Data-Driven Synthesis of Smoke Flows with CNN-based Feature Descriptors

Mengyu Chu, Nils Thuerey

Technical University of Munich

Introduction

- High resolution smoke generation

- Numerical viscosity
- Expensive calculations

Introduction

- Related work

Wavelet Turbulence for Fluid Simulation

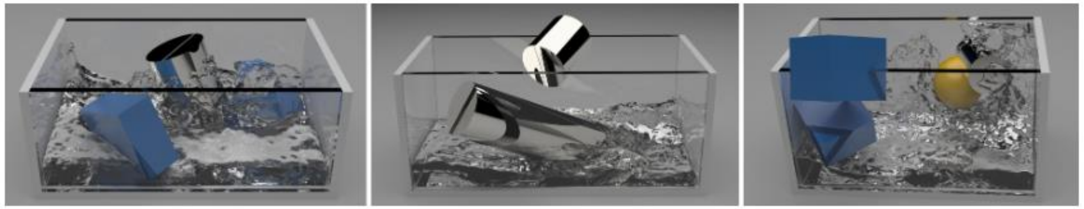
Theodore Kim, Cornell University
Nils Thürey, ETH Zurich
Doug James, Cornell University
Markus Gross, ETH Zurich

LazyFluids: Appearance Transfer for Fluid Animations

Ondřej Jamriška[1]* Jakub Fišer[1] Paul Asente[2] Jingwan Lu[2] Eli Shechtman[2] Daniel Sýkora[1]
[1]CTU in Prague, FEE [2]Adobe Research

Data-driven Fluid Simulations using Regression Forests

Ľubor Ladický*† ETH Zurich
SoHyeon Jeong*† ETH Zurich
Barbara Solenthaler† ETH Zurich
Marc Pollefeys† ETH Zurich
Markus Gross† ETH Zurich, Disney Research Zurich



Data-driven projection method in fluid simulation

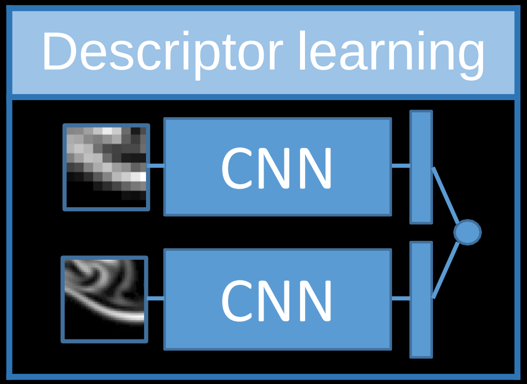
Cheng Yang, Xubo Yang* and Xiangyun Xiao
School of Software, Shanghai Jiao Tong University, Shanghai, China

Accelerating Eulerian Fluid Simulation With Convolutional Networks

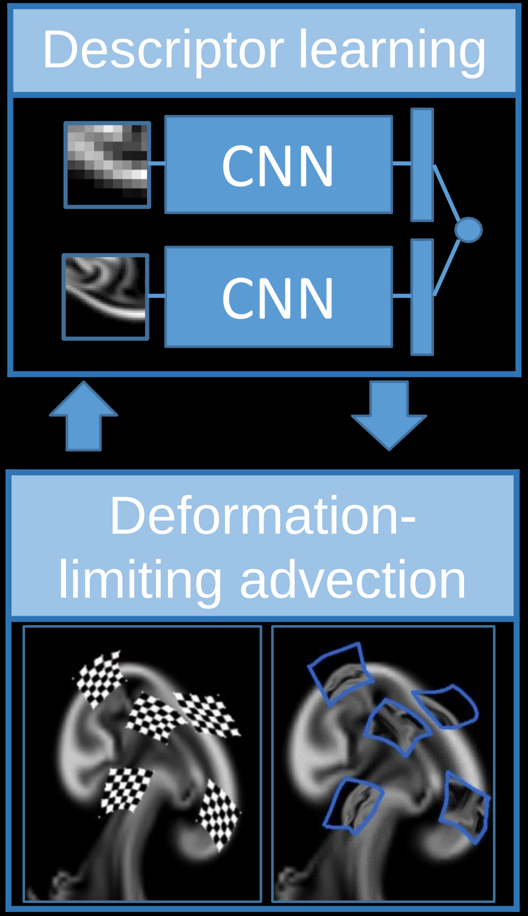
Jonathan Tompson, Google Inc.
and
Kristofer Schlachter, Pablo Sprechmann, Ken Perlin, New York University

Proposed approach

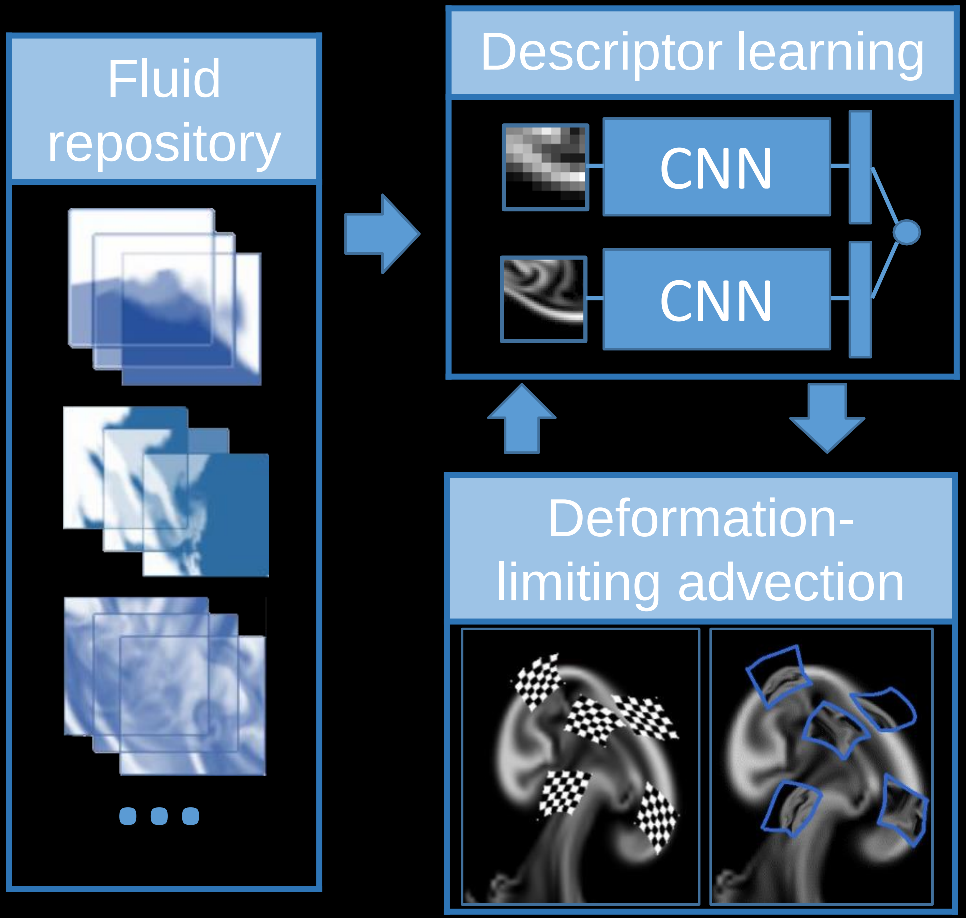
Overview



Overview

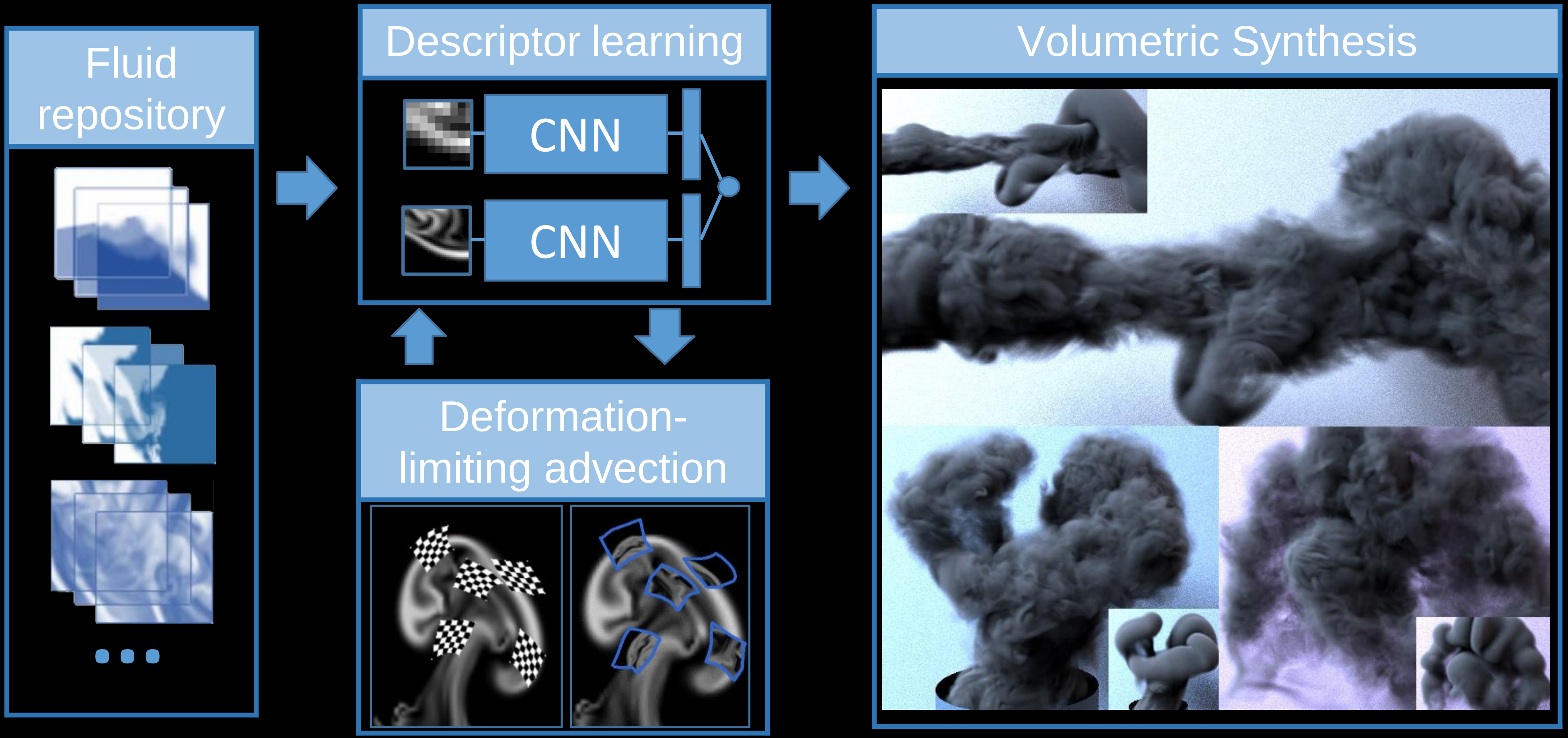

Descriptor learning
CNN
CNN
Deformation-
limiting advection

Overview

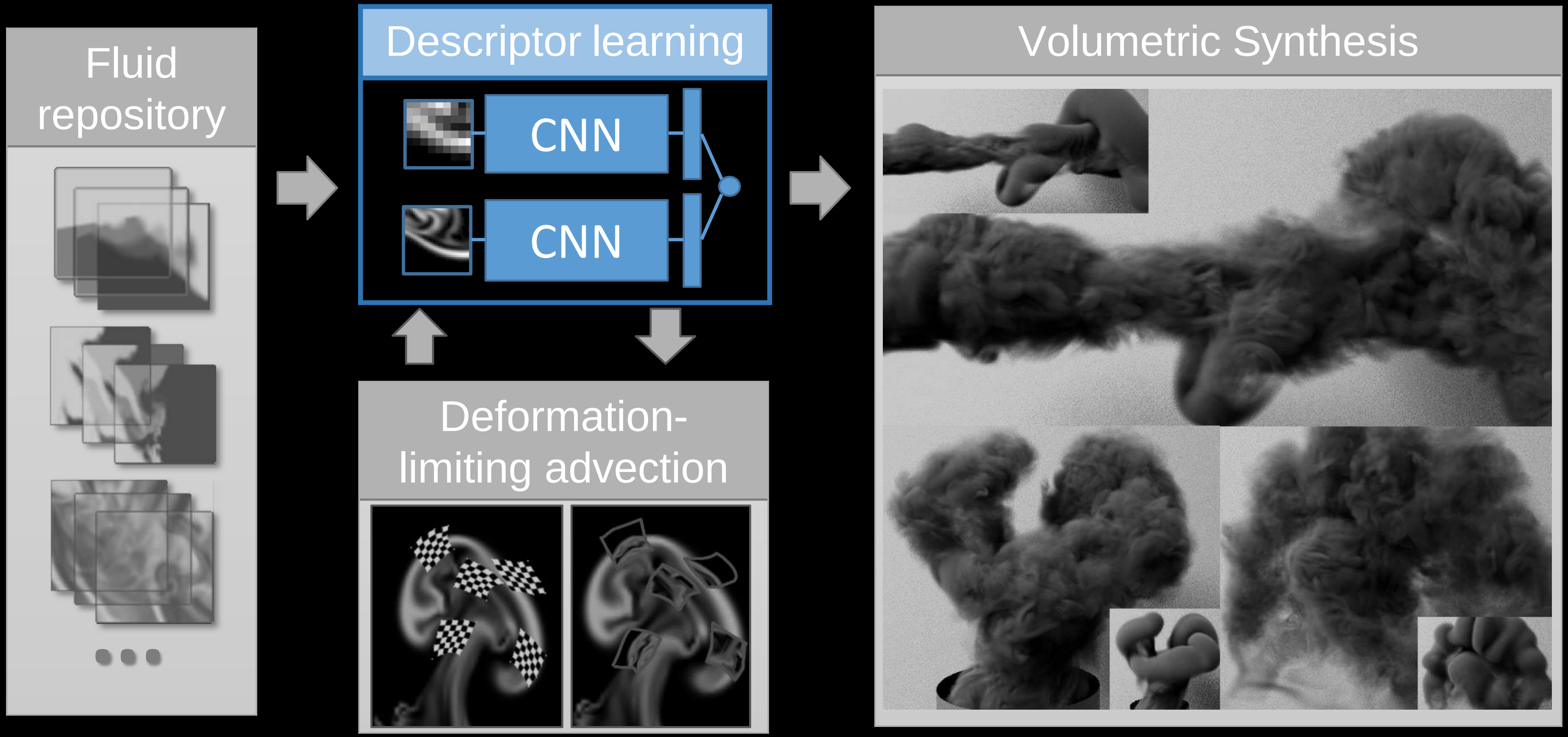


Fluid repository
Descriptor learning
CNN
CNN
Deformation-limiting advection

Overview


Fluid repository
...
Descriptor learning
CNN
CNN
Deformation-limiting advection
Volumetric Synthesis

