Riemannian Flow Matching for Interferometric SAR

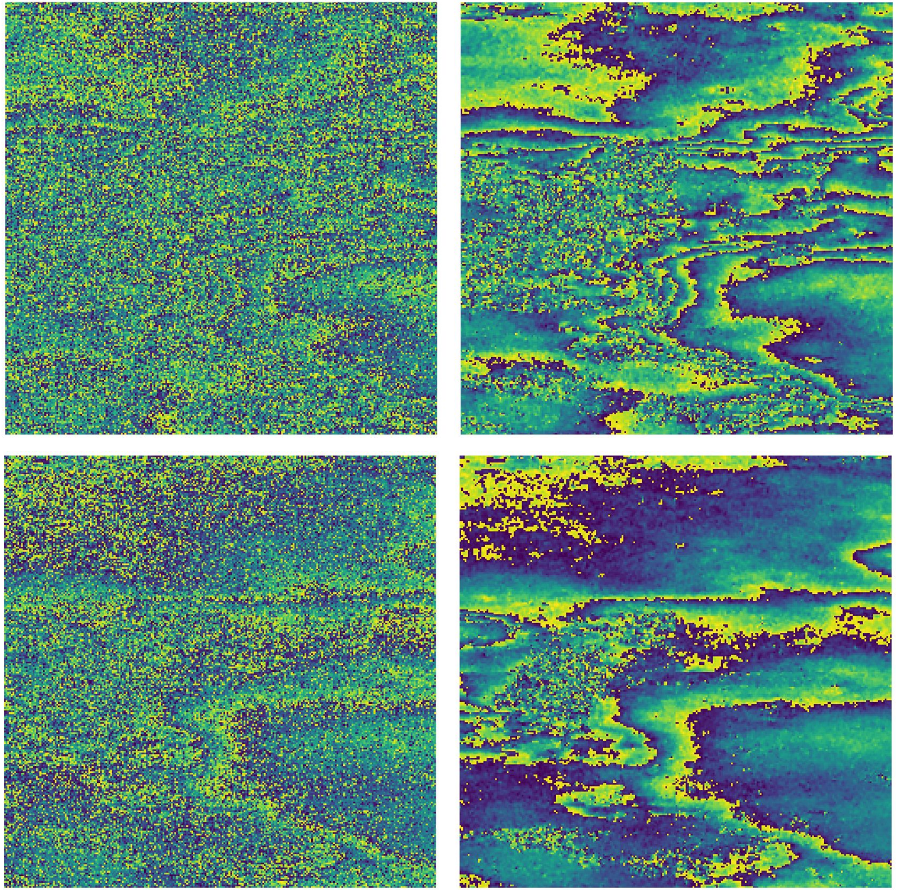
Georges Le Bellier[1]
@lebellig.bsky.social

Dana El Hajjar[3]

Arnaud Breloy[1]

Nicolas Audebert[1,2]

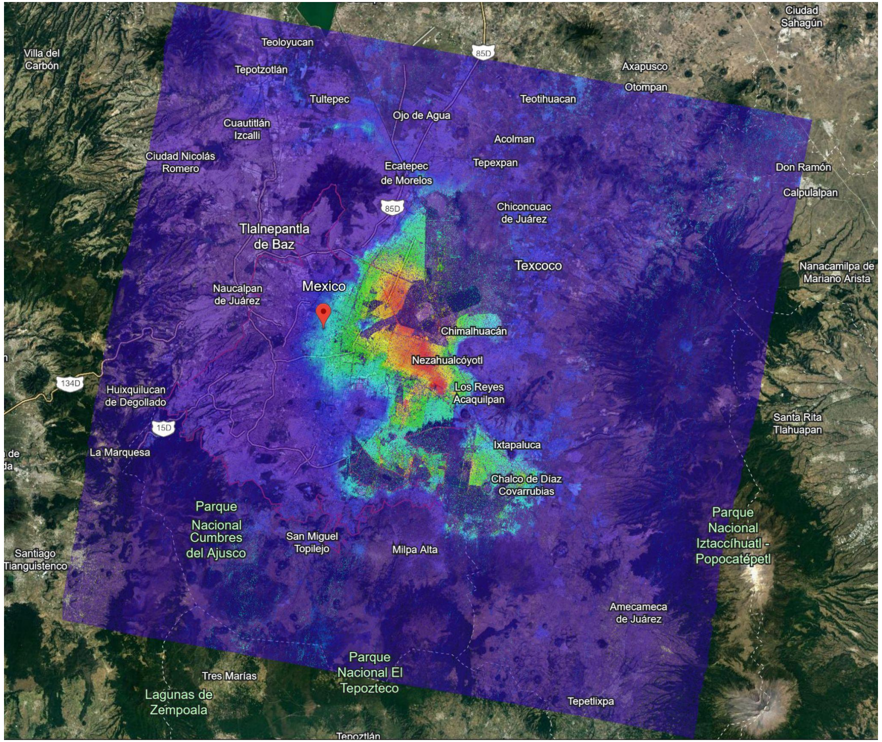
1. CNAM
2. Université Gustave Eiffel, IGN-ENSG
3. Université Savoie Mont Blanc



Introduction

Goal: ground subsidence monitoring

- large scale & long term
- ground-based methods are costly and have limited spatial coverage



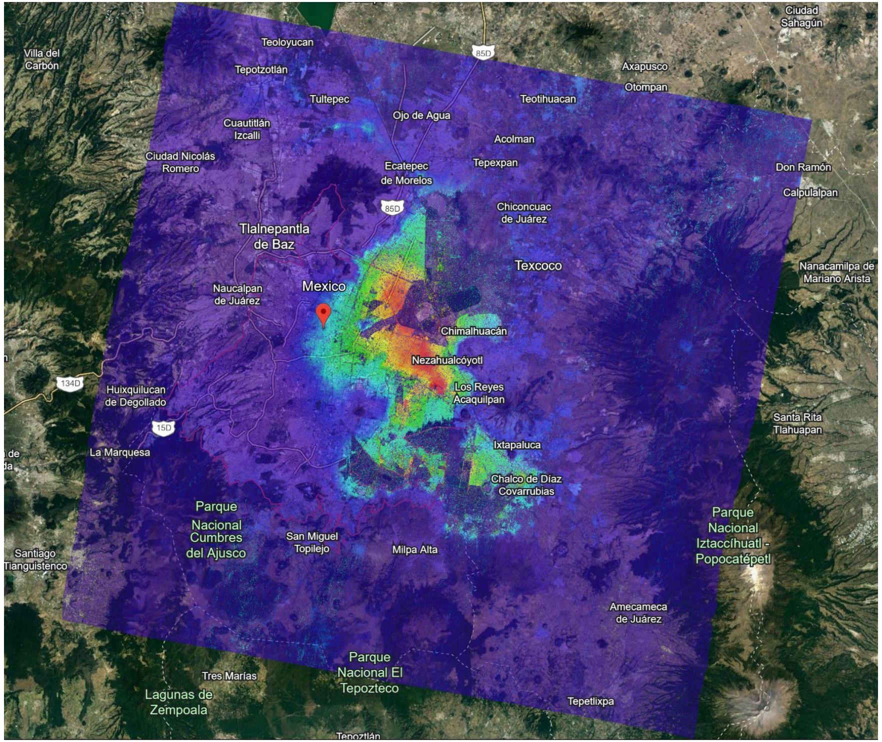
Introduction

Goal: ground subsidence monitoring

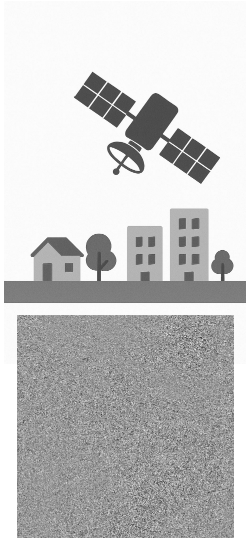
- large scale & long term
- ground-based methods are costly and have limited spatial coverage

