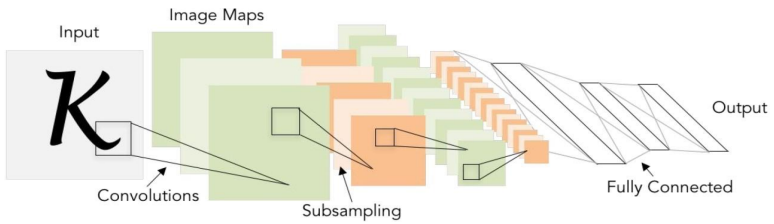
Optimization-Derived Learning with Essential Convergence Analysis of Training and Hyper-training

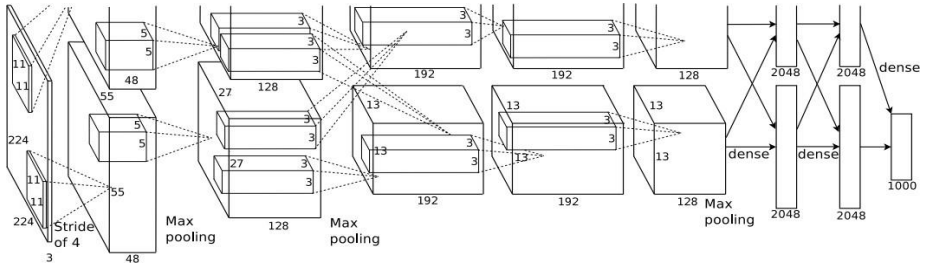
Risheng Liu, Xuan Liu, Shangzhi Zeng, Jin Zhang, Yixuan Zhang

ICML2022

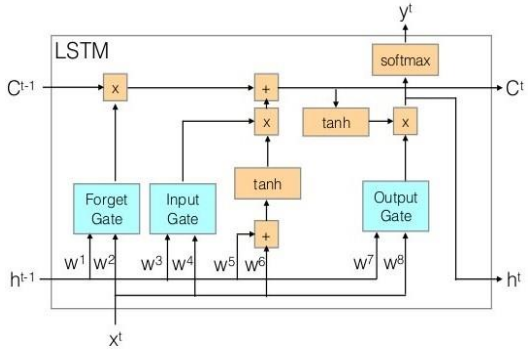
Existing Deep Learning Models



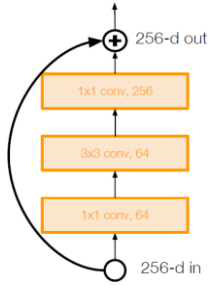
LeNet-5



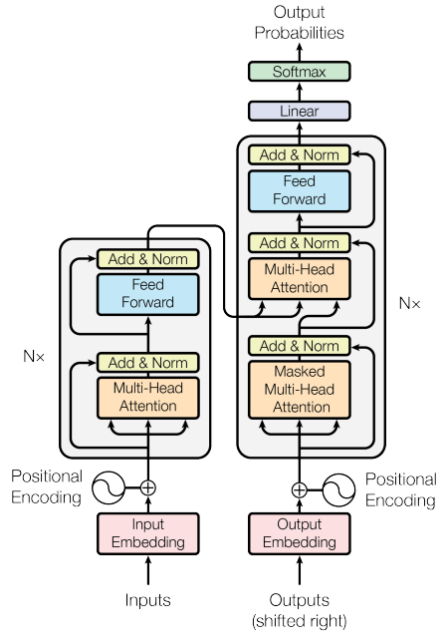
AlexNet



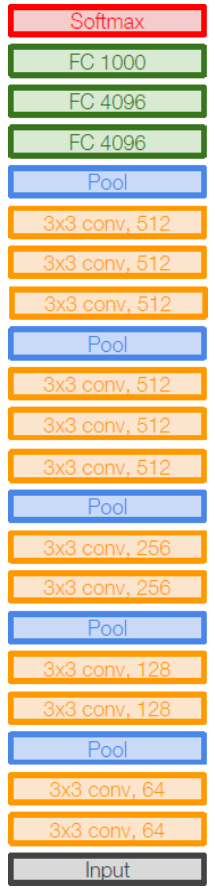
LSTM



ResNet



Transformer



VGG16

Can we design deep models in a more **principled** way and with **theoretical guarantees**?

