Use of a Deep-Learning-Based Geological Parameterization for Production Data Assimilation

Yimin Liu, Wenyue Sun, Louis J. Durlofsky

Department of Energy Resources Engineering

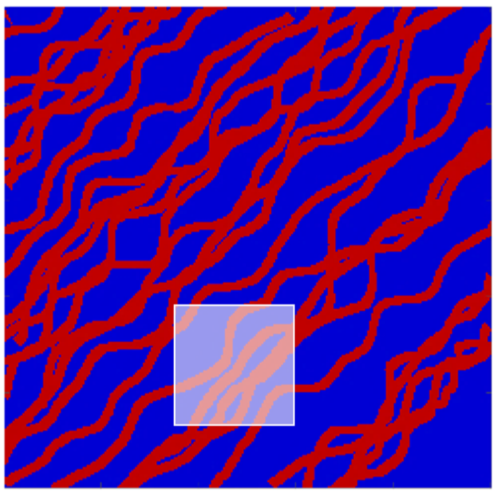
Stanford University

Outline

- PCA-based parameterization for history matching
- Convolutional neural network PCA (CNN-PCA)
- Results for model generation and flow statistics
- History matching results
- Summary and future directions

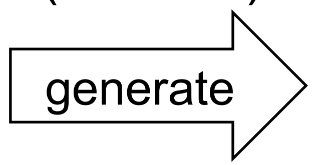
Geomodel Construction

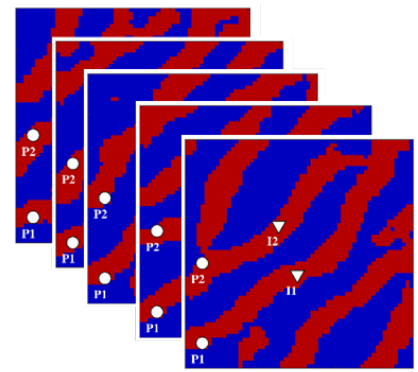
- ❑ Geomodeling tools provide prior realizations that honor
  - Geological concept, described by a training image
  - Hard (well) data and 3D seismic data



Training image (250x250)

Geomodeling tool (SGeMS)



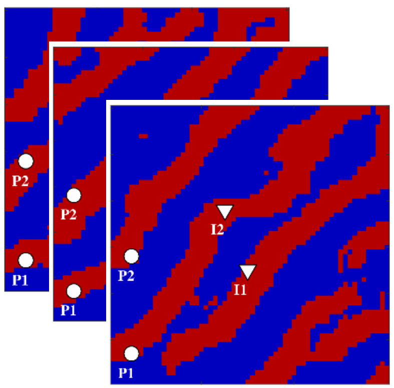


Prior models (60x60)

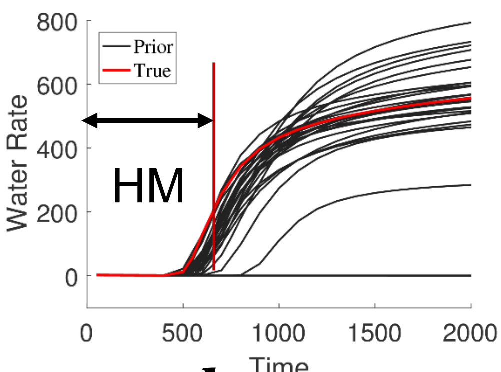
- ❑ Need to modify these models to match production data, while retaining geological realism

Optimization-Based History Matching

- Calibrate realizations $\boldsymbol{m}$ such that mismatch between flow simulation data $\boldsymbol{d}(\boldsymbol{m})$ and production data $\boldsymbol{d}_{\mathbf{obs}}$ is minimized



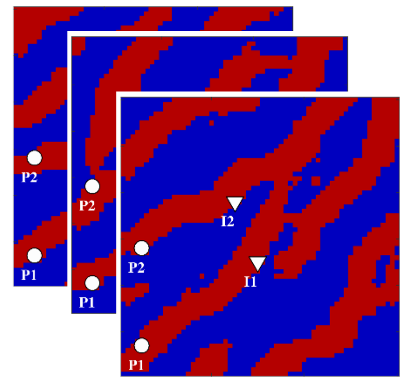
$\boldsymbol{m}_{\mathbf{prior}}$



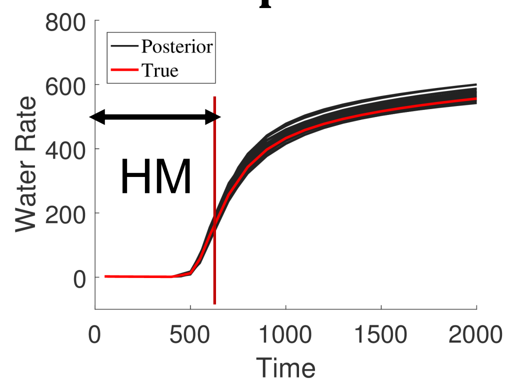
$\boldsymbol{d}_{\mathbf{prior}}$

Modify $\boldsymbol{m}$

flow equations highly nonlinear



$\boldsymbol{m}_{\mathbf{post}}$



$\boldsymbol{d}_{\mathbf{post}}$

Parameterization in History Matching

- Map geological model $\boldsymbol{m}$ to a new variable $\boldsymbol{\xi}$

$$\boldsymbol{m} \approx \widetilde{\boldsymbol{m}} = \boldsymbol{f}(\boldsymbol{\xi})$$

- Favorable properties:
  - Components of $\boldsymbol{\xi}$ are independent
  - $\widetilde{\boldsymbol{m}}$ preserves spatial structure, with $\dim(\boldsymbol{\xi}) \ll \dim(\boldsymbol{m})$
  - Optimization & ensemble methods for post. samples

$$\boldsymbol{\xi}_{\mathbf{post}} = \underset{\boldsymbol{\xi}}{\mathbf{argmin}} \left\{ \begin{aligned} &\frac{\mathbf{1}}{\mathbf{2}}\left(\boldsymbol{d}(\boldsymbol{\xi}) - \boldsymbol{d}^{*}_{\mathbf{obs}}\right)^{T} C_{\mathrm{D}}^{-1}\left(\boldsymbol{d}(\boldsymbol{\xi}) - \boldsymbol{d}^{*}_{\mathbf{obs}}\right) \\ &\qquad + \frac{\mathbf{1}}{\mathbf{2}}\left(\boldsymbol{\xi} - \boldsymbol{\xi}_{uc}\right)^{T}\left(\boldsymbol{\xi} - \boldsymbol{\xi}_{uc}\right) \end{aligned} \right\}$$

Principal Component Analysis (PCA)

- Generate N_r prior realizations using SGeMS

$$\boldsymbol{Y} = \frac{\mathbf{1}}{\sqrt{N_r - \mathbf{1}}}[\boldsymbol{m}_1 - \overline{\boldsymbol{m}}, \boldsymbol{m}_2 - \overline{\boldsymbol{m}}, \dots, \boldsymbol{m}_{N_r} - \overline{\boldsymbol{m}}]$$

- Perform SVD and reduce dimension

$$\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{\Lambda}^{1/2}\boldsymbol{V}^T \approx \boldsymbol{U}_l\boldsymbol{\Lambda}_l^{1/2}\boldsymbol{V}_l^T \qquad \boldsymbol{l} << \boldsymbol{N_C} \qquad (\boldsymbol{N_C}\text{: \# of grid blocks})$$

- Generate new realization

$$\boldsymbol{m}_{\mathbf{pca}} = \boldsymbol{U}_l\boldsymbol{\Lambda}_l^{1/2}\boldsymbol{\xi}_l + \overline{\boldsymbol{m}} \qquad \boldsymbol{\xi}_l \sim \boldsymbol{N}(\mathbf{0}, \boldsymbol{I})$$

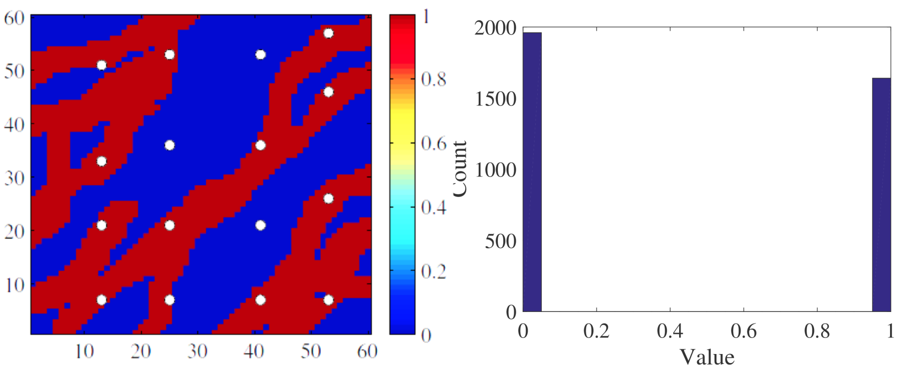
- Works well for Gaussian $\boldsymbol{m}$

(Oliver, 1996; Sarma et al., 2006)

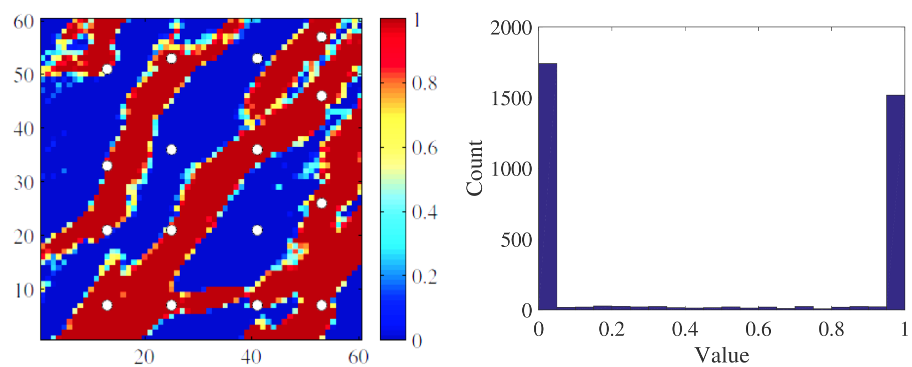
Optimization-based PCA (O-PCA)

$$\boldsymbol{m}_{\text{opca}} = \underset{\boldsymbol{x}}{\textbf{argmin}}\left\{\left\|\boldsymbol{m}_{\text{pca}}(\boldsymbol{\xi}_{\boldsymbol{l}}) - \boldsymbol{x}\right\|_2^2 + \gamma\boldsymbol{x}^{\boldsymbol{T}}(\mathbf{1} - \boldsymbol{x})\right\} \quad x_i \in [x^l, x^u]$$

- Formulate PCA as an optimization problem with regularization
- Essentially post-process $\boldsymbol{m}_{\mathbf{pca}}$ with point-wise mapping
- Honors two-point correlations; less reliable without hard data



SGEMS N_C **= 3600**



O-PCA l **= 70**

(Vo and Durlofsky, 2014, 2015)

Generalized O-PCA

$$\boldsymbol{m}_{\text{opca}} = \underset{\boldsymbol{x}}{\textbf{argmin}}\left\{L_c(\boldsymbol{x}, \boldsymbol{m}_{\text{pca}}(\boldsymbol{\xi_l})) + \gamma L_s(\boldsymbol{x}, \mathbf{TI})\right\} \quad x_i \in [x^l, x^u]$$

- $L_c(\boldsymbol{x}, \boldsymbol{m}_{\text{pca}}(\boldsymbol{\xi_l}))$ – closeness (content) between $\boldsymbol{x}$ and $\boldsymbol{m}_{\text{pca}}(\boldsymbol{\xi_l})$
- $L_s(\boldsymbol{x}, \mathbf{TI})$ – degree to which $\boldsymbol{x}$ reproduces the spatial structure (style) of target training image $\mathbf{TI}$
- Assume $\{\psi_k, k = 1, \ldots, K\}$ is a set of statistical measures for the spatial correlation structure such that

$$\forall k,\ \psi_k(\boldsymbol{m}) = \psi_k(\mathbf{TI}) \quad \leftrightarrow$$

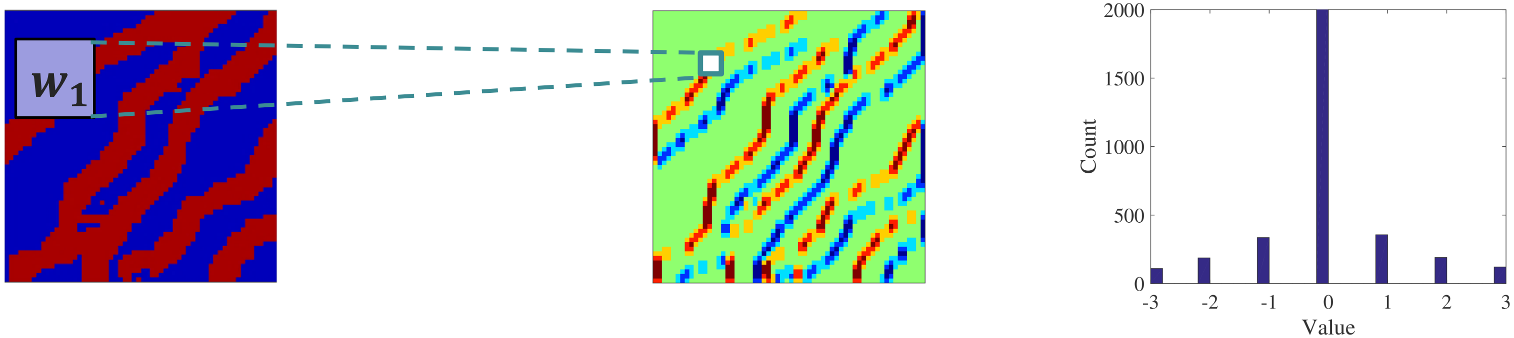
$\boldsymbol{m}$ and $\mathbf{TI}$ have the same spatial correlation structure

- Then define

$$L_s(\boldsymbol{x}, \mathbf{TI}) = \sum_k ||\psi_k(\boldsymbol{x}) - \psi_k(\mathbf{TI})||^2$$

New Regularization Term

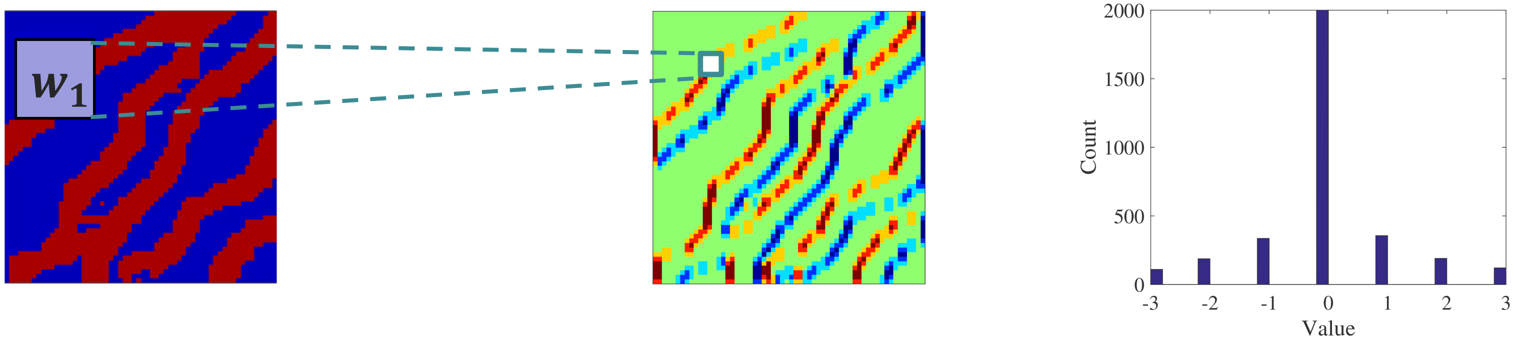
- $\psi_k(\boldsymbol{m})$ – lower-order statistical metrics of filter responses $\boldsymbol{F}(\boldsymbol{m})$



$\boldsymbol{m}$ $F_1(\boldsymbol{m})$ $\psi_1(\boldsymbol{m})$ = Histogram of $F_1(\boldsymbol{m})$

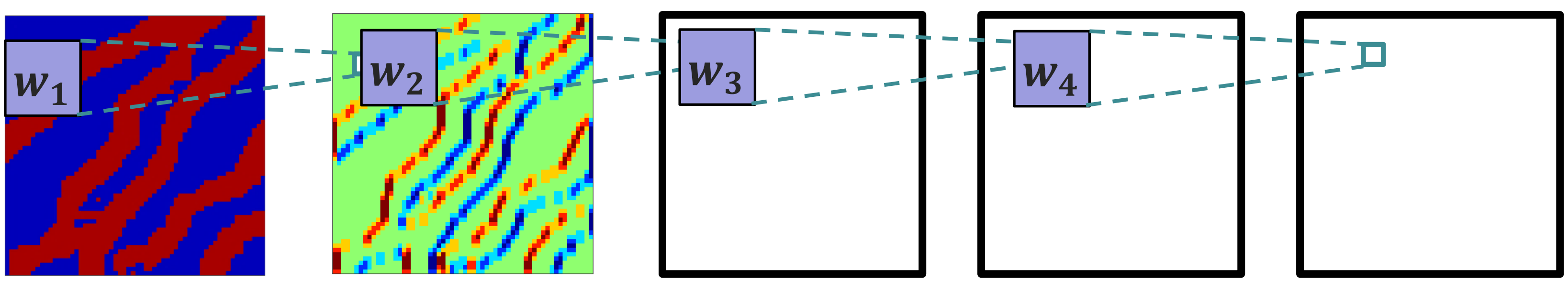
New Regularization Term

- $\psi_k(\boldsymbol{m})$ – lower-order statistical metrics of filter responses $\boldsymbol{F}(\boldsymbol{m})$



$\boldsymbol{m}$ $F_1(\boldsymbol{m})$ $\psi_1(\boldsymbol{m})$ = Histogram of $F_1(\boldsymbol{m})$

