Model-Based Deep Learning

COBREX week 2022

Supervised Learning

- **Goal**: learn a mapping f from an input space $\mathcal{X}$ to an output space $\mathcal{S}$ given a set of n training samples $\{(x_i, s_i)\}_{i=1..n}$ such that $(x_i, s_i) \in \mathcal{X} \times \mathcal{S}$.
- **Parametrized function**: the function f can be parametrized with a set of weights $\theta \in \Theta$, denoted f_θ.

$$f_\theta : \mathcal{X} \mapsto \mathcal{S}$$
$$x \rightarrow \hat{s}$$

- **Optimization problem**: for a given loss function $\mathcal{L}$ (MSE, Cross-Entropy, ...):

$$\min_{\theta \in \Theta} \sum_{i=1}^{n} \mathcal{L}(s_i, f_\theta(x_i)) + \lambda \Omega(\theta)$$

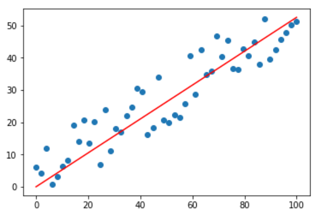


Figure 1: Linear Regression

Deep Learning

- **Artifical Neural Networks (ANN)**: collection of connected nodes with weights on edges
- **Deep Neural Networks (DNN)**: Several layers of ANN stacked together
- **Weights Update**: Gradient descent algorithm:

$$\theta_{t+1} = \theta_t - \eta \sum_i \nabla_\theta \mathcal{L}(s_i, f_\theta(x_i))$$

- **Applications**: Computer vision, image processing, natural language processing, speech recognition ..

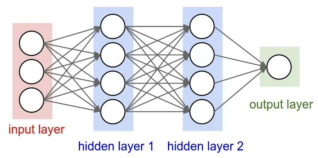


Figure 2: Deep neural network with 3 layers

Deep Learning

- **Upsides**:
  - good performances
  - no need for hand-crafted features
  - scalable
  - fast inference
- **Downsides**:
  - black box (non-interpretable)
  - requires a (large) training set with groundtruth labels
  - requires dedicated hardware (GPUs, TPUs) for training

Model-based method for inverse problems

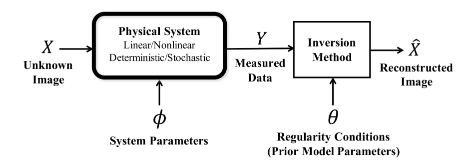


Figure 3: Inverse problem: principle

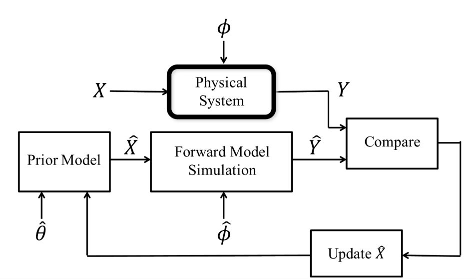


Figure 4: Model-based method

Model-based method for inverse problems

From a statistical standpoint, the problem can be described with **maximum a posteriori (MAP)**:

$$\begin{aligned}
\hat{x}_{MAP} &= \arg\max_x p(x|y) \\
&= \arg\max_x \frac{p(y|x)p(x)}{p(y)} \\
&= \arg\max_x log(p(y|x)) + log(p(x)) - log(p(y)) \\
&= \arg\min_x -log(p(y|x)) - log(p(x)) \\
&= \arg\min_x \underbrace{f(x)}_{\text{Forward model}} + \underbrace{h(x)}_{\text{Prior}}
\end{aligned}$$

Model-Based Approaches

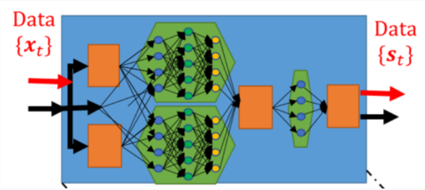
- Dominating type of algorithm in signal processing
- Hand-designed from domain knowledge
- Do not rely on data to learn the mapping, but data is used to estimate a small number of parameters
- Explicit model of the relationship between input and output variables
- **Examples:** Kalman filter, Iterative Shrinkage Thresholding Algorithm (ISTA), Alternating Direction Method of Multipliers (ADMM), etc ..

