MIXSIM3D: a Novel 3D Contrastive Curriculum Learning Method Applied to Digital Rock Physics

Van Thao Nguyen[1,2], Dominique Fourer[2], Souhail Youssef[1], Désiré Sidibé[2]

[1] IFP Energies nouvelles, Rueil-Malmaison, France
[2] IBISC, Univ. Evry Paris-Saclay, Évry-Courcouronnes, France

August 29, 2025

Overview

Digital Rock Physics

How to determine the properties of porous media ?

- Measurement methods in Laboratory: expensive, time-consuming, scale limited with limited conditions.
- Digital experiments use advanced imaging like X-ray microtomography to create digital rock image.

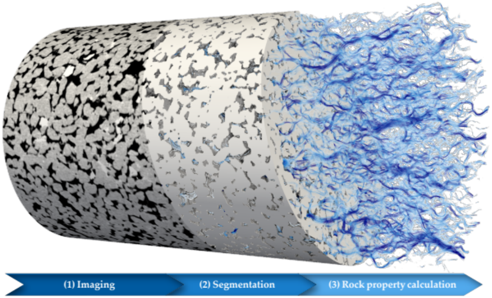


Figure: Illustration of a typical workflow in digital rock physics (DRP) for a core of a Bentheim sandstone. [Wetzel, 2021]

Digital Rock Physics

Traditional simulation methods

- Lattice Boltzmann method (LBM)
- Pore-scale modeling

Cons: Heavy computational burden for simulating multiphase flow, less accurate in complex porous media

Supervised Learning Methods

Cons: Overfitting, require accurately labeled datasets.

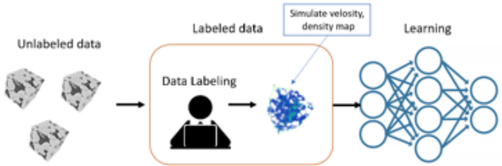


Figure: Supervised Learning workflow

Self-Supervised Representation Learning

- Leverages large amounts of unlabeled data
- Learns general-purpose representations through *pretext tasks*
- Learned representations are transferable and effective for downstream supervised tasks.

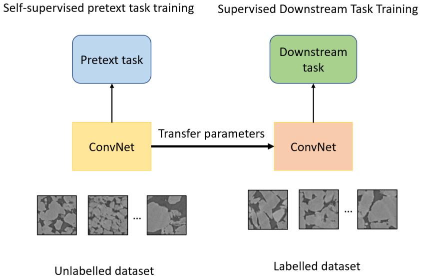
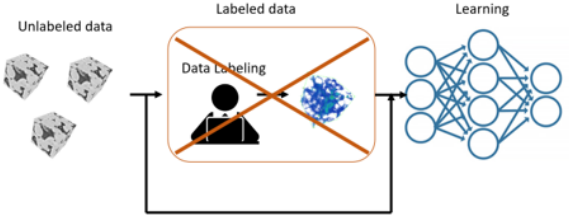


Figure: Supervised Learning workflow



- Step 1: Define a task that learn from data itself
- Step 2: Transfer to the specific task with few labeled samples.

State-of-the-art SSL methods

Contrastive methods (SimCLR, MoCo, CLIP), **Masking / prediction** (BERT, MAE, GPT), **Distillation / autoencoding** (BYOL, DINO).

Figure: SimCLR [Chen et al., 2020a]

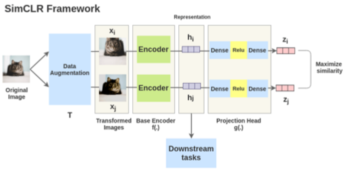


Figure: MoCo [Chen et al., 2020b]

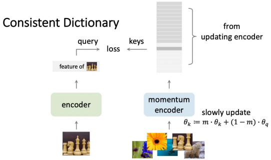


Figure: SimSiam[Chen and He, 2021]

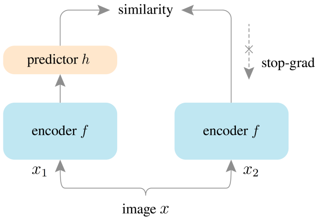
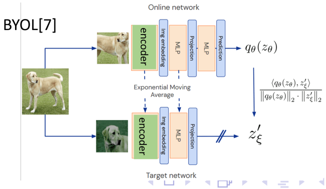


Figure: BYOL[Grill et al., 2020]