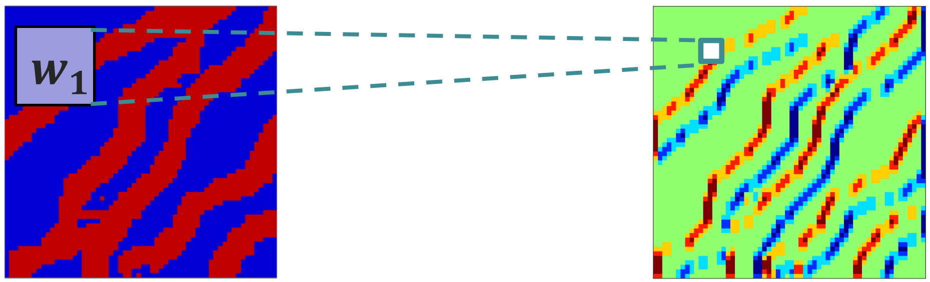
- Multiscale filter responses

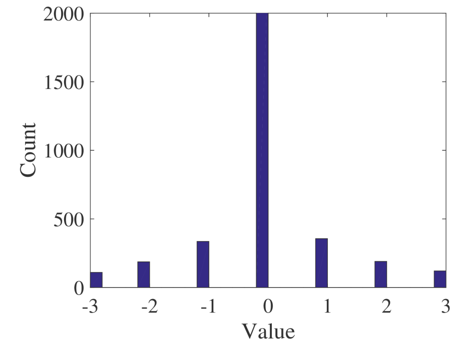


$\boldsymbol{m}$ $F_1(\boldsymbol{m})$ $F_2(\boldsymbol{m})$ $F_3(\boldsymbol{m})$ $F_4(\boldsymbol{m})$

Convolutional Neural Network

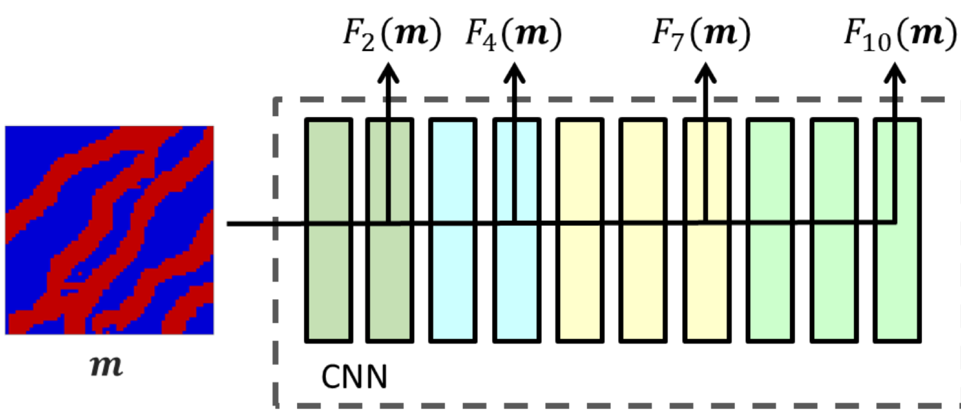
- Convolutional layer





$\boldsymbol{m}$ $F_1(\boldsymbol{m})$ Histogram of $F_1(\boldsymbol{m})$

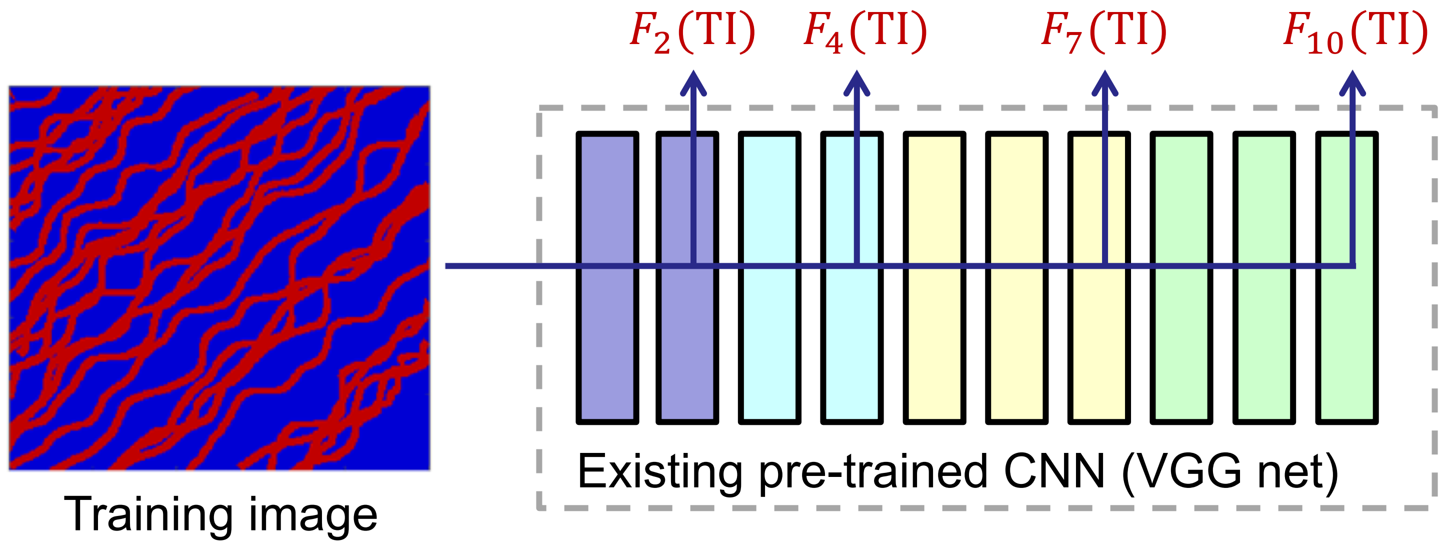
- Convolutional neural network



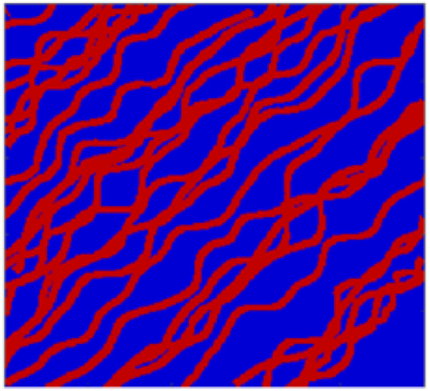
$$F_k(\boldsymbol{m}) = f(W^T F_{k-1}(\boldsymbol{m}) + b)$$

- F: feature matrix
- W: weight matrix
- f : nonlinear activation (e.g., ReLU, sigmoid)

O-PCA with CNN-based Regularization

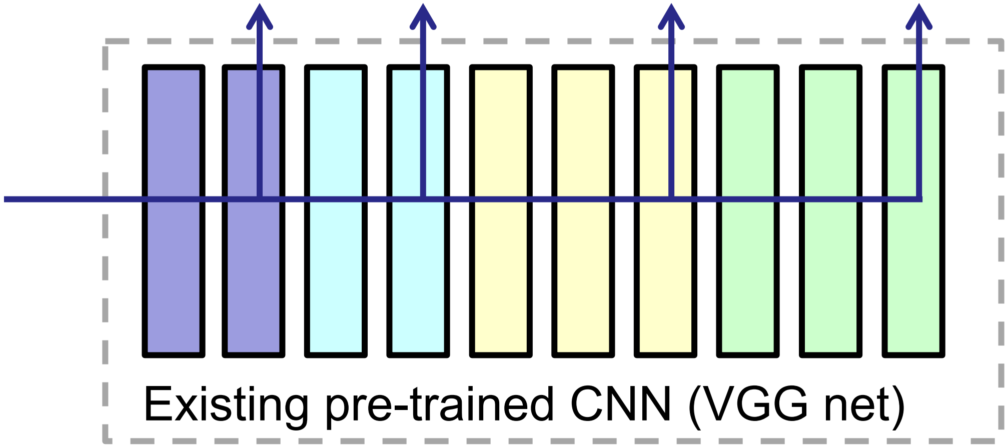


O-PCA with CNN-based Regularization

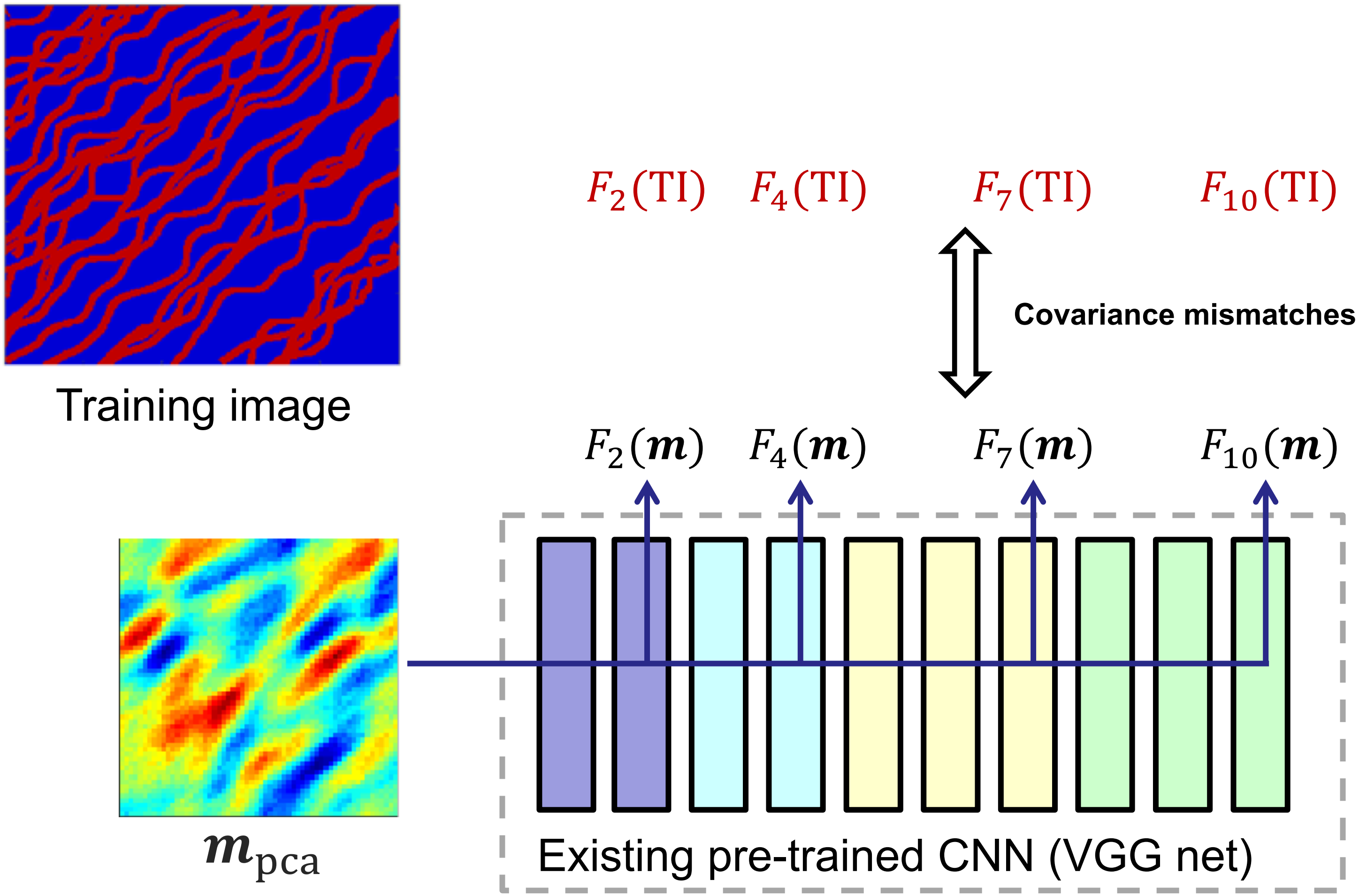


Training image

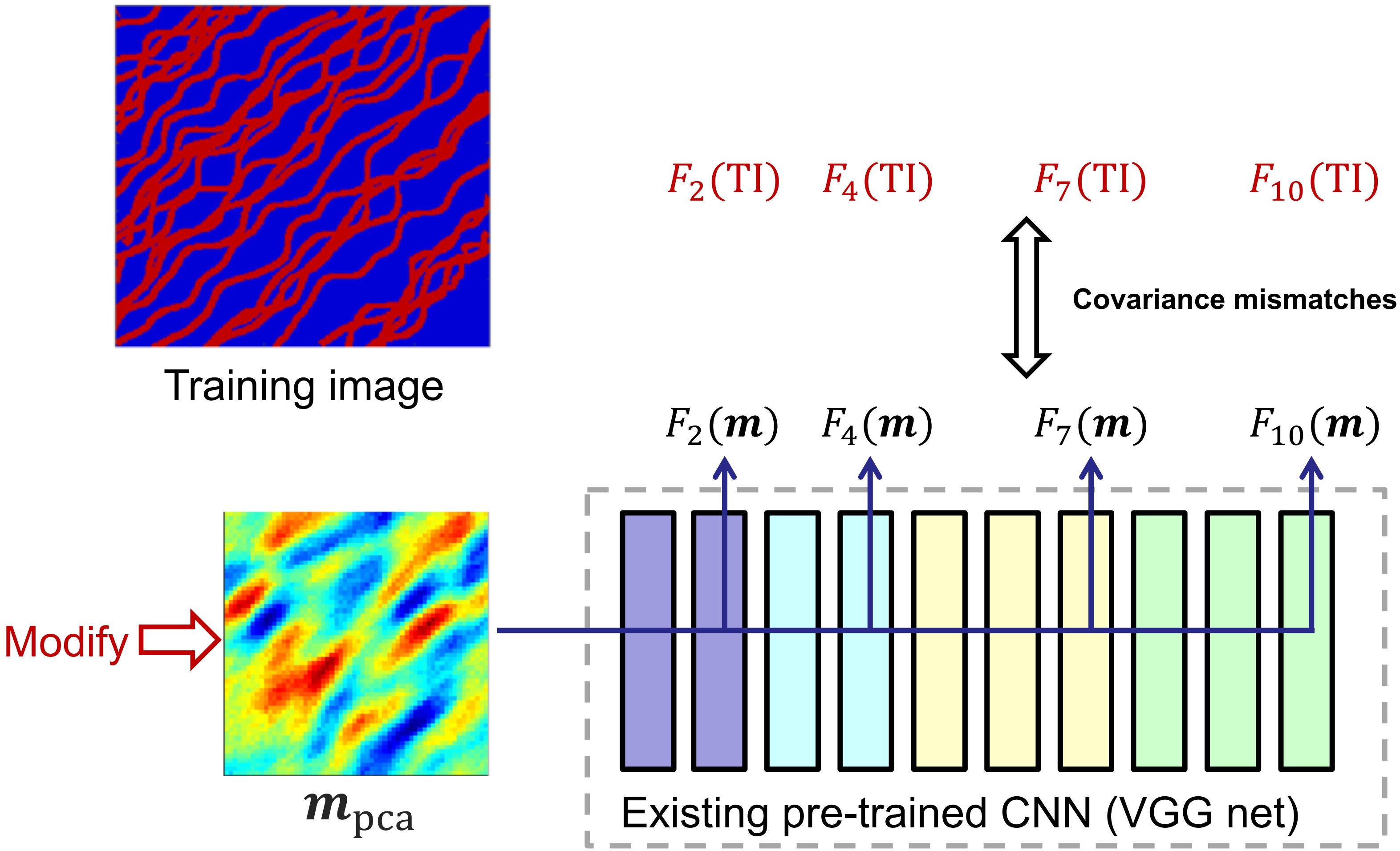
$F_2(\mathrm{TI})$ $F_4(\mathrm{TI})$ $F_7(\mathrm{TI})$ $F_{10}(\mathrm{TI})$