Overview

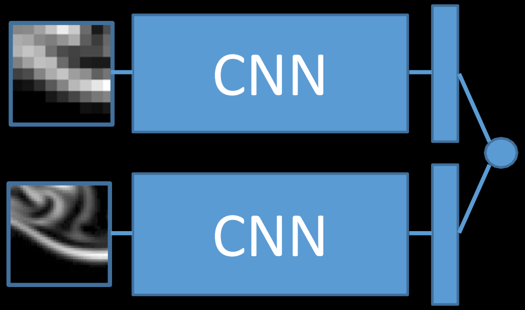

Fluid repository
Descriptor learning
CNN
CNN
Deformation-limiting advection
Volumetric Synthesis

Learning flow similarity

- Descriptor learning

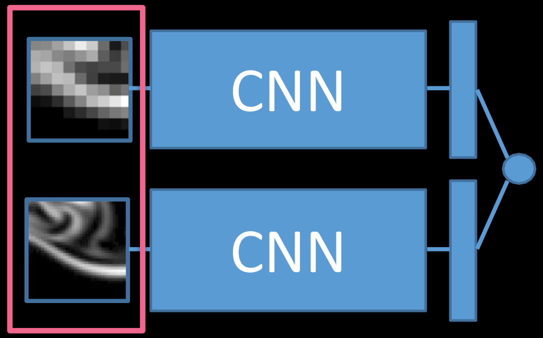
Learning flow similarity

- Descriptor learning



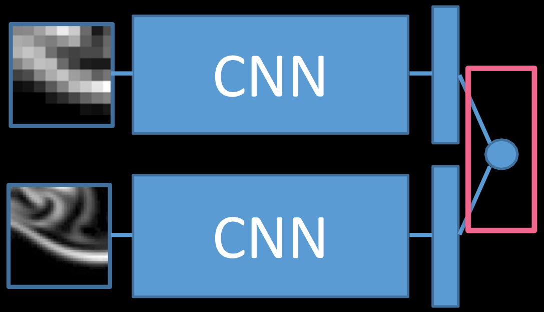
Learning flow similarity

- Descriptor learning
  - Input: pair of fluid data



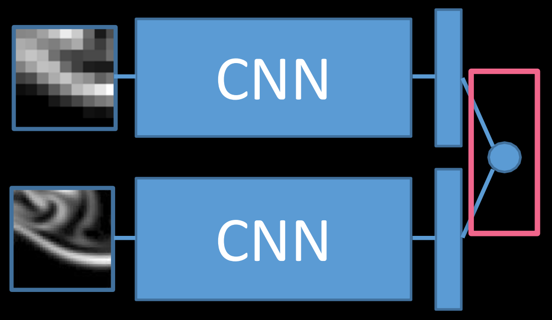
Learning flow similarity

- Descriptor learning
  - Input: pair of fluid data
  - Output: similarity (scalar)



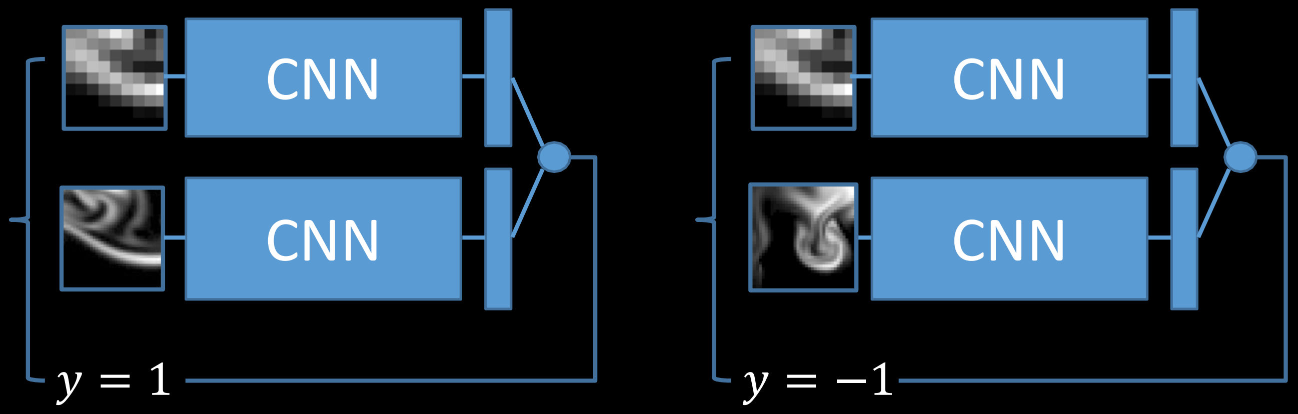
Learning flow similarity

- Descriptor learning
  - Input: pair of fluid data
  - Output: similarity (scalar)
  - Flow similarity, 1 as similar, -1 as dissimilar



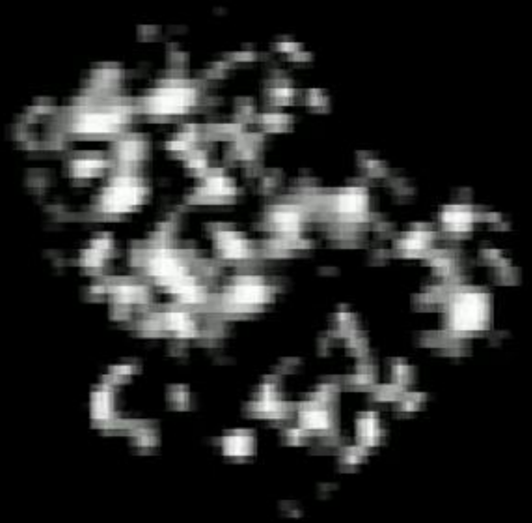
Learning flow similarity

- Descriptor learning
  - Input: pair of fluid data
  - Output: similarity (scalar)
  - Flow similarity, 1 as similar, -1 as dissimilar
  - Labelled input pairs

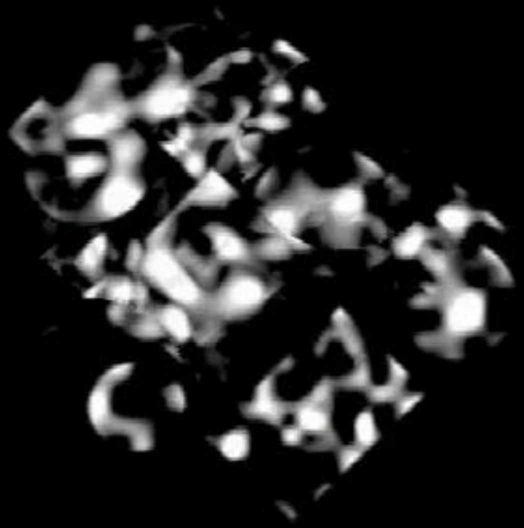


Learning flow similarity

Data generation



Low resolution
(Re-synchronized every 20 steps)

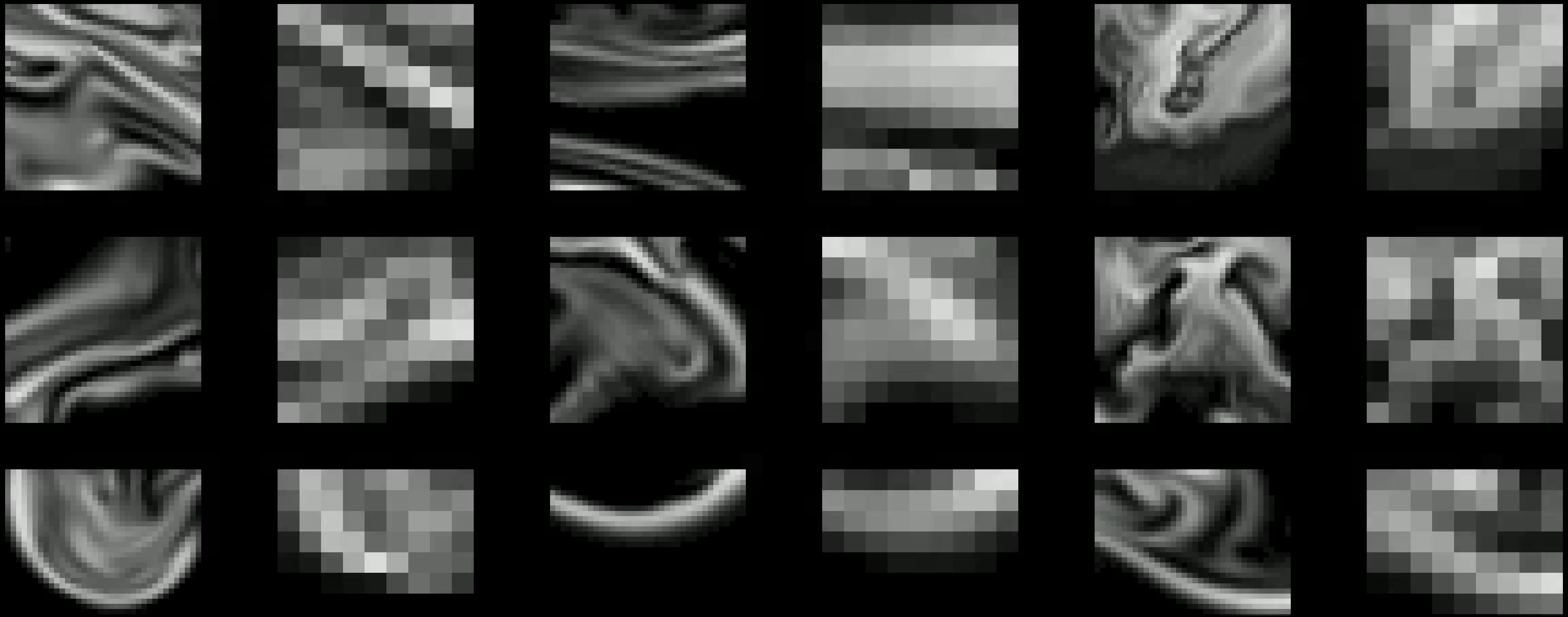


High resolution

Learning flow similarity

Data generation

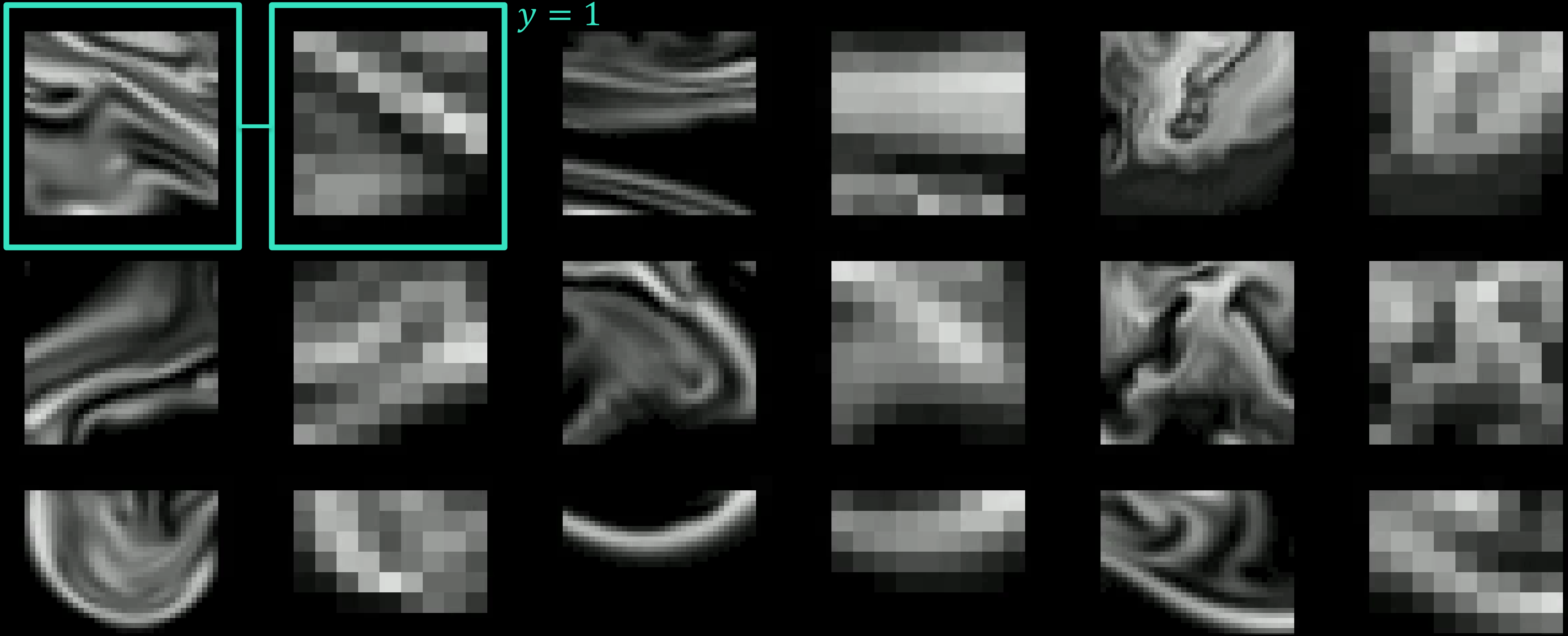
Example of input pairs
Extracted per frame, 120 frames per patch



