Advancing Trustworthy & Efficient AI for Science, Engineering, and Medicine

Ju Sun

Jun 04, 2025

Systems, Information, Learning and Optimization (SILO) Seminar, University of Wisconsin, Madison

University of Minnesota
Driven to Discover℠

Success of deep learning (DL) not news anymore

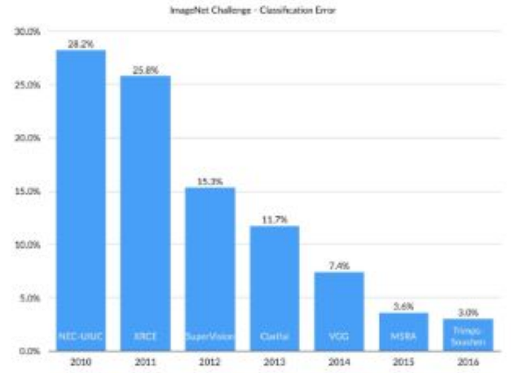


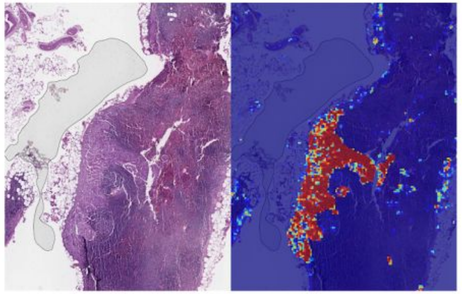
image classification

Go game (2017)

Commercial breakthroughs ...