Interferometry with SAR (InSAR)

- covers large areas
- unaffected by weather
- sub-centimeter accuracy

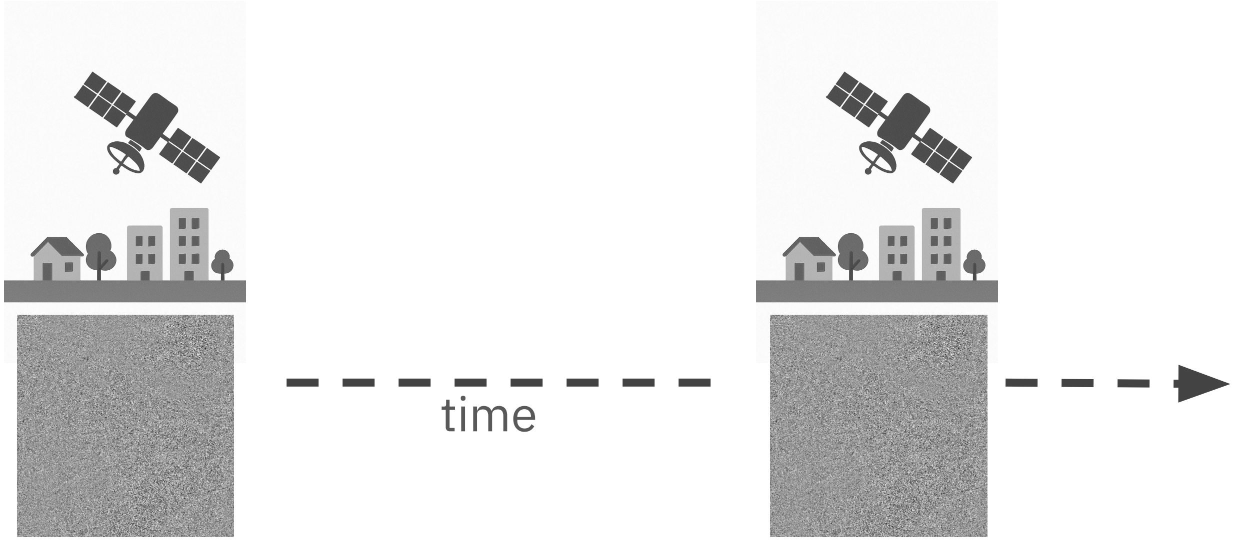


InSAR



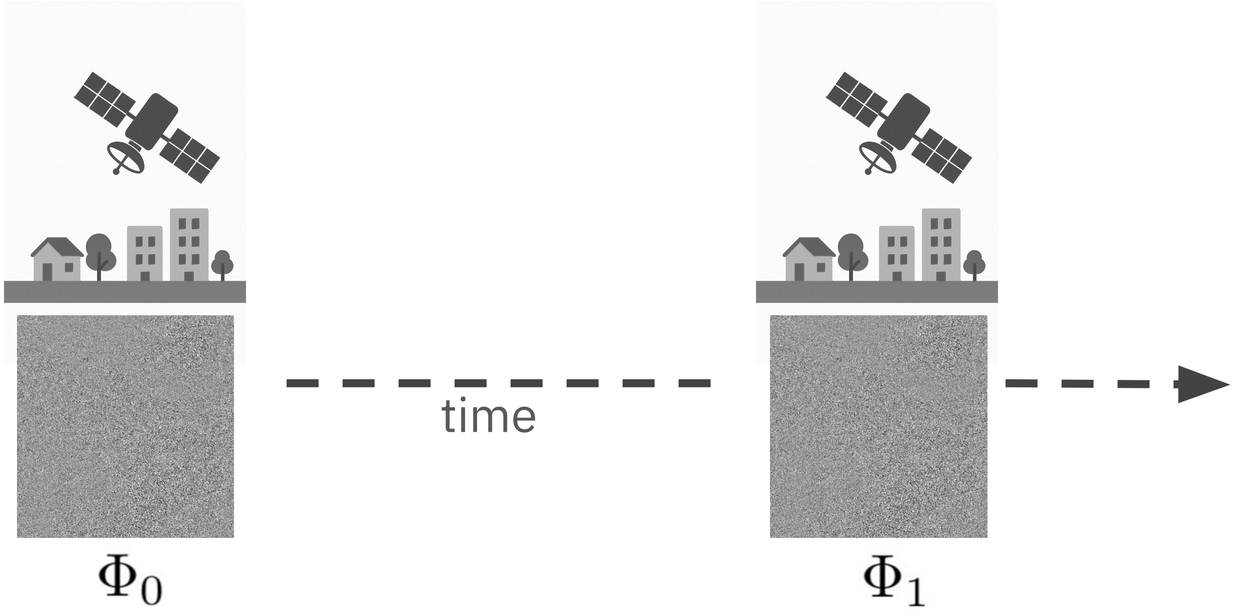
Φ_0

InSAR



Φ_0 Φ_1

InSAR



Interferograms

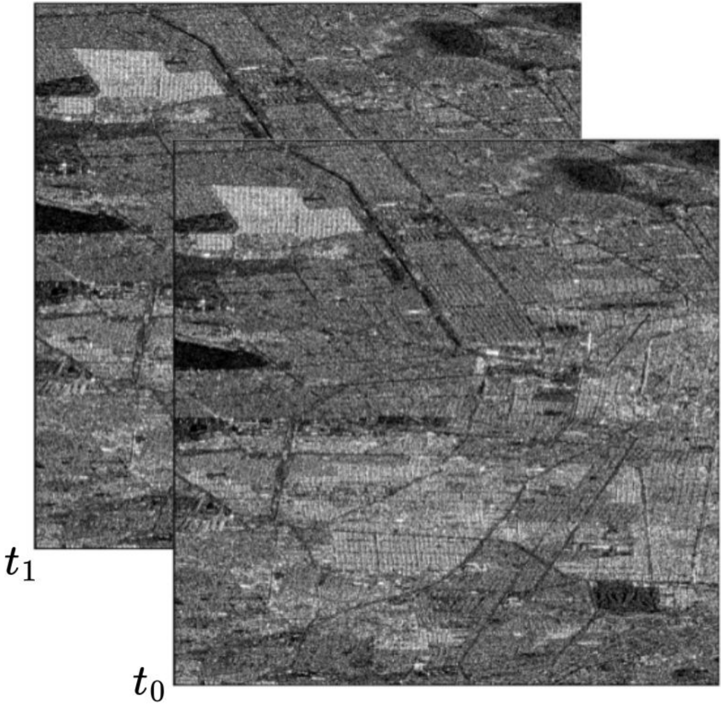
$$\mathbf{w} = \Delta\Phi = \Delta\Phi_{disp} + \Delta\Phi_{topo} + \Delta\Phi_{orb} + \Delta\Phi_{atm} + \Delta\Phi_{noise}$$