Learning flow similarity

Data generation

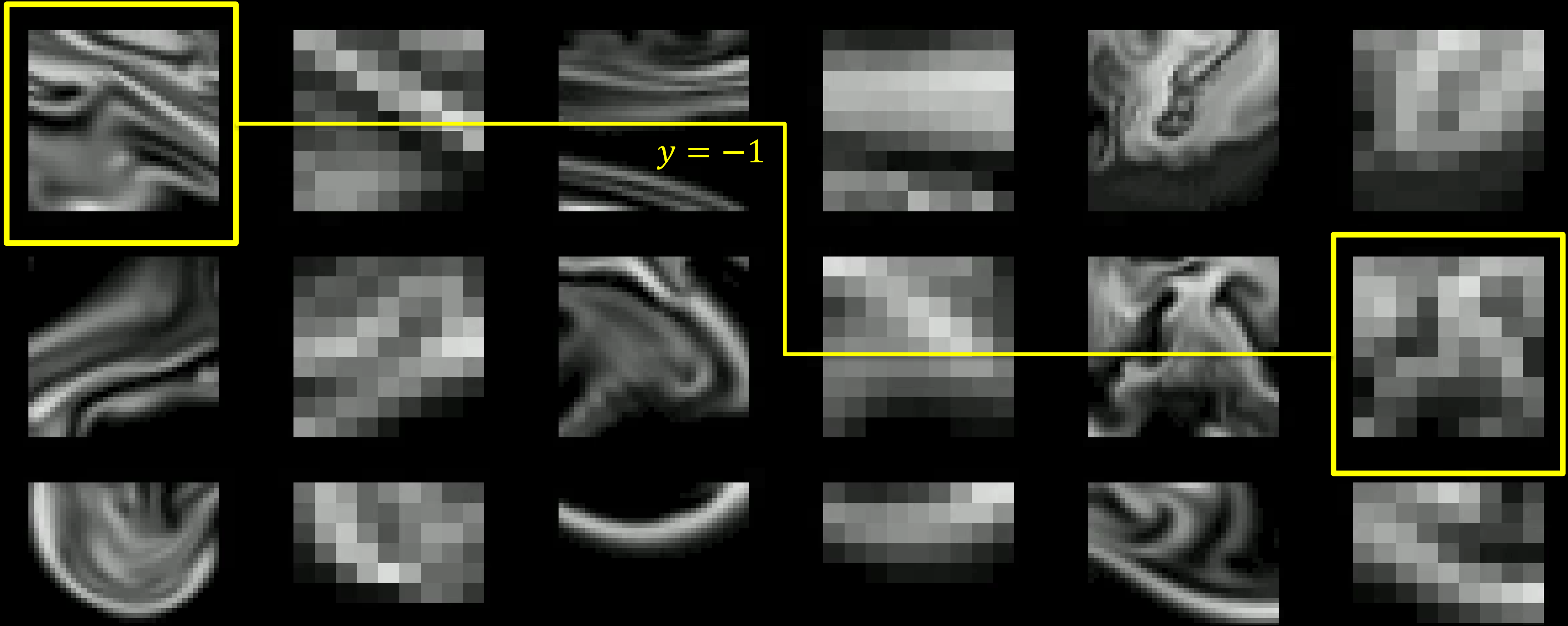
Example of input pairs
Extracted per frame, 120 frames per patch