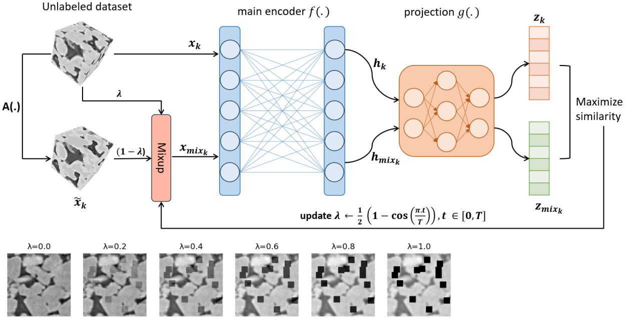


Proposed MixSim3D Algorithm

- MixSim3D applies progressive mixing between original and augmented samples to improve representation learning.
- Initially, the model emphasizes original data, but over time it smoothly integrates augmentations using a cosine schedule.
- Curriculum learning strategy to learn from easy (original) to more challenging (mixed) samples over time.

$$x_{\text{mix}} = (1-\lambda)x_{\text{original}} + \lambda x_{\text{augmented}} \tag{1}$$

where $\lambda \in [0,1]$ follows a cosine scheduler



Mixup Augmentation

Mixup: an augmentation technique to enhance the generalization capabilities of image classification models [Zhang et al., 2017]

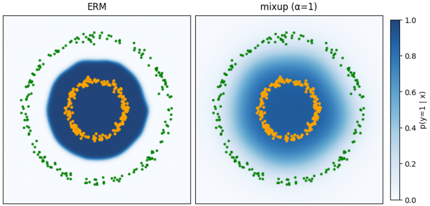
Formulation:

$$x_{\text{mix}} = \lambda x_i + (1 - \lambda) x_j, \tag{2}$$

$$y_{\text{mix}} = \lambda y_i + (1 - \lambda) y_j, \tag{3}$$

where x_i and x_j denote two distinct samples, and y_i and y_j are their corresponding labels.

Figure: Effect of mixup ($\alpha = 1$) on a toy problem. Green: Class 0. Orange: Class 1. Blue shading indicates $p(y = 1 \mid x)$. [Zhang et al., 2017]



Algorithm Overview

Algorithm 1: MixSim3D Pseudo-Code (Simplified)

Input: $f(\cdot)$: Encoder, $g(\cdot)$: Projection function
τ: Temperature, T: Number of epochs
`loader`: Mini-batch generator
Output: Trained networks $f(\cdot)$ and $g(\cdot)$
for $x_k \in \mathtt{loader}$ **do**

```
// For each mini-batch
```

$\tilde{x}_k \leftarrow A(x_k)$ `// Augmented sample`

$h_k \leftarrow f(x_k)$, $z_k \leftarrow g(h_k)$

$\lambda \leftarrow \frac{1}{2}\left(1 - \cos\left(\frac{\pi \cdot t}{T}\right)\right)$, at epoch $t \in [1, T]$

$x_{\text{mix}_k} \leftarrow (1-\lambda)x_k + \lambda\tilde{x}_k$

$h_{\text{mix}_k} \leftarrow f(x_{\text{mix}_k})$

$z_{\text{mix}_k} \leftarrow g(h_{\text{mix}_k})$

$$L_{\text{sim}} \leftarrow \frac{-1}{N}\sum_{k=1}^{N} \log\left(\frac{\exp\left(\frac{\text{sim}(z_k, z_{\text{mix}_k})}{\tau}\right)}{\sum_{\substack{m=1\\ m\neq k}}^{N} \exp\left(\frac{\text{sim}(z_k, z_{\text{mix}_m})}{\tau}\right)}\right)$$

Update $f(\cdot)$ and $g(\cdot)$ to minimize L_{sim}

return Trained networks $f(\cdot)$ and $g(\cdot)$

with $\text{sim}(x, y) = \frac{x^T y}{||x|| \cdot ||y||}$

Dataset

- 20,000 samples of size $100 \times 100 \times 100$ are randomly extracted, each associated with its corresponding permeability value.
- 6 rock types (blue points), are selected for training and validation, split in an 8:2 ratio
- 3 rock types (red points), are used to evaluate generalization performance.

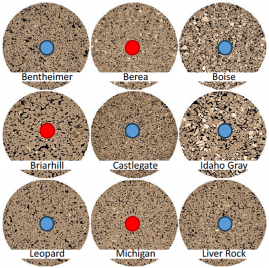
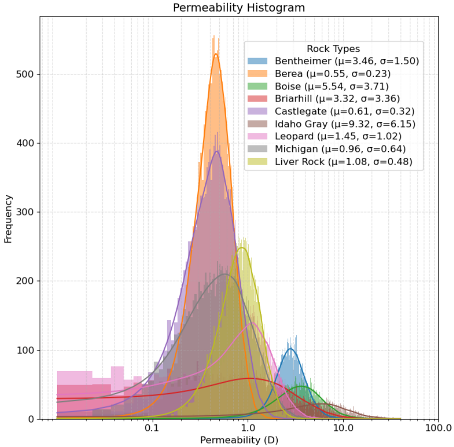


Figure: Permeability distribution of the dataset. The mean (μ) and standard deviation (σ) of permeability are indicated for each rock type in the legend.