Model-Based Approaches

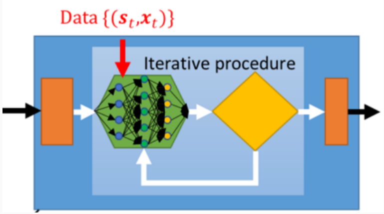
- **Upsides:**
  - Interpretability
  - Good performances if the model accurate and perfectly known
- **Downsides:**
  - Requires domain knowledge (statistical models, or deterministic rules)
  - Rely on some assumptions about the underlying statistics, which do not always hold (linear system, Gaussian and independant noise, etc..)

Model-Based Deep Learning

- **Model-Aided network:** specific DNN architecture tailored for the problem at hand
- **DNN-Aided inference:** specific parts of the model-based algorithm are augmented with deep learning tools



(a) Model-Aided network



(b) DNN-Aided inference

Model-Based Deep Learning[1]

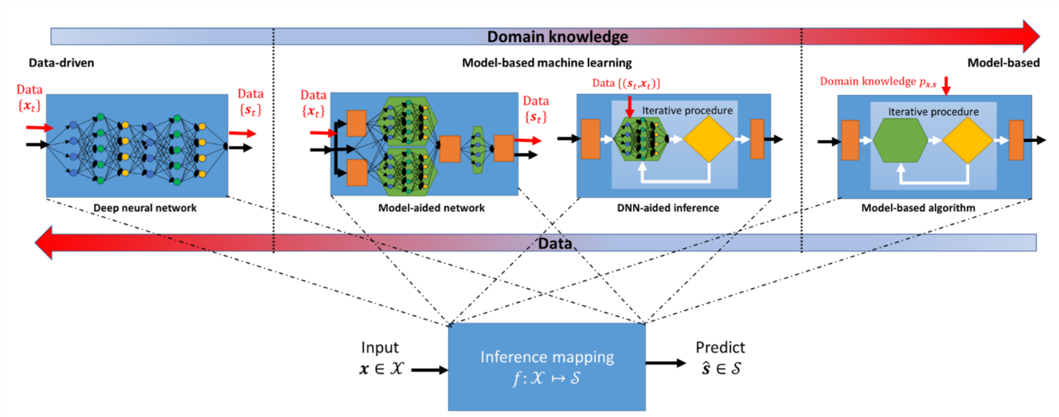


Figure 6: Illustration of model-based versus data-driven inference. The red arrows correspond to computation performed before the particular inference data is received.

[1] Nir Shlezinger et al. "Model-based deep learning". In: *arXiv preprint arXiv:2012.08405* (2020).

Model-Based Deep Learning

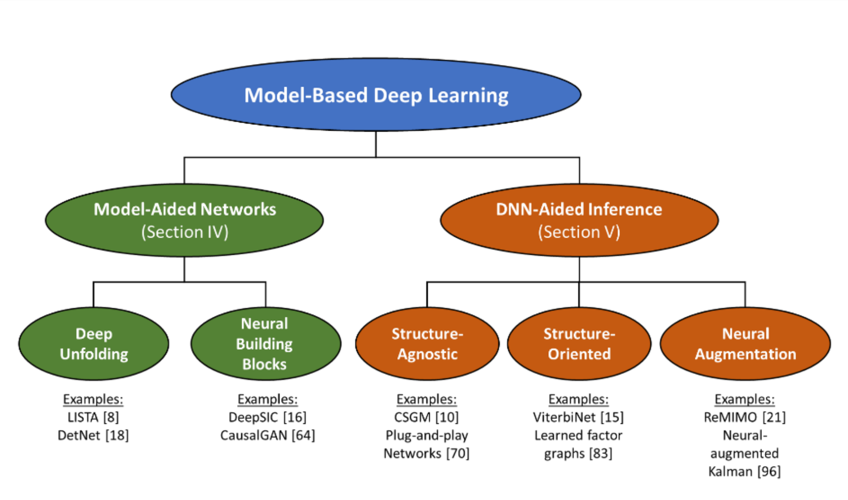


Figure 7: Division of model-based deep learning techniques into categories and sub-categories.