Learning flow similarity

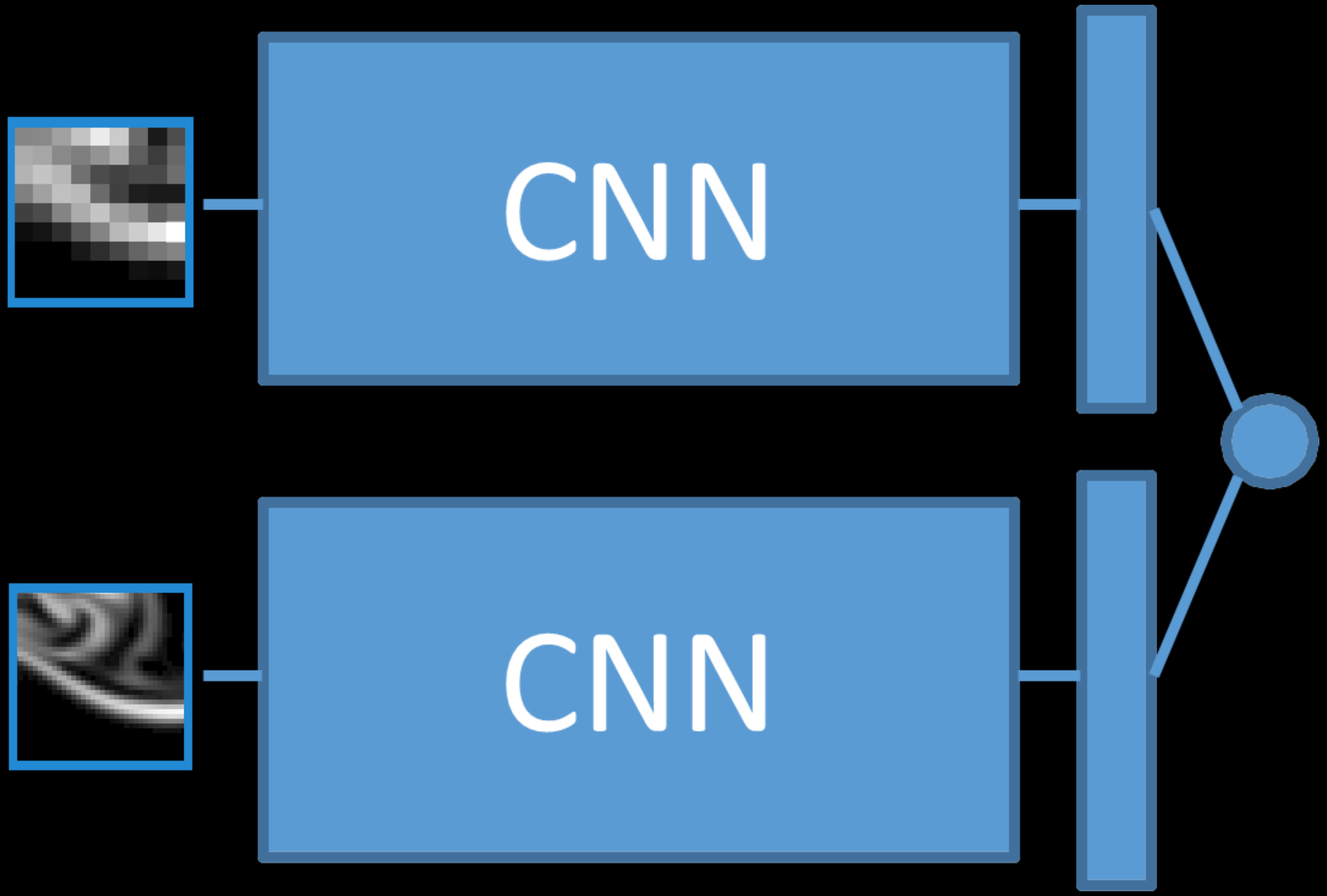
Data generation

Example of input pairs
Extracted per frame, 120 frames per patch



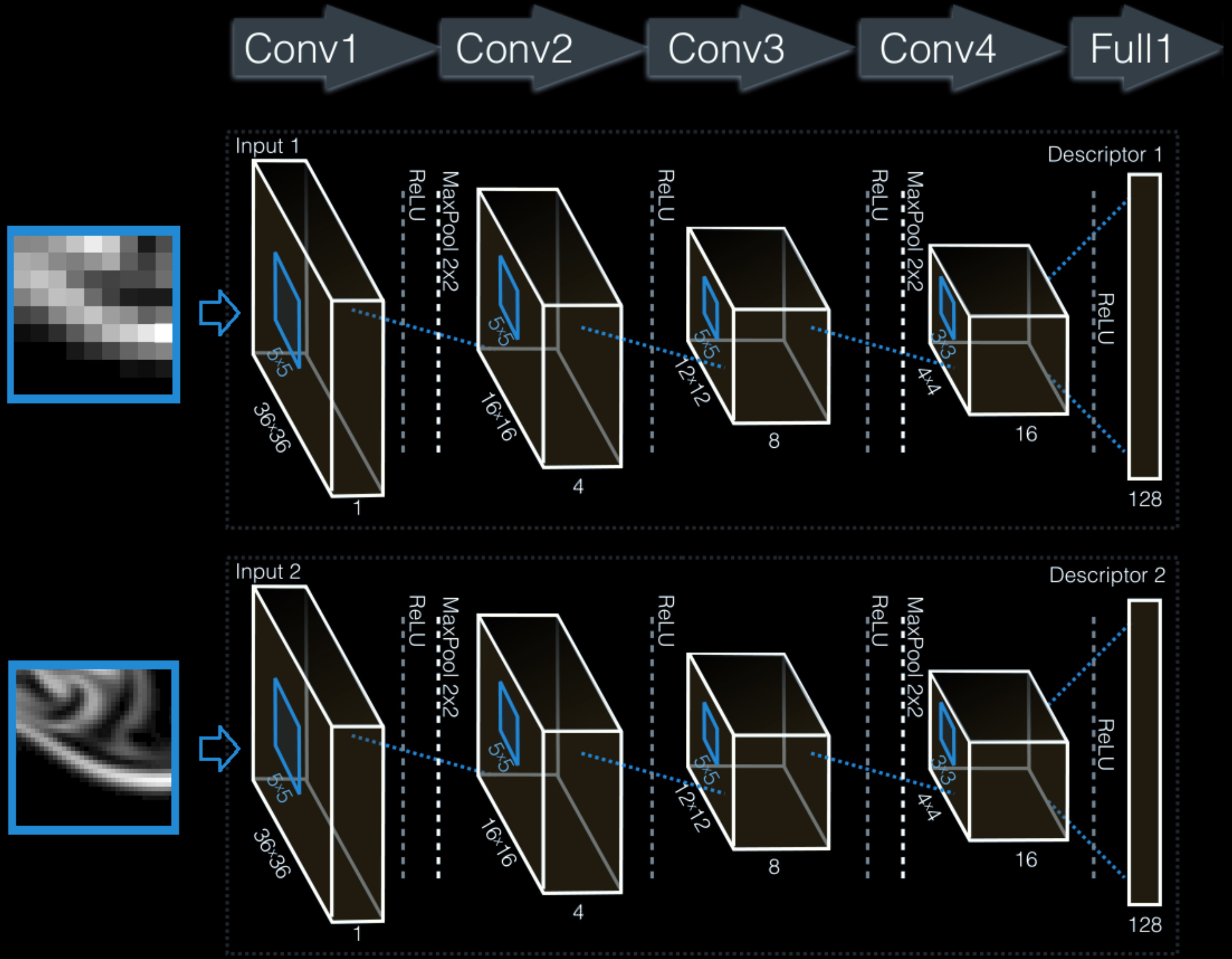
Learning flow similarity

- Structure



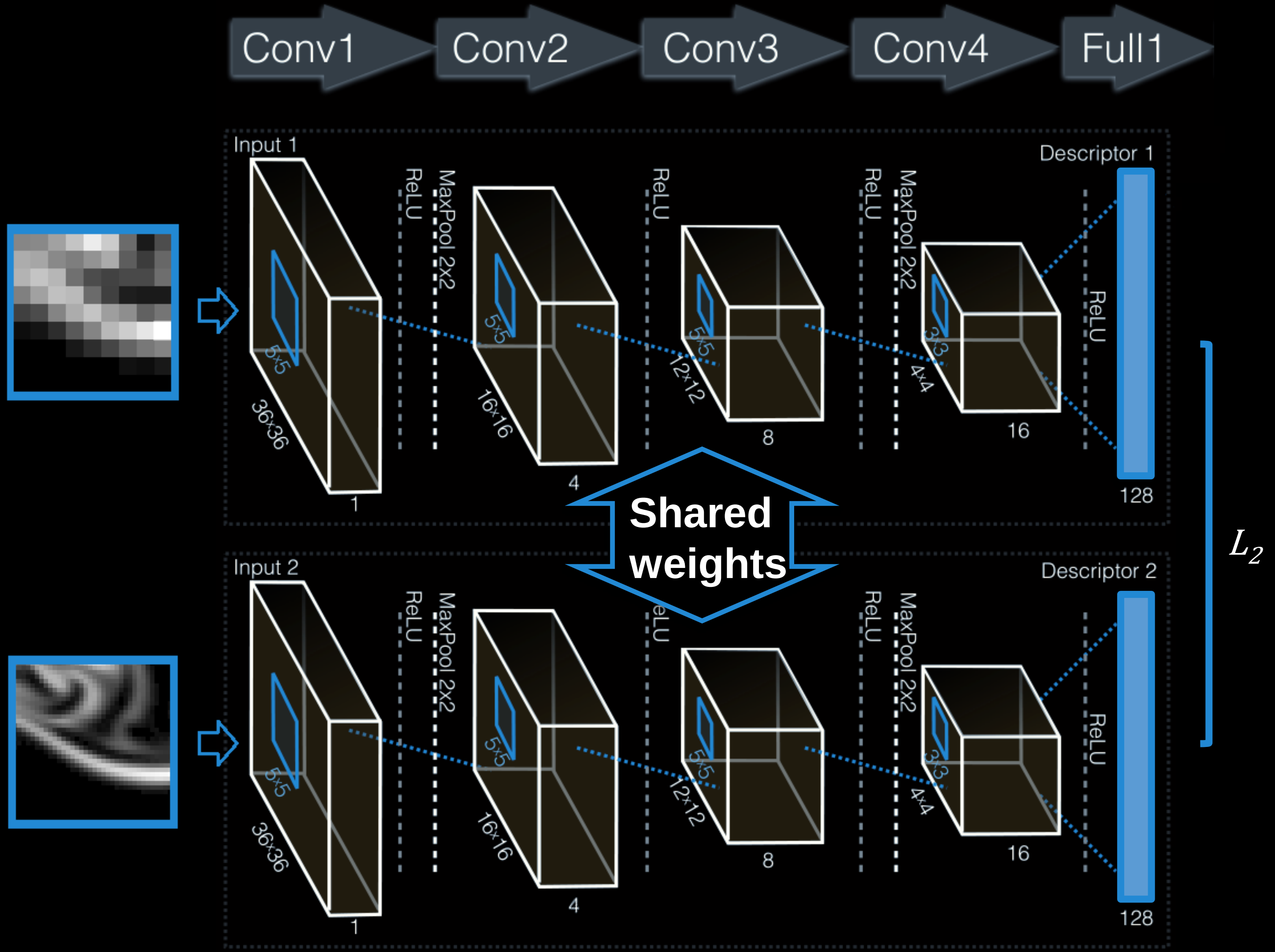
Learning flow similarity

- Structure
  - Siamese structure



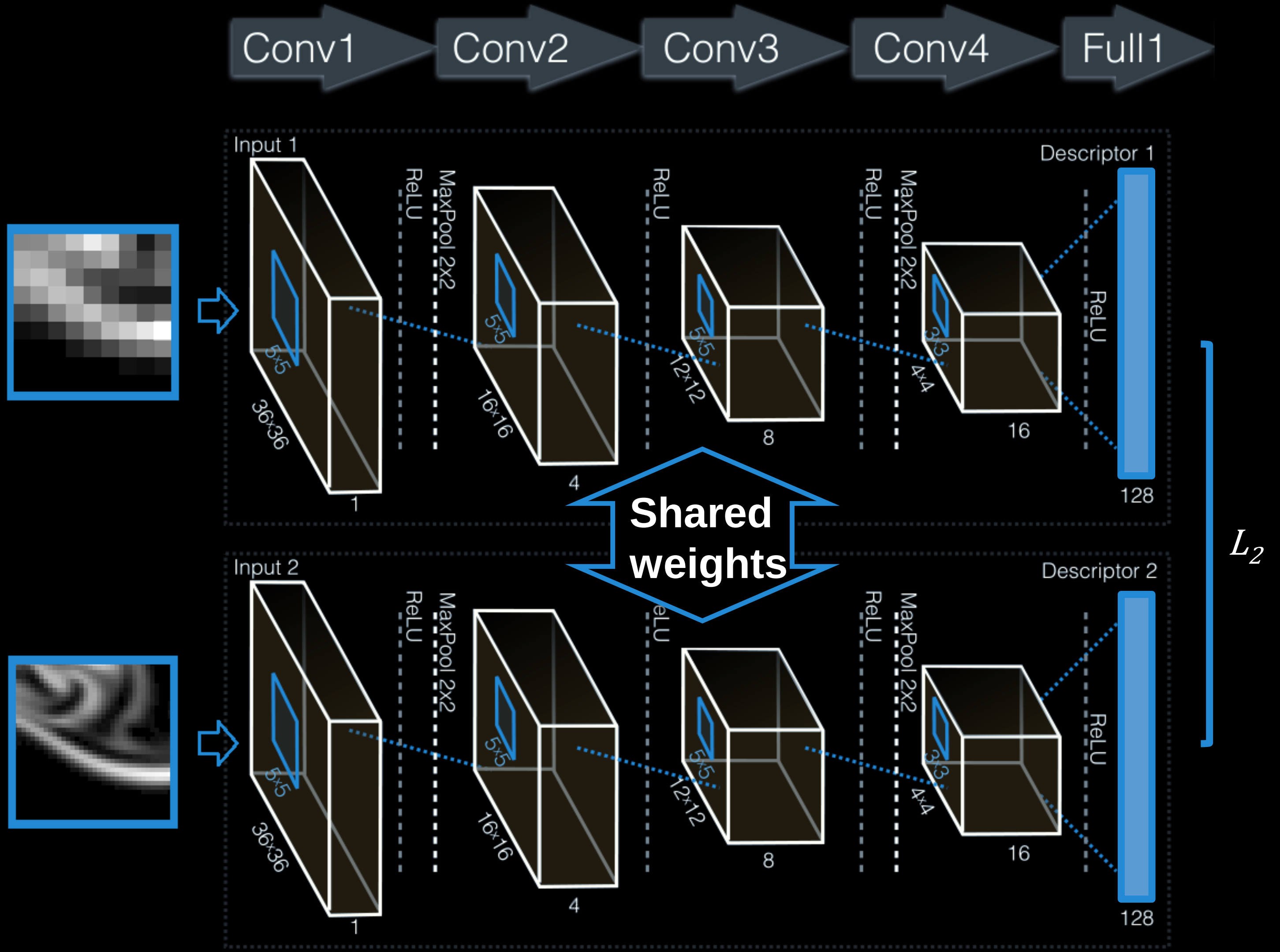
Learning flow similarity

- Structure
  - Siamese structure



Learning flow similarity

- Structure
  - Siamese structure
  - Descriptor learning
    - – Invariants
      - resolution
      - numerical viscosity



Learning flow similarity

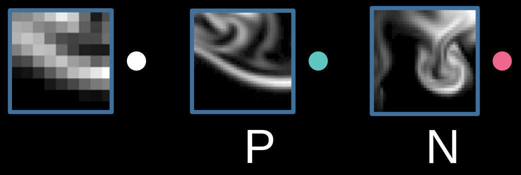
- CNN structure —— Siamese structure
- Loss function

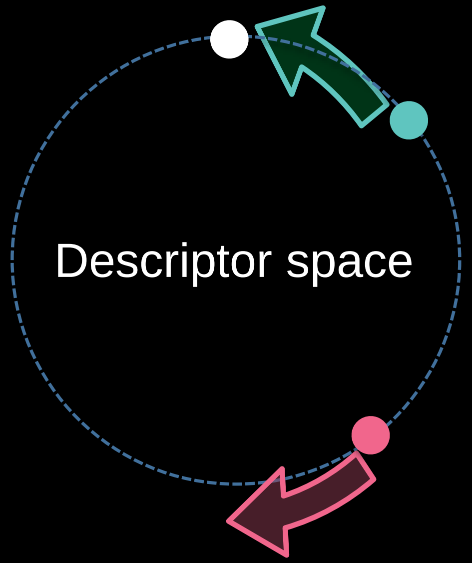
$$l\ (x_1, x_2) = \begin{cases} +\|d_w(x_1) - d_w(x_2)\|\ , & y = 1 \\ -\|d_w(x_1) - d_w(x_2)\|\ , & y = -1 \end{cases}$$

Learning flow similarity

- CNN structure —— Siamese structure
- Loss function

$$l\ (x_1, x_2) = \begin{cases} +\ \|d_w(x_1) - d_w(x_2)\|\ , & y = 1 \\ -\ \|d_w(x_1) - d_w(x_2)\|\ , & y = -1 \end{cases}$$

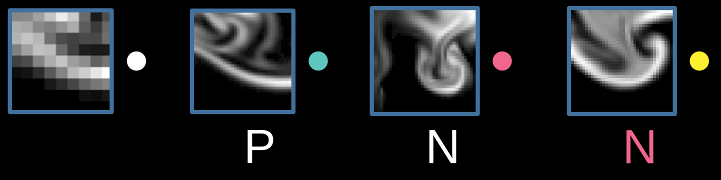


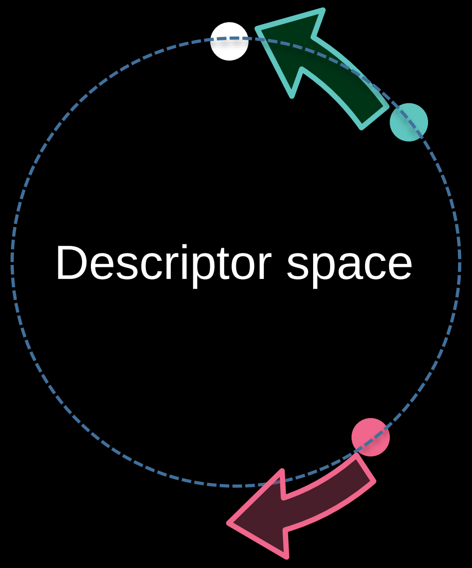


Learning flow similarity

- CNN structure —— Siamese structure
- Loss function

$$l\ (x_1, x_2) = \begin{cases} +\,\|d_w(x_1) - d_w(x_2)\|\ , & y = 1 \\ -\,\|d_w(x_1) - d_w(x_2)\|\ , & y = -1 \end{cases}$$

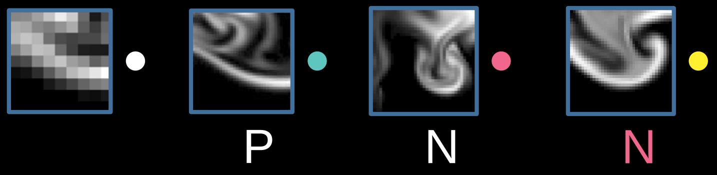


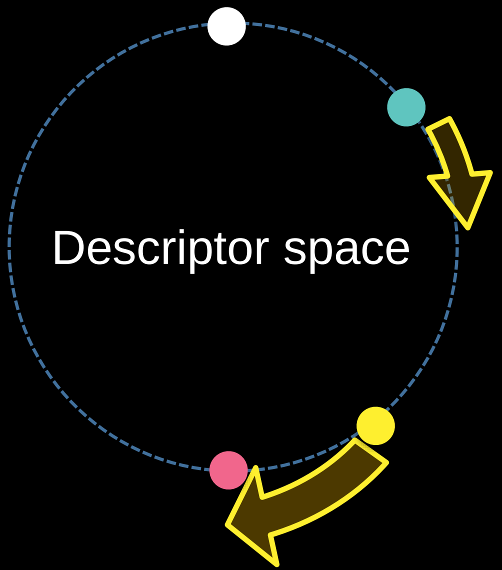


Learning flow similarity

- CNN structure —— Siamese structure
- Loss function

$$l\ (x_1, x_2) = \begin{cases} +\,\|d_w(x_1) - d_w(x_2)\|\ , & y = 1 \\ -\,\|d_w(x_1) - d_w(x_2)\|\ , & y = -1 \end{cases}$$

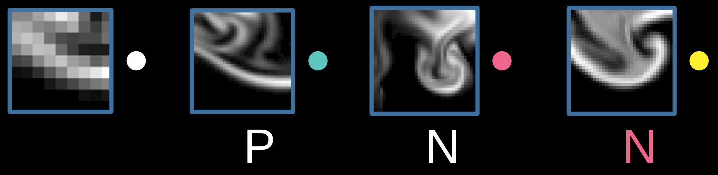


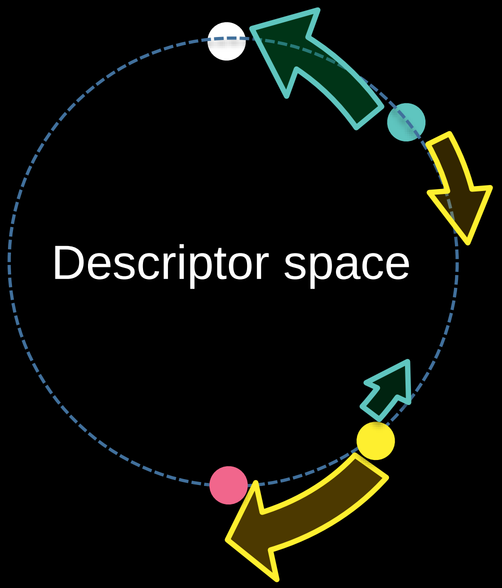


Learning flow similarity

- CNN structure —— Siamese structure
- Loss function

$$l\ (x_1, x_2) = \begin{cases} +\ \|d_w(x_1) - d_w(x_2)\|\ , & y = 1 \\ -\ \|d_w(x_1) - d_w(x_2)\|\ , & y = -1 \end{cases}$$

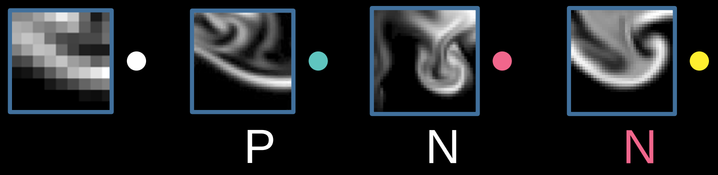


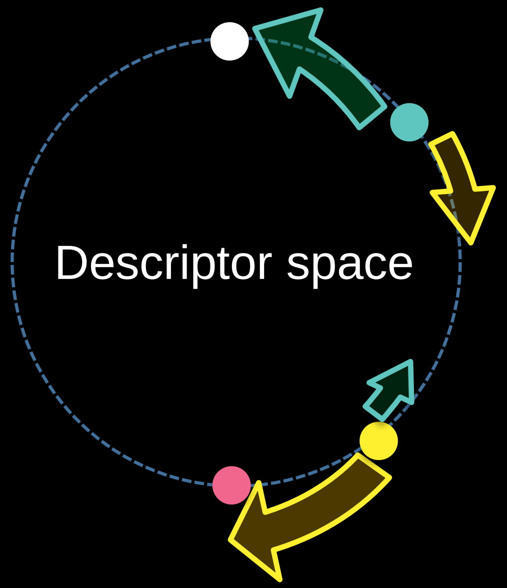


Learning flow similarity

- CNN structure —— Siamese structure
- Loss function

$$l\ (x_1, x_2) = \begin{cases} +\|d_w(x_1) - d_w(x_2)\| , & y = 1 \\ -\|d_w(x_1) - d_w(x_2)\| , & y = -1 \end{cases}$$

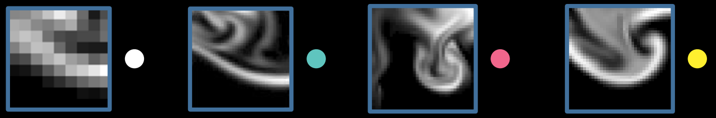




Learning flow similarity

- CNN structure —— Siamese structure
- Loss function —— Hinge loss

$$l_e(x_1, x_2) = \begin{cases} \max\left(0, -a_p + \|d_w(x_1) - d_w(x_2)\|\right), & y = 1 \\ \max\left(0, \; a_n - \|d_w(x_1) - d_w(x_2)\|\right), & y = -1 \end{cases}$$

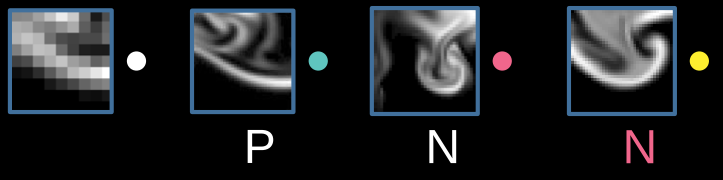


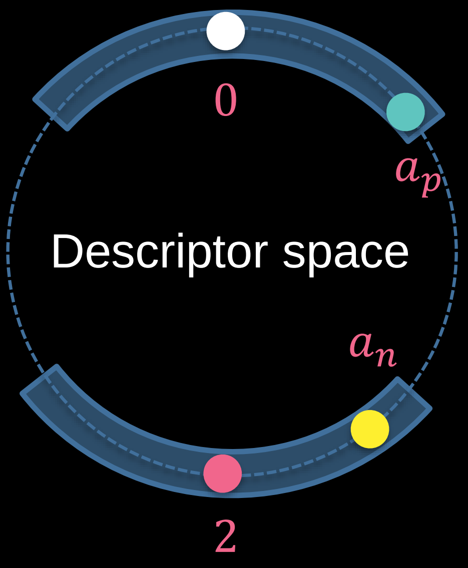
P N N

Learning flow similarity

- CNN structure —— Siamese structure
- Loss function —— Hinge loss

$$l_e(x_1, x_2) = \begin{cases} \max(0, -a_p + \|d_w(x_1) - d_w(x_2)\|), & y = 1 \\ \max(0, \ a_n - \|d_w(x_1) - d_w(x_2)\|), & y = -1 \end{cases}$$