Permeability computation - Ground truth

1. **Segmentation:** CT rock volume $\rightarrow$ pore/solid mask.
2. **LBM computation:** simulate flow on pores to obtain velocity/permeability heatmap.
3. **Sampling:** extract 3D patches (i, j, k) from the large volume, and label each patch with its corresponding permeability and rock-type class.
4. **3D CNN training:** predict permeability/rock-type.

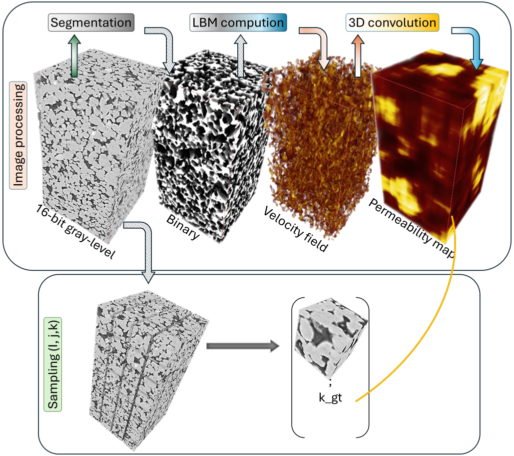
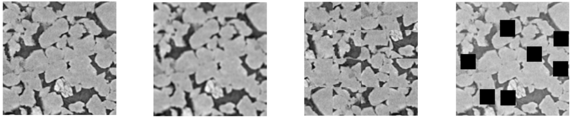


Figure: From segmentation and LBM method to generate permeability heatmap.

Implementation Details

- **Data Augmentation:**
  - Gaussian Blur, Gaussian Noise, Random Jigsaw, Cutout with probability $p = 0.5$
- **Training Setup:**
  - 30 epochs
  - Computation server: up to 12 nodes with $4 \times$A100 32GB GPUs per node.
  - Data parallelism across nodes
  - Each epoch requires $\approx$ 6 hours of training
- **Self-Supervised Pretraining:**
  - Dataset: 20,000 unlabeled 3D μCT $100 \times 100 \times 100$ images on 9 distinct rock types
- **Fine-tuning:**
  - Supervised training on 6 rock types
  - Data split: 80% training, 20% validation



Original image | Gaussian Blur | Random Jigsaw | Coarse droupout

Results for rock types classification task

We perform two experiments using 1% and 10% of the labeled data for training, respectively.

Table: Evaluation metrics using 1% of the dataset

Model	**F1 Score**	**Recall**	**Precision**	**Top1 Acc**
ResNet18[Feichtenhofer et al., 2019]	73.75	71.98	75.61	71.85
SimCLR [Chen et al., 2020a]	74.58	72.91	76.33	72.85
MoCo-v2 [He et al., 2020]	**81.18**	**81.04**	**81.32**	**81.04**
BYOL [Grill et al., 2020]	80.30	80.19	80.46	80.23
SimSiam [Chen and He, 2021]	75.45	75.01	75.64	75.56
MixSim3D	<u>80.65</u>	<u>80.64</u>	<u>80.65</u>	<u>80.62</u>

Table: Evaluation metrics using 10% of the dataset

Model	**F1 Score**	**Recall**	**Precision**	**Top1 Acc**
ResNet18[Feichtenhofer et al., 2019]	93.80	93.72	93.89	93.74
SimCLR [Chen et al., 2020a]	94.25	94.23	94.27	94.26
MoCo-v2 [He et al., 2020]	95.10	95.03	95.18	95.05
BYOL [Grill et al., 2020]	**95.47**	**95.46**	**95.48**	**95.47**
SimSiam [Chen and He, 2021]	93.91	93.85	93.98	93.87
MixSim3D	<u>95.26</u>	<u>95.24</u>	<u>95.29</u>	<u>95.33</u>

Additional Qualitative Results

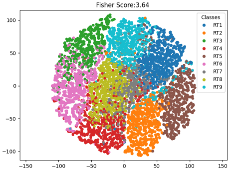
t-SNE & Fisher Score

- Plot t-SNE of latent embeddings for the nine rock types (RT1–RT9).
- Compute the Fisher score to quantify class separability:

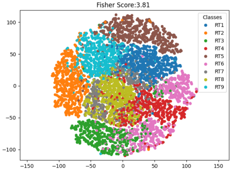
$$\mathcal{F} = \frac{\sum_{c=1}^{9} n_c \left\|\boldsymbol{\mu}_c - \boldsymbol{\mu}\right\|^2}{\sum_{c=1}^{9} \sum_{i \in c} \|\mathbf{z}_i - \boldsymbol{\mu}_c\|^2},$$

  where $\mathbf{z}_i$ are embeddings, $\boldsymbol{\mu}_c$ the class means, $\boldsymbol{\mu}$ the global mean, and n_c class sizes.
- Higher $\mathcal{F}$ indicates better separation.