LISTA[2]

- **Sparse Coding** Let $x \in \mathbb{R}^n$ the input noisy signal. We try to solve the following sparse decomposition problem:

$$\min_{\alpha} \frac{1}{2}||x - D\alpha||^2 + \lambda||\alpha||_1 \tag{1}$$

  where $D = [d_1, \cdots, d_p] \in \mathbb{R}^{n \times p}$ is the dictionary, the sparse vector $\alpha \in \mathbb{R}^p$ is the code, and λ is the regularization constant.

[2] Karol Gregor and Yann LeCun. "Learning fast approximations of sparse coding". In: *International Conference on Machine Learning*. 2010.

LISTA

- **Iterative Shrinkage Thresholding Algorithm (ISTA)**[3]
  ISTA and FISTA [Beck & Teboulle, 2009] : model-based iterative algorithm to solve problem (1):

$$\alpha^{(k+1)} = S_\lambda\left[\alpha^{(k)} + \eta \mathrm{D}^\top(\mathrm{x} - \mathrm{D}\alpha^{(k)})\right] \tag{2}$$

such that $\mathrm{D} \in \mathbb{R}^{n \times p}$, $\lambda, \eta \in \mathbb{R}$ and S_λ the thresholding function defined by:

$$\forall j \in [\![1, p]\!], \quad S_\eta(\alpha)_j = sign(\alpha_j) max(0, |\alpha_j| - \lambda) \tag{3}$$

[3] Amir Beck and Marc Teboulle. "A fast iterative shrinkage-thresholding algorithm for linear inverse problems". In: *SIAM journal on imaging sciences* (2009).

LISTA

- **Iterative Shrinkage Thresholding Algorithm (ISTA)**

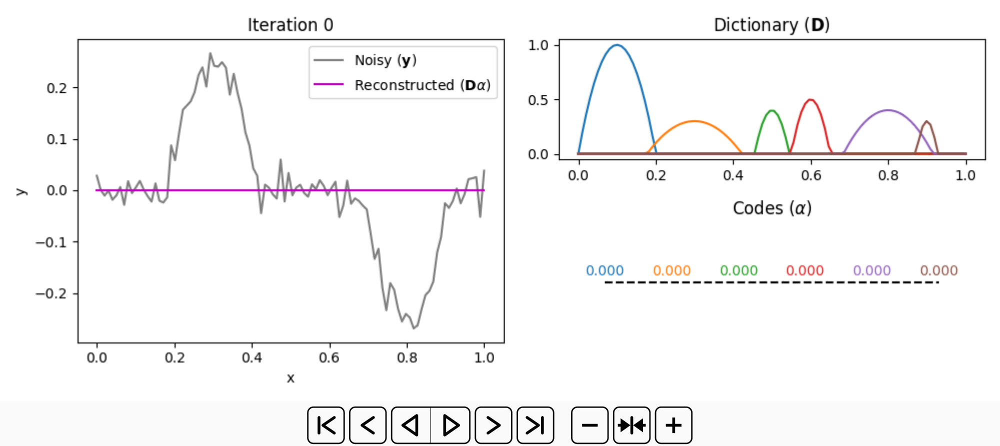


Figure 8: ISTA for signal denoising on a toy example.

- **Learned ISTA (LISTA)**
  The iterative steps to solve (1) (gradient descent + soft-thresholding) can be unrolled to a DNN of fixed depth K, such that $\mathrm{D} \in \mathbb{R}^{n \times p}$, $\lambda \in \mathbb{R}^p$, and $\eta \in \mathbb{R}$ become learnable parameters (weights of the network).

$$\alpha^{(k+1)} = S_\lambda\left[\alpha^{(k)} + \eta \mathrm{D}^\top(\mathrm{x} - \mathrm{D}\alpha^{(k)})\right] \tag{4}$$