self-driving vehicles credit: wired.com

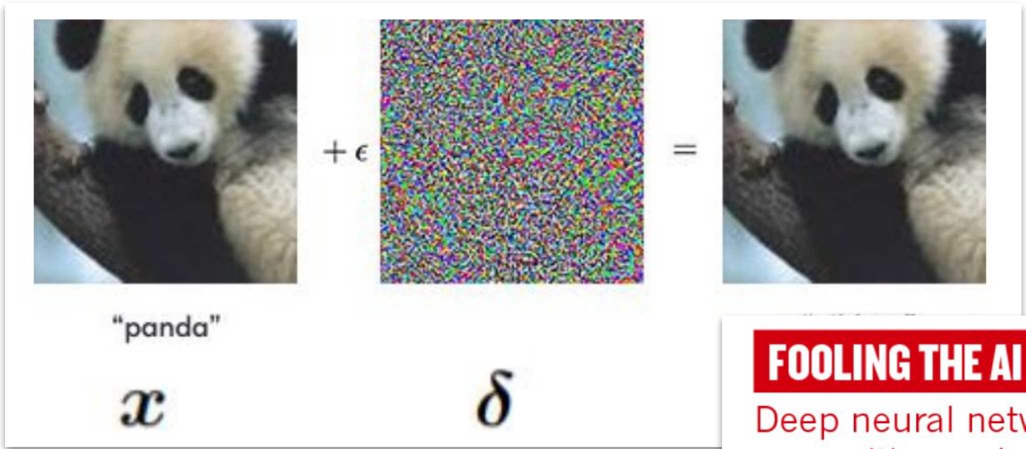
smart-home devices credit: Amazon

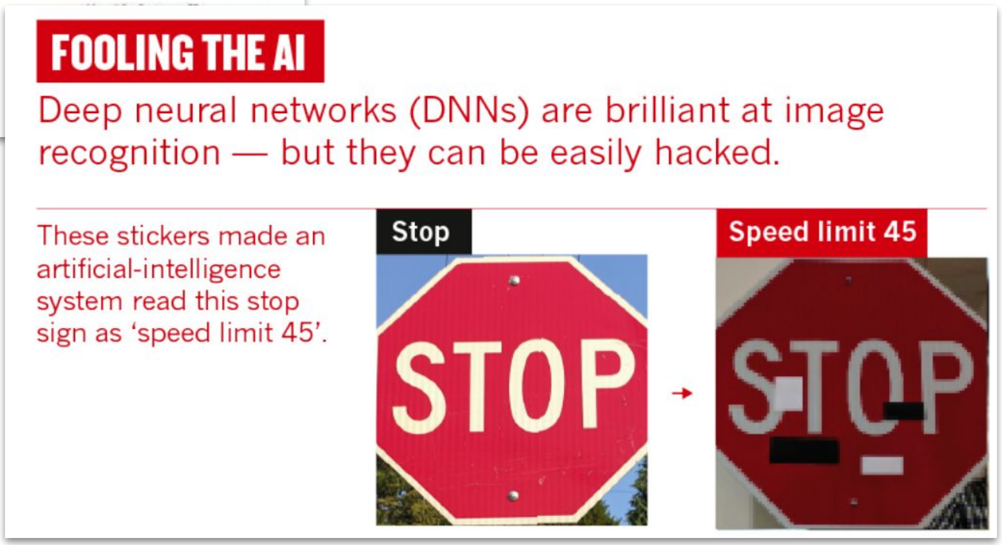


healthcare credit: Google AI

robotics credit: Cornell U.

Robustness issues of DL not news anymore



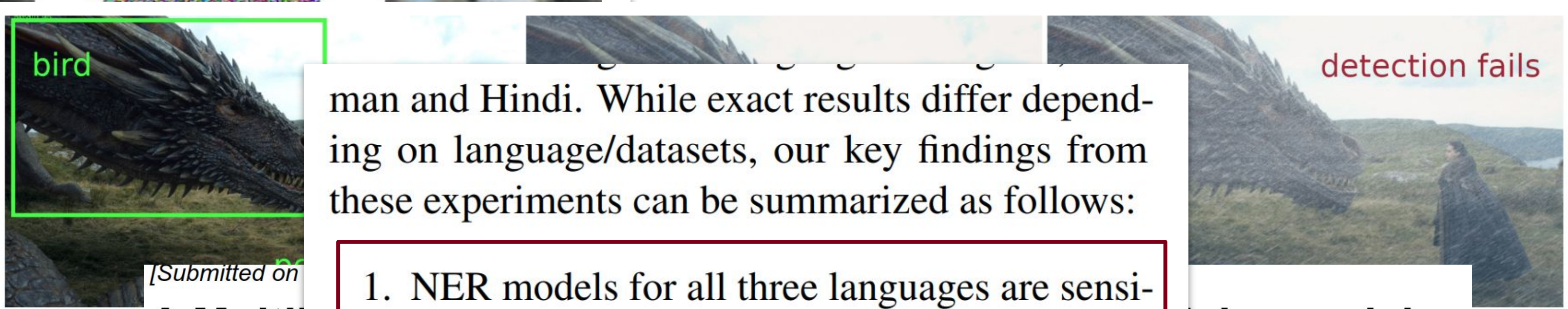

FOOLING THE AI

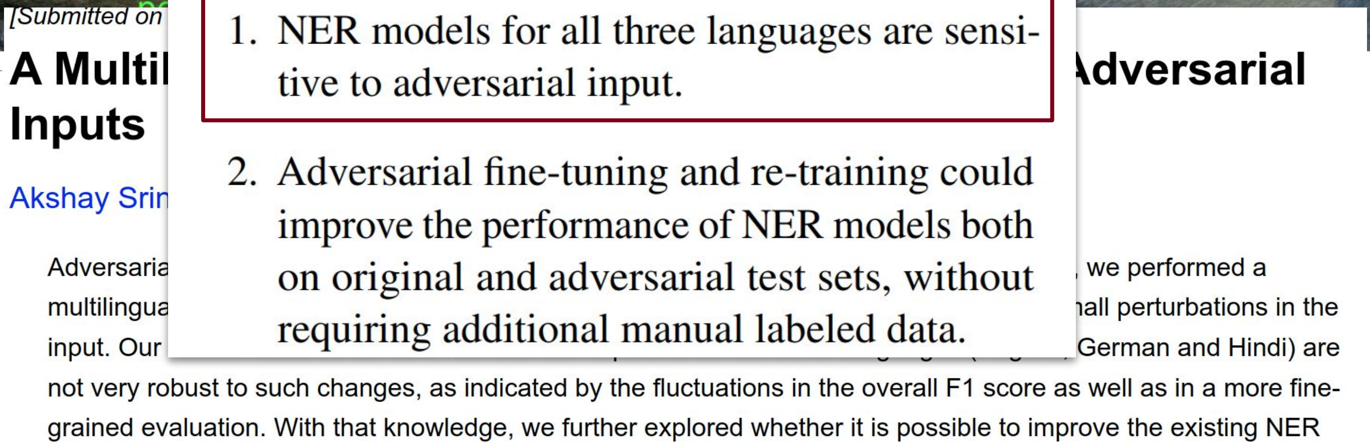
Deep neural networks (DNNs) are brilliant at image recognition — but they can be easily hacked.