Optimization-Drived Learning (ODL)

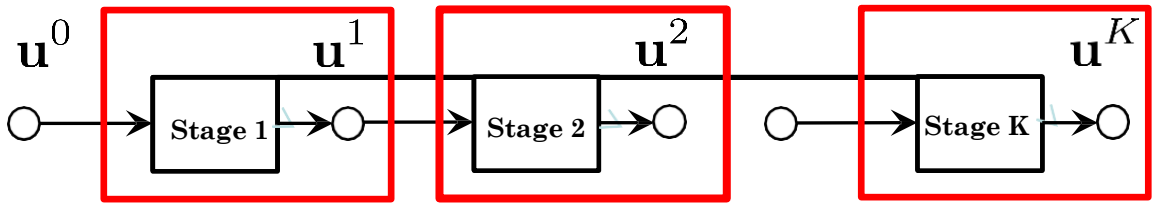
- **Both have "deep" structures**
  - Hierarchical propagation process
  - Many layers

- **But designed based on**
  - Numerical rules ($\mathcal{D}_{\text{num}}$)
  - Experiences and data ($\mathcal{D}_{\text{net}}$)

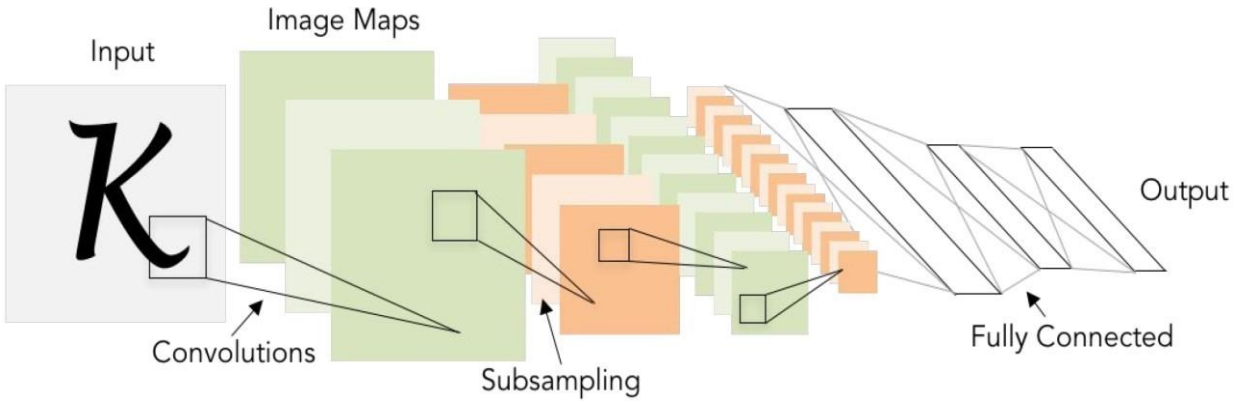
- **Optimization**
  - Principals and priors
  - White box
  - Theoretical investigation
  - **Less flexibility and capability**