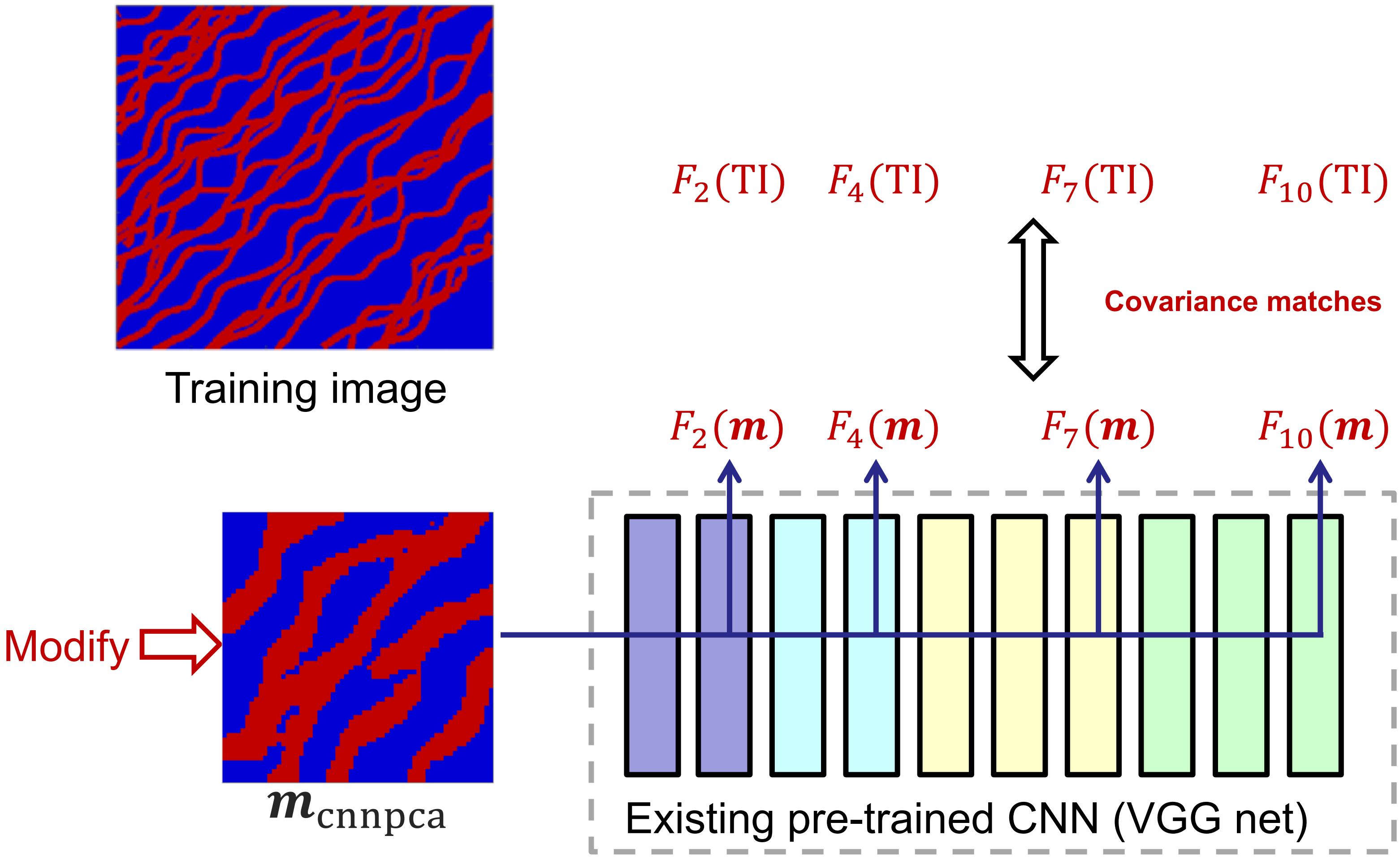
O-PCA with CNN-based Regularization



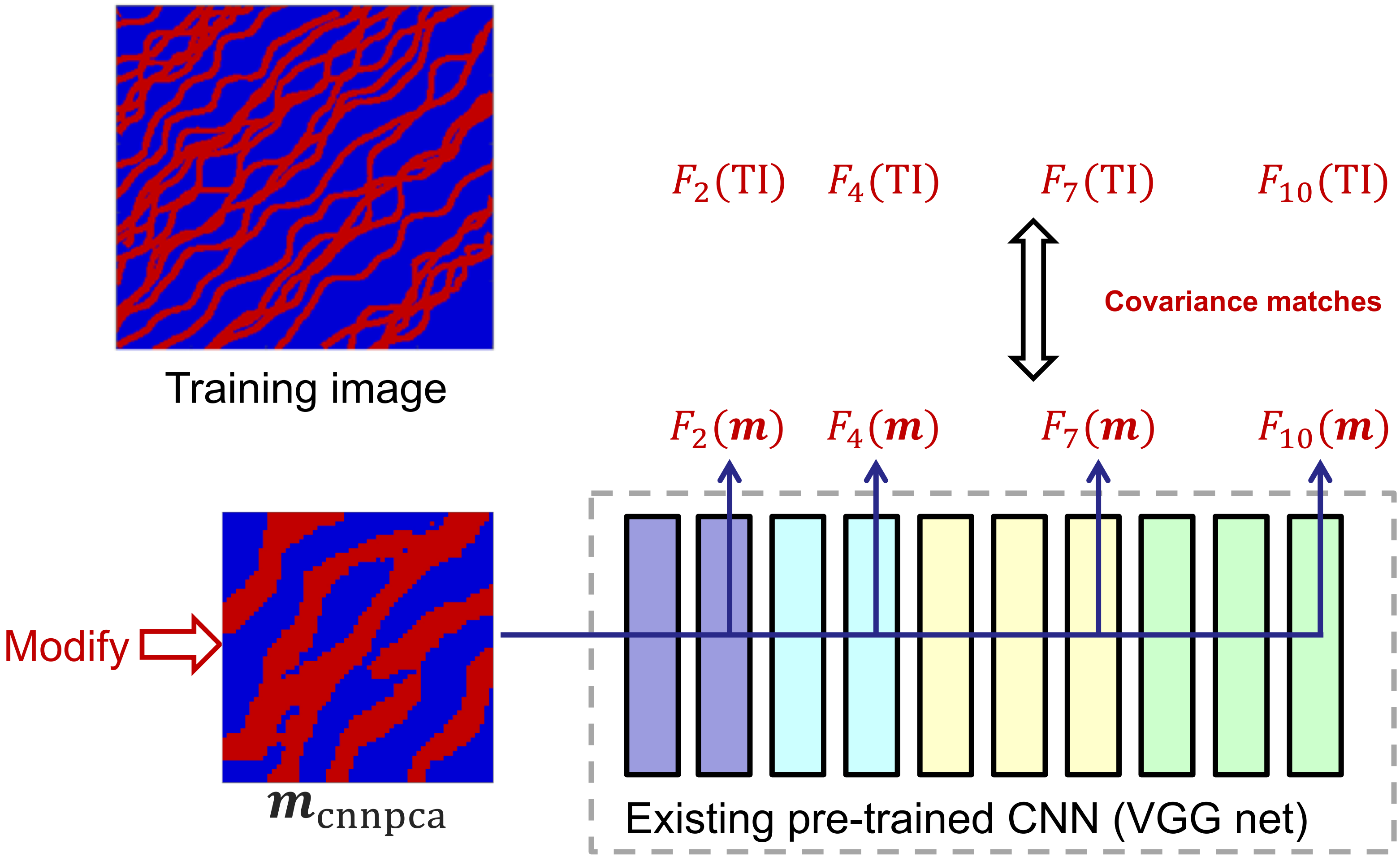
O-PCA with CNN-based Regularization



O-PCA with CNN-based Regularization



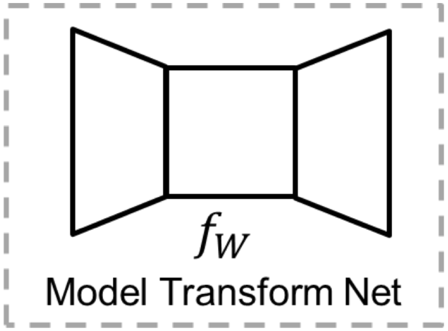
O-PCA with CNN-based Regularization



Need to solve an expensive optimization for every PCA model

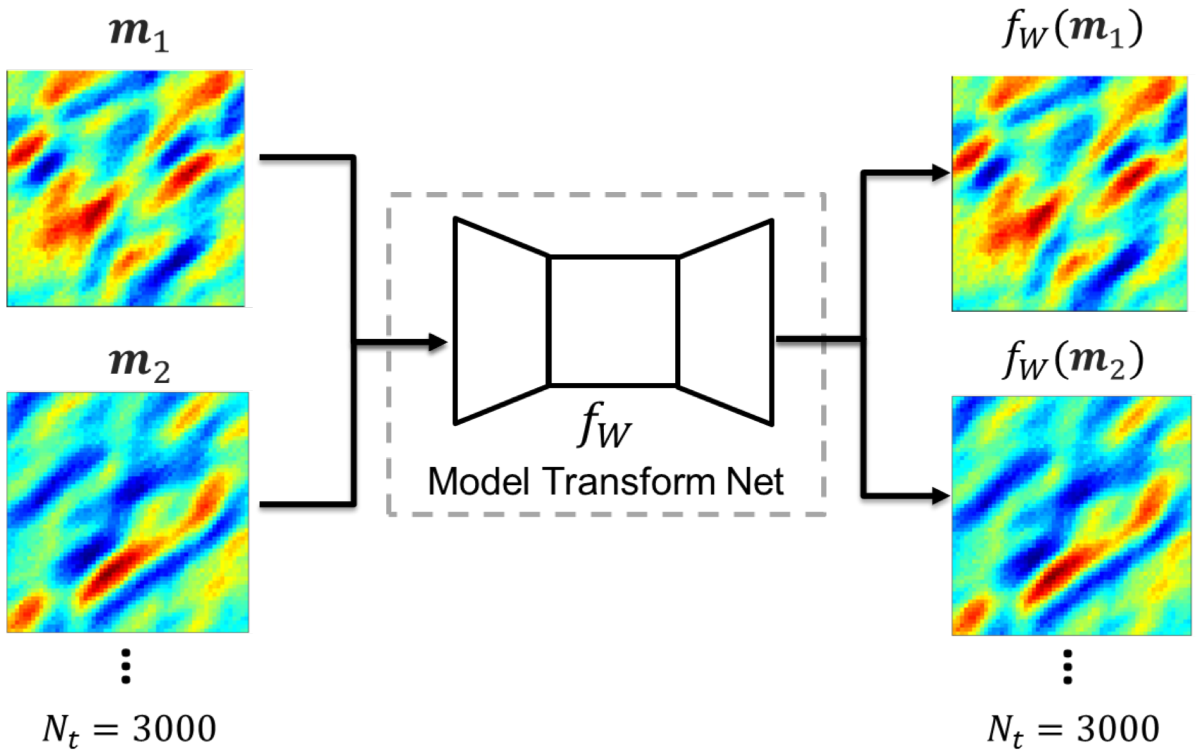
Model Transform Net and CNN-PCA

- Train another transform CNN f_W



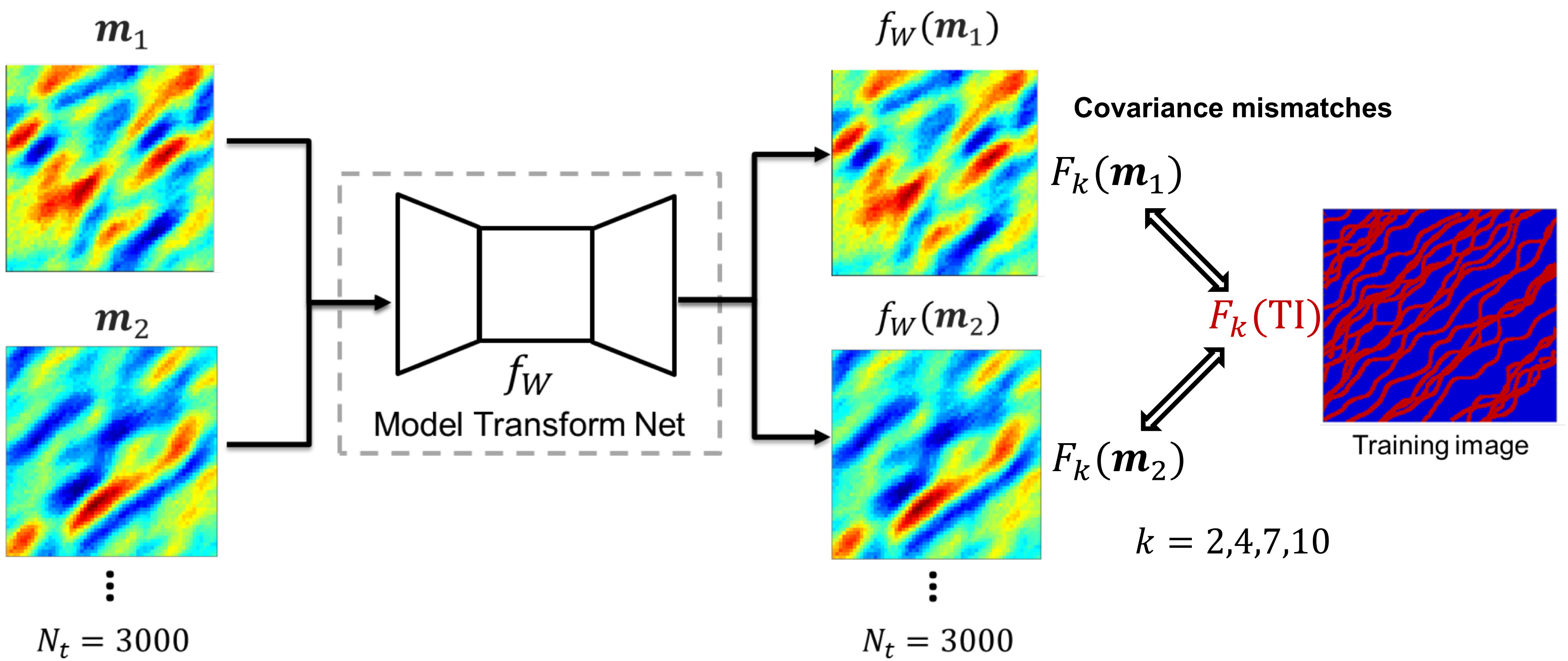
Model Transform Net and CNN-PCA

- Train another transform CNN f_W



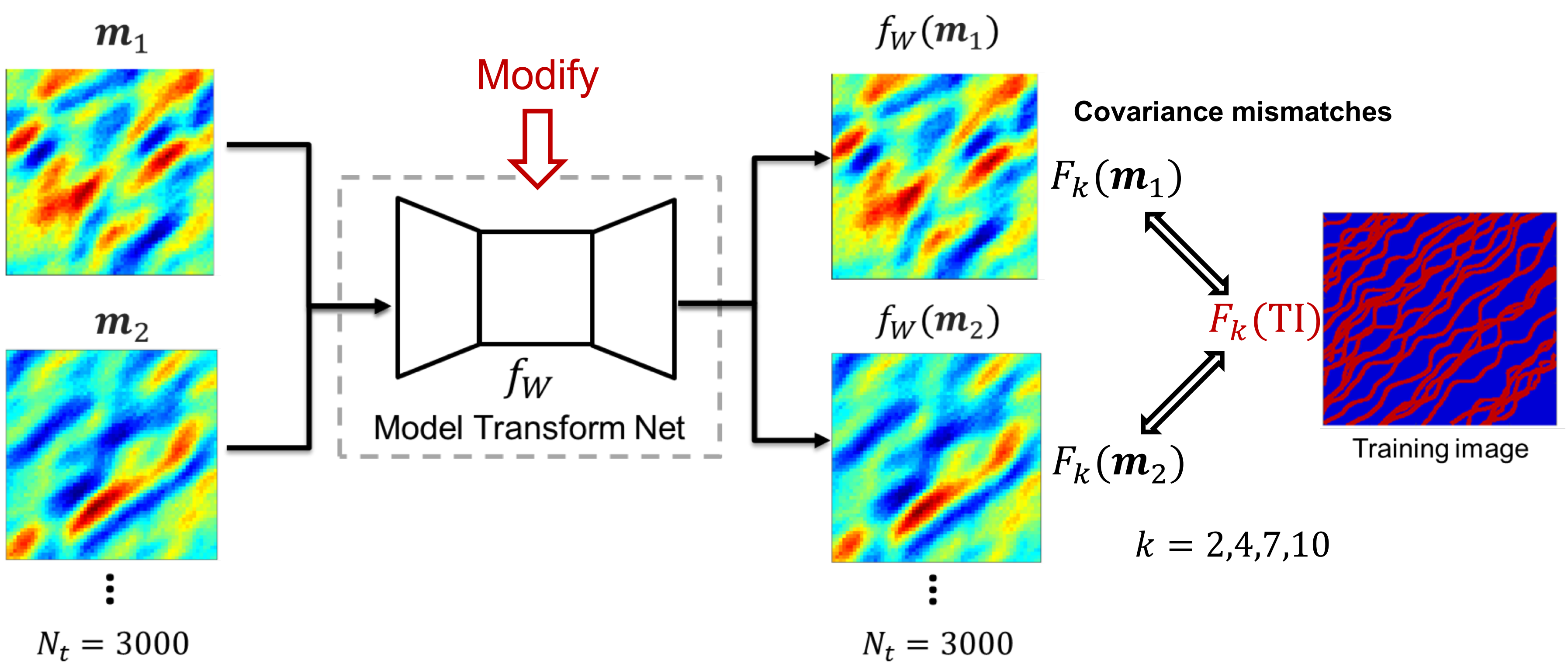
Model Transform Net and CNN-PCA

- Train another transform CNN f_W



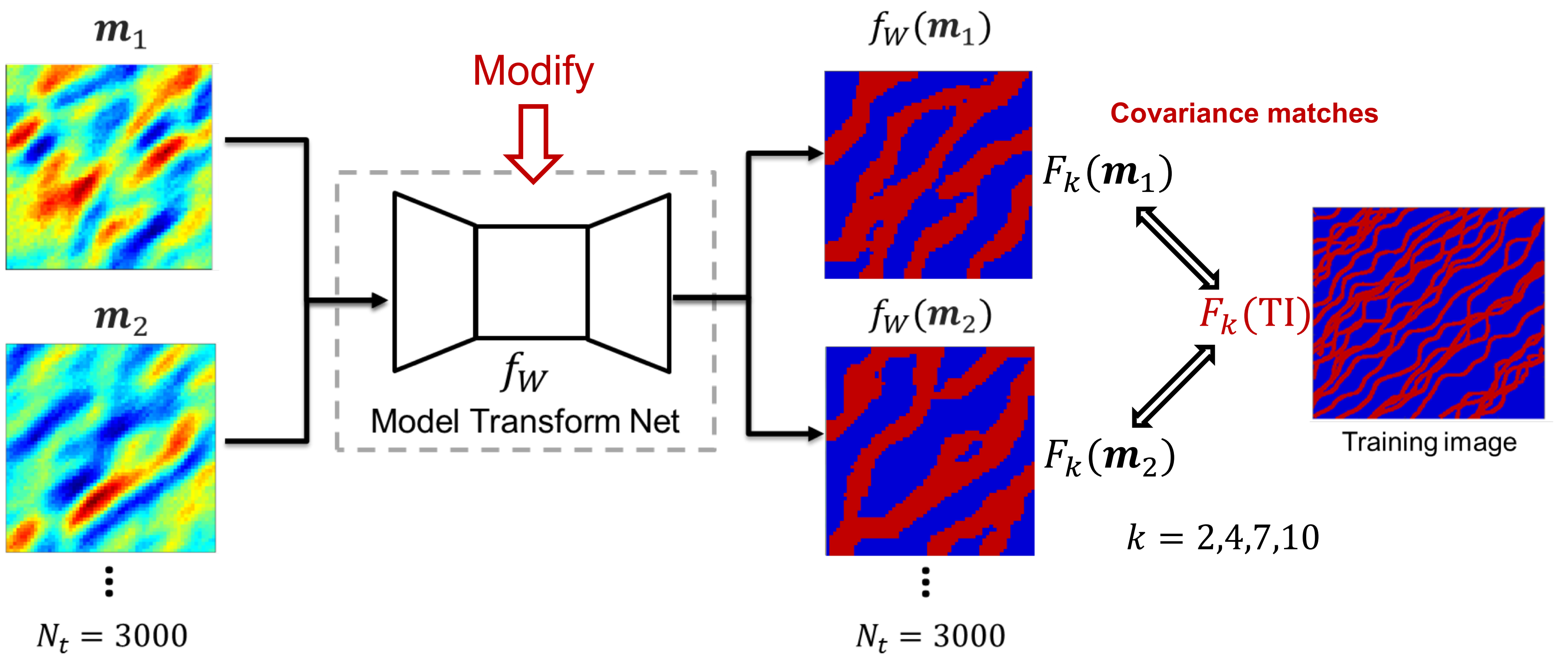
Model Transform Net and CNN-PCA

- Train another transform CNN f_W



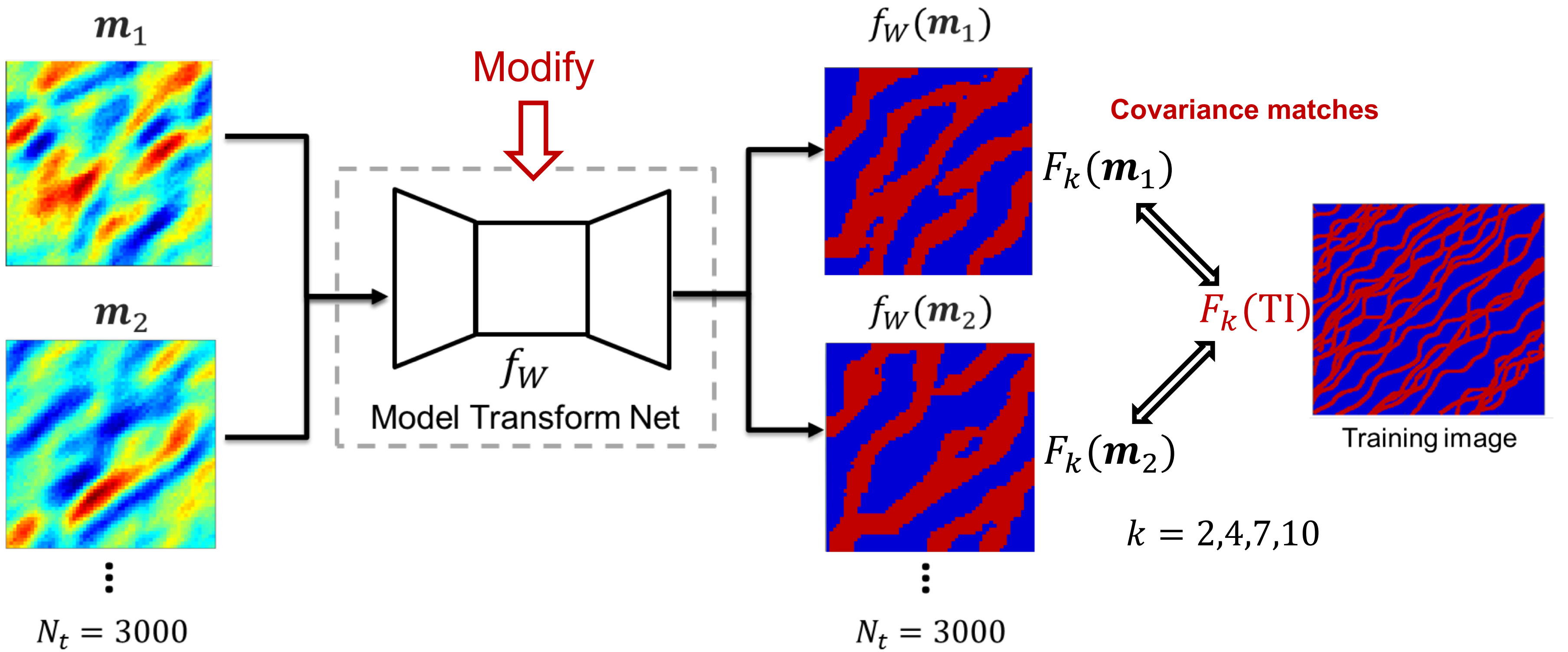
Model Transform Net and CNN-PCA

- Train another transform CNN f_W



Model Transform Net and CNN-PCA

- Train another transform CNN f_W



After training, transforming PCA models is almost real-time

Related approaches: Laloy et al. (2017, 2018), Chan & Elsheikh (2017, 2018), Canchumuni et al. (2017, 2018), Mosser et al. (2017, 2018), Khaninezhad et al. (2019)

Transform-Net Training

$$\underset{f_W}{\mathbf{argmin}}\left\{\sum_{i}^{N_t} L_c\left(f_W(\boldsymbol{m}_{\mathbf{pca}}^{\boldsymbol{i}}), \boldsymbol{m}_{\mathbf{pca}}^{\boldsymbol{i}}\right) + \gamma_{\boldsymbol{s}} L_S\left(f_W(\boldsymbol{m}_{\mathbf{pca}}^{\boldsymbol{i}}), \mathbf{TI}\right)\right\}$$

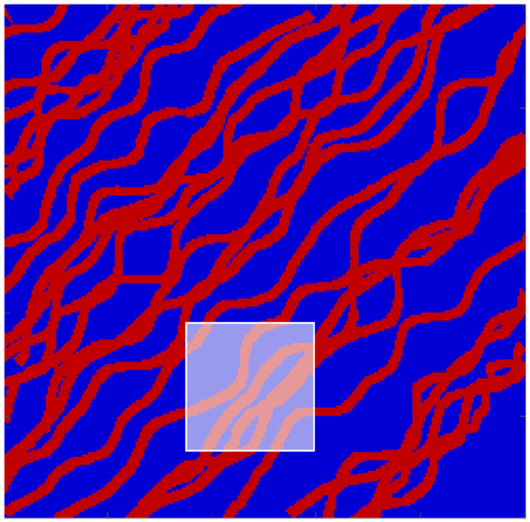
$$L_c\left(f_W(\boldsymbol{m}_{\mathbf{pca}}^{\boldsymbol{i}}), \boldsymbol{m}_{\mathbf{pca}}^{\boldsymbol{i}}\right) = a||F_4\left(f_W(\boldsymbol{m}_{\mathbf{pca}}^{\boldsymbol{i}})\right) - F_4\left(\boldsymbol{m}_{\mathbf{pca}}^{\boldsymbol{i}}\right)||_{\boldsymbol{F_r}}^{\mathbf{2}}$$