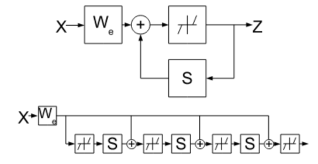


Figure 9: Unrolled iterations (original notations)

LISTA

- Reduced number of parameters w.r.t. DNN-based denoiser
- Fast inference speed
- Interpretable model parameters
- Smaller amount of training data needed
- Can be trained on non-gaussian noise
- Can be extended to 2D of 3D data with image patches

Deep Unfolded Projected Gradient Descent[4]

- **System model:** Symbol detection

$$\mathbf{x} = \mathbf{H}\mathbf{s} + \mathbf{w}$$

with $\mathbf{x} \in \mathbb{R}^n$ the observation, $\mathbf{s} \in \mathcal{S} = \{\pm 1\}^K$ the signal to recover, and $\mathbf{w} \in \mathbb{R}^n$ i.i.d Gaussian noise. The channel matrix $\mathbf{H} \in \mathbb{R}^{n \times K}$ is known.

- **Problem:**

$$\hat{s} = \arg\min_{\mathbf{s} \in \{\pm 1\}^K} ||\mathbf{x} - \mathrm{Hs}||^2$$

The search space becomes too large for large values of K (2^K).

[4] Neev Samuel, Tzvi Diskin, and Ami Wiesel. "Learning to detect". In: *IEEE Transactions on Signal Processing* (2019).

Deep Unfolded Projected Gradient Descent

- **Model-based algorithm:** Projected Gradient Descent

$$\hat{\boldsymbol{s}}_{q+1} = \mathcal{P}_{\mathcal{S}}\left(\hat{\boldsymbol{s}}_q - \eta_q \frac{\partial \|\boldsymbol{x} - \boldsymbol{H}\boldsymbol{s}\|^2}{\partial \boldsymbol{s}}\bigg|_{\boldsymbol{s}=\hat{\boldsymbol{s}}_q}\right)$$
$$= \mathcal{P}_{\mathcal{S}}\left(\hat{\boldsymbol{s}}_q - \eta_q \boldsymbol{H}^T\boldsymbol{x} + \eta_q \boldsymbol{H}^T\boldsymbol{H}\hat{\boldsymbol{s}}_q\right)$$

- **Unfolded DetNet:** Projected Gradient Descent

$$\boldsymbol{z}_q = \mathrm{ReLU}\left(\boldsymbol{W}_{1,q}\left((\boldsymbol{I}+\delta_{2,q}\boldsymbol{H}^T\boldsymbol{H})\hat{\boldsymbol{s}}_{q-1} - \delta_{1,q}\boldsymbol{H}^T\boldsymbol{x}\right) + \boldsymbol{b}_{1,q}\right)$$

$$\hat{\boldsymbol{s}}_q = \mathrm{soft\ sign}\left(\boldsymbol{W}_{2,q}\boldsymbol{z}_q + \boldsymbol{b}_{2,q}\right)$$

with trainable parameters

$$\boldsymbol{\theta} = \{(\boldsymbol{W}_{1,q}, \boldsymbol{W}_{2,q}, \boldsymbol{b}_{1,q}, \boldsymbol{b}_{2,q}, \delta_{1,q}, \delta_{2,q})\}_{q=1}^{Q}$$

- **Results:**
  - requires an order of magnitude less iterations (layers), improved runtime
  - competitive performances

- **System model:** reconstruct clean signal $\mu \in \mathbb{R}^n$ disturbed with **Poisson noise** from a noisy measurement $x \in \mathbb{R}^n$, with some a priori knowledge:

$$\log(\boldsymbol{\mu}) = \sum_{c=1}^{C} \boldsymbol{h}_c * \boldsymbol{s}^c = \boldsymbol{H}\boldsymbol{s}$$