translate knowledge into optimization

$$\min_{\mathbf{u}} f(\mathbf{u}) + g(\mathbf{u})$$



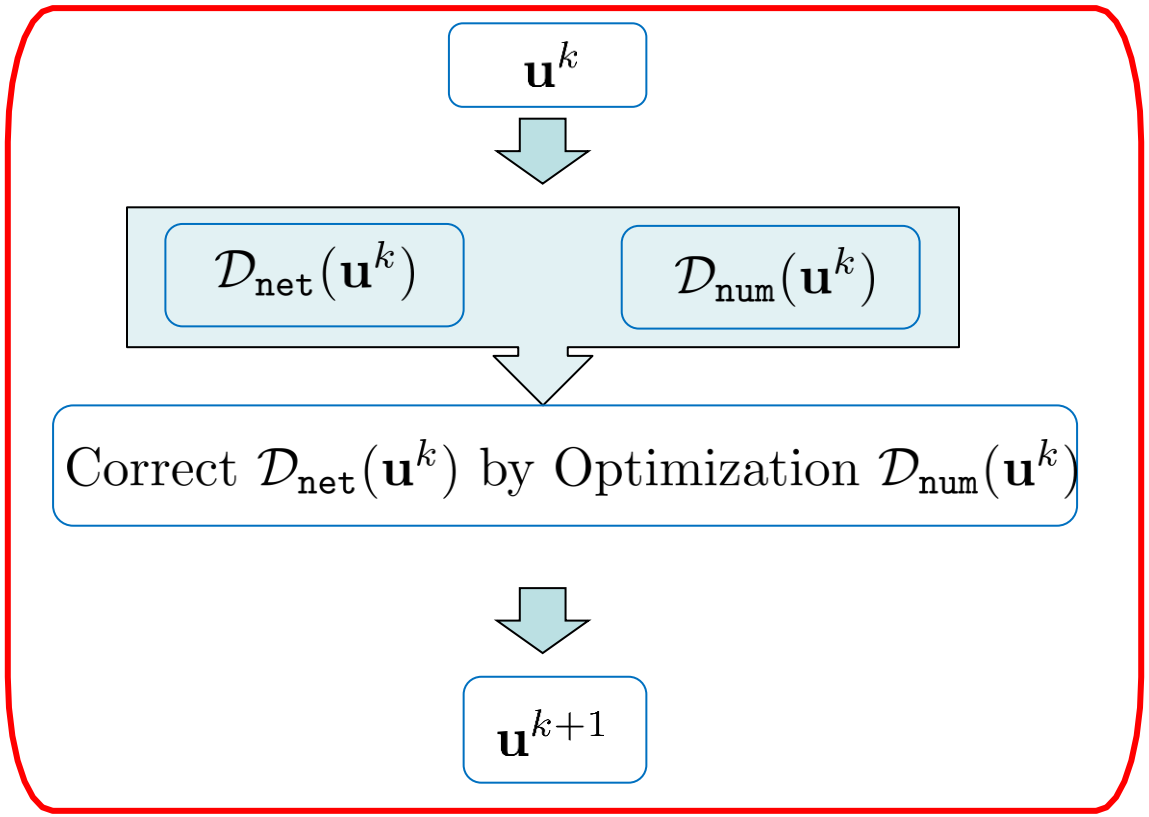
- **Neural networks**
  - Experiences/heuristic
  - Black box
  - Training data
  - **Weak interpretability and control**



Existing ODL Methods

Unrolling with Numerical Hyper-parameters (UNH)