$$L_S\left(f_W(\boldsymbol{m}_{\mathbf{pca}}^{\boldsymbol{i}}), \mathbf{TI}\right) = \sum_{k} \beta_k||G_k\left(f_W(\boldsymbol{m}_{\mathbf{pca}}^{\boldsymbol{i}})\right) - G_k(\mathbf{TI})||_{\boldsymbol{F_r}}^{\mathbf{2}}$$

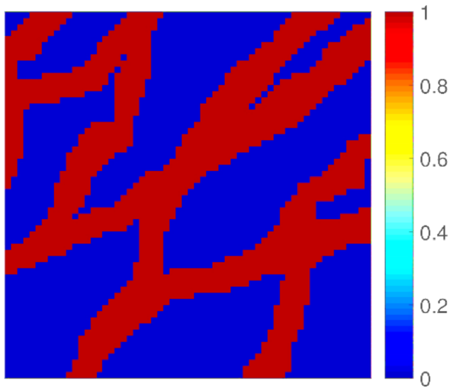
- f_W: set of parameters in the model transform net
- N_t: # of training models, G_k: covariance of F_k, $\gamma_{\boldsymbol{s}}$: style weight
- Identical to ‘fast neural style transfer’ in computer vision

(Johnson et al., 2016)

Unconditional Binary System



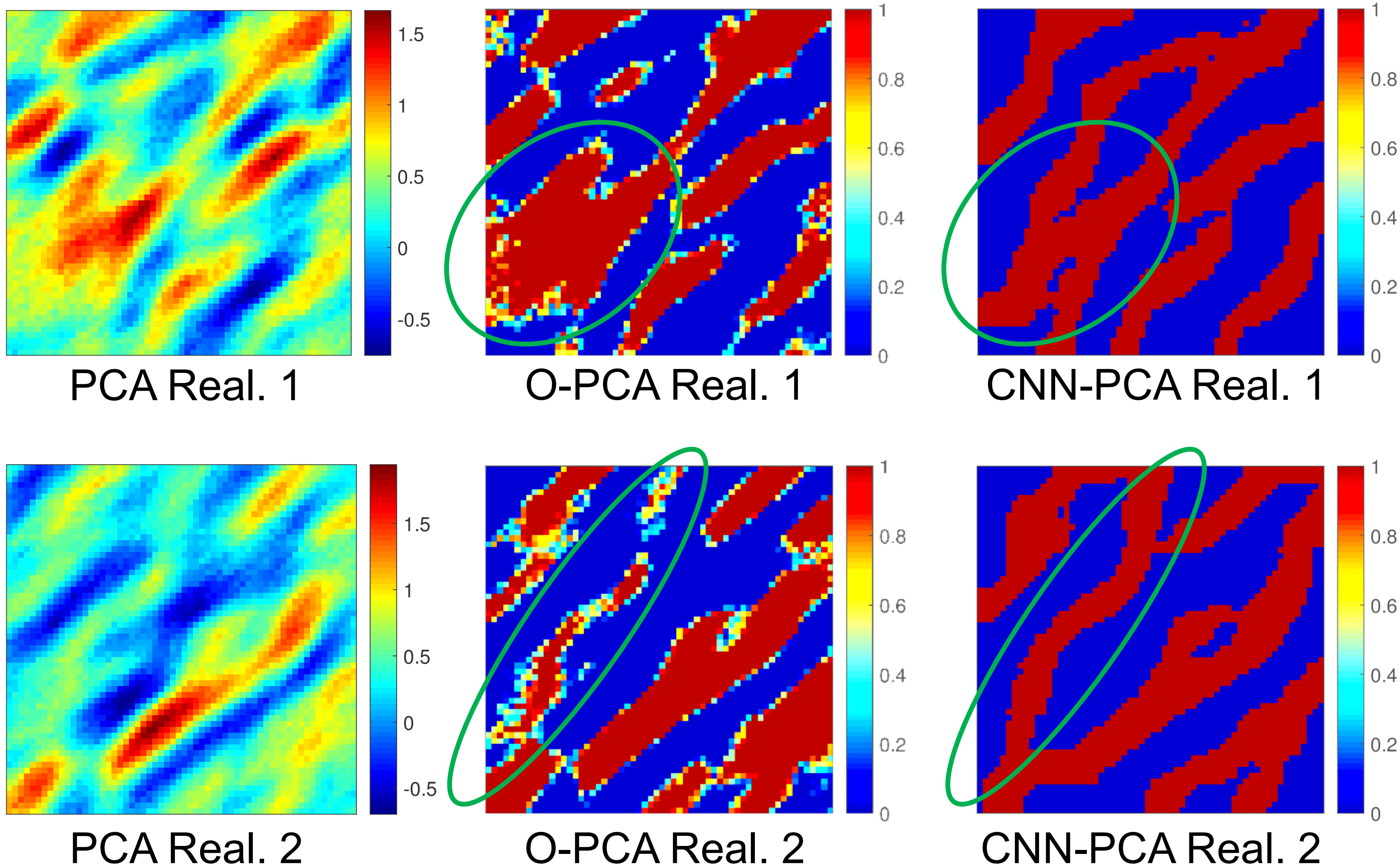
Training image



One SGeMS realization

- Training image size: 250 x 250
- Model size: 60 x 60, $N_c = 3600$
- No hard data
- Goal: low-dimensional representation $l = 70$
- Construct PCA with 1000 SGeMS realizations
- Train f_W with 3000 PCA models (3 min on NVIDIA Tesla K80 GPU)
- $\gamma_s = 0.3$

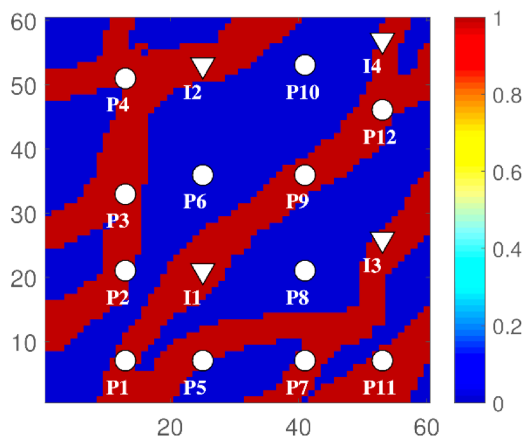
Unconditional Binary System



Conditional Binary System

$$\underset{f_W}{\mathbf{argmin}}\left\{\sum_{i}^{N_t} L_c\left(f_W\left(\boldsymbol{m}_{\mathbf{pca}}^{i}\right),\boldsymbol{m}_{\mathbf{pca}}^{i}\right)+\gamma_{\boldsymbol{s}} L_s\left(f_W\left(\boldsymbol{m}_{\mathbf{pca}}^{i}\right),\mathbf{TI}\right)+\gamma_{\boldsymbol{h}} L_h\left(f_W\left(\boldsymbol{m}_{\mathbf{pca}}^{i}\right)\right)\right\}$$

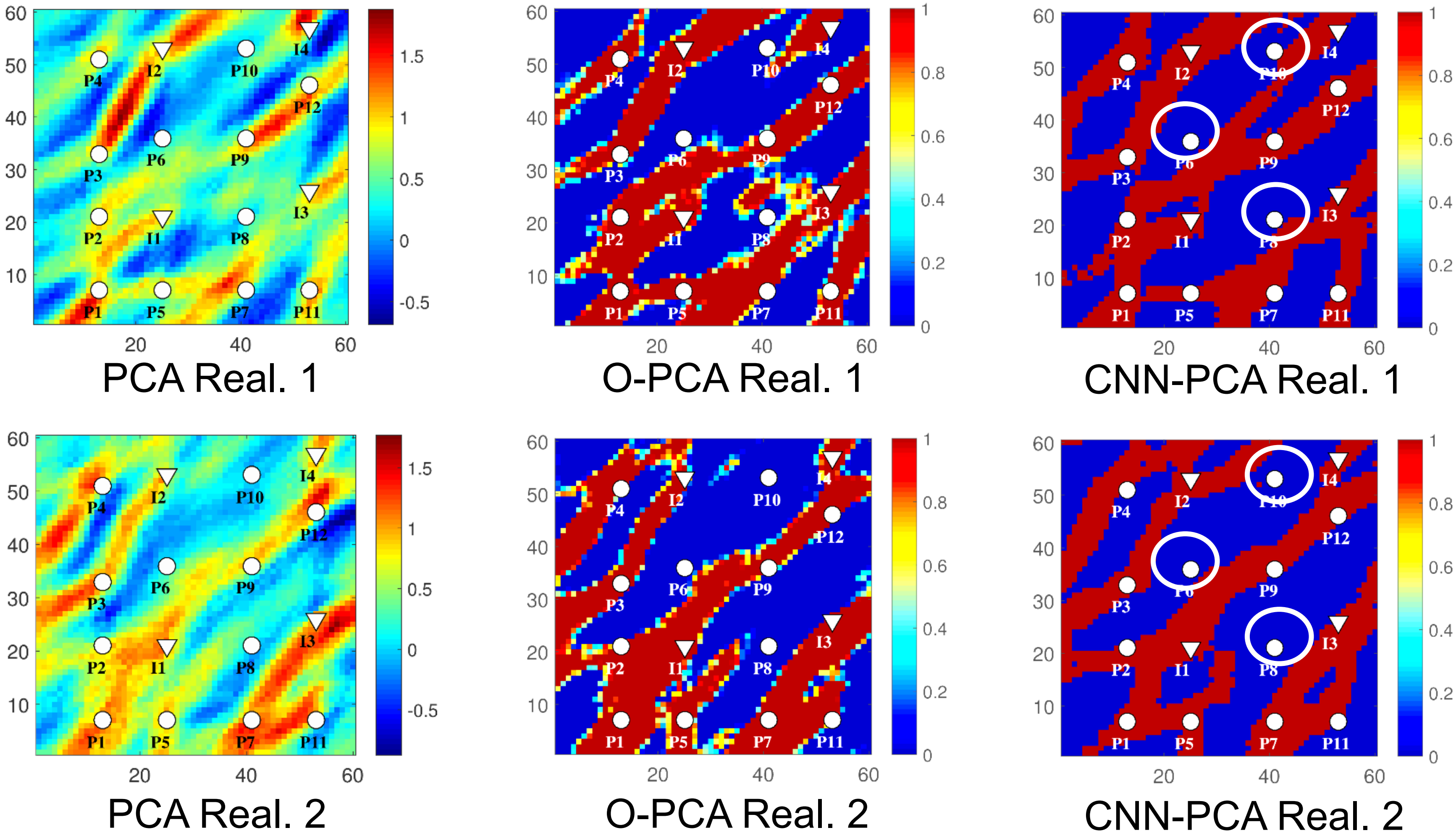
(additional hard data loss term)



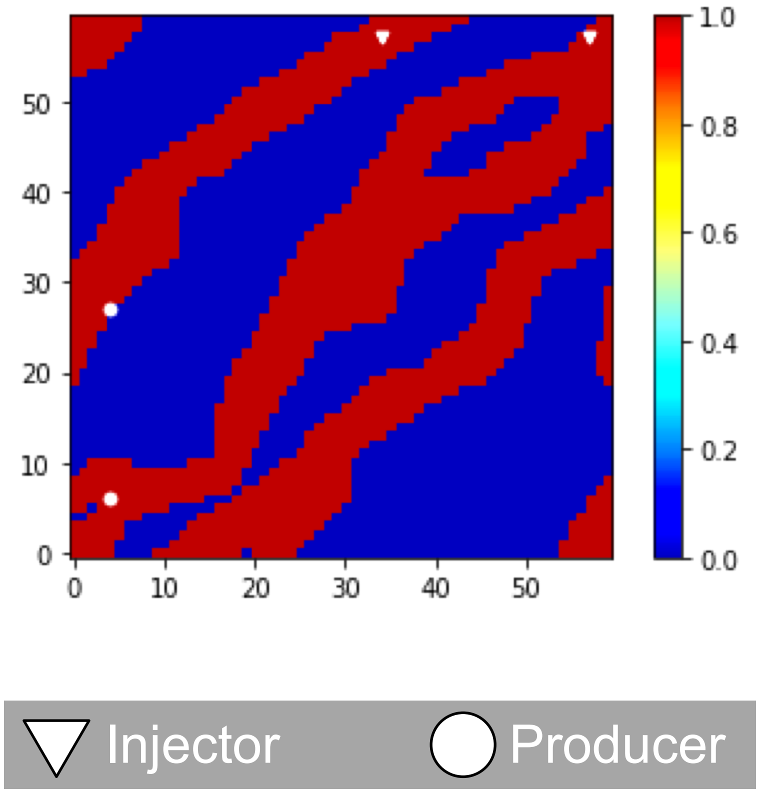
One SGeMS realization

- Training image size: 250 x 250
- Model size: 60 x 60, $N_c = 3600$
- Hard data at 16 well locations
- Reduced dimension: $l = 70$
- $\gamma_{\boldsymbol{s}} = 0.3$ and $\gamma_{\boldsymbol{h}} = 1.0$