These stickers made an artificial-intelligence system read this stop sign as 'speed limit 45'.

Stop
Speed limit 45
STOP
STOP

Robustness issues across domains/tasks



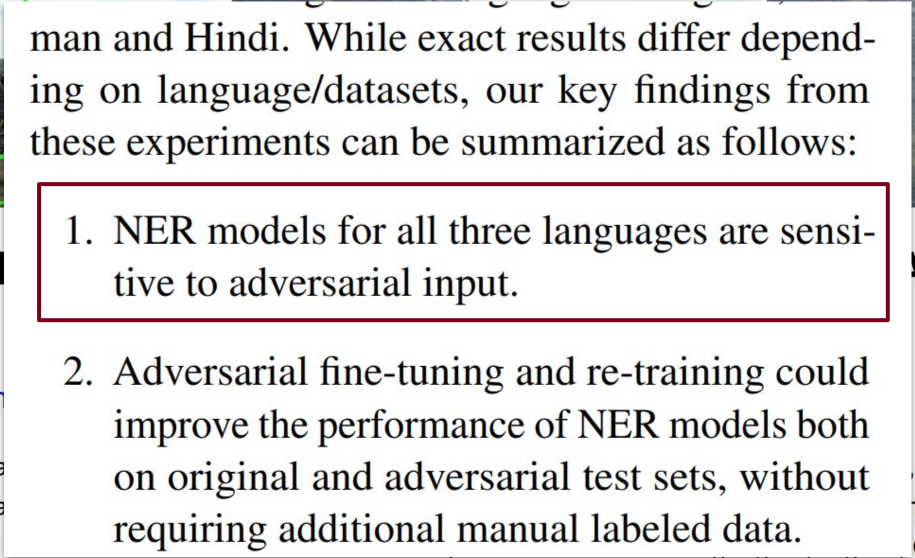


[Submitted on

A Multil … Adversarial Inputs

Akshay Srin

Adversaria … , we performed a multilingua … all perturbations in the input. Our … German and Hindi) are not very robust to such changes, as indicated by the fluctuations in the overall F1 score as well as in a more fine-grained evaluation. With that knowledge, we further explored whether it is possible to improve the existing NER

