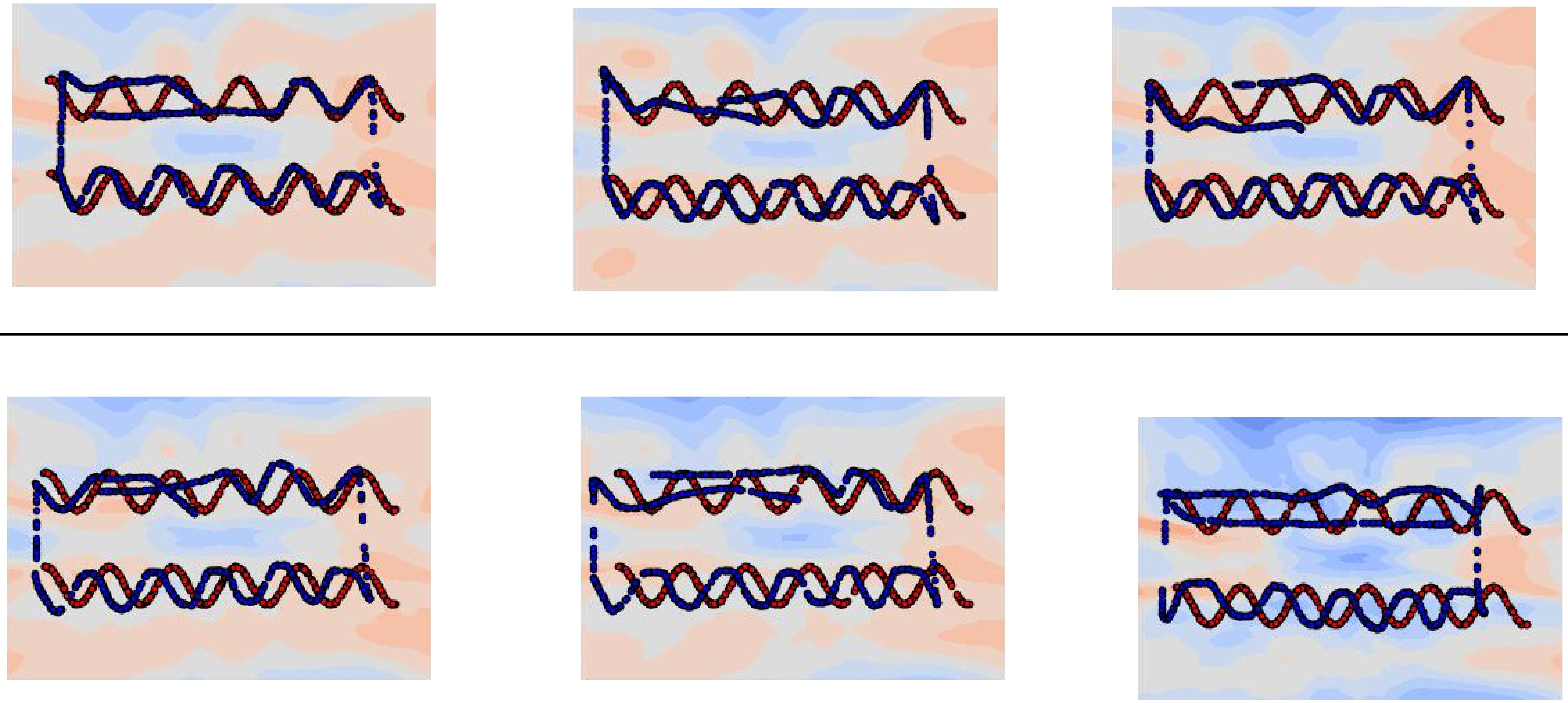
Interesting behavior



Bigger Batch



War between head and tail



Thanks!