$$\min_{\mathbf{u}} f(\mathbf{u}) + g(\mathbf{u})$$

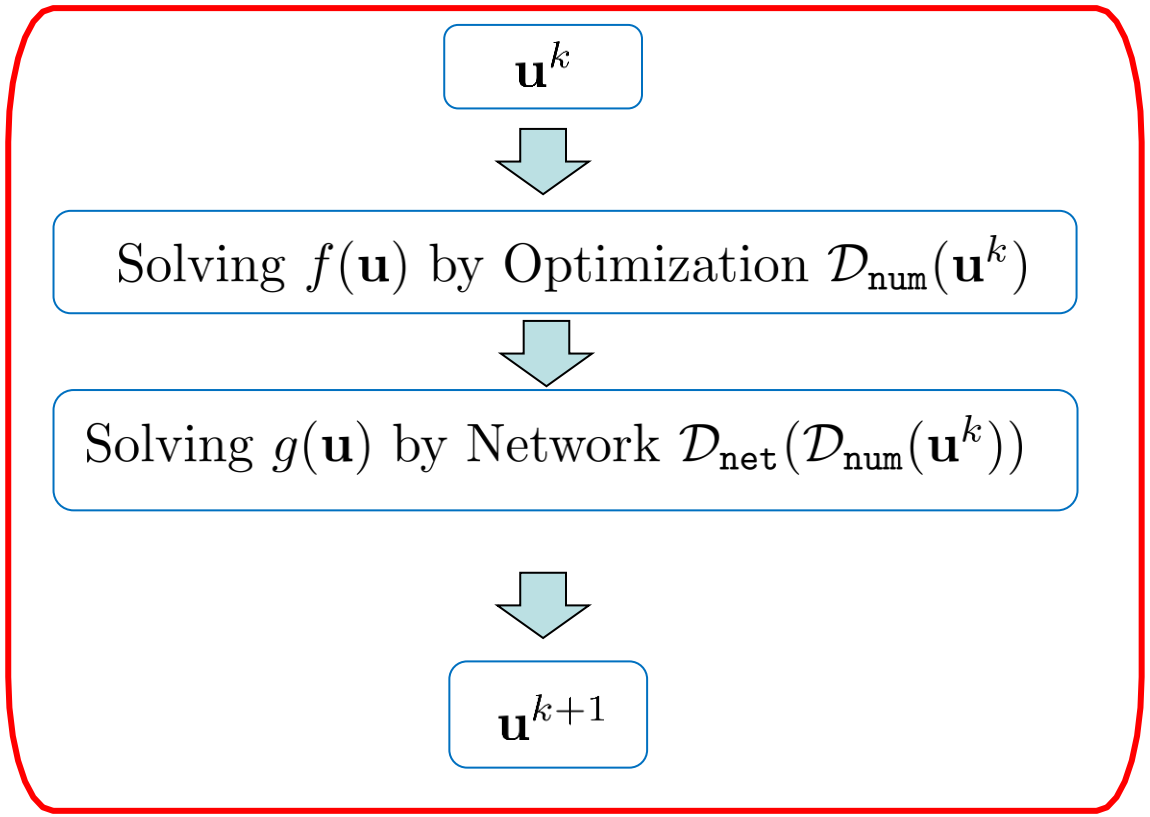


Minimizing Training Objective $f + g$

Ignore Hyper-Training Objective ℓ

Embedded with Network Architectures (ENA)