Conditional Binary System



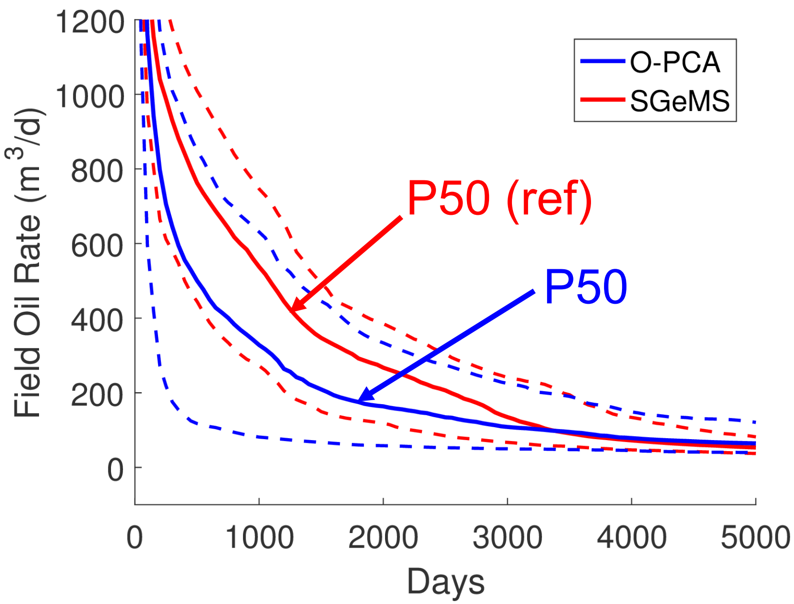
Assessment of Prior Flow Statistics



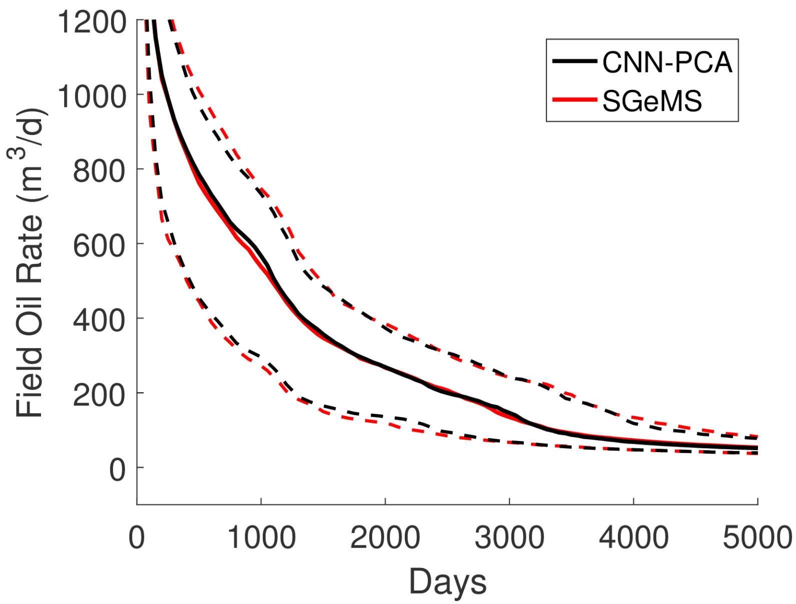
- Oil-water, 60 x 60 grid
- 2 injectors, 2 producers, BHP controlled; hard data at wells
- k_{sand} = 2000 md, k_{mud} = 2 md
- Flow simulation with 200 conditional SGeMS, O-PCA and CNN-PCA prior models
- Challenging case for O-PCA

Field Oil Rate P10, P50, P90
(200 Prior Models)

O-PCA Results

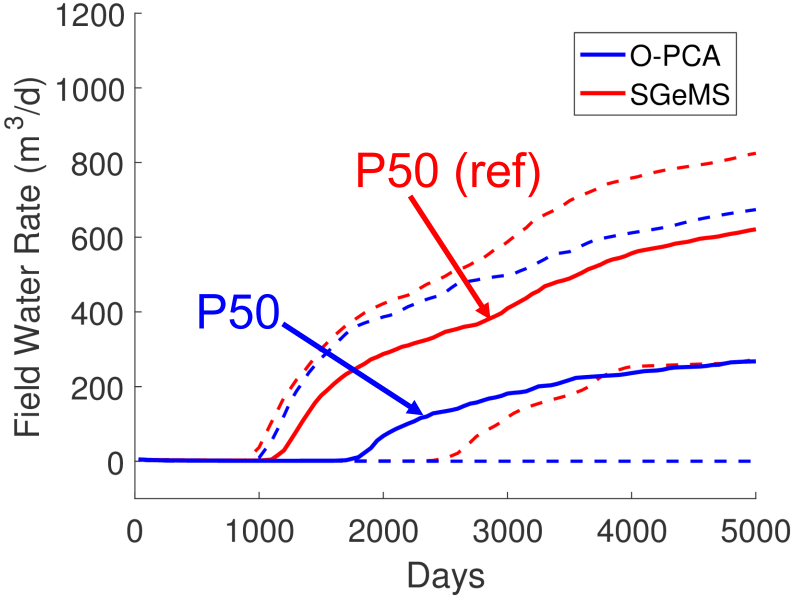


CNN-PCA Results

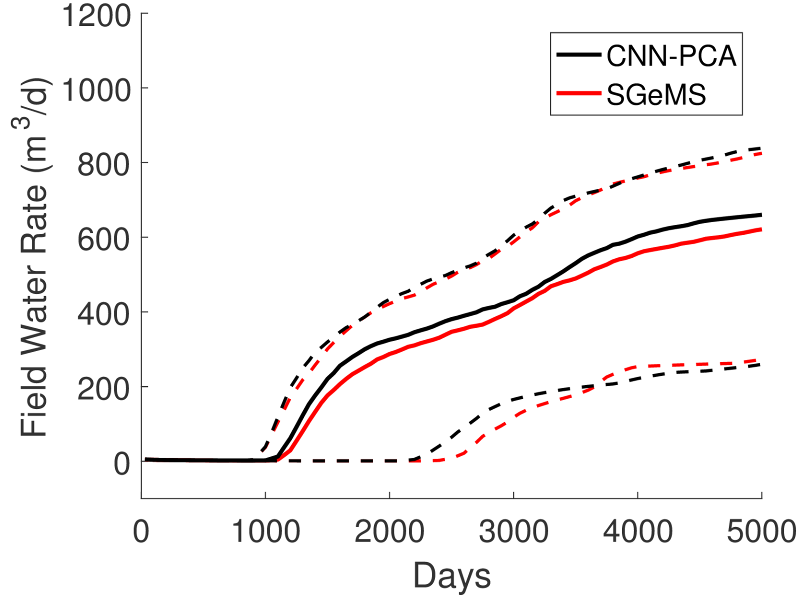


Field Water Rate P10, P50, P90
(200 Prior Models)

O-PCA Results

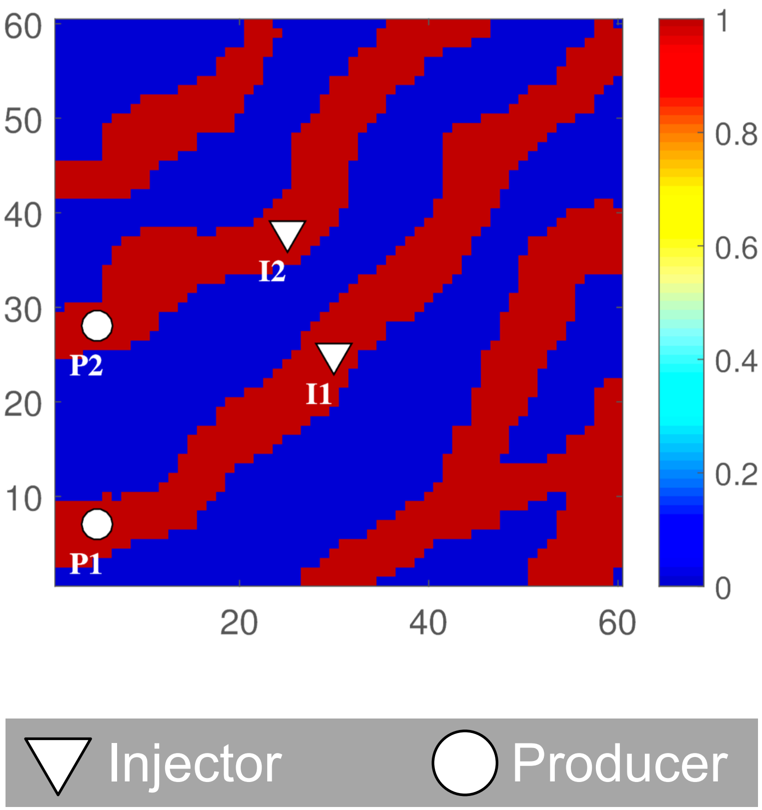


CNN-PCA Results



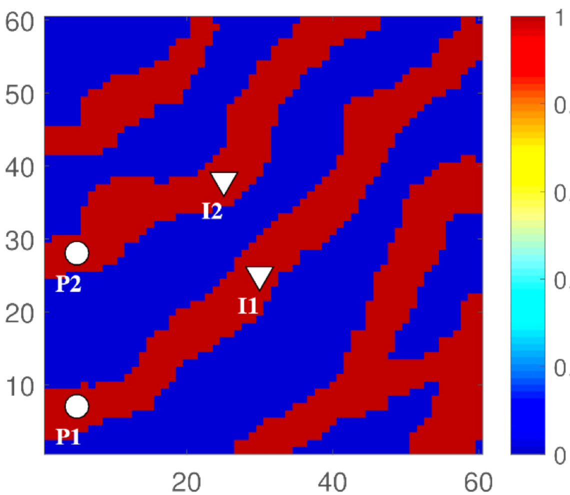
History Matching – Conditional Models

$$\boldsymbol{\xi}_{\mathbf{post}} = \underset{\boldsymbol{\xi}}{\mathbf{argmin}} \left\{ \frac{\mathbf{1}}{\mathbf{2}} (\boldsymbol{d}(\boldsymbol{\xi}) - \boldsymbol{d}^*_{\mathbf{obs}})^T C_{\mathrm{D}}^{-1} (\boldsymbol{d}(\boldsymbol{\xi}) - \boldsymbol{d}^*_{\mathbf{obs}}) + \frac{\mathbf{1}}{\mathbf{2}} (\boldsymbol{\xi} - \boldsymbol{\xi}_{uc})^T (\boldsymbol{\xi} - \boldsymbol{\xi}_{uc}) \right\}$$

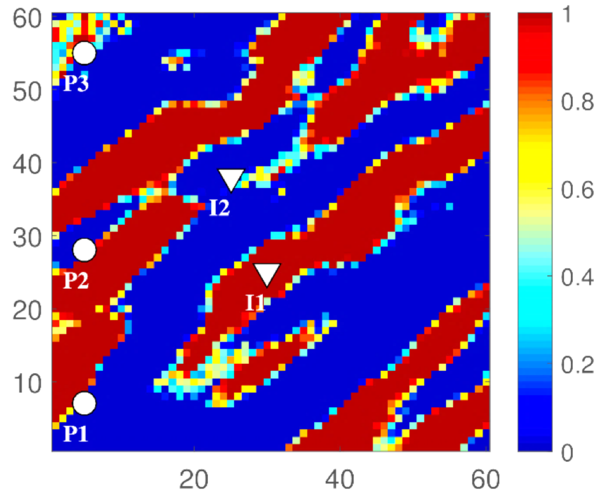


- Oil-water, 60 x 60 grid
- 4 wells, BHP controlled
- Data: production and injection rates for 1000 days
- Optimizer: PSO-MADS (Isebor et al. 2014), 30 RML models
- $l = 70$, $N_{\mathrm{MADS}} = 2l = 140$ simulations per iteration (multilevel treatment also applied)