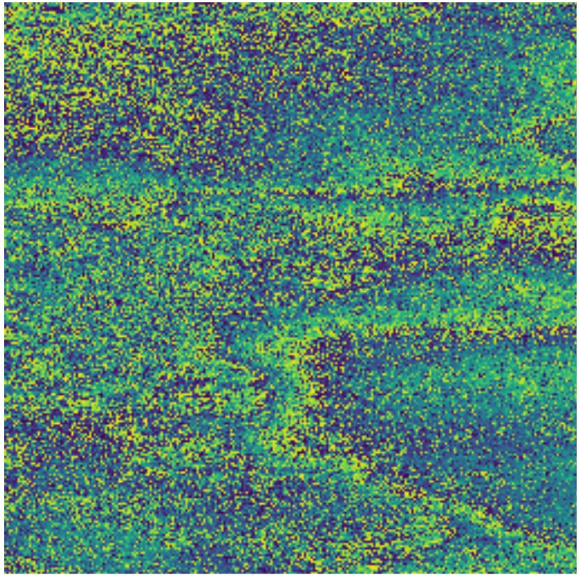
surface displacement

removed with preprocessing

we want to remove it

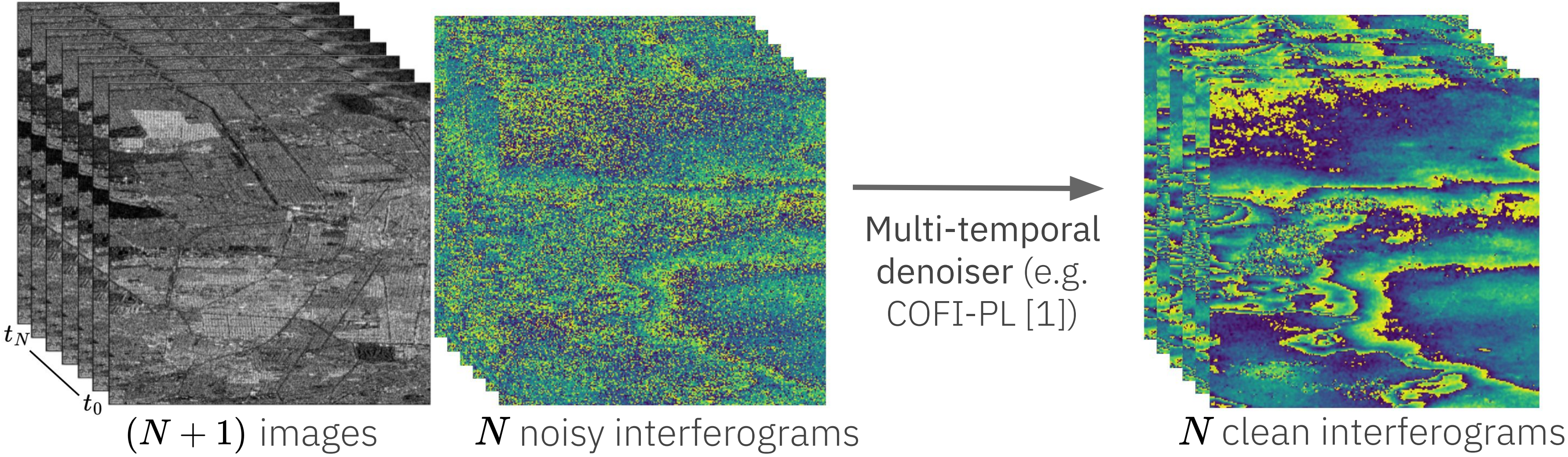
InSAR





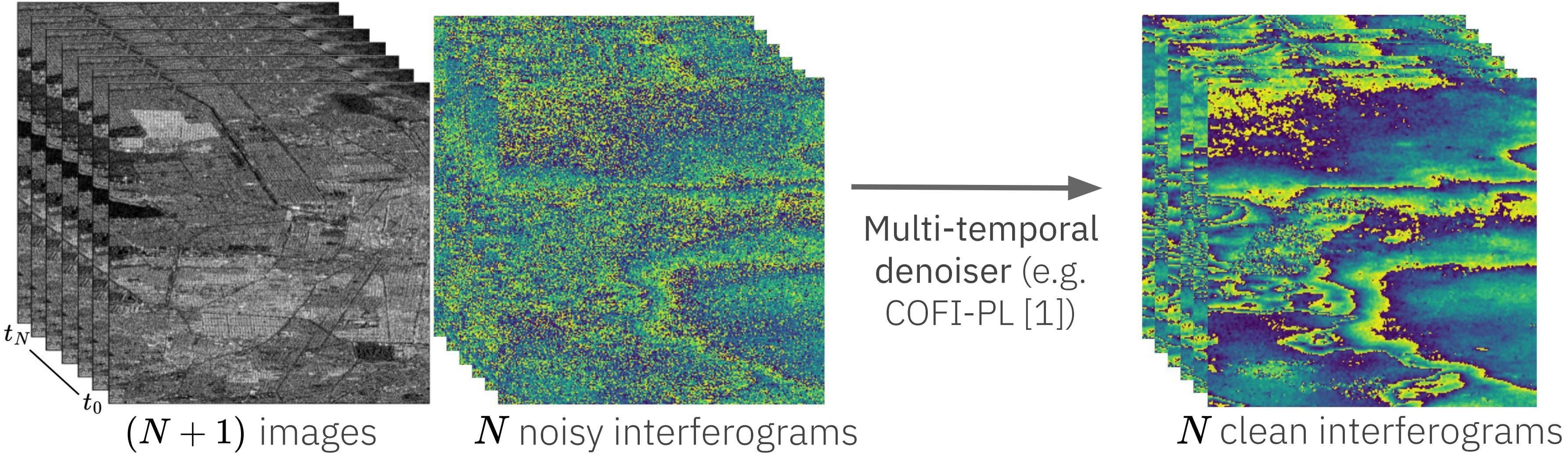
Noisy interferogram $\mathbf{w}$

Denoising



[1] Covariance Fitting Interferometric Phase Linking: Modular Framework and Optimization Algorithms, Phan Viet Hoa Vu et al.

Denoising

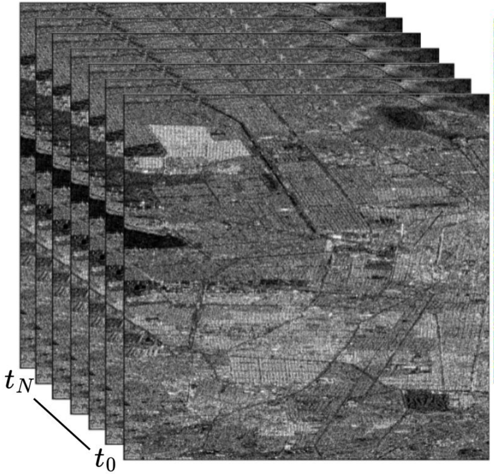


Multi-temporal InSAR:

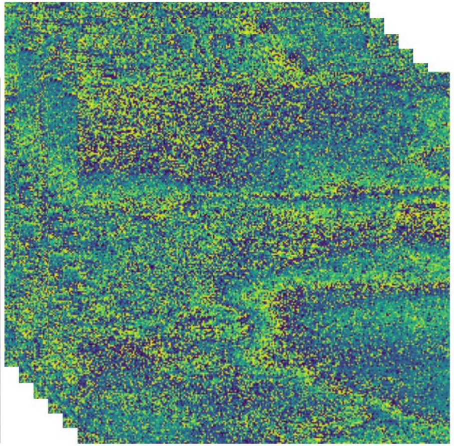
- clean denoised interferograms
- leverage temporal dependencies

[1] Covariance Fitting Interferometric Phase Linking: Modular Framework and Optimization Algorithms, Phan Viet Hoa Vu et al.