$$\min_{\mathbf{u}} f(\mathbf{u}) + g(\mathbf{u})$$



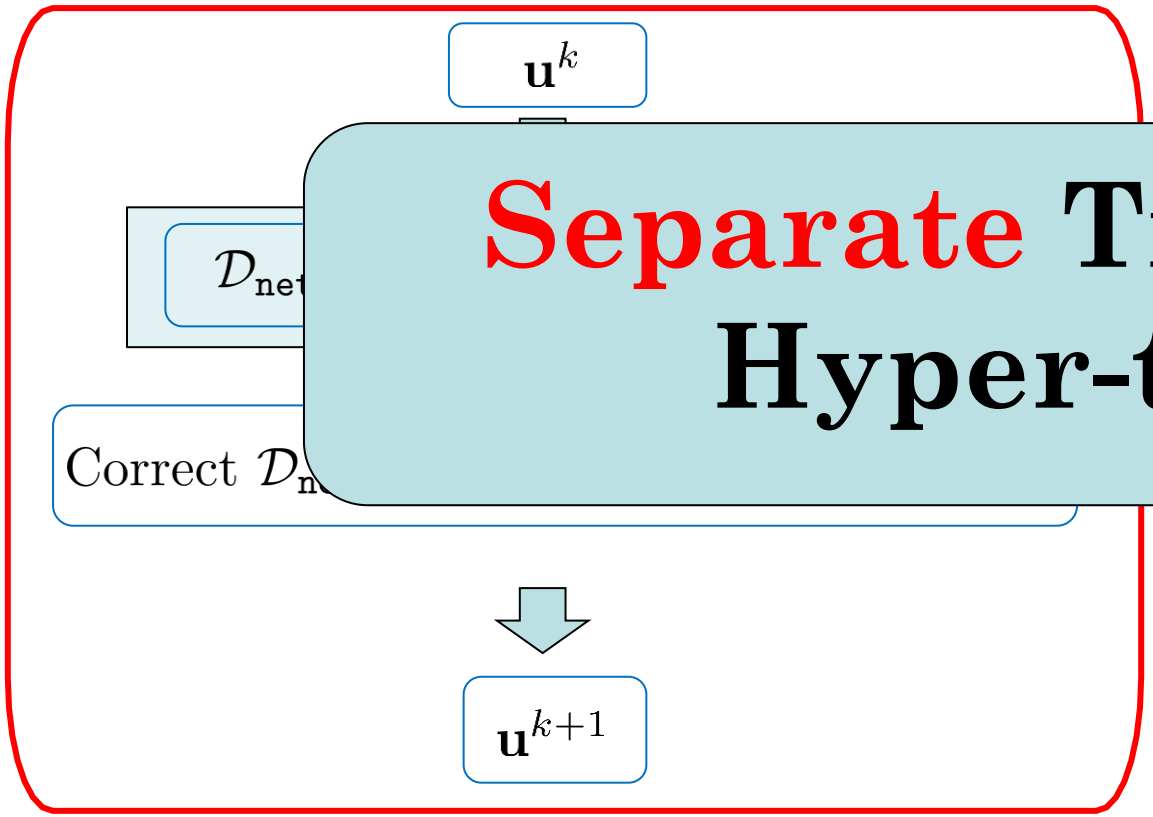
Minimizing Hyper-Training Objective ℓ

Ignore Training Objective $f + g$

Existing ODL Methods

Unrolling with Numerical Hyper-parameters (UNH)