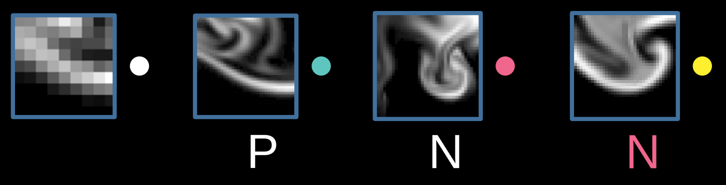


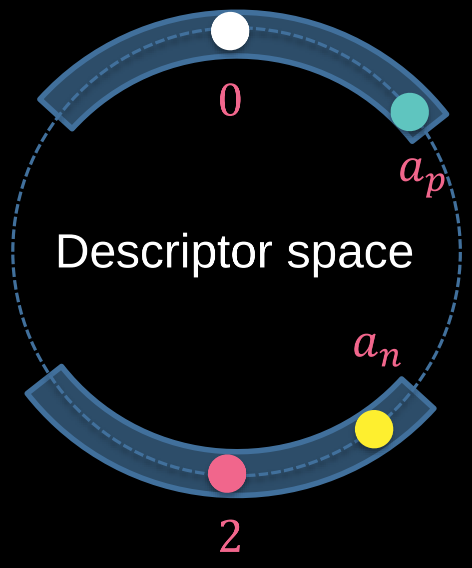


Learning flow similarity

- CNN structure —— Siamese structure
- Loss function —— Hinge loss

$$l_e(x_1, x_2) = \begin{cases} \max(0, -a_p + \|d_w(x_1) - d_w(x_2)\|), & y = 1 \\ \max(0, a_n - \|d_w(x_1) - d_w(x_2)\|), & y = -1 \end{cases}$$

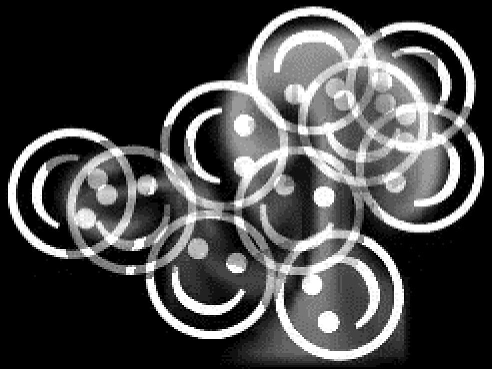




Patch advection

- Error minimization problem

$$E = \lambda E_{defo} + E_{adv}$$

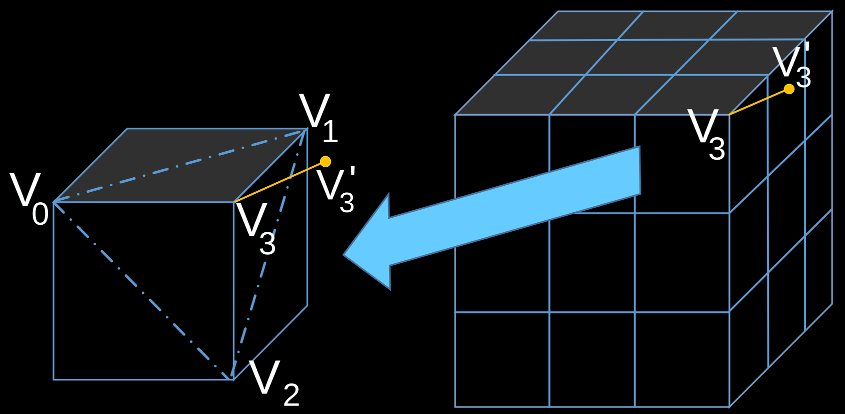


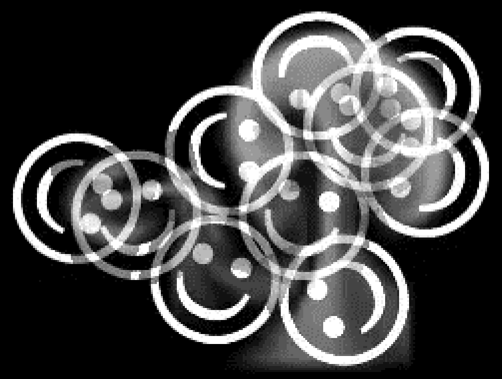
Patch advection

- Error minimization problem

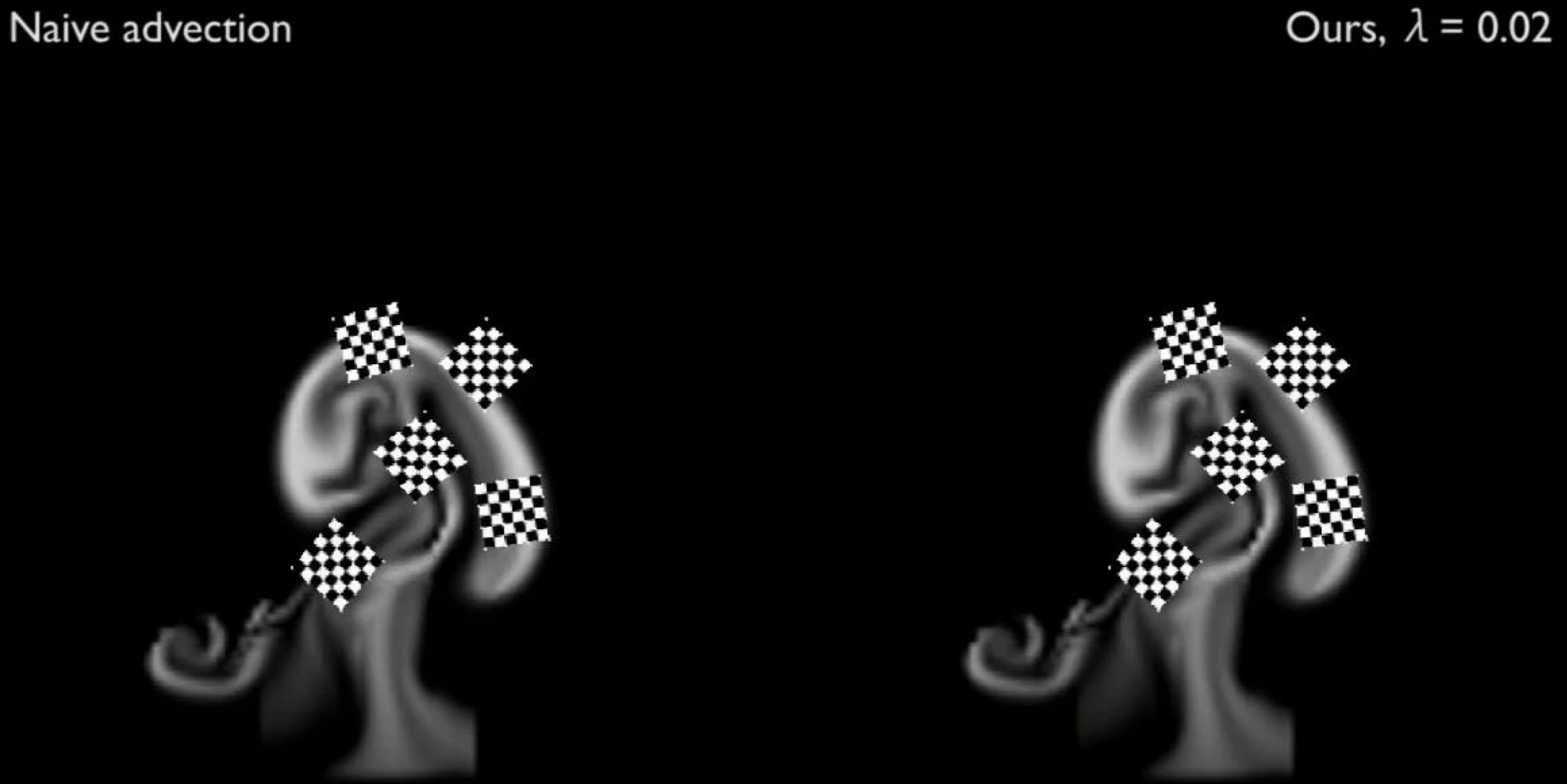
$$E = \lambda E_{defo} + E_{adv}$$

  - $E_{adv} = \sum \|v_i - v_i'\|^2, \; v' = adv(v_{t-1})$
  - $E_{defo} = \sum \|v_i - v_i^*\|^2 = \sum \left\| v_i - \sum A_j v_j \right\|^2$
    - v_i^*, based on Laplacian coordinates [Sorkine et al. 2004]





Patch advection



Patch anticipation

- Fading in → Anticipation

Patch anticipation

- Fading in → Anticipation
- Fading out ill-suited ones

Patch anticipation

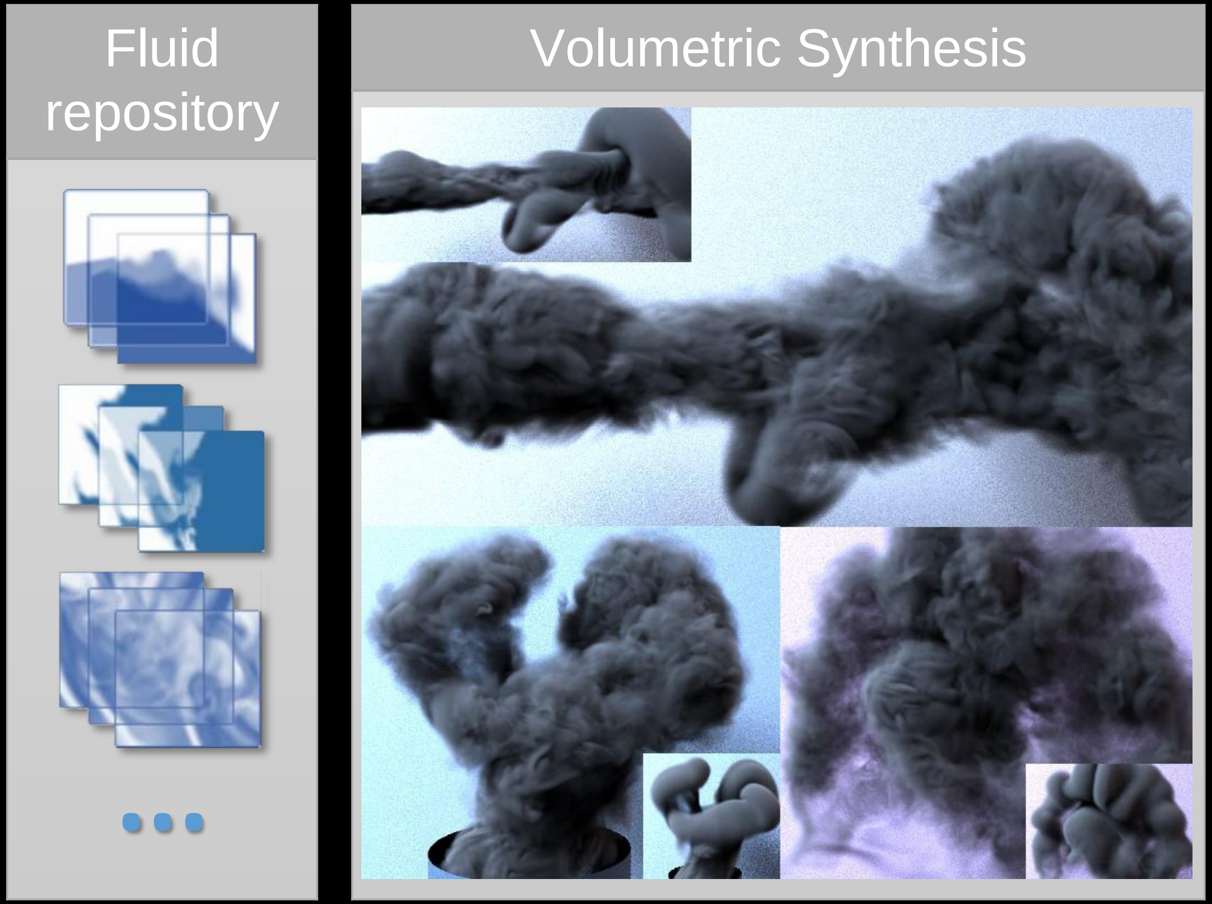
- Fading in → Anticipation
- Fading out ill-suited ones

Normal fading in

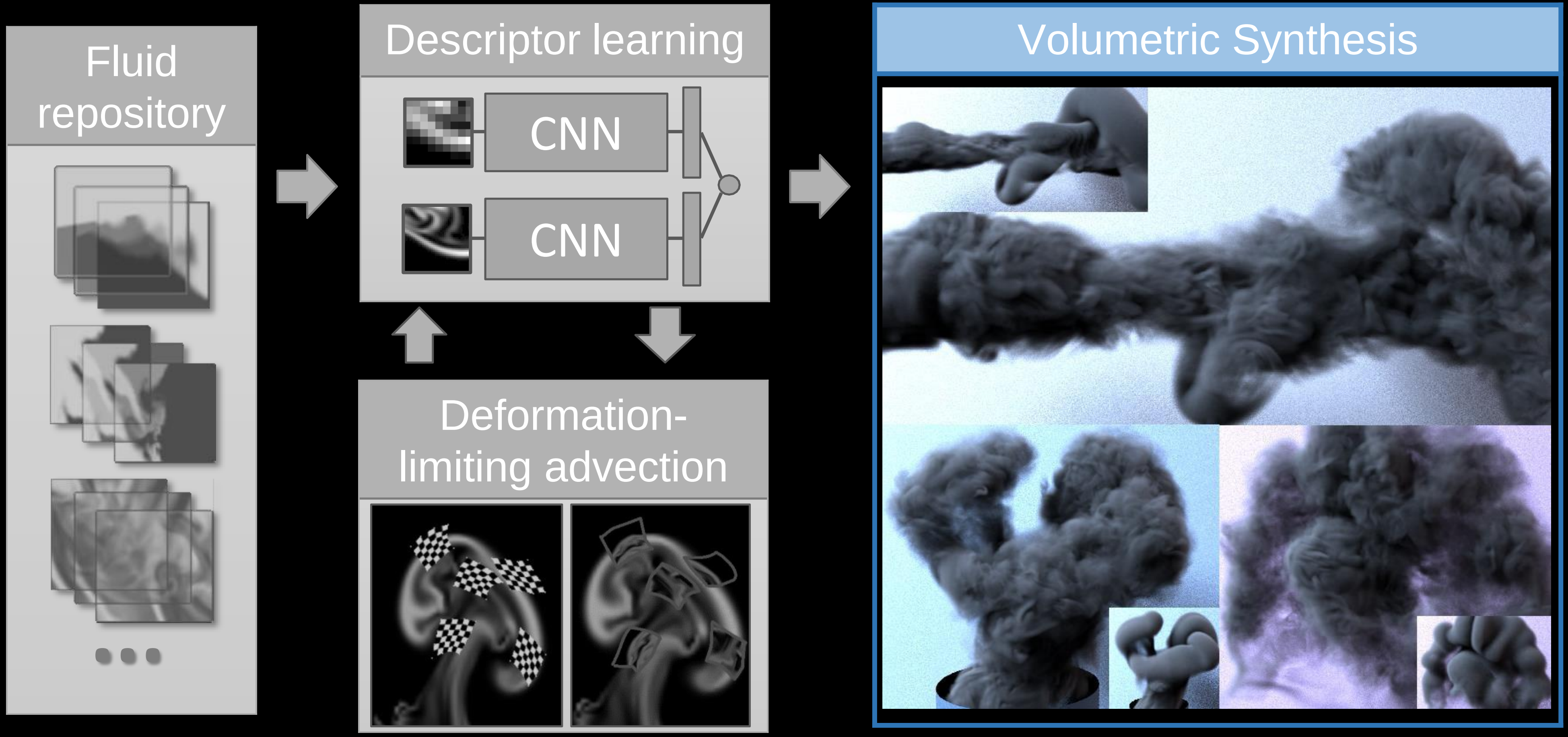
Patch anticipation

Patch advection

- Fluid repository
  - Space-time data
- Synthesis
  - Reusing the repository
- Lagrangian
  - Stable & reusable
  - Resolution independent