Figure 10: Convolutional generative model (CGM), $s \in \mathbb{R}^n$ is sparse

- **Problem:** Poisson noise + CGM

$$\begin{aligned}(\hat{\boldsymbol{s}}, \{\hat{\boldsymbol{h}}_c\}_{c=1}^{C}) &= \underset{\boldsymbol{s},\{\boldsymbol{h}_c\}}{\arg\min} - \log p_{\boldsymbol{x}|\boldsymbol{\mu}}(\boldsymbol{x}|\boldsymbol{\mu} = \boldsymbol{H}\boldsymbol{s}) + \lambda\|\boldsymbol{s}\|_1 \\ &= \underset{\boldsymbol{s},\{\boldsymbol{h}_c\}}{\arg\min} \, \mathbf{1}^T \exp\left(\boldsymbol{H}\boldsymbol{s}\right) - \boldsymbol{x}^T\boldsymbol{H}\boldsymbol{s} + \lambda\|\boldsymbol{s}\|_1,\end{aligned}$$

In this case, the matrix H is not known.

- **Model-based algorithm:** Proximal Gradient mapping, 2-steps process

$$\hat{\boldsymbol{s}}_{q+1} = \mathcal{T}_b\left(\hat{\boldsymbol{s}}_q + \eta \boldsymbol{H}^T\left(\boldsymbol{x} - \exp\left(\boldsymbol{H}\hat{\boldsymbol{s}}_q\right)\right)\right)$$

Figure 11: First step: update of the code s

$$\hat{\boldsymbol{H}}_{l+1} = \arg\min_{\boldsymbol{H}} \mathbf{1}^T \exp\left(\boldsymbol{H}\boldsymbol{s}\right) - \boldsymbol{x}^T \boldsymbol{H}\boldsymbol{s}, \quad \text{subject to } \boldsymbol{s} = \hat{\boldsymbol{s}}_{l+1}.$$

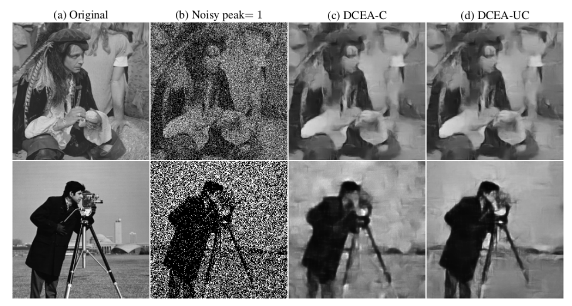
Figure 12: Second step: update of the dictionary H

Deep Unfolded Dictionary Learning

- **Deep Convolutional Exponential-Family Autoencoder (DCEA):** Unrolled iterations:

$$\hat{\boldsymbol{s}}_{q+1} = \mathcal{T}_b\left(\hat{\boldsymbol{s}}_q + \eta \boldsymbol{W}_2^T\left(\boldsymbol{x} - \exp\left(\boldsymbol{W}_1 \hat{\boldsymbol{s}}_q\right)\right)\right)$$

  Two variants: DCEA-C ($W_1 = W_2$) and DCEA-UC ($W_1 \neq W_2$).
- **Results:**



DNN-aided inference

Plug-and-Play Networks for Image Restoration[5]

- **System Model:**

$$\mathbf{x} = \mathbf{H}\mathbf{s} + \mathbf{w}$$

with $\mathbf{x} \in \mathbb{R}^m$ the observation, $\mathbf{s} \in \mathbb{R}^n$ the signal to recover, and $\mathbf{w} \in \mathbb{R}^m$ i.i.d Gaussian noise, and $\mathbf{H} \in \mathbb{R}^{m\times n}$.

- **Problem:**

$$\begin{aligned}\hat{\boldsymbol{s}} &= \arg\min_{\boldsymbol{s}} -\log p(\boldsymbol{s}|\boldsymbol{x}) \\ &= \arg\min_{\boldsymbol{s}} -\log p(\boldsymbol{x}|\boldsymbol{s}) - \log p(\boldsymbol{s}) \\ &= \arg\min_{\boldsymbol{s}} \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{H}\boldsymbol{s}\|^2 + \phi(\boldsymbol{s})\end{aligned}$$

[5] Singanallur V Venkatakrishnan, Charles A Bouman, and Brendt Wohlberg. "Plug-and-play priors for model based reconstruction". In: *2013 IEEE Global Conference on Signal and Information Processing*.