$$\min_{\mathbf{u}} f(\mathbf{u}) + g(\mathbf{u})$$

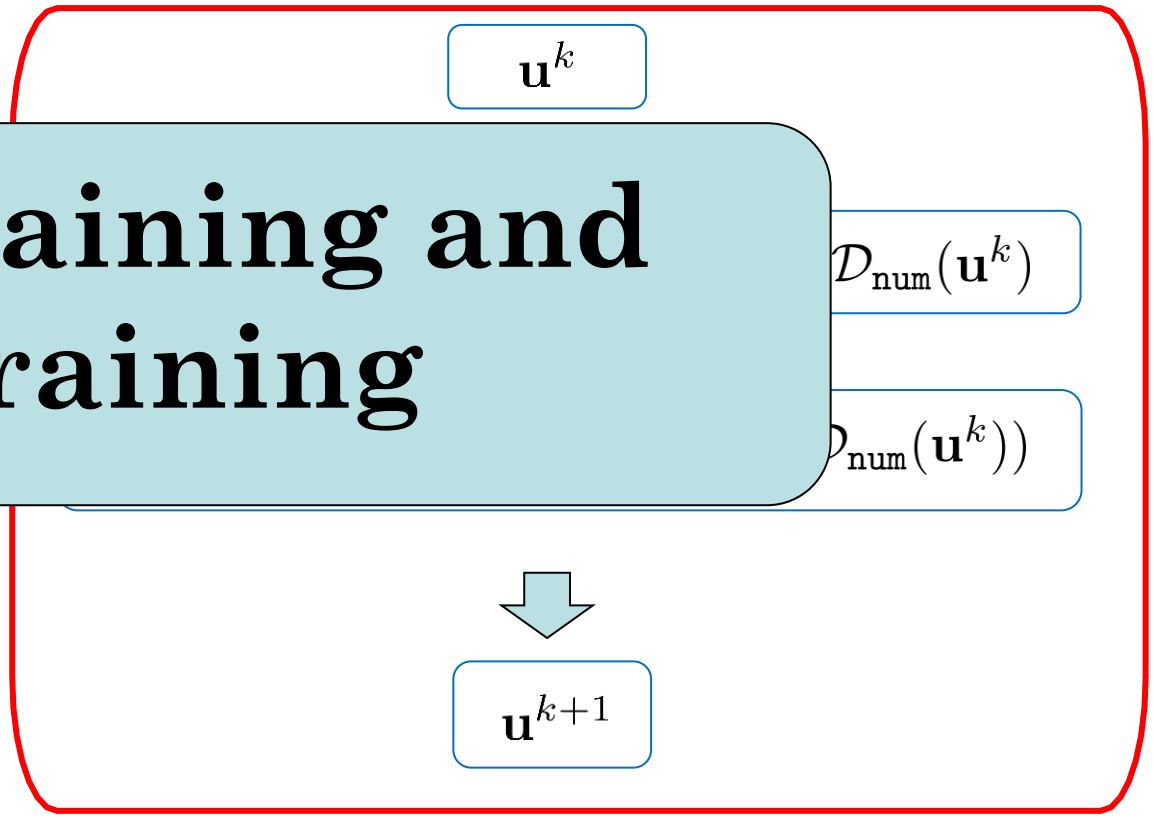


Minimizing Training Objective $f + g$

Ignore Hyper-Training Objective ℓ

Embedded with Network Architectures (ENA)

$$\min_{\mathbf{u}} f(\mathbf{u}) + g(\mathbf{u})$$



Minimizing Hyper-Training Objective ℓ

Ignore Training Objective $f + g$

Separate Training and Hyper-training

Bilevel Meta Optimization (BMO)

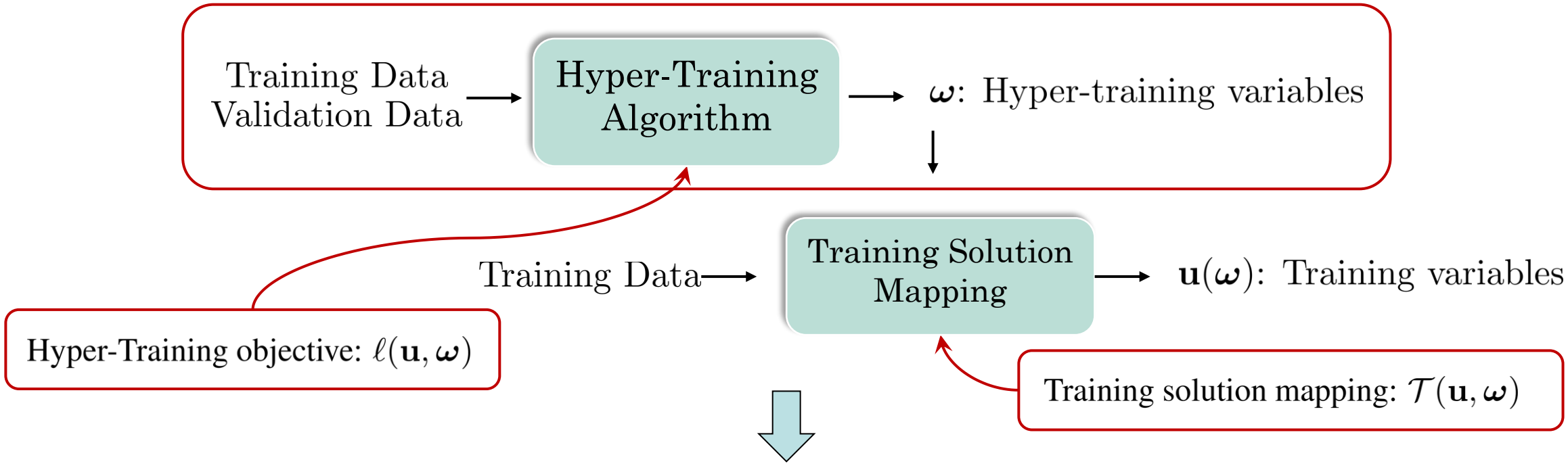
We consider the following **BMO** formulation:

- A **hierarchical optimization** problem, where an optimization problem contains another problem **as the constraint**
- More **general** than a traditional bilevel optimization problem, since the lower level for training is the solution mapping of a broader **fixed-point** iteration

$$\min_{\mathbf{u}\in U,\boldsymbol{\omega}\in\Omega} \ell(\mathbf{u};\boldsymbol{\omega}),\ \text{s.t.}\ \mathbf{u}\in \mathrm{Fix}(\mathcal{T}(\cdot,\boldsymbol{\omega}))$$

- $\ell : U \times \Omega \to \mathbb{R}$ is called the Hyper-Training objective.
- $\mathcal{T} : U \times \Omega \to U$ is called the Training solution mapping.
  - $\mathcal{T}$ can be both $\mathcal{D}_{\texttt{num}}$ and $\mathcal{D}_{\texttt{net}}$.