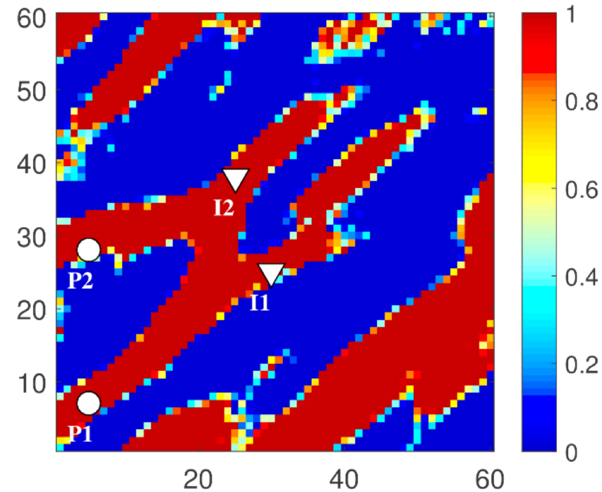
Permeability Estimation



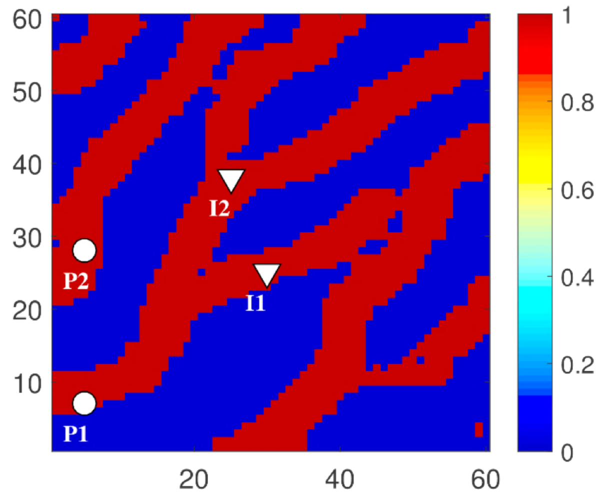
True model



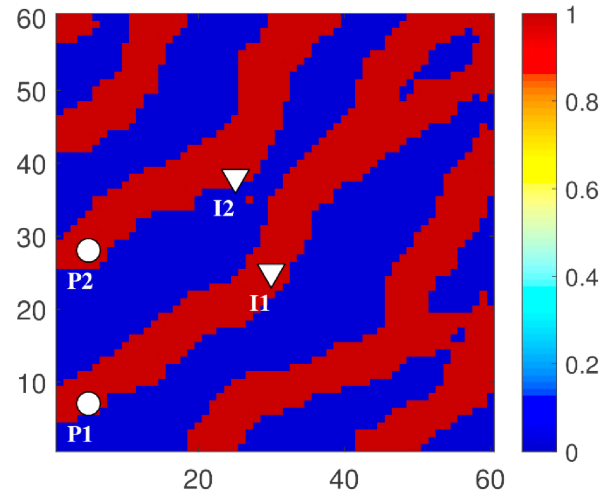
O-PCA prior model



O-PCA posterior model



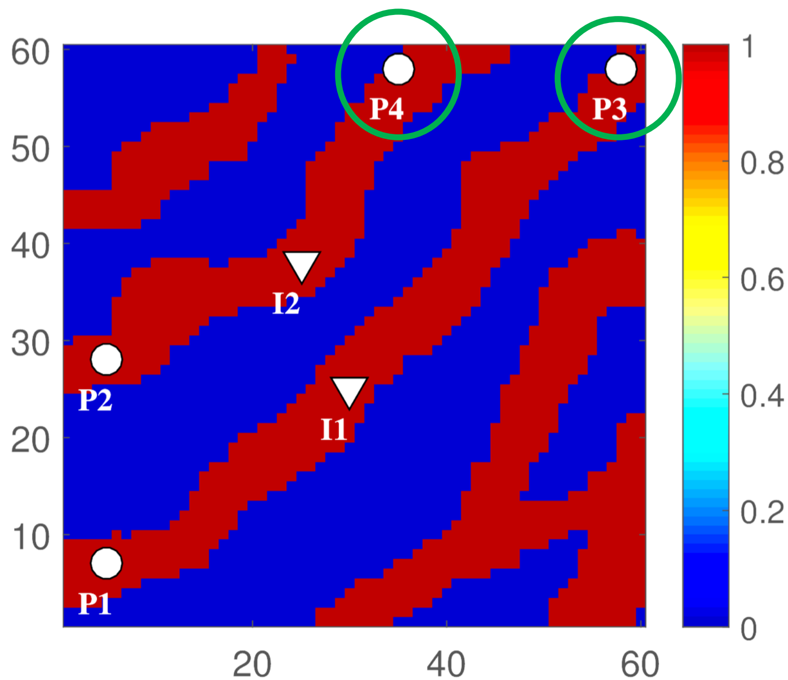
CNN-PCA prior model



CNN-PCA posterior model

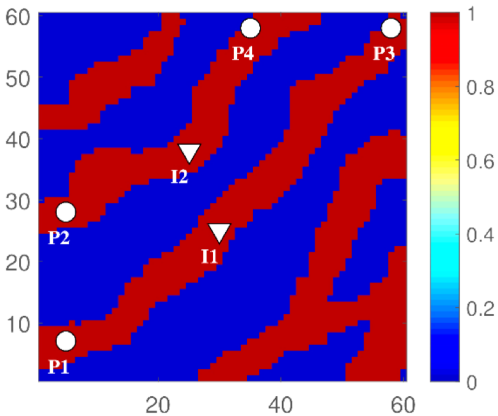
Infill Well Prediction

- Two infill wells P3, P4
- Drilled at 1000 days, prediction to 2000 days

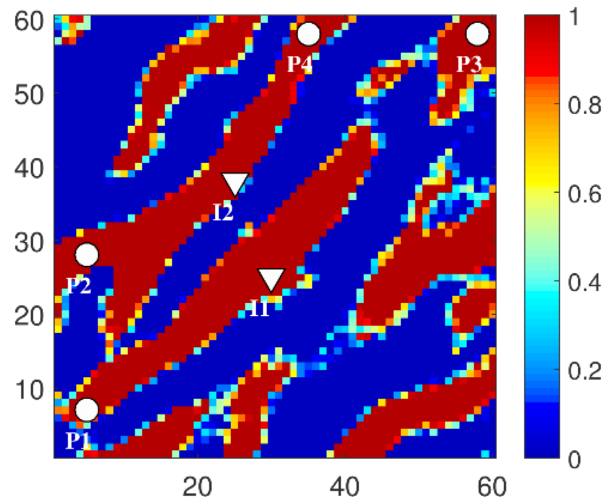


infill well locations

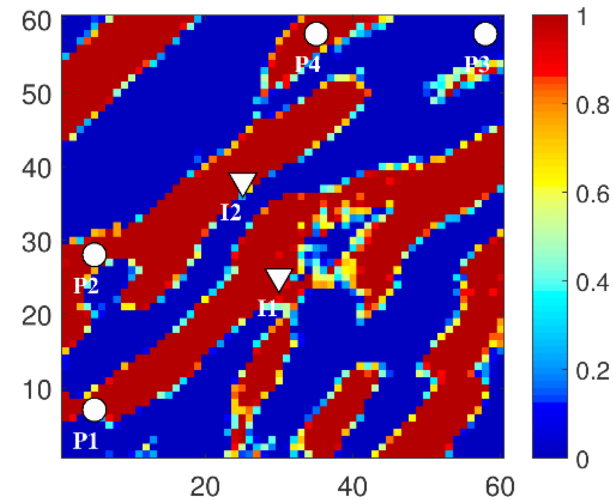
Some Posterior Models



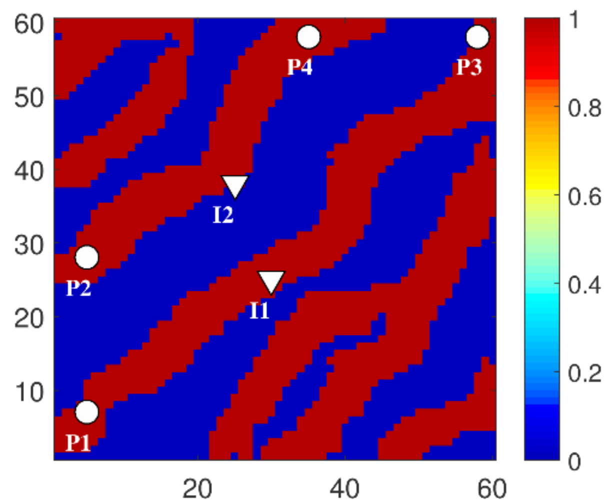
True model



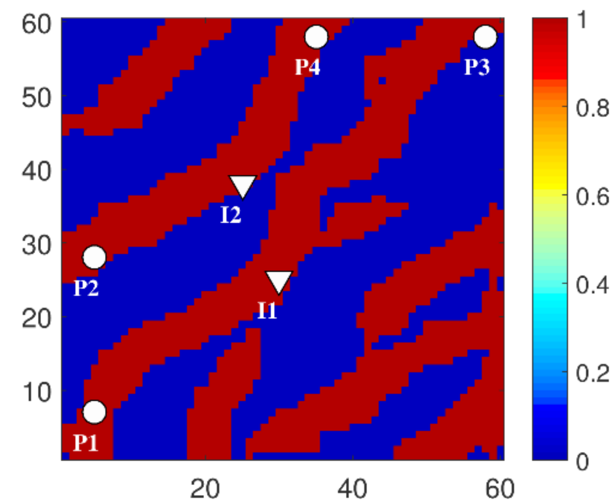
O-PCA #1



O-PCA #2



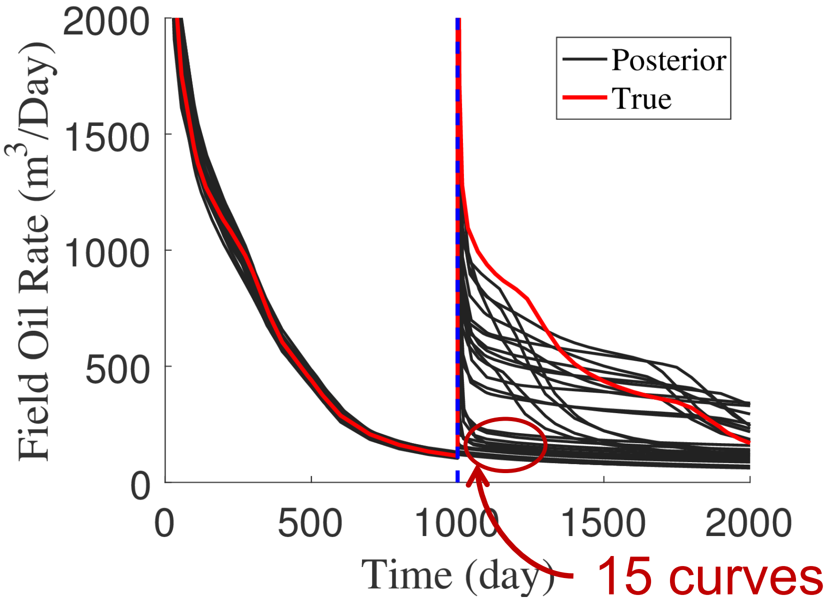
CNN-PCA #1



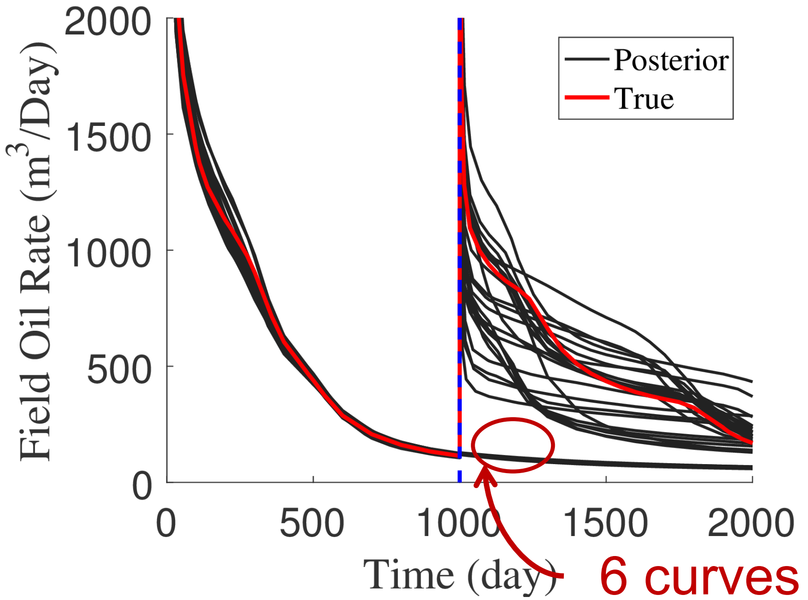
CNN-PCA #2

Field Oil Prediction
(30 posterior models, predict for *t* > 1000 days)

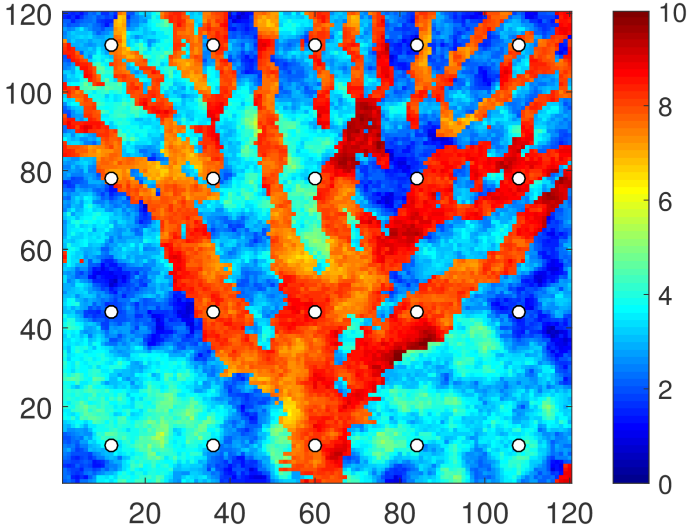
O-PCA Results



CNN-PCA Results



Bimodal Deltaic Fan System

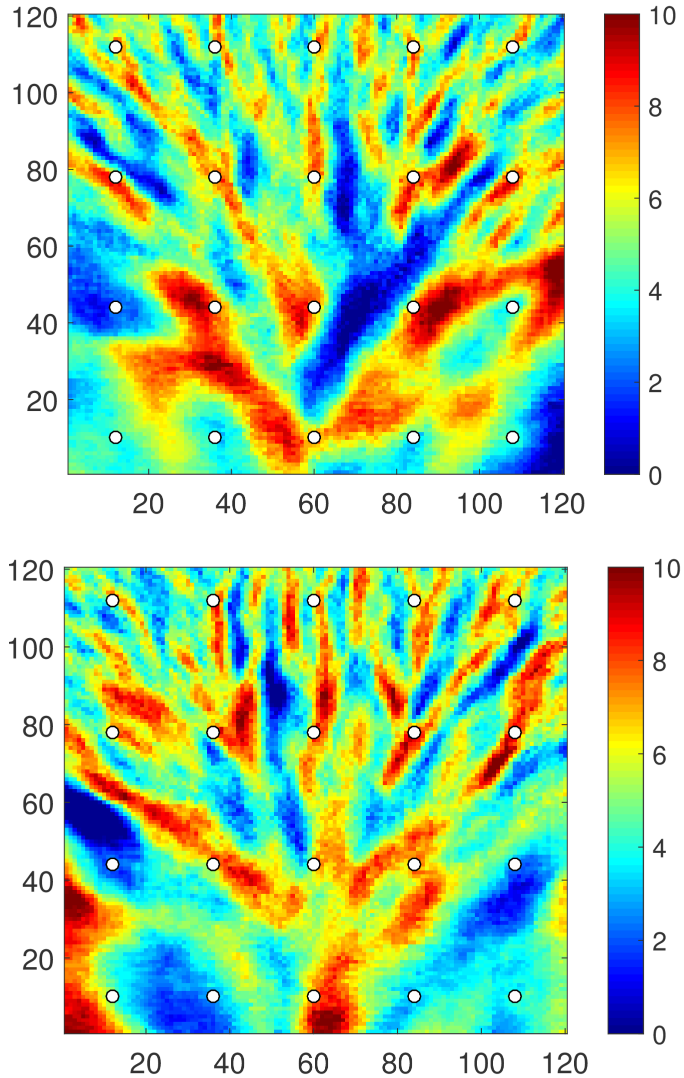


Training image
(one SGeMS realization)

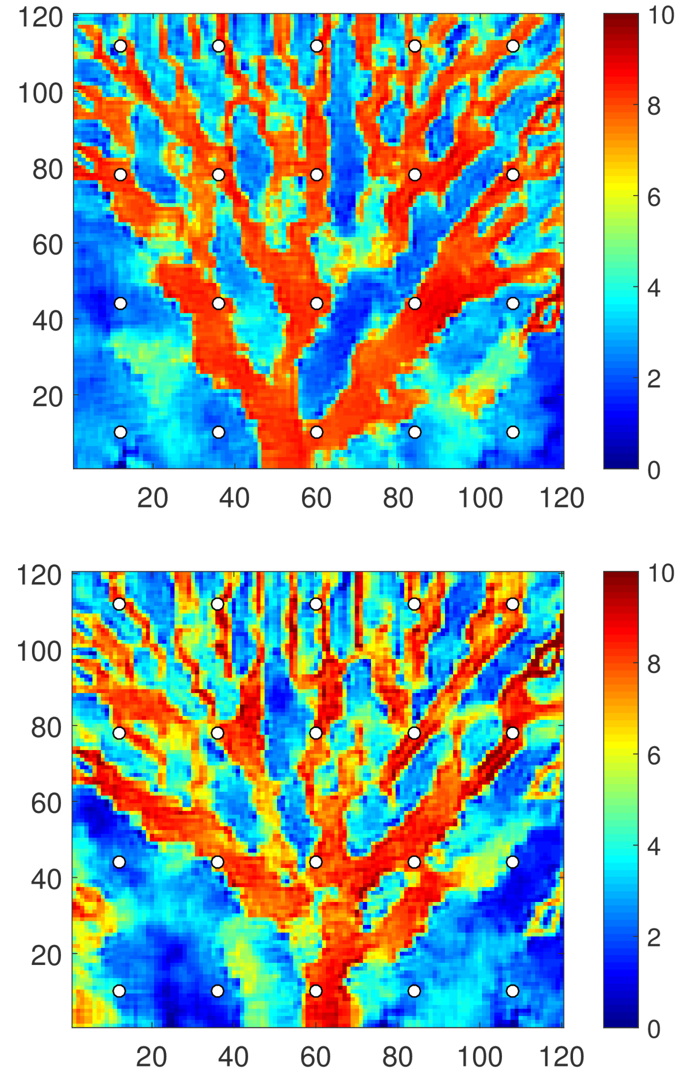
- Model size: 120 x 120, $N_c = 14{,}400$
- Hard data at 16 well locations
- Goal: represent with $l = 150$
- PCA from 1500 SGeMS realizations
- Train f_W with 3000 random PCA models (10 minutes on 1 GPU)
- $\gamma_s = 7.5$ and $\gamma_h = 10$
- CNN-PCA output further post-processed with bimodal O-PCA

Conditional Bimodal Deltaic Fan Models

PCA Realizations



CNN-PCA Realizations



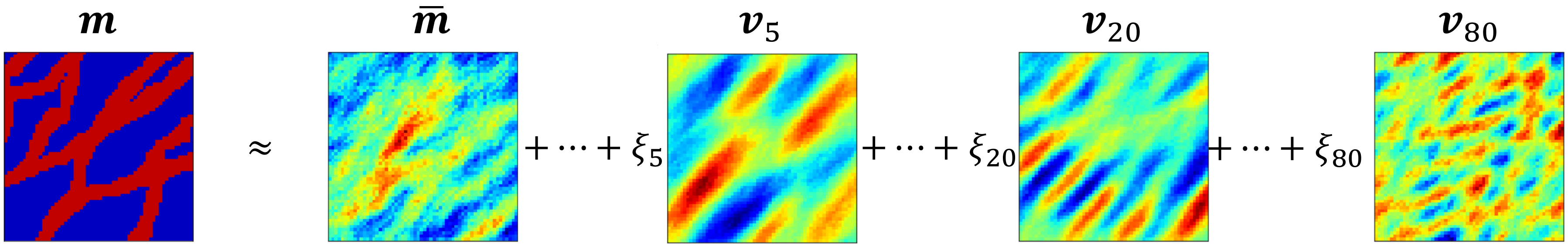
Summary & Future Directions

- Developed CNN-PCA by combining deep-learning-based neural style transfer algorithm with PCA
- CNN-PCA preserves spatial features better than O-PCA
- CNN-PCA prior models provide flow statistics consistent with SGeMS prior models
- History matching with CNN-PCA models achieved
- Extension to 3D; apply with adjoint-gradient and ensemble-based history matching methods; further testing

Acknowledgments

- Open source code:
  - O-PCA: Hai Vo (Chevron)
  - Neural style transfer (PyTorch): Hang Zhang
  - Fast neural style transfer (PyTorch): Abhishek Kadian
- PSO-MADS: Obi Isebor
- Stanford Smart Fields Consortium
- Stanford CEES

Multilevel History Matching Strategy



- $N_{\text{MADS}} = 2l = 160$
- Principal components $\boldsymbol{v}_i$ are of decreasing scale
- Determine PCA coefficients ξ_i separately
- Estimate ξ_i associated with larger-scale features first
- Accomplished previously for O-PCA (Liu, 2017)

Multilevel History Matching Strategy

- Assume $l = 80$
- 1 level strategy needs 160 simulations per MADS iteration
- 4 level strategy needs 40 simulations per MADS iteration

$$[\underbrace{\xi_1^0, \dots, \xi_{20}^0}_{\text{Optimize}}, \overbrace{\xi_{21}^0, \dots, \xi_{80}^0}^{\text{Fix}}]$$

$$[\underbrace{\xi_1^*, \dots, \xi_{20}^*}_{\text{Fix}}, \overbrace{\xi_{21}^0, \dots, \xi_{40}^0}^{\text{Optimize}}, \underbrace{\xi_{41}^0, \dots, \xi_{80}^0}_{\text{Fix}}]$$

$$[\underbrace{\xi_1^*, \dots, \xi_{40}^*}_{\text{Fix}}, \overbrace{\xi_{41}^0, \dots, \xi_{60}^0}^{\text{Optimize}}, \underbrace{\xi_{61}^0, \dots, \xi_{80}^0}_{\text{Fix}}]$$

$$[\underbrace{\xi_1^*, \dots, \xi_{60}^*}_{\text{Fix}}, \overbrace{\xi_{61}^0, \dots, \xi_{80}^0}^{\text{Optimize}}]$$