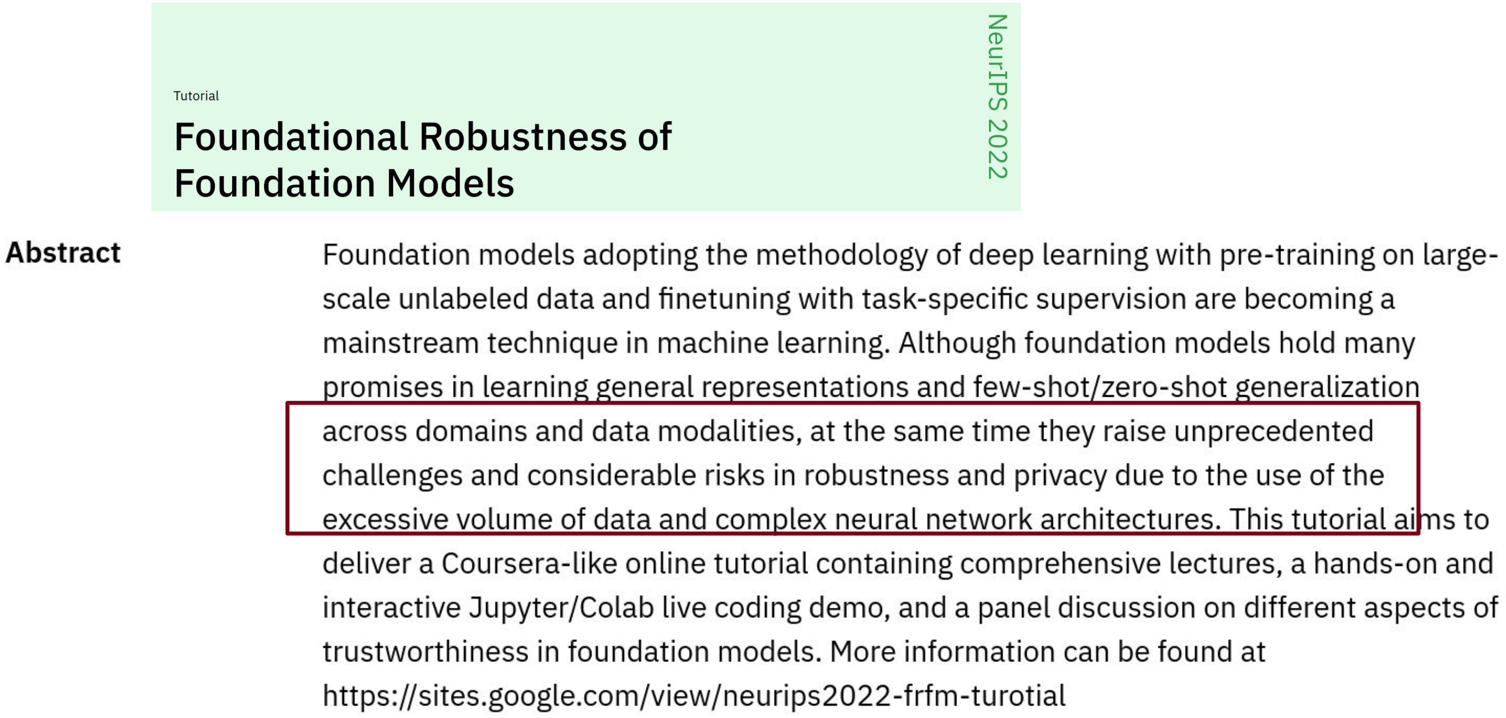


man and Hindi. While exact results differ depending on language/datasets, our key findings from these experiments can be summarized as follows:

1. NER models for all three languages are sensitive to adversarial input.
2. Adversarial fine-tuning and re-training could improve the performance of NER models both on original and adversarial test sets, without requiring additional manual labeled data.