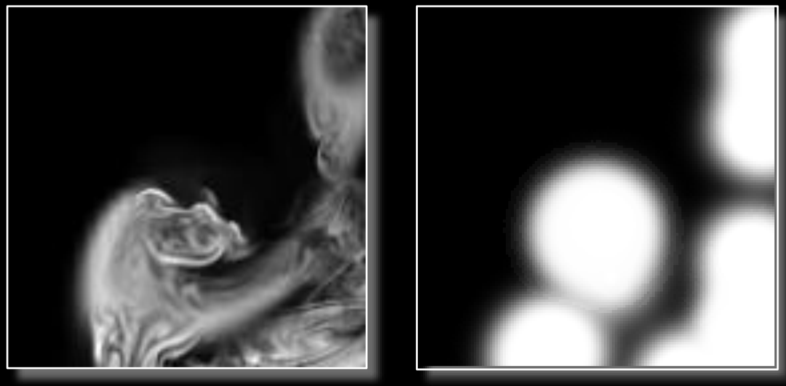
Overview


Fluid repository
Descriptor learning
CNN
CNN
Deformation-limiting advection
Volumetric Synthesis

Synthesis

Simulation:

- Forward pass
  - Sampling, matching

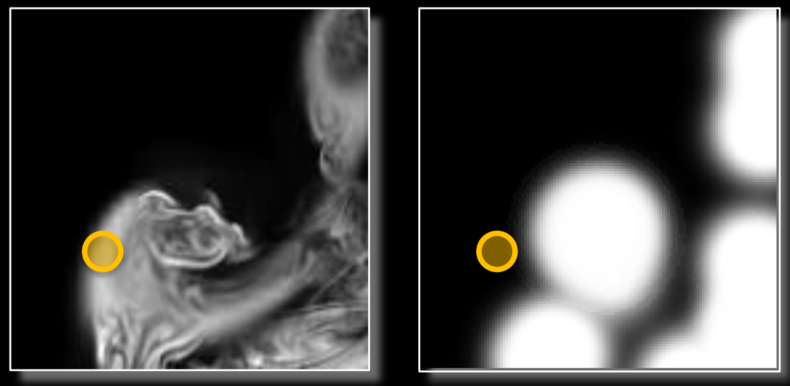


- Backward pass

Synthesis

Simulation:

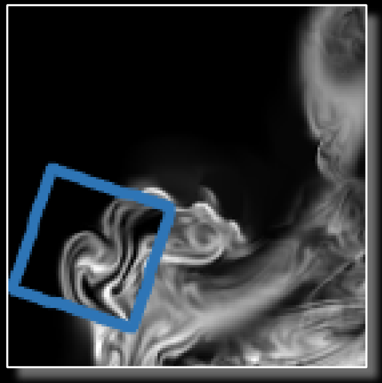
- Forward pass
  - Sampling, matching
- Backward pass

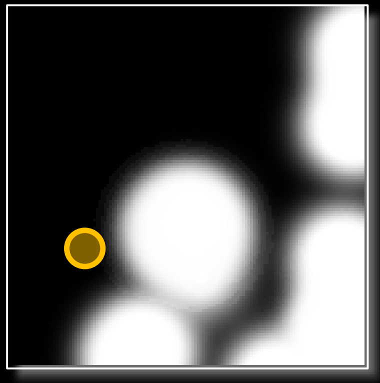


Synthesis

Simulation:

- Forward pass
  - Sampling, matching
- Backward pass

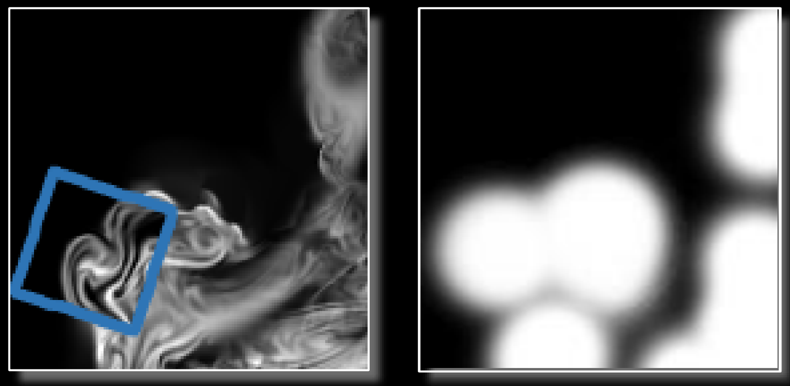




Synthesis

Simulation:

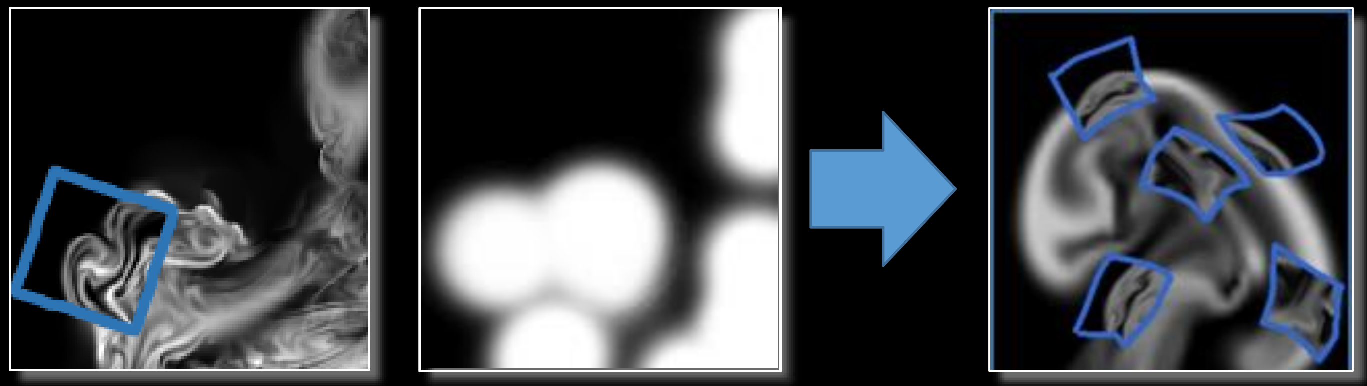
- Forward pass
  - Sampling, matching
- Backward pass



Synthesis

Simulation:

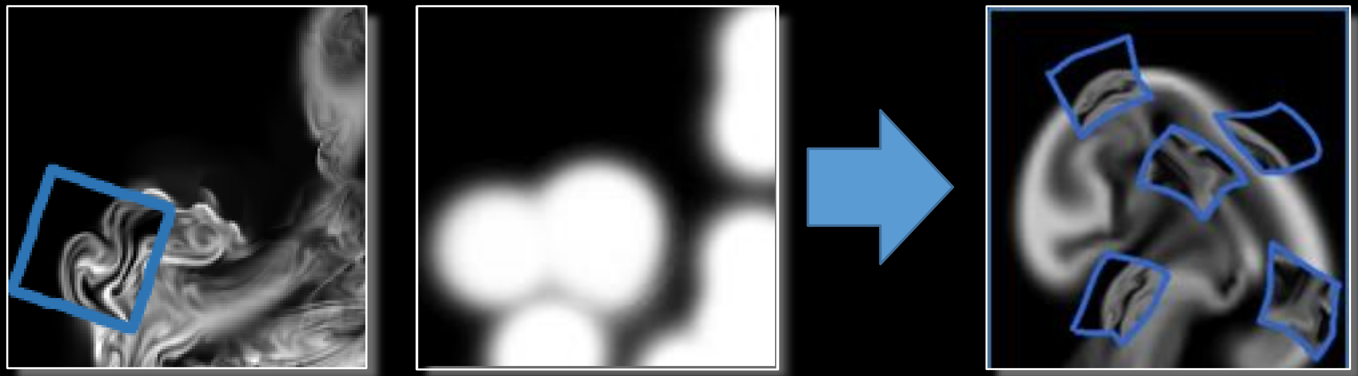
- Forward pass
  - Sampling, matching
  - Forward advection
- Backward pass



Synthesis

Simulation:

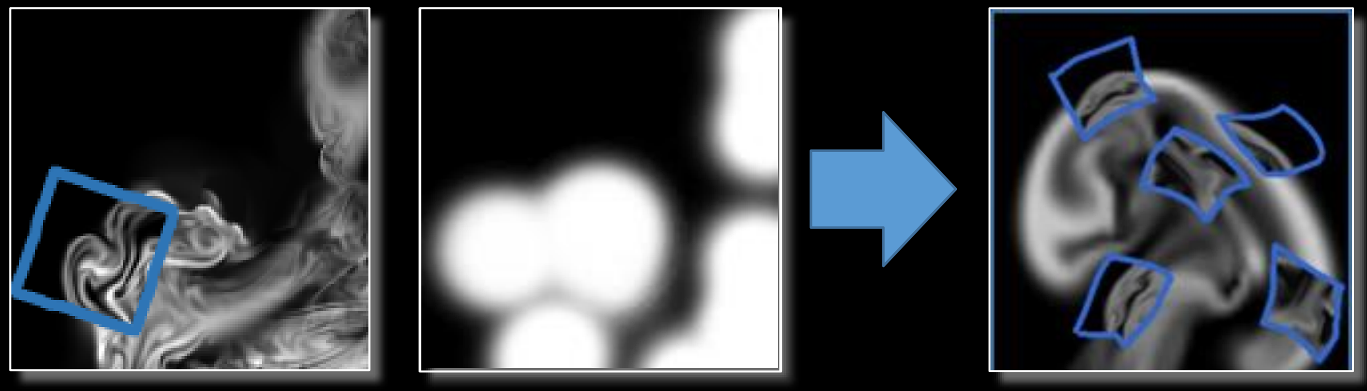
- Forward pass
  - Sampling, matching
  - Forward advection
  - Fading out ill-suited
- Backward pass



Synthesis

Simulation:

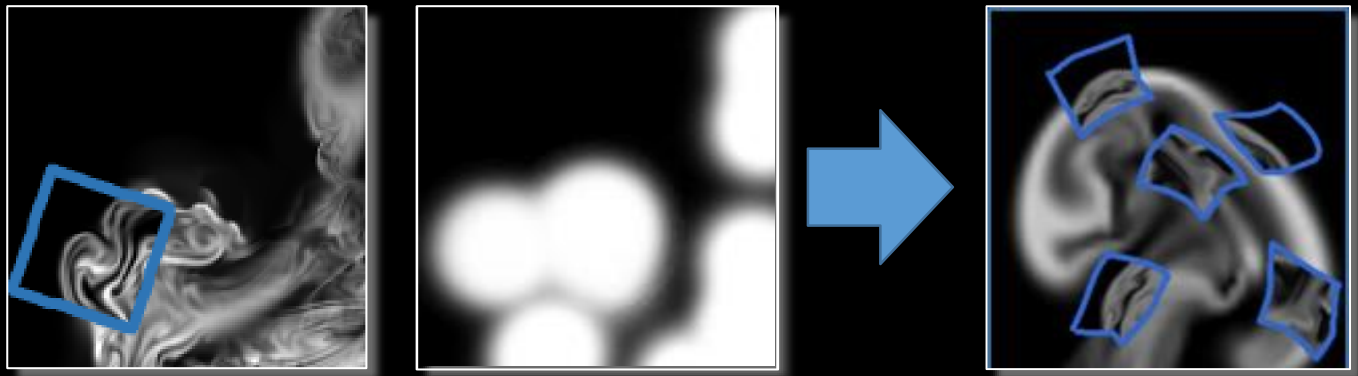
- Forward pass
  - Sampling, matching
  - Forward advection
  - Fading out ill-suited
- Backward pass
  - Backward anticipation & advection



Synthesis

Simulation:

- Forward pass
  - Sampling, matching
  - Forward advection
  - Fading out ill-suited
- Backward pass
  - Backward anticipation & advection



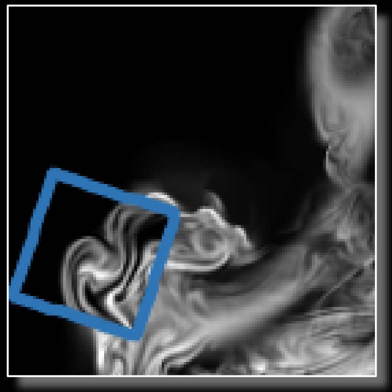
Advantages:

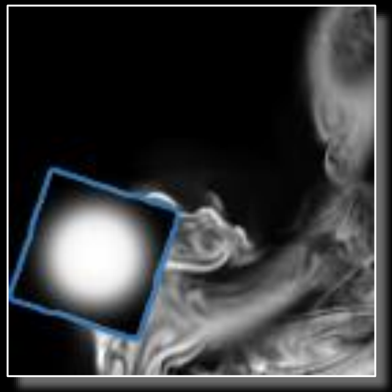
- Calculation: Coarse resolution
- Storage:
  - Descriptors only
  - Output: patch ID, cage vertices' pos, fading weights

Synthesis

Rendering:

- Loading patches,
  - fading weights
  - spatial weights



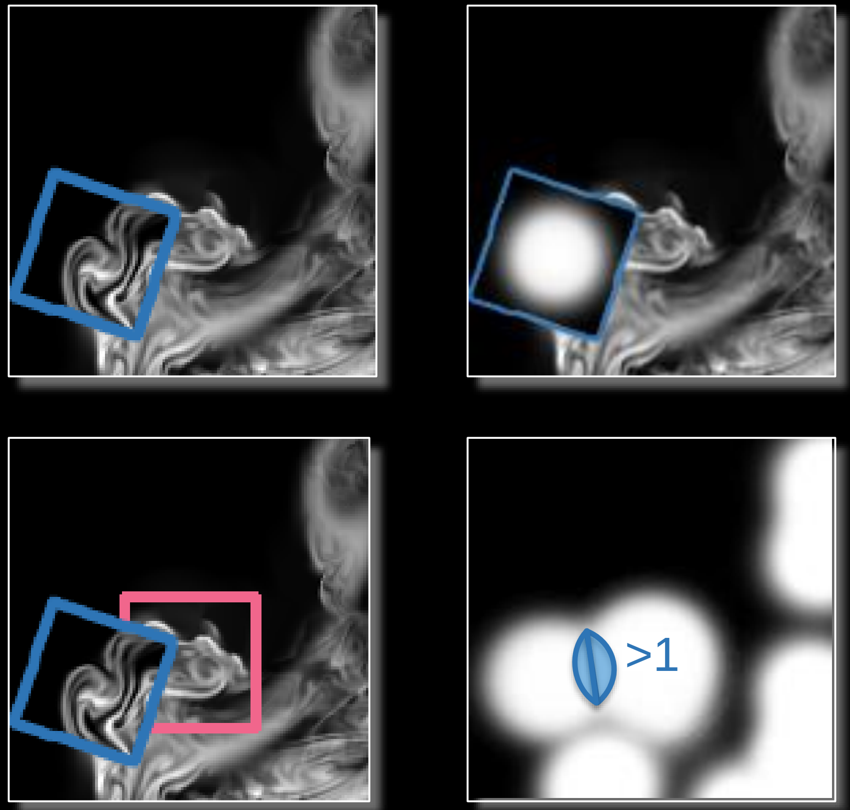


Synthesis

Rendering:

- Loading patches,
  - fading weights
  - spatial weights

- Normalization

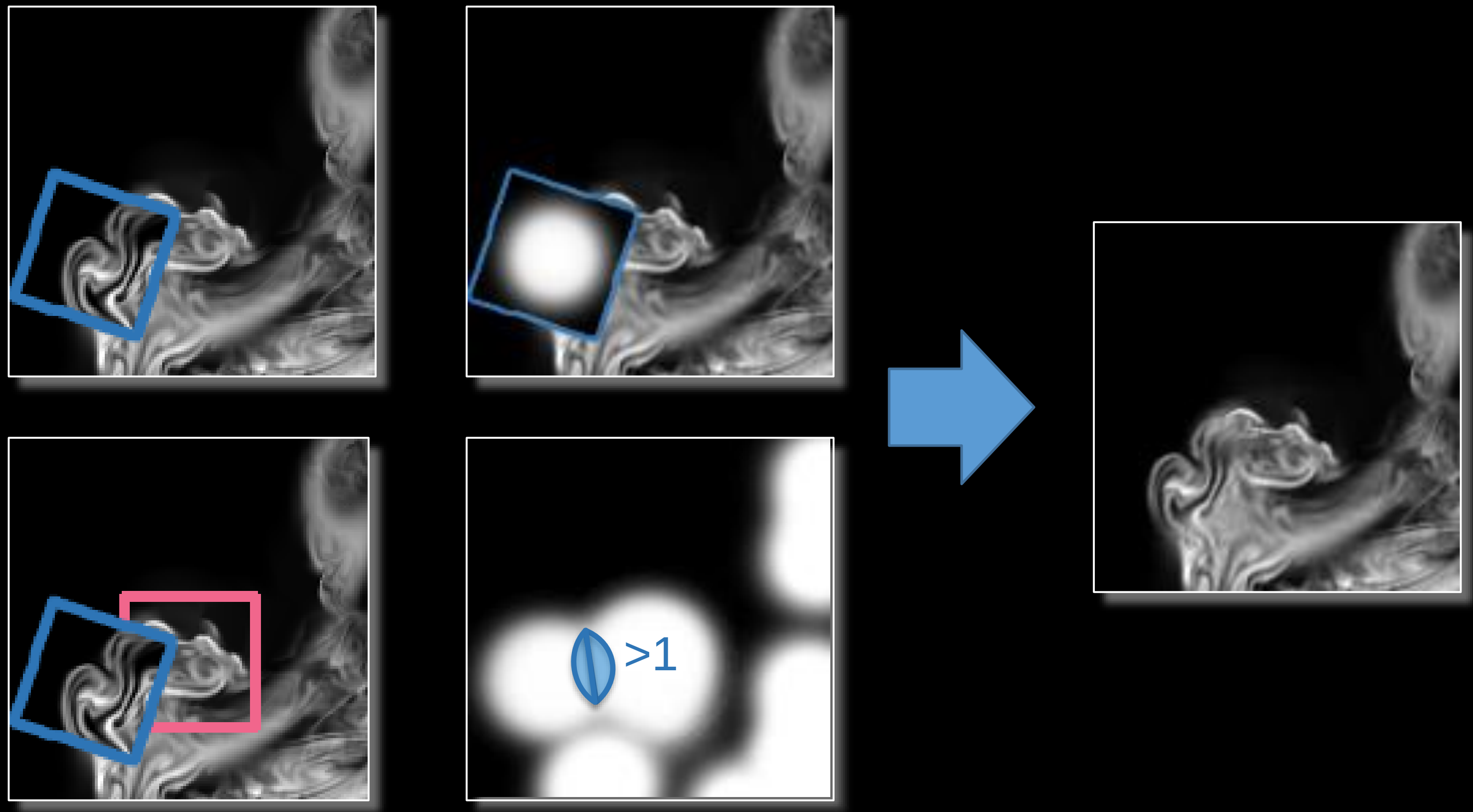


Synthesis

Rendering:

- Loading patches,
  - fading weights
  - spatial weights

- Normalization

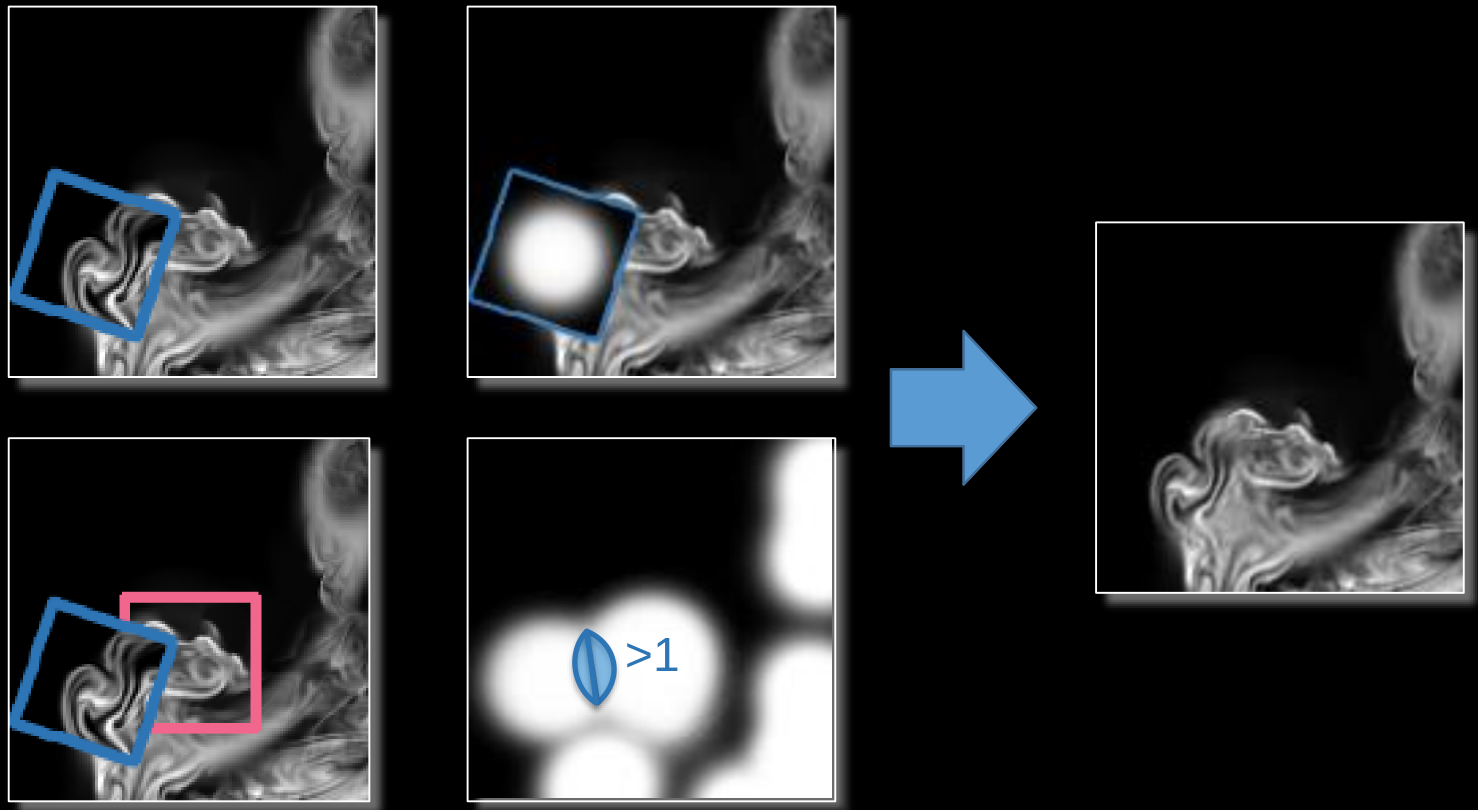


Synthesis

Rendering:

- Loading patches,
  - fading weights
  - spatial weights

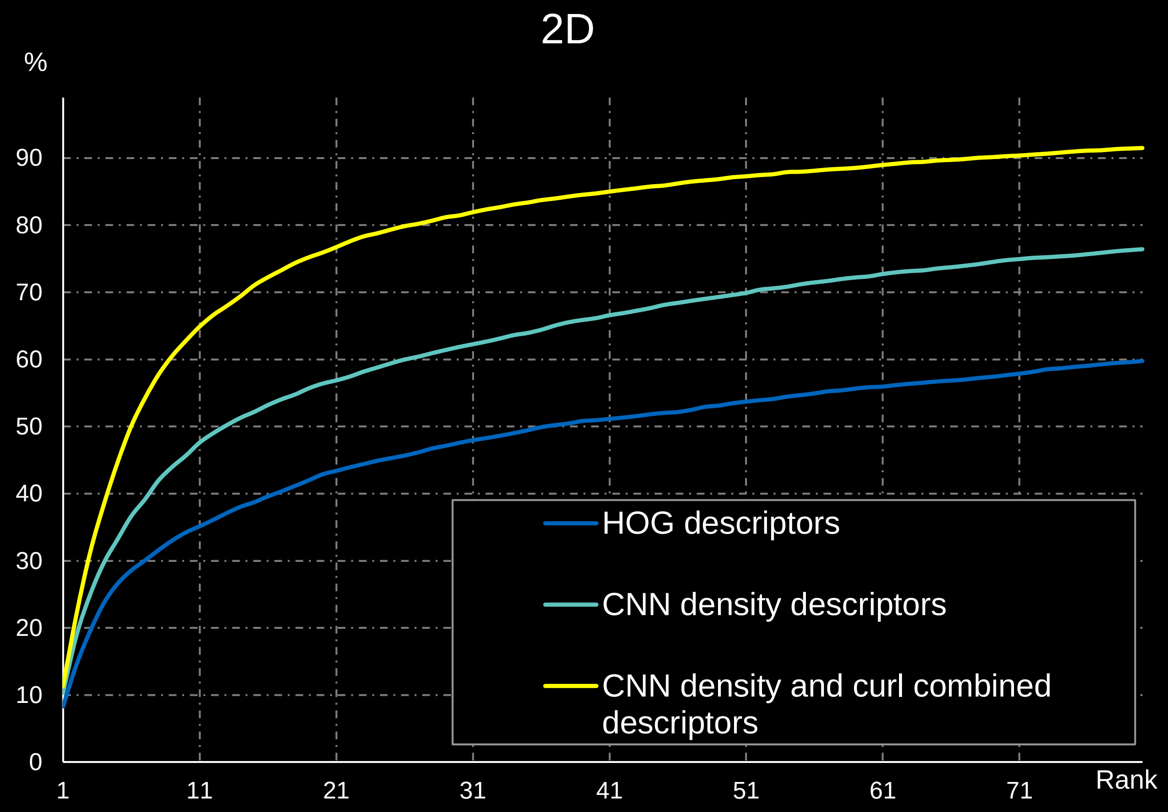
- Normalization
- Independent frames

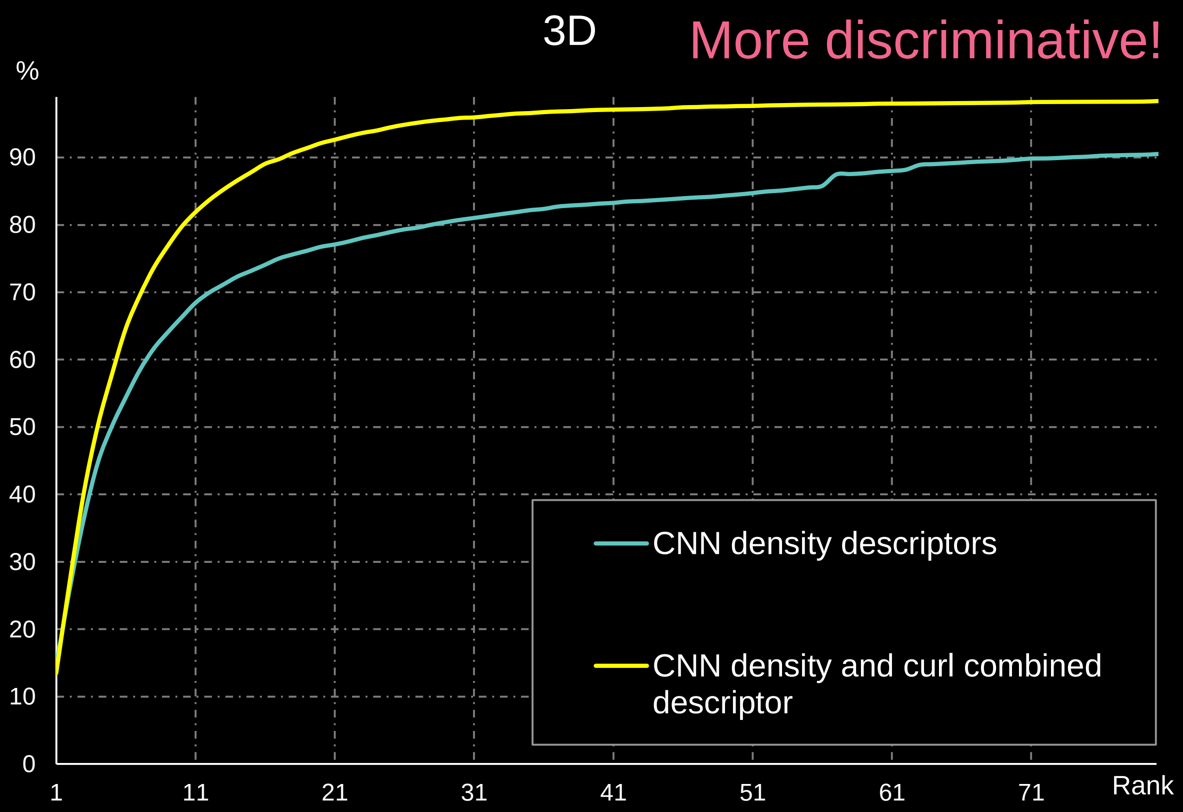


Evaluation

Descriptor evaluation

- Recall over rank —— the percentage of correctly matched pairs within a given rank





More discriminative!