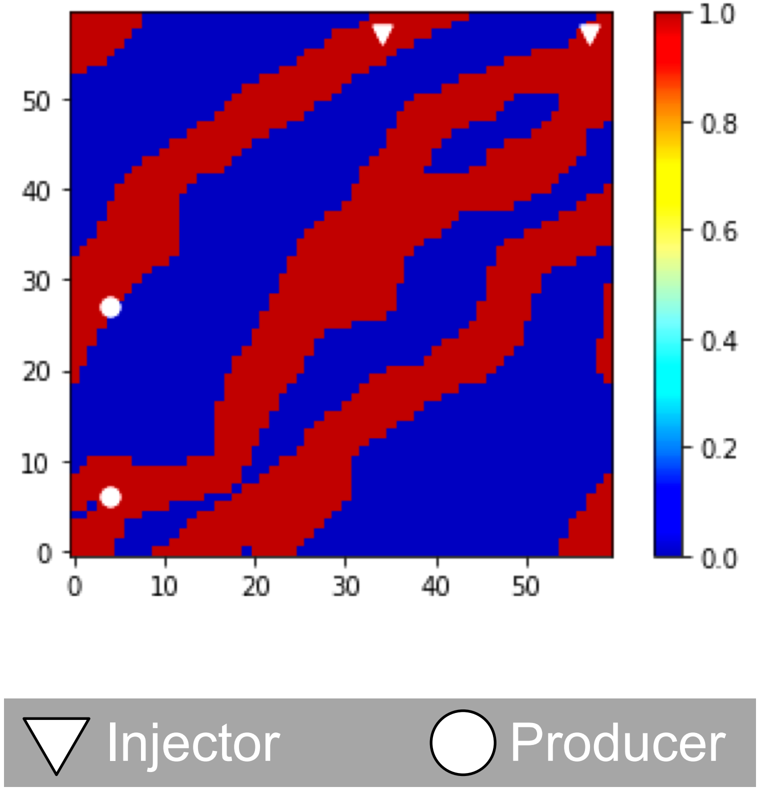
Multilevel History Matching Strategy

$$\xi_{\mathrm{RML}}^{*} = \underset{\xi}{\operatorname{argmin}}\{S(\xi)\}$$

$$S(\xi) = \frac{1}{2}(d(\xi) - d_{\mathrm{obs}}^{*})^{T} C_{d}^{-1}(d(\xi) - d_{\mathrm{obs}}^{*}) + \frac{1}{2}(\xi - \xi_{\mathrm{uc}})^{T}(\xi - \xi_{\mathrm{uc}})$$

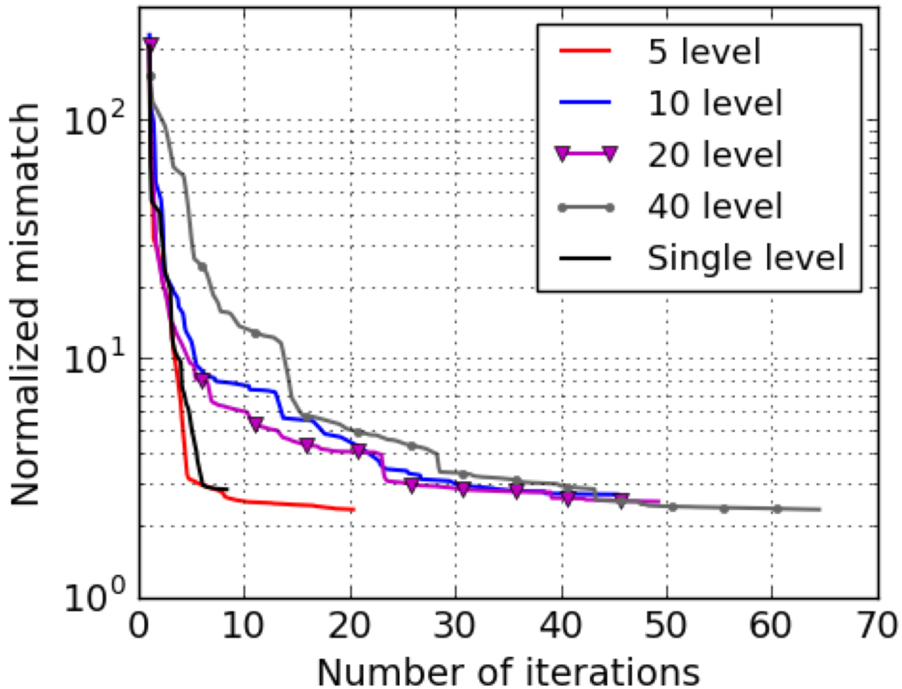
- N_L levels, evenly divide ξ
- $2l/N_L$ simulations at each MADS iteration
- Move to next level when little or no improvement
- Terminate when converged or at 200 iterations

Multilevel History Matching with CNN-PCA



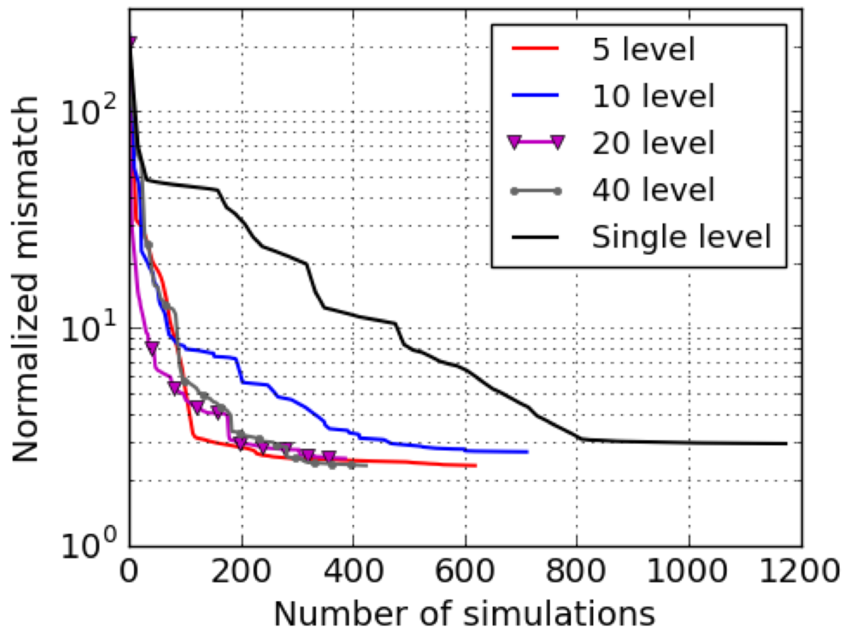
- Oil-water, 60 x 60 grid
- 2 injectors, 2 producers, BHP controlled
- No hard data
- k_{sand} = 2000 md, k_{mud} = 2 md
- Data: production and injection rates for 1000 days
- Goal: 10 RML posterior models
- Optimizer: MADS

Convergence Rate



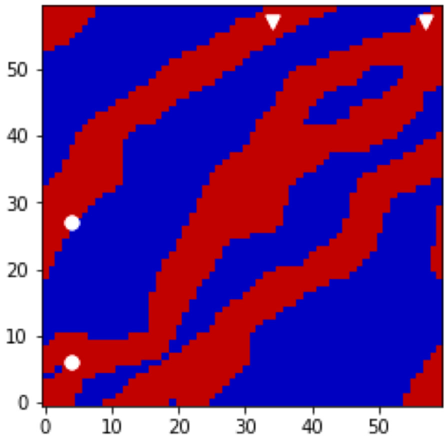
- Single level strategy converges in less than 10 iterations (average)
- Multilevel strategy with 5 levels convergence rate is comparable to single level

Number of Simulations

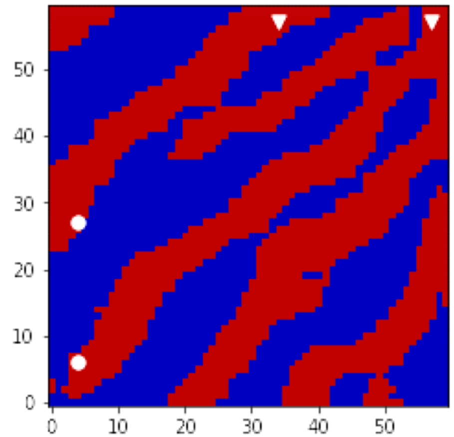


- From 1 level: ~1,188 simulations
- To 20 levels: ~380 simulations
- To 40 levels: ~423 simulations
- Multilevel strategy results in more rejected local minima (rejected runs included in above numbers)

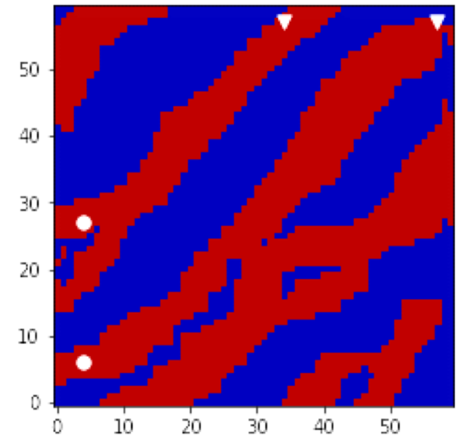
Permeability Estimation



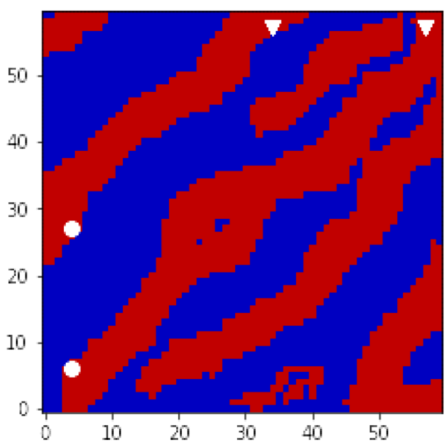
True model



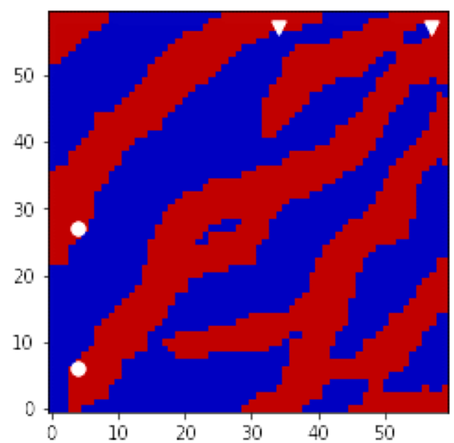
Single level



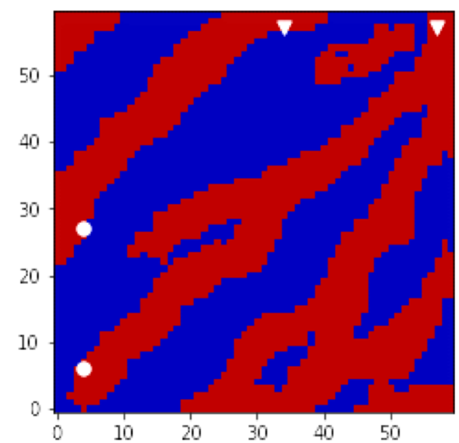
5 level



10 level



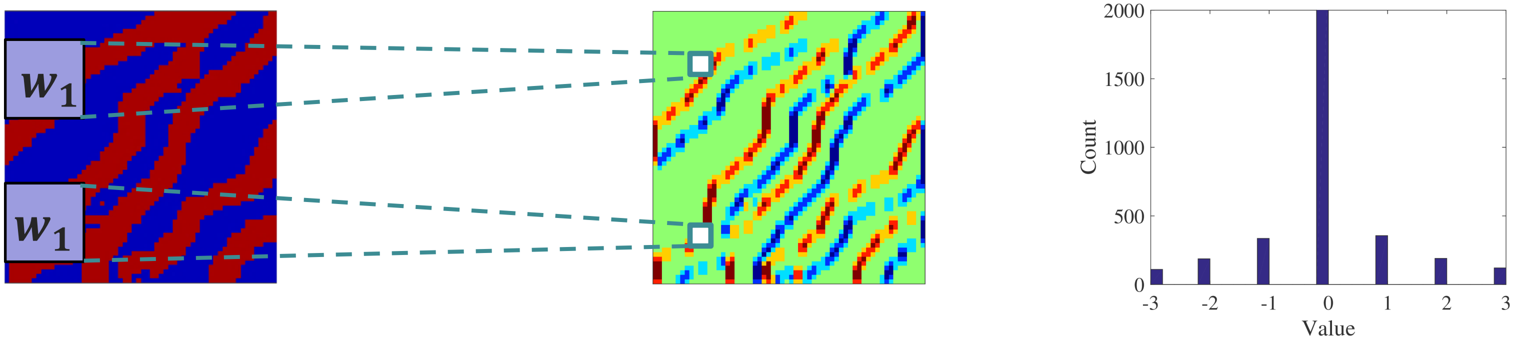
20 level



40 level

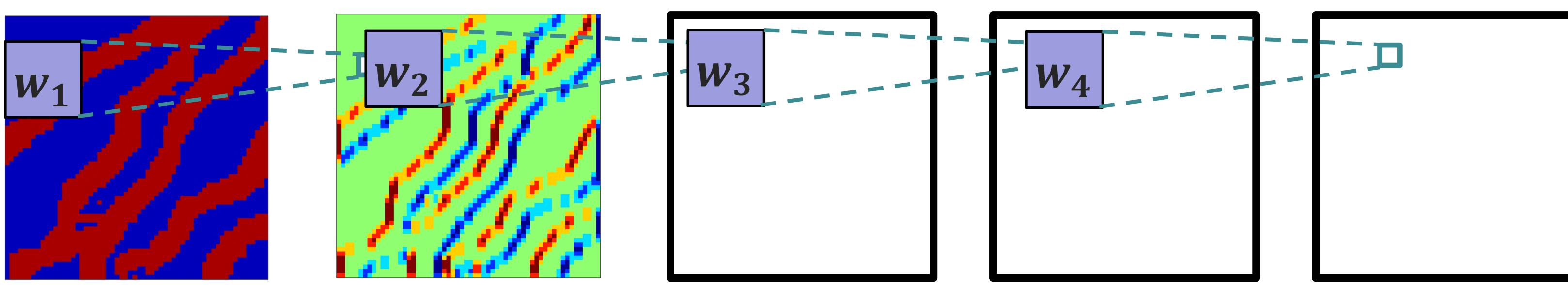
New Regularization Term

- $\psi_k(\boldsymbol{m})$ - lower-order statistical metrics of filter responses $\boldsymbol{F}(\boldsymbol{m})$



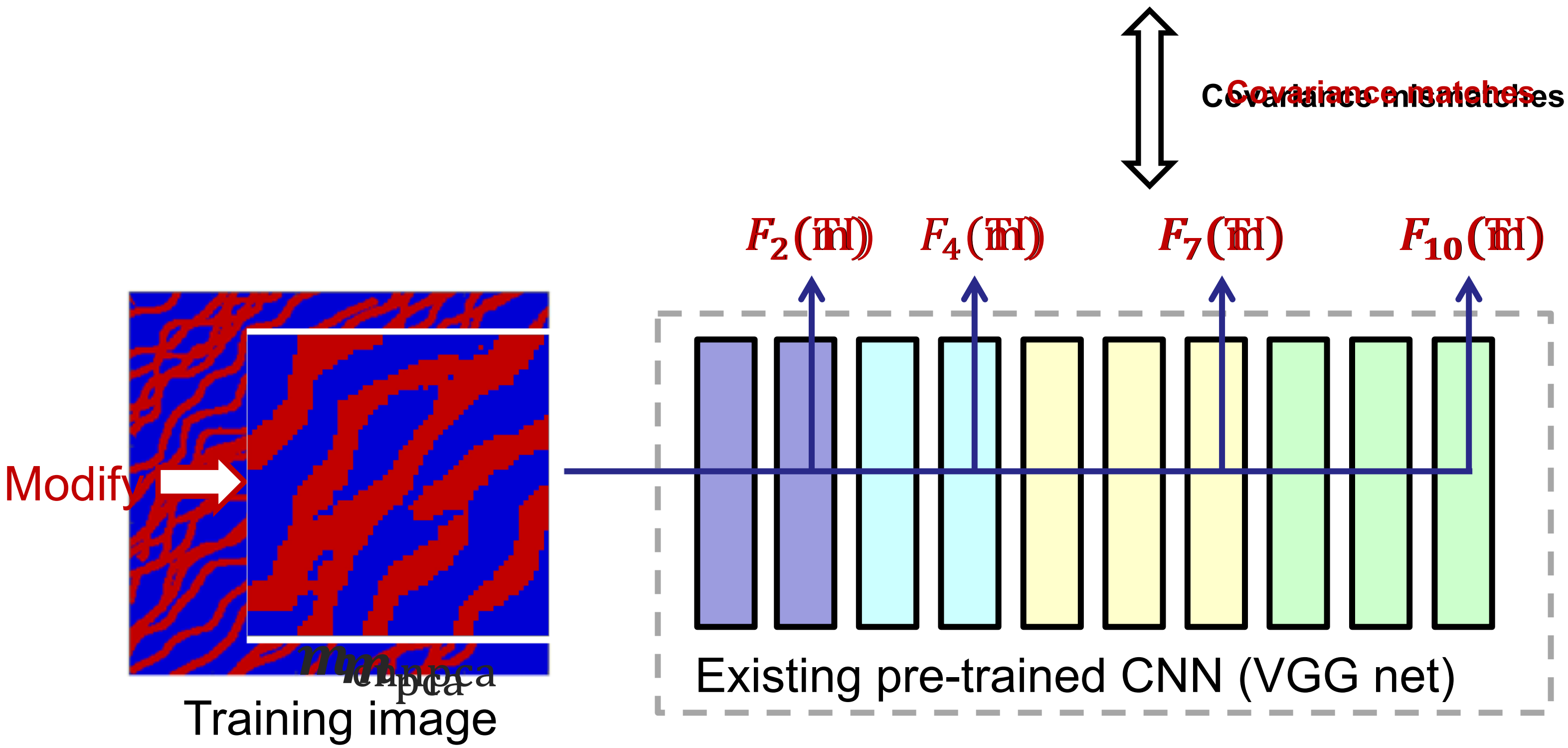
$\boldsymbol{m}$ $F_1(\boldsymbol{m})$ $\psi_1(\boldsymbol{m})$ = Histogram of $F_1(\boldsymbol{m})$

- Multiscale filter responses



$\boldsymbol{m}$ $F_1(\boldsymbol{m})$ $F_2(\boldsymbol{m})$ $F_3(\boldsymbol{m})$ $F_4(\boldsymbol{m})$