Denoising

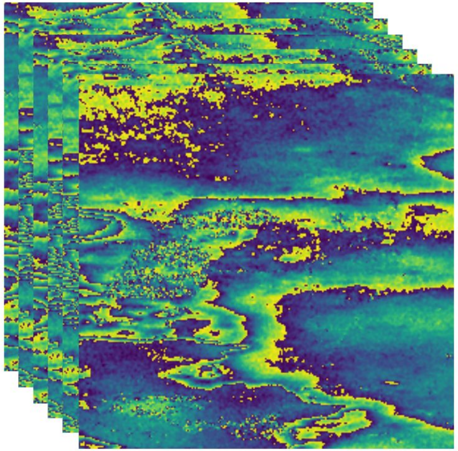


$(N+1)$ images



N noisy interferograms

Multi-temporal denoiser (e.g. COFI-PL [1])



N clean interferograms

Multi-temporal InSAR:

- clean denoised interferograms
- leverage temporal dependencies

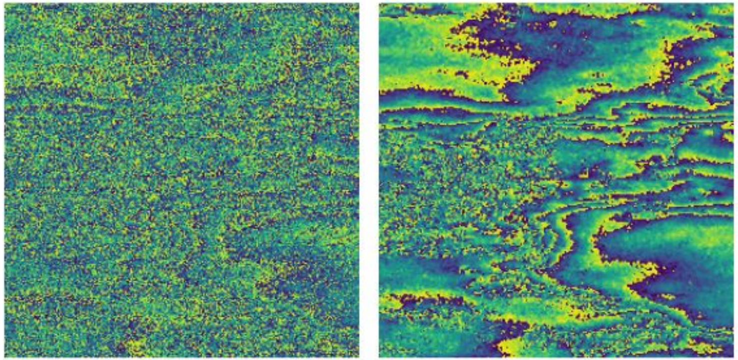
But...

- you need the full timeseries
- computational cost

[1] Covariance Fitting Interferometric Phase Linking: Modular Framework and Optimization Algorithms, Phan Viet Hoa Vu et al.

Questions?

1. Denoising

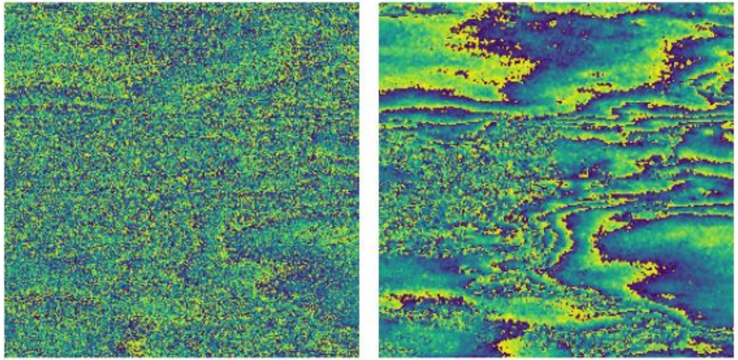


Can we generate the output of the multi-temporal algorithm from a **unique noisy interferogram?**

No need for the full timeseries

Questions?

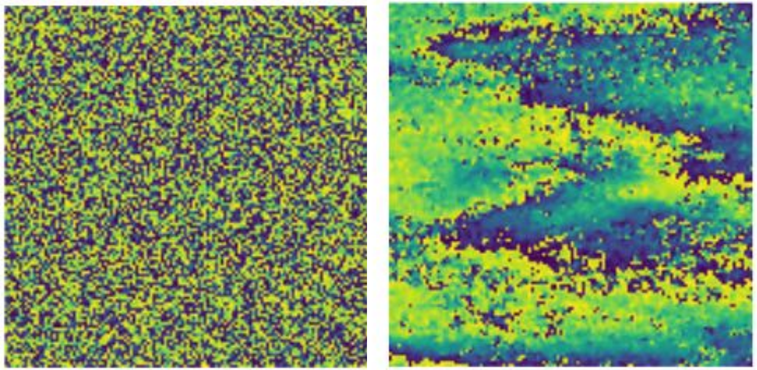
1. Denoising



Can we generate the output of the multi-temporal algorithm from a **unique noisy interferogram?**

No need for the full timeseries

2. Generation



Can we generate **valid** $\mathbf{w}_i \in [0, 2\pi[$ interferograms from noise?

Data augmentation/dataset generation

Flow Matching

- Generalization of diffusion models introduced in 2022 [2, 3, 4]
- Bridge arbitrary distributions p_0 and p_1 by learning a velocity field $u_t(\cdot)$

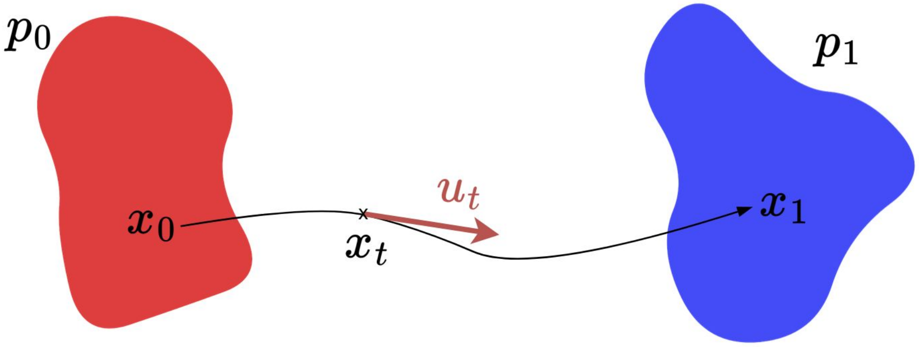
$$\frac{d}{dt}\varphi_t(x) = u_t(\varphi_t(x))$$

[2] Flow Matching for Generative Modeling, Y.Lipman et al.
[3] Building Normalizing Flows with Stochastic Interpolants, M.Albergo et al.
[4] Non-Denoising Forward-Time Diffusions, S.Peluchetti