BMO for Training and Hyper-training



$$\min_{\mathbf{u}\in U, \boldsymbol{\omega}\in\Omega} \ell(\mathbf{u}; \boldsymbol{\omega}), \text{ s.t. } \mathbf{u} \in \text{Fix}(\mathcal{T}(\cdot, \boldsymbol{\omega}))$$

Joint Convergence

Algorithm and Convergence Analysis

- Algorithm

Algorithm 1 The Solution Strategy of BMO

Require: Step sizes $\{s_k\}$, γ and parameter μ

Initialize $\boldsymbol{\omega}^0$.
for $t = 1 \to T$ **do**
  Initialize $\mathbf{u}^0$.
  for $k = 1 \to K$ **do**
    $\mathbf{v}_l^k = \mathcal{T}\left(\mathbf{u}^{k-1}, \boldsymbol{\omega}^{t-1}\right)$.
    $\mathbf{v}_u^k = \mathbf{u}^{k-1} - s_k \mathbf{H}_{\boldsymbol{\omega}}^{-1} \frac{\partial}{\partial \mathbf{u}} \ell(\mathbf{u}^{k-1}, \boldsymbol{\omega}^{t-1})$.[1].
    $\mathbf{u}^k = \texttt{Proj}_{U,\mathbf{H}_{\boldsymbol{\omega}}}\left(\mu \mathbf{v}_u^k + (1-\mu)\mathbf{v}_l^k\right)$.
  end for
  $\boldsymbol{\omega}^t = \boldsymbol{\omega}^{t-1} - \gamma \frac{\partial}{\partial \boldsymbol{\omega}} \ell(\mathbf{u}^K, \boldsymbol{\omega}^{t-1})$.
end for

[1] $\mathbf{H}$ is a correction matrix related to $\mathcal{T}$, please see the paper for details

- Convergence Analysis
  - Approximation Quality

Theorem 1 *Under appropriate conditions, we have*

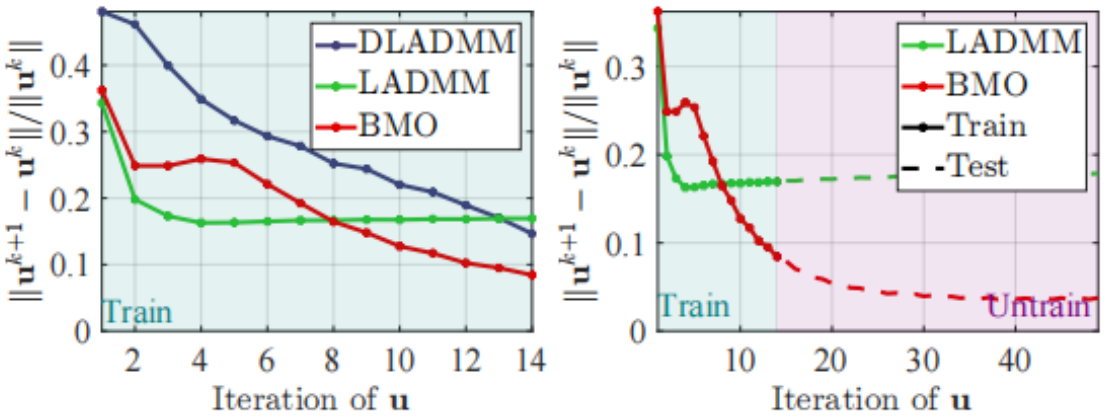
(1) any limit point $(\bar{\mathbf{u}}, \bar{\boldsymbol{\omega}})$ *of the sequence* $\{(\mathbf{u}^K(\boldsymbol{\omega}^K), \boldsymbol{\omega}^K)\}$ *is a solution to the problem, i.e.,* $\bar{\boldsymbol{\omega}} \in \operatorname{argmin}_{\boldsymbol{\omega}\in\Omega}\varphi(\boldsymbol{\omega})$ *and* $\bar{\mathbf{u}} = \mathcal{T}(\bar{\mathbf{u}}, \bar{\boldsymbol{\omega}})$.