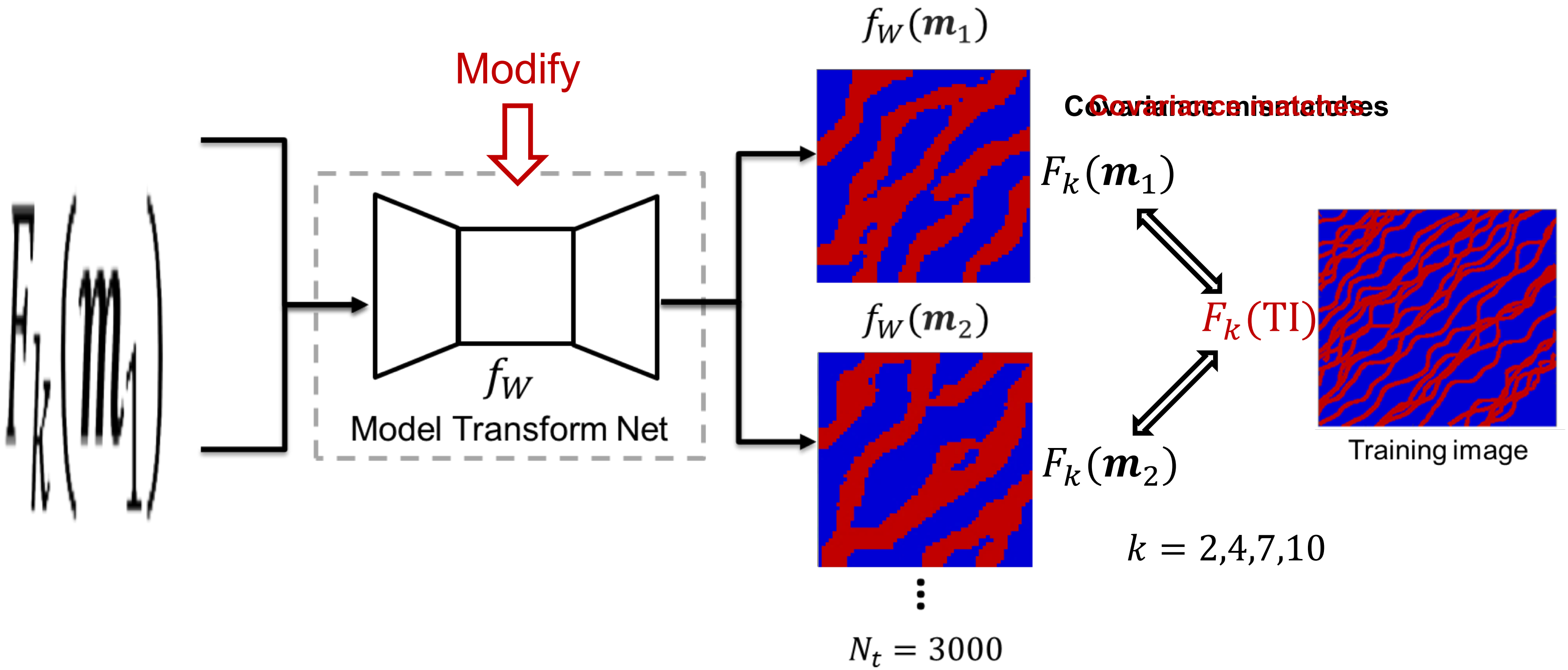
O-PCA with CNN-based Regularization



Need to solve an expensive optimization for every PCA model

Model Transform Net and CNN-PCA

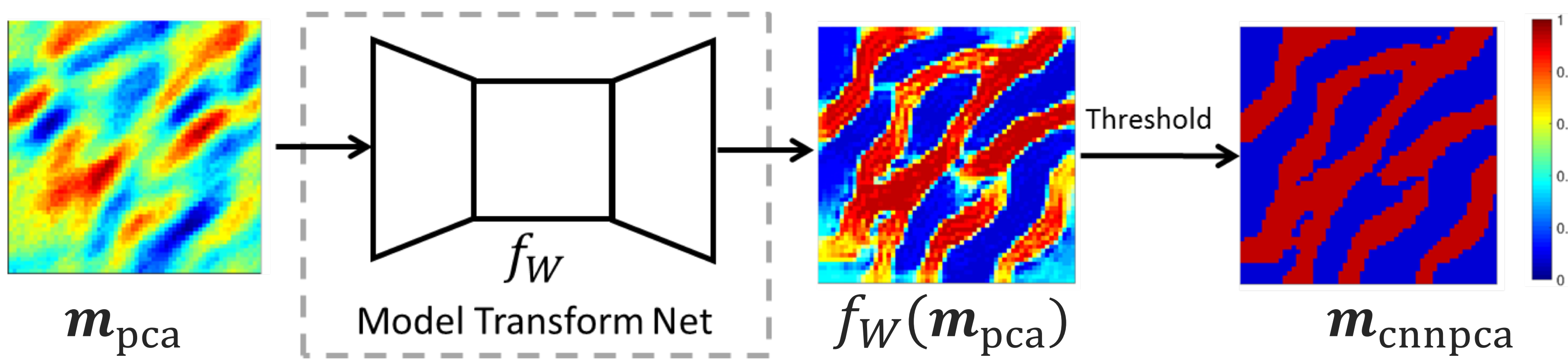
- Train another transform CNN f_W



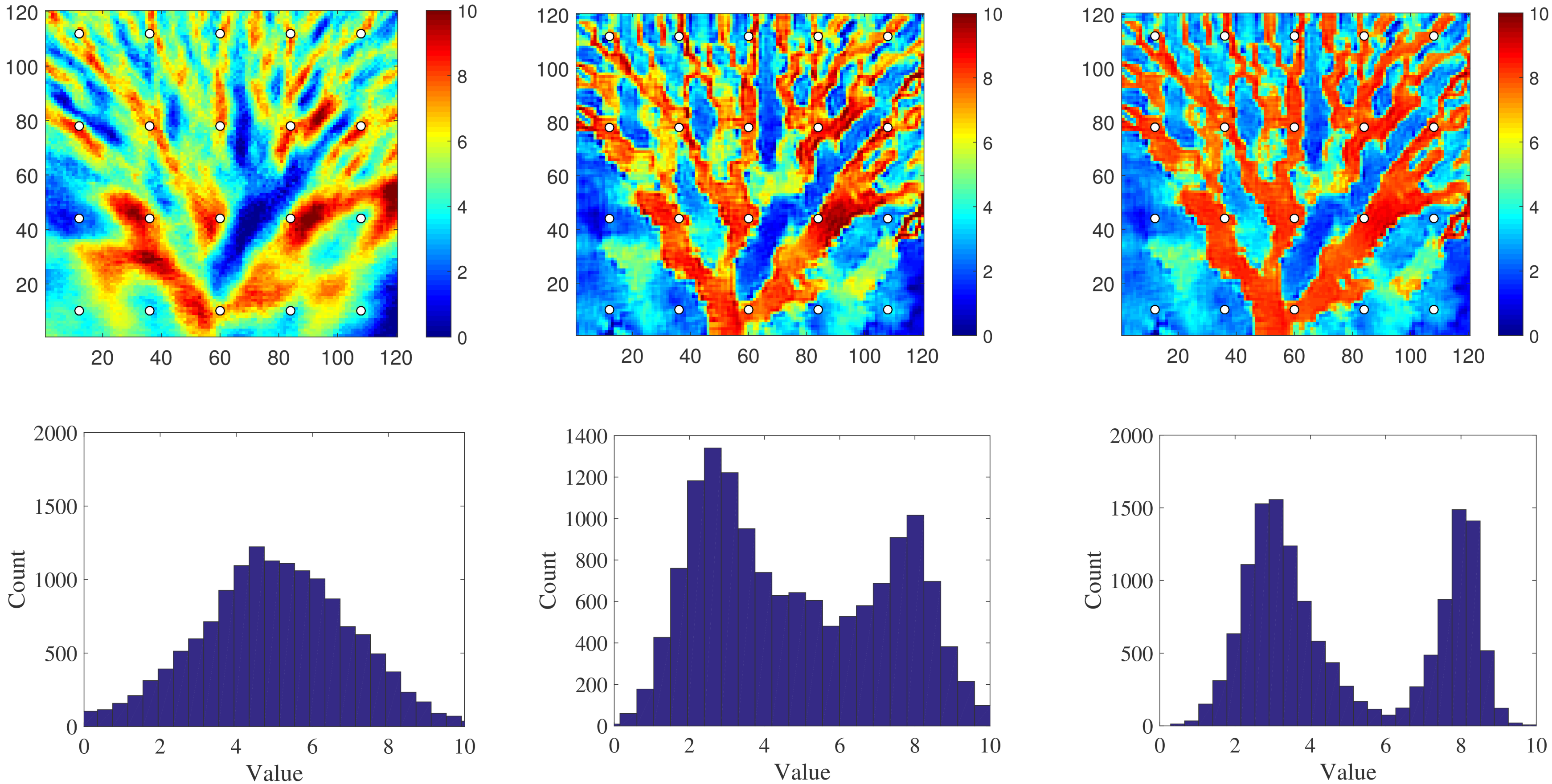
After training, transforming PCA models is almost real-time

CNN-PCA Final Threshold

- Output from the model transform net is not strictly binary
- Perform hard threshold to enforce binary output model



Bimodal Deltaic Fan System



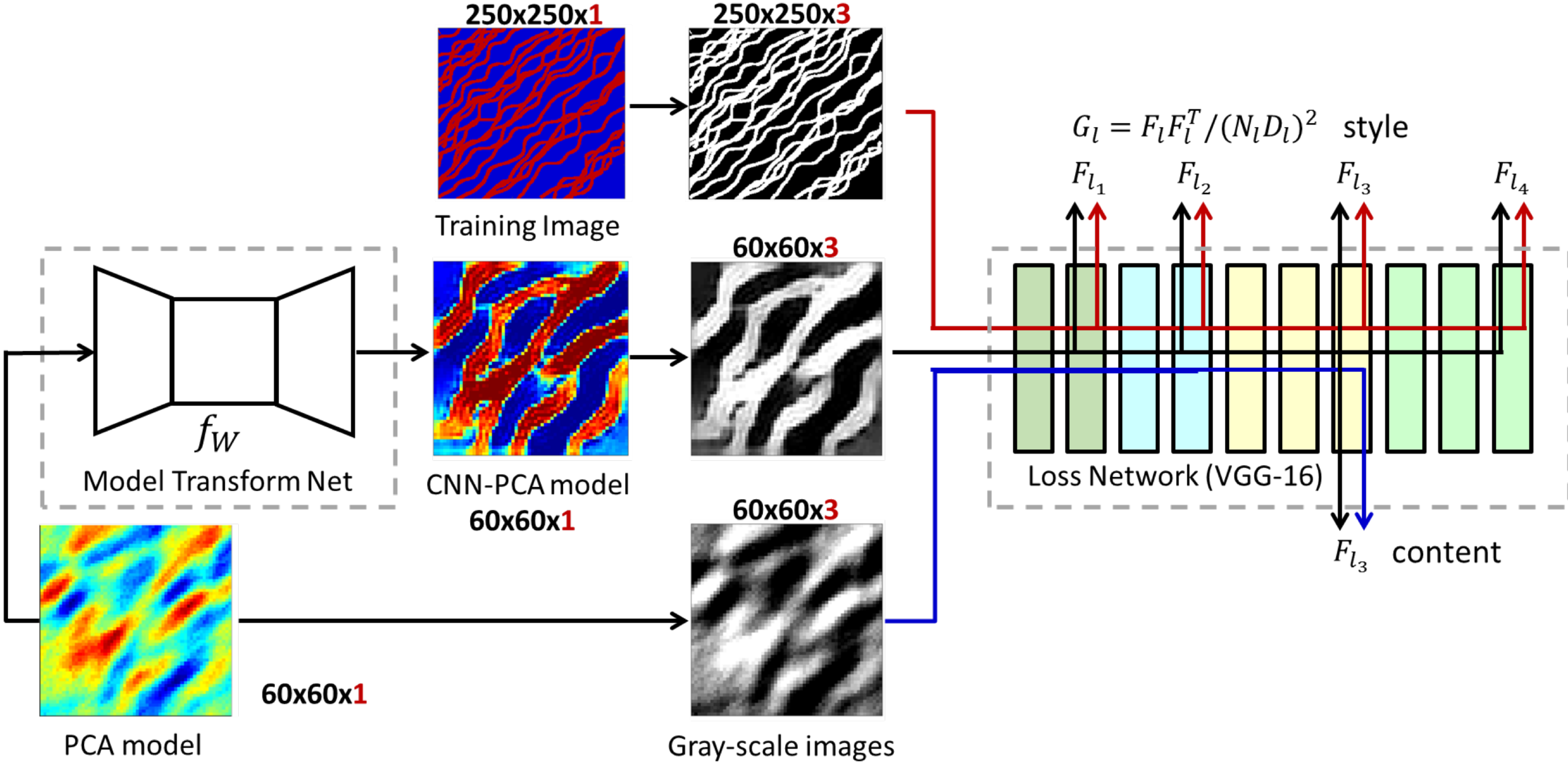
PCA CNN-PCA CNN-PCA + O-PCA

CNN-PCA for Reparameterization

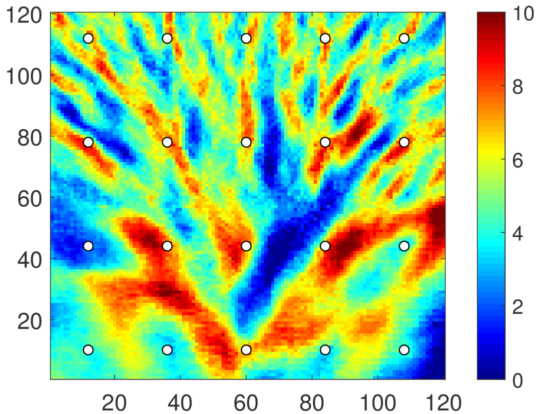
- Construct PCA with N_r SGeMS realizations

$$\boldsymbol{m}_{\mathbf{pca}} = \boldsymbol{U}_l \boldsymbol{\Lambda}_l^{1/2} \boldsymbol{\xi}_l + \bar{\boldsymbol{m}} \qquad \boldsymbol{\xi}_l \sim \boldsymbol{N}(\boldsymbol{0}, \boldsymbol{I})$$

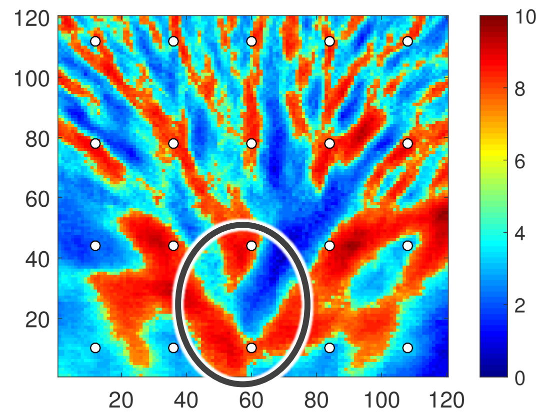
- Post-process $\boldsymbol{m}_{\mathbf{pca}}$ with fast neural style transfer algorithm



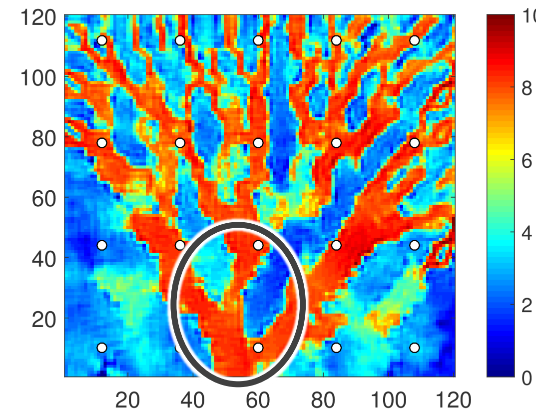
Conditional Bimodal Deltaic Fan Models



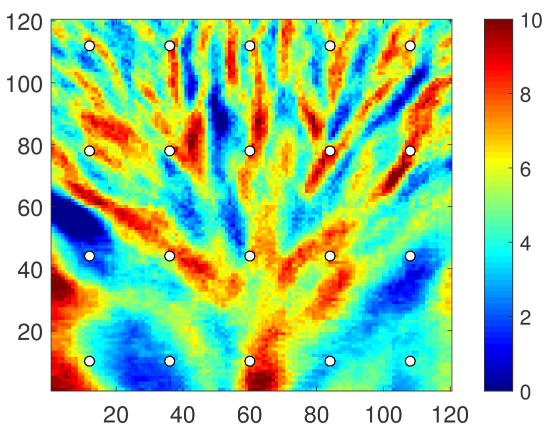
PCA Real. 1



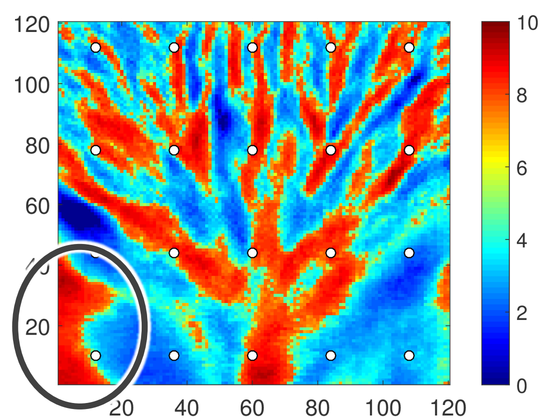
O-PCA Real. 1



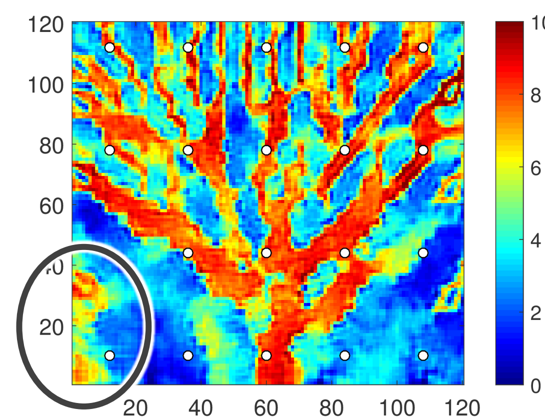
CNN-PCA Real. 1



PCA Real. 2



O-PCA Real. 2



CNN-PCA Real. 2