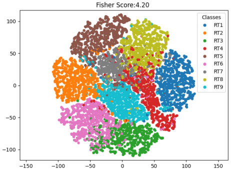
Figure: t-SNE visualizations of latent embeddings for state-of-the-art models and MixSim.



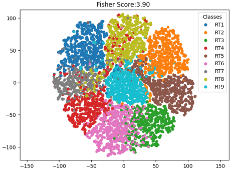
(a) ResNet-Fisher: 3.64



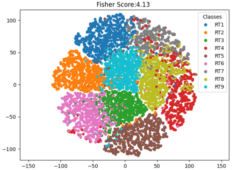
(b) SimCLR-Fisher: 3.81



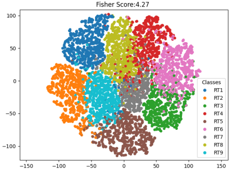
(c) BYOL-Fisher: 4.20



(d) SimSiam-Fisher: 3.9



(e) MoCo-Fisher: 4.13

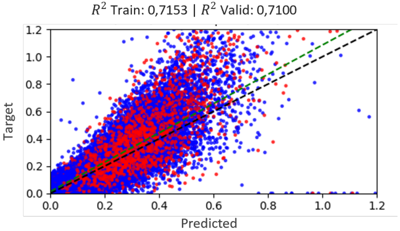


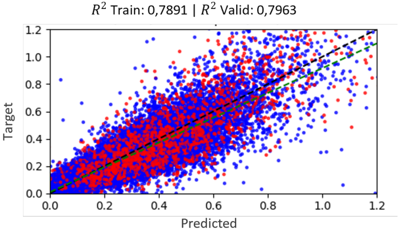
(f) MixSim-Fisher: 4.27

Rock permeability estimation results

Table: Permeability prediction results

Model	R^2 (Train)	R^2 (Validation)	L_2 (Train)	L_2 (Validation)
ResNet18 [Feichtenhofer et al., 2019]	0.7153	0.7100	7.4465	7.8241
SimCLR [Chen et al., 2020a]	0.7456	0.7478	7.0958	6.5604
SimSiam [Chen and He, 2021]	0.7387	0.7465	6.8354	6.6257
BYOL [Grill et al., 2020]	0.7795	0.7860	6.3073	5.5659
MoCo-v2 [He et al., 2020]	0.7648	0.7714	6.4059	6.3526
MixSim3D	**0.7819**	**0.7963**	**5.8450**	**5.3613**





Conclusion

- We introduced **MixSim3D**, a novel self-supervised learning method adapted for **3D data representation**.
- MixSim3D is especially adapted for 3D data where volumetric structure enriches the learning context.
- **MixSim3D** achieves comparable results on a real rock dataset to other **state-of-the-art contrastive methods** in both classification and regression tasks.

Future work

- Physics informed model through loss regularization
- Extending the dataset
- New applications of the method

Thanks

Contact: van-thao.nguyen@ifpen.fr

Pytorch Code: https://github.com/nguyenva04/mixsim3d_gretsi

Link to the paper:

References I

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020a).
A simple framework for contrastive learning of visual representations.
In *International conference on machine learning*, pages 1597–1607. PMLR.

Chen, X., Fan, H., Girshick, R., and He, K. (2020b).
Improved baselines with momentum contrastive learning.
arXiv preprint arXiv:2003.04297.

Chen, X. and He, K. (2021).
Exploring simple siamese representation learning.
In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758.

Feichtenhofer, C., Fan, H., Malik, J., and He, K. (2019).
Slowfast networks for video recognition.
In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al. (2020).
Bootstrap your own latent-a new approach to self-supervised learning.
Advances in neural information processing systems, 33:21271–21284.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020).
Momentum contrast for unsupervised visual representation learning.
In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.

References II

Wetzel, M. (2021).
Pore space alterations and their impact on hydraulic and mechanical rock properties quantified by numerical simulations.
PhD thesis, Universität Potsdam.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2017).
mixup: Beyond empirical risk minimization.
arXiv preprint arXiv:1710.09412.