(2) $\inf_{\boldsymbol{\omega}\in\Omega} \varphi_K(\boldsymbol{\omega}) \to \inf_{\boldsymbol{\omega}\in\Omega} \varphi(\boldsymbol{\omega})$ *as* $K \to \infty$.

  - Stationary Analysis

Theorem 2 *Let* $\boldsymbol{\omega}^K$ *be an* ε_K*-stationary point of* $\varphi_K(\boldsymbol{\omega})$*, i.e.,* $\|\nabla\varphi_K(\boldsymbol{\omega}^K)\| = \varepsilon_K$*. Then under appropriate conditions, if* $\varepsilon_K \to 0$*, we have that any limit point* $\bar{\boldsymbol{\omega}}$ *of the sequence* $\{\boldsymbol{\omega}^K\}$ *is a stationary point of* φ*, i.e.,* $\nabla\varphi(\bar{\boldsymbol{\omega}}) = 0$.

Experiments



Toy Example:

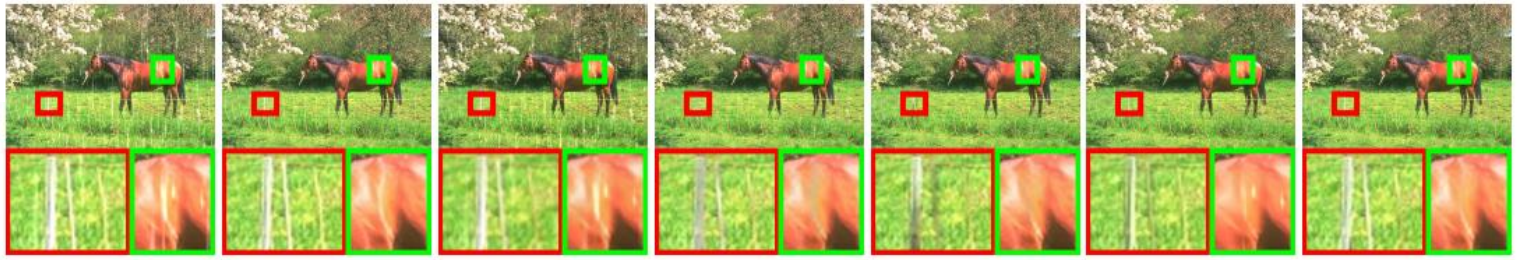
- Sparse Coding
  - **Joint** convergence
  - Convergence on **untrained** layers

Noise level	$\sigma = 1\%$			$\sigma = 3\%$		
Image	Butterfly	Leaves	Starfish	Butterfly	Leaves	Starfish
EPLL	20.55	19.22	24.84	18.64	17.54	22.47
FDN	27.40	26.51	27.48	24.27	23.53	24.71
IRCNN	32.74	33.22	33.53	28.53	28.45	28.42
IRCNN+	32.48	33.59	32.18	28.40	28.14	28.20
DPIR	**34.18**	35.12	33.91	29.45	30.27	29.46
BMO	33.67	**35.39**	**33.98**	**29.46**	**30.69**	**29.64**

More results are in the paper.

Image processing:

- Image Deconvolution
- Rain Streak Removal



Take Home Messages

- We establish a new formulation as a general form of various ODL methods.
- BMO optimizes training and hyper-training variables simultaneously to obtain the true solution.
- BMO provides strict essential convergence analysis of both training and hyper-training variables.