Plug-and-Play Networks for Image Restoration

- **Model-Based:** Alternating Direction Method of Multipliers (ADMM)

  The problem can be reformulated as:

$$\hat{\boldsymbol{s}} = \arg\min_{\boldsymbol{s}} \min_{\boldsymbol{v}} \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{H}\boldsymbol{s}\|^2 + \phi(\boldsymbol{v}) \quad \text{subject to } \boldsymbol{v} = \boldsymbol{s}.$$

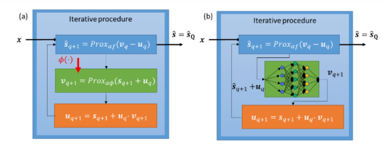
  which can be processed with ADMM.

$$\begin{aligned}
\hat{\boldsymbol{s}}_{q+1} &= \arg\min_{\boldsymbol{s}} \frac{\alpha}{2}\|\boldsymbol{x} - \boldsymbol{H}\boldsymbol{s}\|^2 + \frac{1}{2}\|\boldsymbol{s} - (\boldsymbol{v}_q - \boldsymbol{u}_q)\|^2, \\
\boldsymbol{v}_{q+1} &= \arg\min_{\boldsymbol{v}} \alpha\phi(\boldsymbol{v}) + \frac{1}{2}\|\boldsymbol{v} - (\hat{\boldsymbol{s}}_{q+1} + \boldsymbol{u}_q)\|^2, \\
\boldsymbol{u}_{q+1} &= \boldsymbol{u}_q + (\hat{\boldsymbol{s}}_{q+1} - \boldsymbol{v}_{q+1}).
\end{aligned}$$

  The second step can be replaced by a pre-trained DNN denoiser f_θ:

$$\boldsymbol{v}_{q+1} = f_{\boldsymbol{\theta}}(\hat{\boldsymbol{s}}_{q+1} + \boldsymbol{u}_q; \alpha_q)$$

Plug-and-Play Networks for Image Restoration



- **Results:**

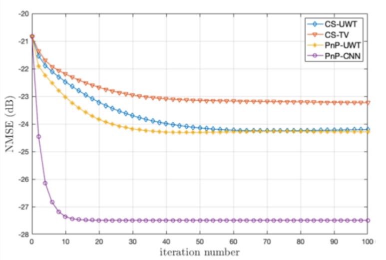


Figure 13: Normalized MSE versus iteration for the recovery of cardiac MRI images.

Image burst super-resolution[6]

Output of Dcraw | Camera's ISP (hi-quality jpeg) | Joint demosaick+single-image SR | Proposed method

[6] Bruno Lecouat, Jean Ponce, and Julien Mairal. "Lucas-kanade reloaded: End-to-end super-resolution from raw image bursts". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 2370–2379.

Image burst super-resolution

- **Image Formation Model:** k low-resolution frames y_k, one high-resolution latent image x.

$$\mathbf{y}_k = DBW_{\mathbf{p}_k}\mathbf{x} + \varepsilon_k \text{ for } k = 1, \ldots, K,$$

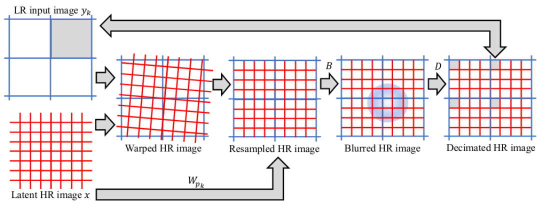


Image burst super-resolution

- **Objective Function:**

$$\frac{1}{2}\|\mathbf{y} - U_{\mathbf{p}}\,\mathbf{x}\|^2 + \lambda\phi_\theta(\mathbf{x}),$$

Which can be re-written (variable splitting) as:

$$E_\mu(\mathbf{x}, \mathbf{z}, \mathbf{p}) = \frac{1}{2}\|\mathbf{y} - U_{\mathbf{p}}\,\mathbf{z}\|^2 + \frac{\mu}{2}\|\mathbf{z} - \mathbf{x}\|^2 + \lambda\phi_\theta(\mathbf{x}),$$