Flow Matching

- Generalization of diffusion models introduced in 2022 [2, 3, 4]
- Bridge arbitrary distributions p_0 and p_1 by learning a velocity field $u_t(\cdot)$

$$\frac{d}{dt}\varphi_t(x) = u_t(\varphi_t(x))$$



$$x_1 = \varphi_1(x_0) = \mathrm{ODE}_u(x_0, t : 0 \to 1)$$

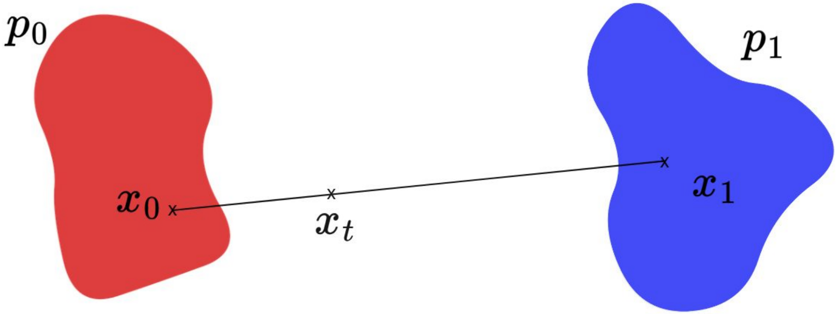
[2] Flow Matching for Generative Modeling, Y.Lipman et al.
[3] Building Normalizing Flows with Stochastic Interpolants, M.Albergo et al.
[4] Non-Denoising Forward-Time Diffusions, S.Peluchetti

Flow Matching

1. Sample $(x_0, x_1) \sim p(x_0, x_1)$

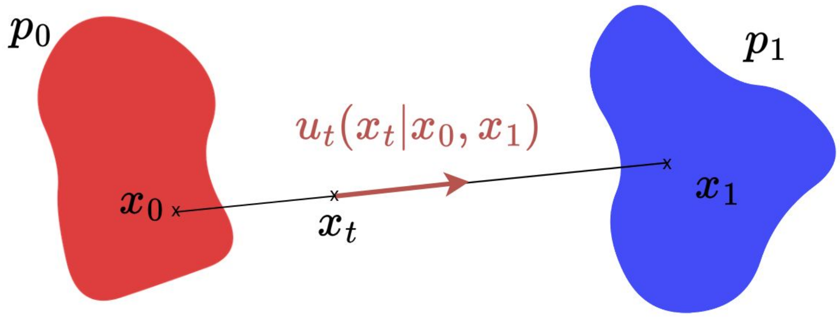
Flow Matching

1. Sample $(x_0, x_1) \sim p(x_0, x_1)$
2. Sample time $t \sim U(0,1)$
3. Interpolant $x_t = (1-t)x_0 + tx_1$



Flow Matching

1. Sample $(x_0, x_1) \sim p(x_0, x_1)$
2. Sample time $t \sim U(0,1)$
3. Interpolant $x_t = (1-t)x_0 + tx_1$
4. Simple regression loss



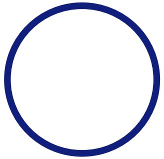
$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}\|v_\theta(t, x_t) - u_t(x_t|x_0, x_1)\|^2$$

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}\|v_\theta(t, x_t) - (x_1 - x_0)\|^2$$

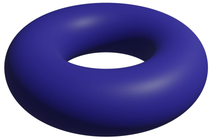
Riemannian Manifold

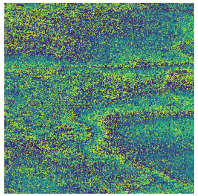
Interferogram's pixels are in $[0, 2\pi[$

1-pixel interferogram

$\in$  $= S_1$

2-pixel interferogram

$\in$  $= (S_1)^2 = \mathbb{T}_2$



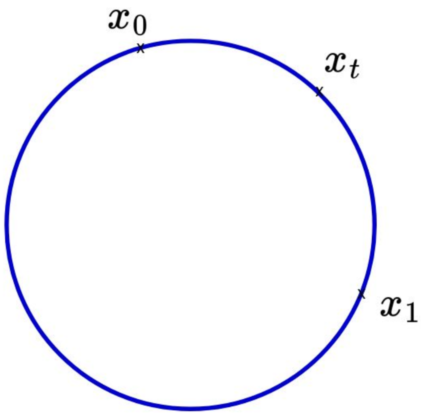
d-pixel interferogram

$\in$ ... $= (S_1)^d = \mathbb{T}_d$

Riemannian Flow Matching [4]

1. Sample $(x_0, x_1) \sim p(x_0, x_1)$
2. Sample time $t \sim U(0, 1)$
3. Interpolant $x_t = \exp_{x_1}\left(\kappa(t) \log_{x_1}(x_0)\right)$

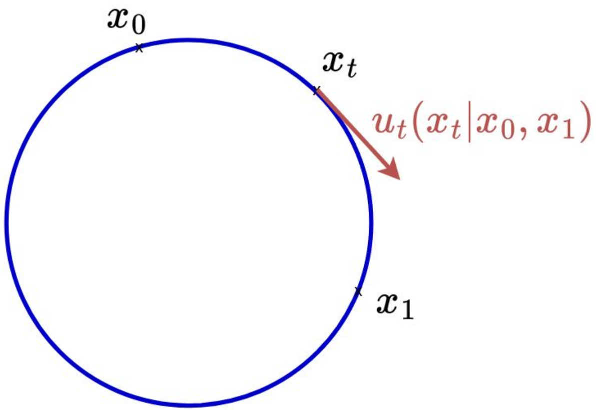
use the geodesic on the torus



[5] Flow Matching on General Geometries, R.T.Q Chen and Y.Lipman.

Riemannian Flow Matching [4]

1. Sample $(x_0, x_1) \sim p(x_0, x_1)$
2. Sample time $t \sim U(0,1)$
3. Interpolant $x_t = \exp_{x_1}\left(\kappa(t)\log_{x_1}(x_0)\right)$
4. Simple regression loss



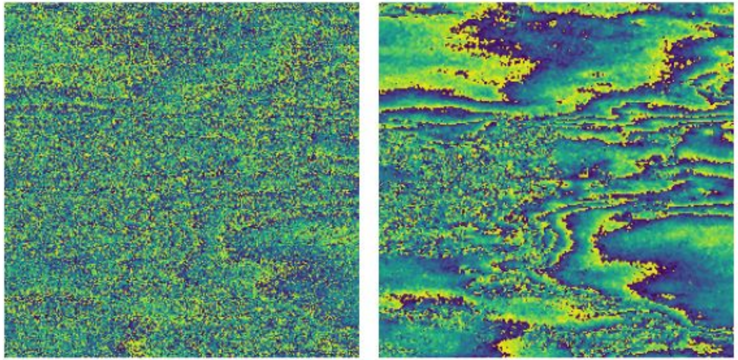
$$\mathcal{L}_{\text{RFM}}(\theta) = \mathbb{E}\|v_\theta(t, x_t) - u_t(x_t|x_0, x_1)\|^2$$

$$\mathcal{L}_{\text{RFM}}(\theta) = \mathbb{E}\|v_\theta(t, x_t) - \dot{x}_t\|^2$$

[5] Flow Matching on General Geometries, R.T.Q Chen and Y.Lipman.