Descriptor evaluation

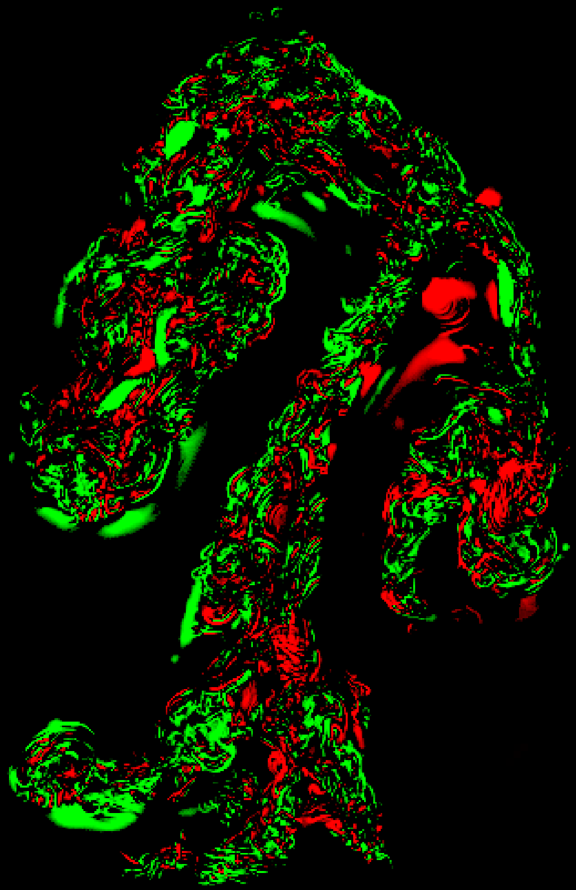
Input

Density descriptor only

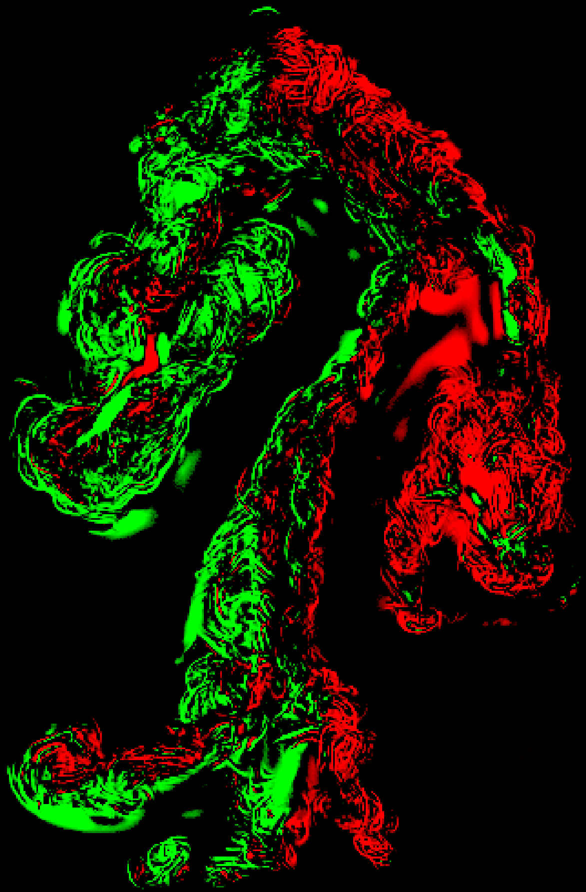
Density and curl descriptors

Descriptor evaluation

Input

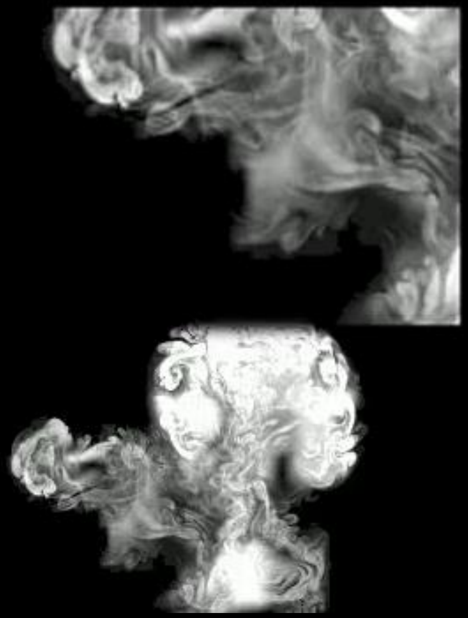


Density descriptor only

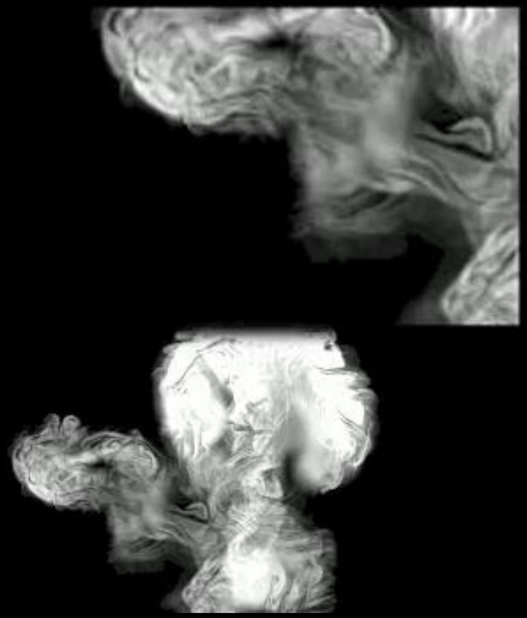


Density and curl descriptors

Descriptor evaluation



Density descriptor only

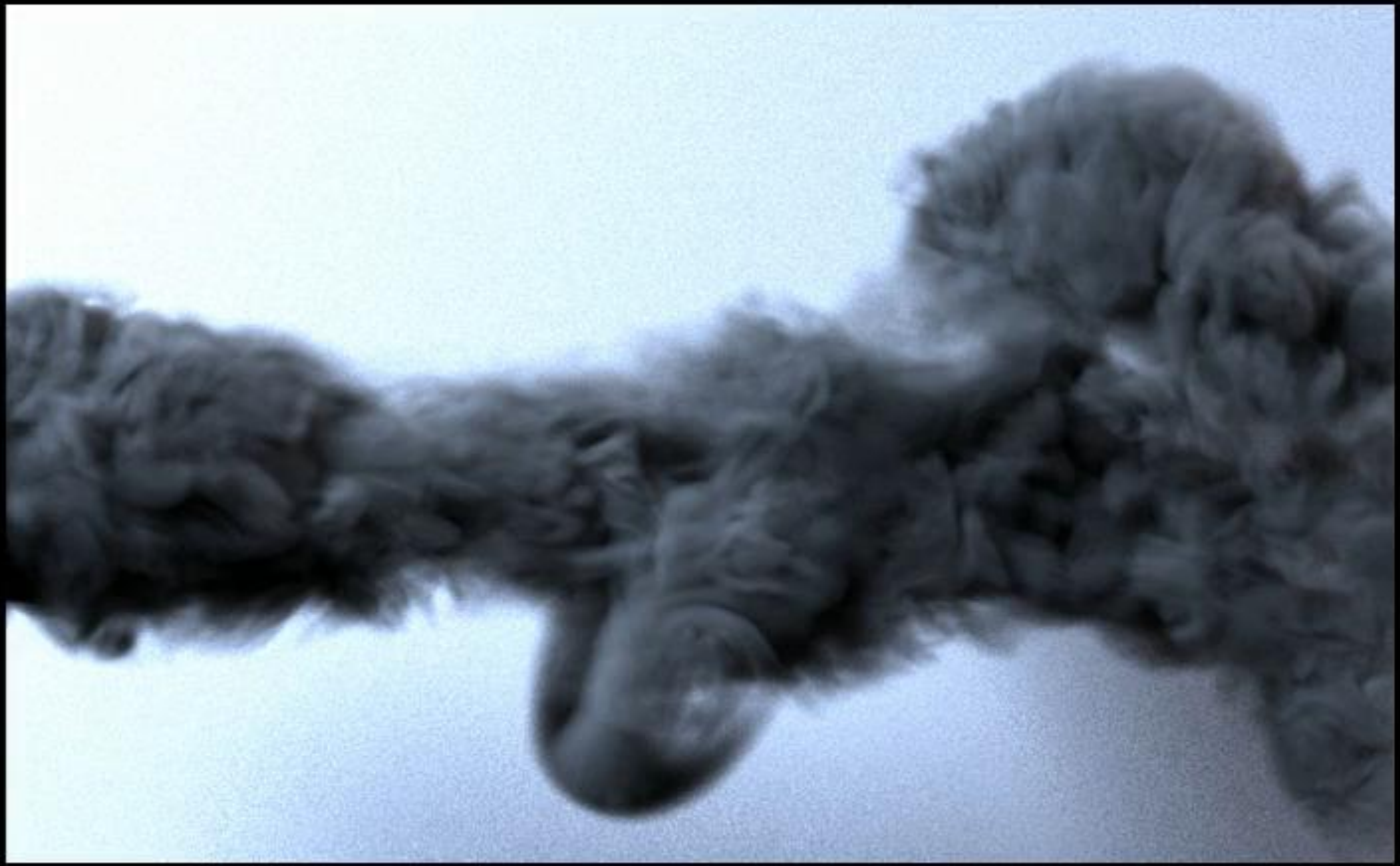


Density and curl descriptors

More results

Horizontal Plume

Resolution: 108x60x60
Avg. no of patches: 388
Avg. time per frame: 5.3s

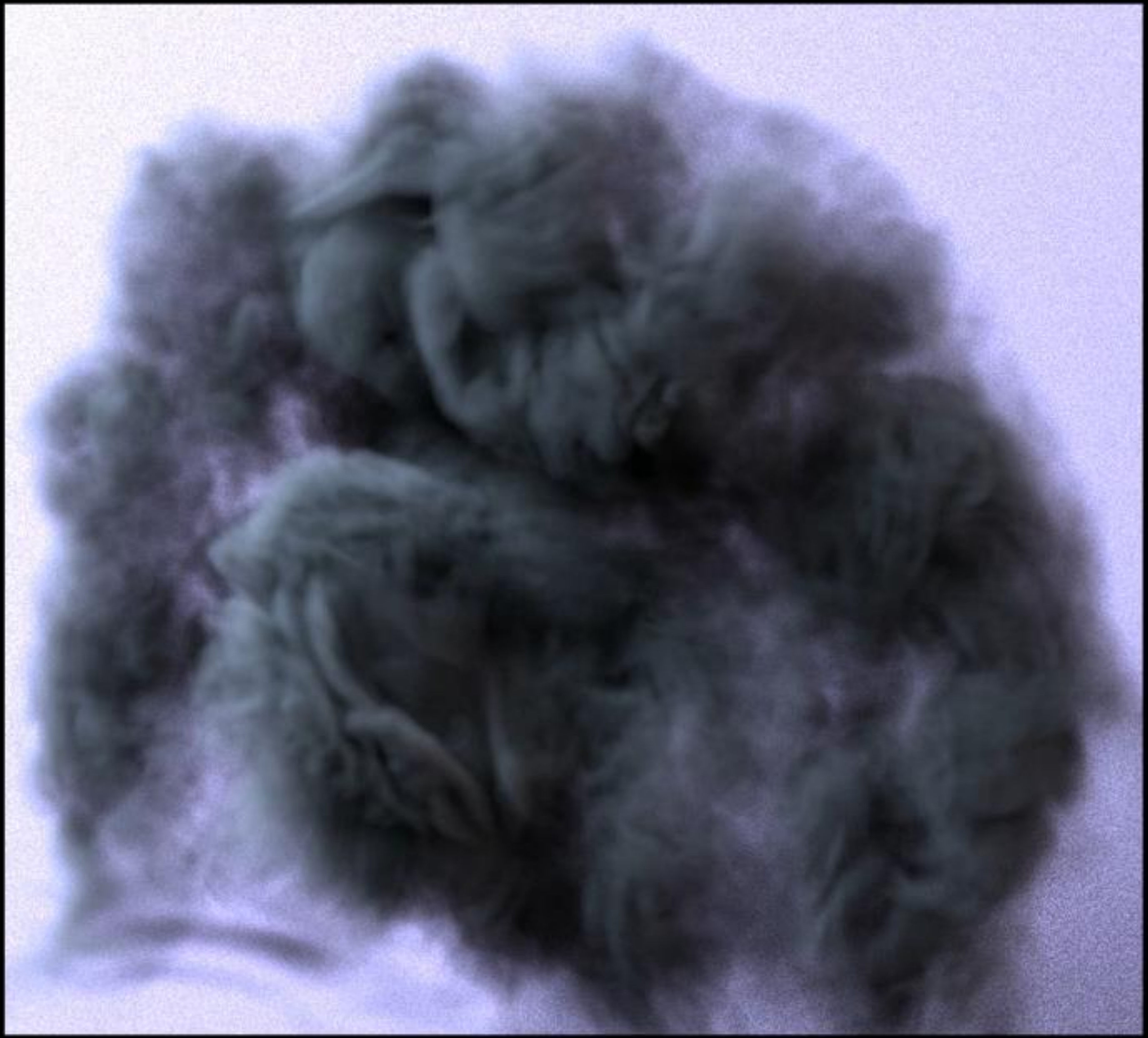


Obstacle Flow

Resolution: 76x64x64
Avg. no of patches: 362
Avg. time per frame: 3.9s

Colliding Jets

Resolution: 90x60x60
Avg. no of patches: 486
Avg. time per frame: 4.0s




Wavelet turbulence
Ours

Conclusion

Discussions

- Contributions
  - CNN fluid descriptors
  - Patch advection
  - Fluid repository
  - Synthesis
- Limitations
  - Fully divergence-free
    - Velocity synthesis
  - Spatial blending
  - Storage

Future directions

Neural networks

- More data-driven approaches
- Neural networks

Thank you!

More information: http://ge.in.tum.de/publications/2017-sig-chu/

Code online: https://github.com/RachelCmy/mantaPatch/