- **Updating latent variables:**

$$\mathbf{z}^t \leftarrow \mathbf{z}^{t-1} - \eta_t\left[U_{\mathbf{p}^{t-1}}^\top(U_{\mathbf{p}^{t-1}}\mathbf{z}^{t-1} - \mathbf{y}) + \mu(\mathbf{z}^{t-1} - \mathbf{x}^{t-1})\right]$$

$$\mathbf{p}_k^t \leftarrow \mathbf{p}_k^{t-1} - \left(\mathbf{J}_k^{t\top}\mathbf{J}_k^t\right)^{-1}\mathbf{J}_k^{t\top}\mathbf{r}_k^t$$

$$\mathbf{x}^t \leftarrow \arg\min_{\mathbf{x}} \frac{\mu_{t-1}}{2}\|\mathbf{z}^t - \mathbf{x}\|^2 + \lambda\phi_\theta(\mathbf{x})$$

The last step is replaced by a CNN: $\mathbf{x}^t = f_\theta(\mathbf{z}^t)$

Conclusion

- The integration of deep learning facilitates inference in **complex environments**, where accurately capturing the underlying model may be be infeasible
- Model-based deep learning systems require notably **less data** in order to learn an accurate mapping
- M system combining DNNs with model-based inference often provides the ability to **analyze its resulting predictions**, yielding interpretability and confidence which are commonly challenging to obtain with conventional **black-box deep learning**.

Application to Direct Imaging

- **Local approach (patch-based)**

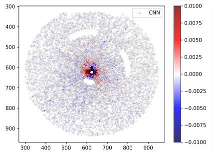
$$(\hat{x}, \hat{\alpha}, \hat{p}) = \underset{(x,\alpha,p)}{\arg\min} ||y - x - \alpha H(p)||^2 + \lambda \phi_\theta(x)$$

- **Update:**

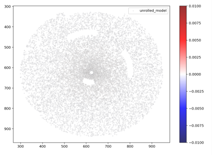
$$\begin{aligned} b^t &= y - \alpha^{t-1} H(p) \\ \beta^t &= S_\lambda[\beta^{t-1} + C^T(b^t - D\beta^{t-1})] \\ r^t &= D\beta^t \\ \alpha^t &= \alpha^{t-1} + \rho_t H(p)^T(r^t - H(p)) \end{aligned}$$

Application to Direct Imaging

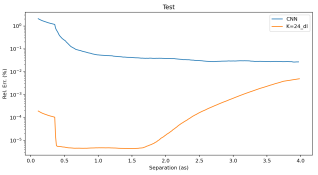
- **Motivation:** Failure case of CNN (self-subtraction)



(a) CNN



(b) Unrolled Model

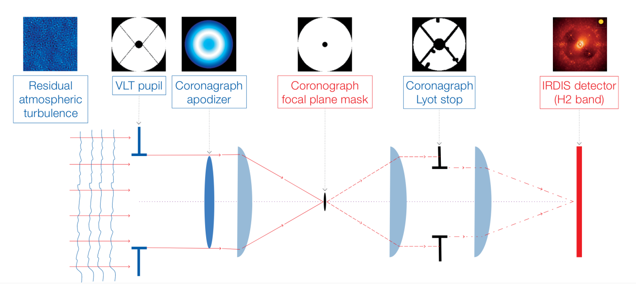


(c) Rel. Err. vs separation

Figure 14: Rel. Err. on **Blank cube**

Application to Direct Imaging

- **Global approach (optical model)**[7] Apodized Lyot Coronograph (APLC)



[7] Faustine Cantalloube et al. "Peering through SPHERE Images: A Glance at Contrast Limitations". In: *arXiv preprint arXiv:1907.03624* (2019).

Application to Direct Imaging

- **Upsides**:
  - Leverage the symmetries in the image
  - Model the high contrast inherent to direct imaging
  - Integration of metadata in the optical model (wind halo, waffle pattern)
  - Temporally and spatially varying off-axis PSF
- **Potential issues:**
  - Inversion problem close to phase retrieval, which is notoriously difficult
  - The true image formation model is the **long exposure** PSF, integrated over **multiple wavelengths**