Experiments

1. Denoising



p_0 : noisy interferograms

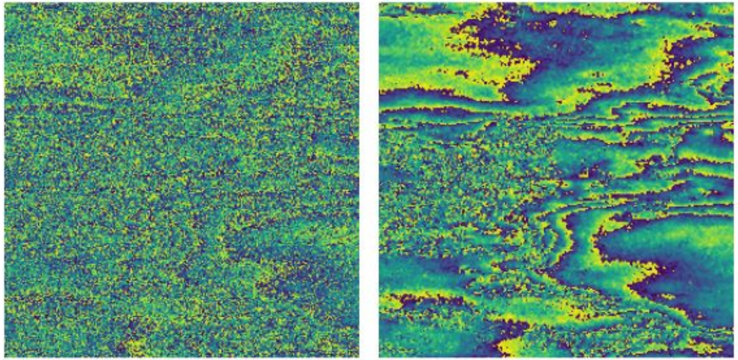
p_1 : interferograms denoised by COFI-PL

$p(x_0, x_1)$: clean/noisy pairs [5]

[6] Stochastic interpolants with data-dependent couplings, M.Albergo et al

Experiments

1. Denoising

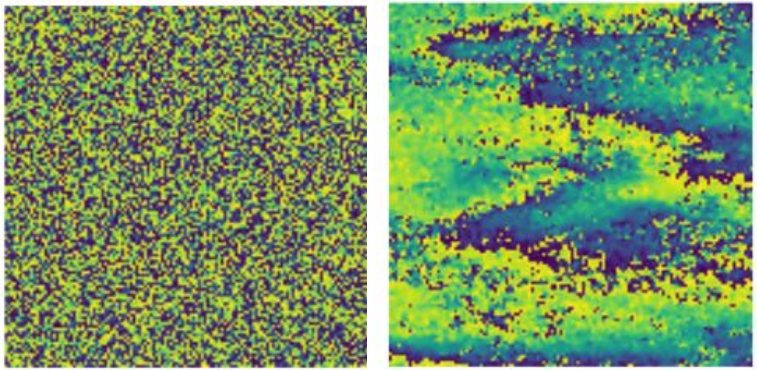


p_0 : noisy interferograms

p_1 : interferograms denoised by COFI-PL

$p(x_0, x_1)$: clean/noisy pairs [5]

2. Generation



p_0 : wrapped gaussian

p_1 : interferograms denoised by COFI-PL

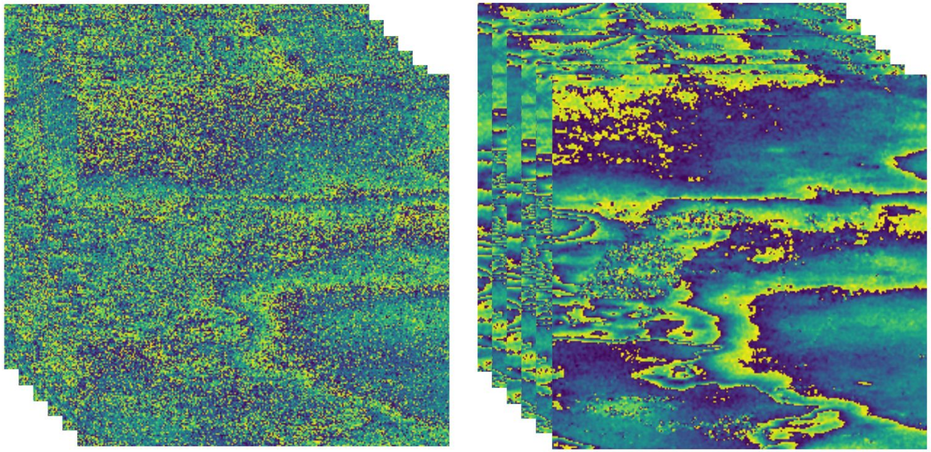
$p(x_0, x_1)$: independent

[6] Stochastic interpolants with data-dependent couplings, M.Albergo et al

Implementation

Dataset:

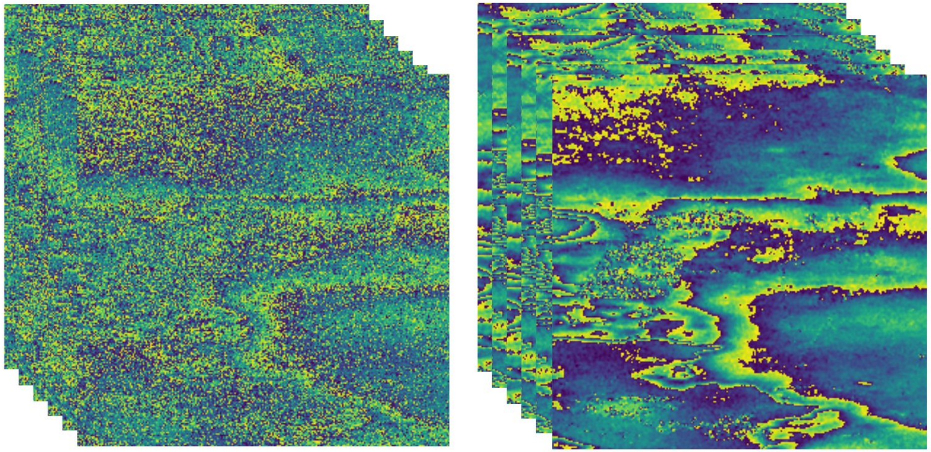
- 40 SAR Sentinel-1 SLC images of Mexico City
- Interferograms cleaned with COFI-PL
- Every 12 day between 14/08/19 and 6/12/20
- Downsampled (x4)
- 128x128px patches
- Spatial split between train/test



Implementation

Dataset:

- 40 SAR Sentinel-1 SLC images of Mexico City
- Interferograms cleaned with COFI-PL
- Every 12 day between 14/08/19 and 6/12/20
- Downsampled (x4)
- 128x128px patches
- Spatial split between train/test



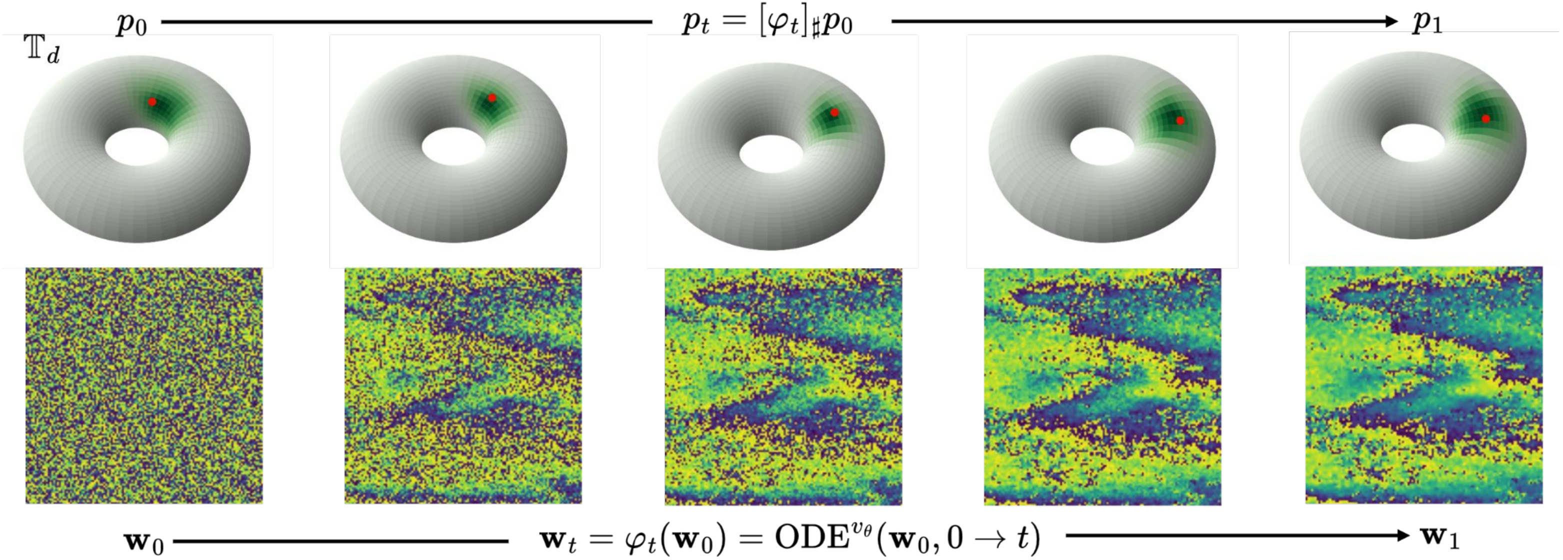
Neural Network:

- U-Net backbone (60M params)
- 100 000 training steps with batch size of 32
- 50 sampling steps (Euler solver)

Generation

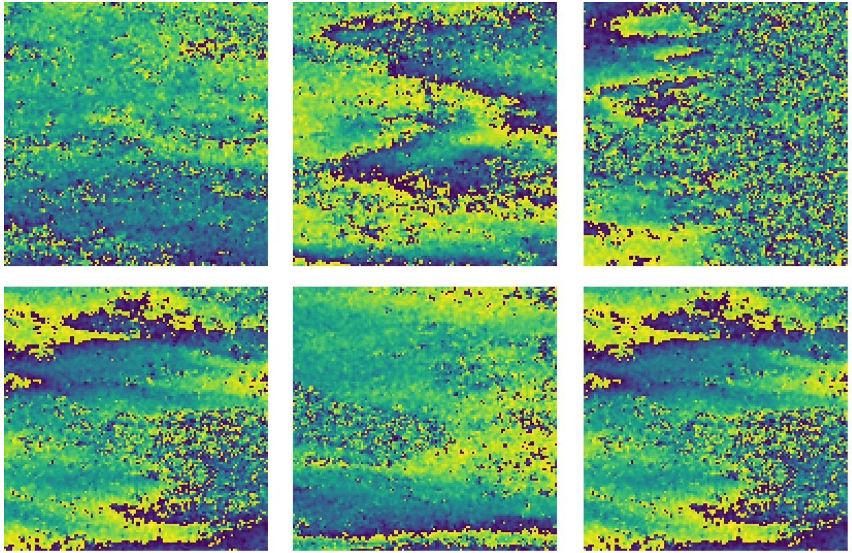
p_0 : wrapped gaussian

p_1: clean interferograms



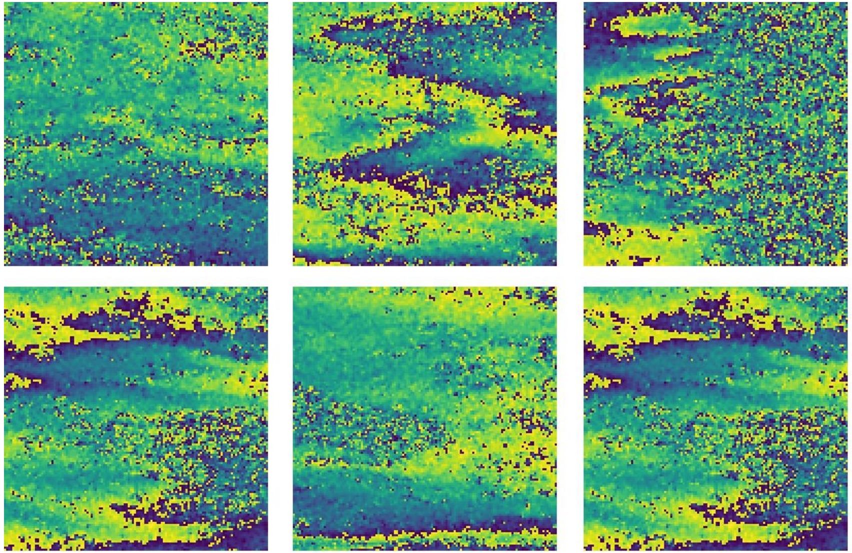
Generation

Riemannian Flow Matching



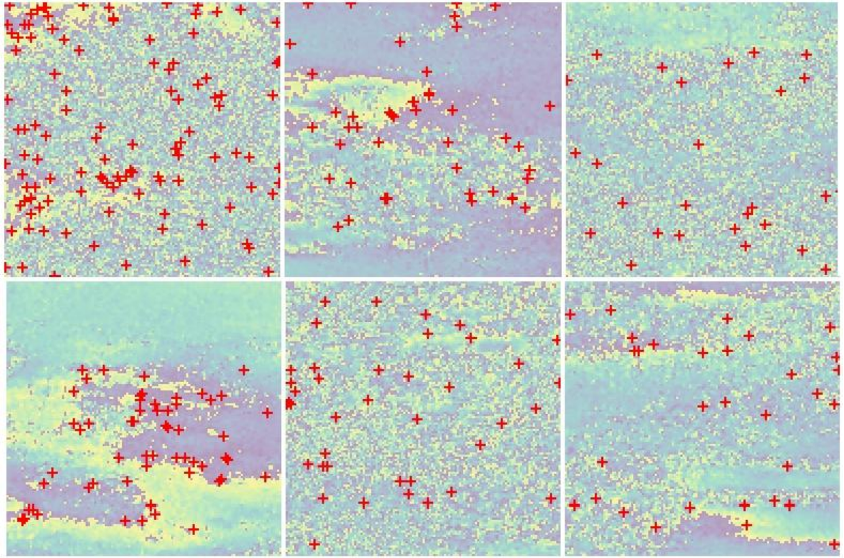
Generation

Riemannian Flow Matching



Valid interferograms

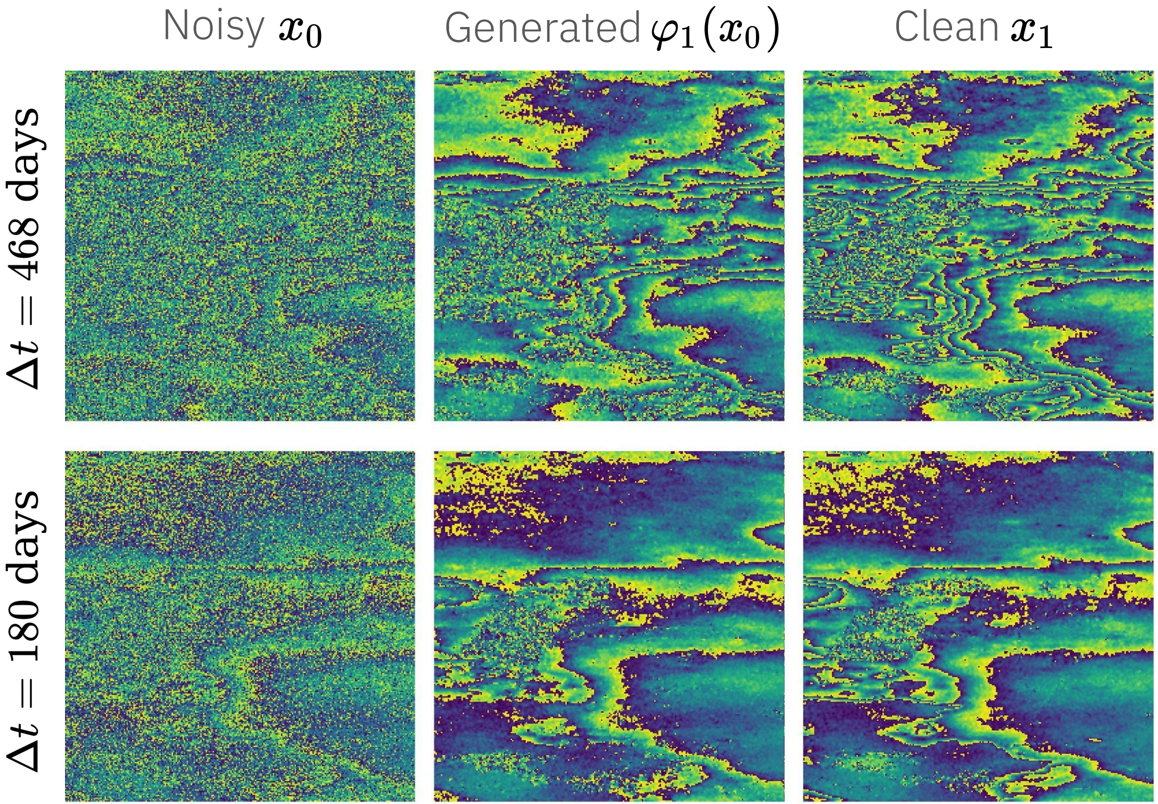
Euclidian Flow Matching



Invalid points $\mathbf{w}^{(i)} \notin [0, 2\pi[$

Denoising

test set



Conlusion

- Riemannian Flow Matching generates valid interferograms from noise.
- RFM produces expected patterns and textures.
- RFM generates interesting denoised interferograms from a single noisy phase difference.

Conlusion

- Riemannian Flow Matching generates valid interferograms from noise.
- RFM produces expected patterns and textures.
- RFM generates interesting denoised interferograms from a single noisy phase difference.

Next steps

- Evaluation on synthetic data for comparison with baselines.
- Condition RFM on the time period between two SAR images.
- Use multiple MT-denoisers as supervision.
- Can we learn the denoised interferograms without supervision?

Conlusion

- Riemannian Flow Matching generates valid interferograms from noise.
- RFM produces expected patterns and textures.
- RFM generates interesting denoised interferograms from a single noisy phase difference.

Next steps

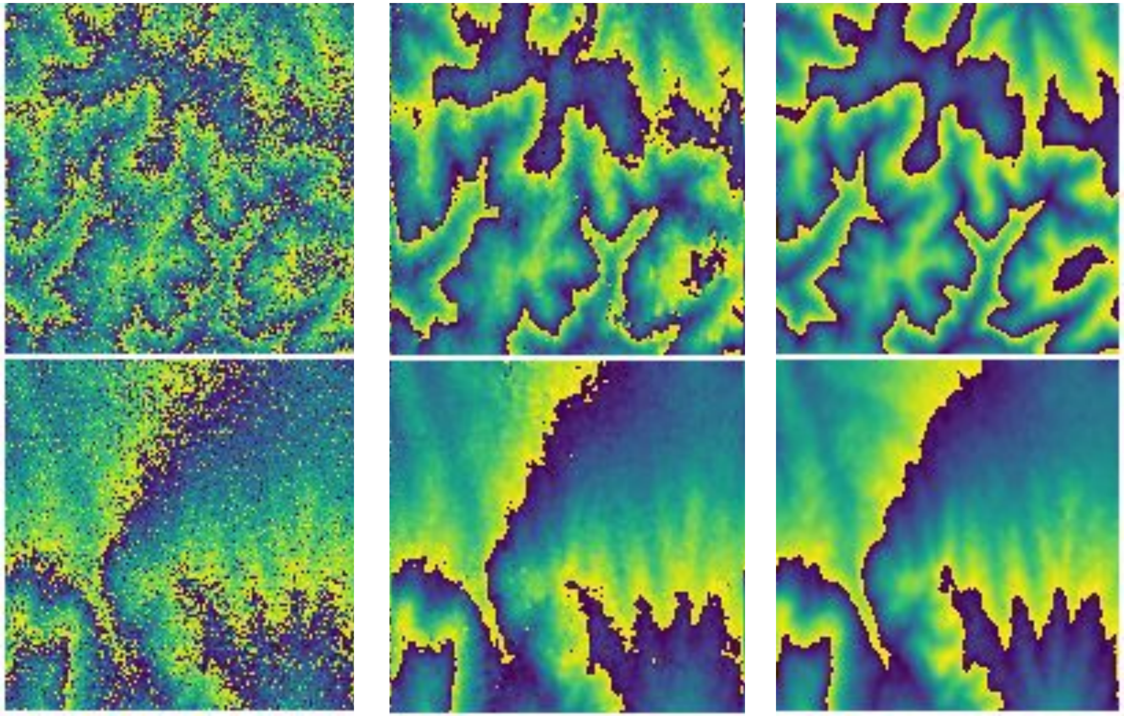
- Evaluation on synthetic data for comparison with baselines.
- Condition RFM on the time period between two SAR images.
- Use multiple MT-denoisers as supervision.
- Can we learn the denoised interferograms without supervision?

Questions?

Denoising

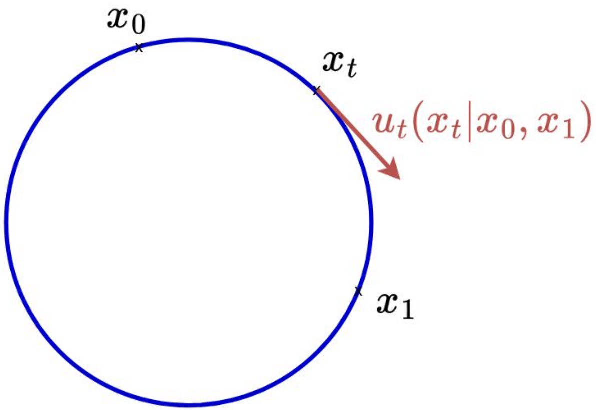
synthetic [4]

Noisy $\boldsymbol{x}_0$ Generated $\varphi_1(\boldsymbol{x}_0)$ Clean $\boldsymbol{x}_1$



[4] InSAR-MONet: Interferometric SAR Phase Denoising Using a Multiobjective, Neural Network, S. Vitale et al.

Riemannian Flow Matching [4]



1. Sample $(x_0, x_1) \sim p(x_0, x_1)$
2. Sample time $t \sim U(0, 1)$
3. Interpolant $x_t = \exp_{x_1}\left(\kappa(t) \log_{x_1}(x_0)\right)$

$$\exp_{\mathbf{w}}(\cdot) : T_{\mathbf{w}}\mathbb{T}_d \to \mathbb{T}_d$$
$$\log_{\mathbf{w}}(\cdot) : \mathbb{T}_d \to T_{\mathbf{w}}\mathbb{T}_d$$

$$\begin{cases} \exp_w(u) = (w + u) \pmod{2\pi} \\ \log_{w_a}(w_b) = \text{arctan2}(\sin(w_b - w_a), \cos(w_b - w_a)) \end{cases}$$