Name entry Recognition

Robustness issues across models



Tutorial

Foundational Robustness of Foundation Models

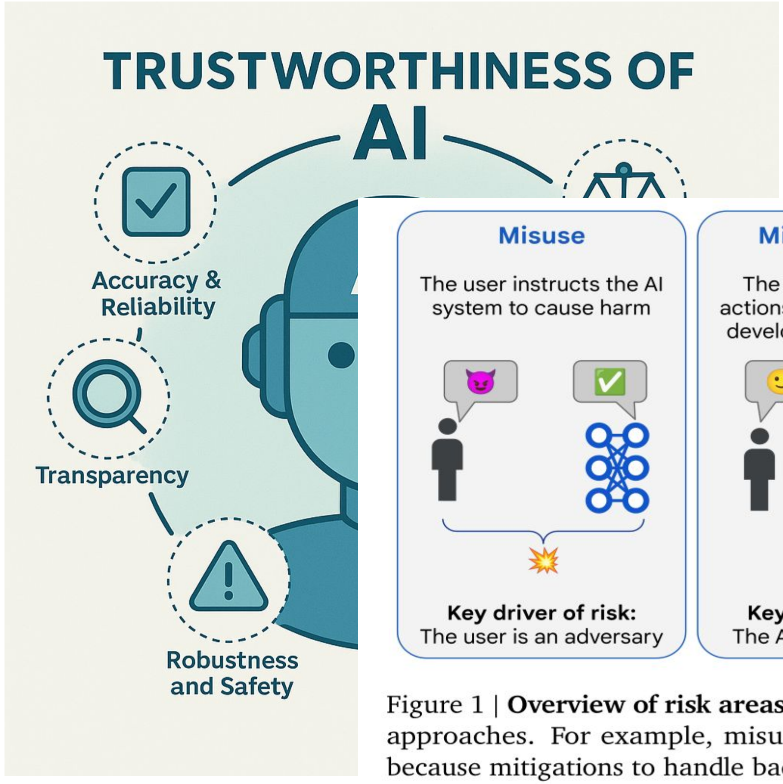
NeurIPS 2022

Abstract

Foundation models adopting the methodology of deep learning with pre-training on large-scale unlabeled data and finetuning with task-specific supervision are becoming a mainstream technique in machine learning. Although foundation models hold many promises in learning general representations and few-shot/zero-shot generalization across domains and data modalities, at the same time they raise unprecedented challenges and considerable risks in robustness and privacy due to the use of the excessive volume of data and complex neural network architectures. This tutorial aims to deliver a Coursera-like online tutorial containing comprehensive lectures, a hands-on and interactive Jupyter/Colab live coding demo, and a panel discussion on different aspects of trustworthiness in foundation models. More information can be found at https://sites.google.com/view/neurips2022-frfm-turotial

https://research.ibm.com/publications/foundational-robustness-of-foundation-models

Trustworthiness issues not news anymore



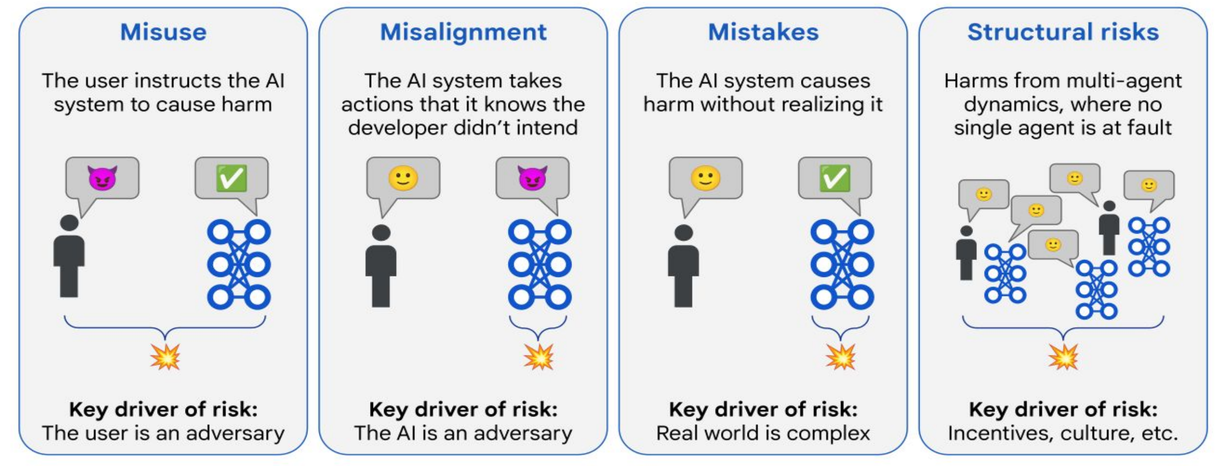
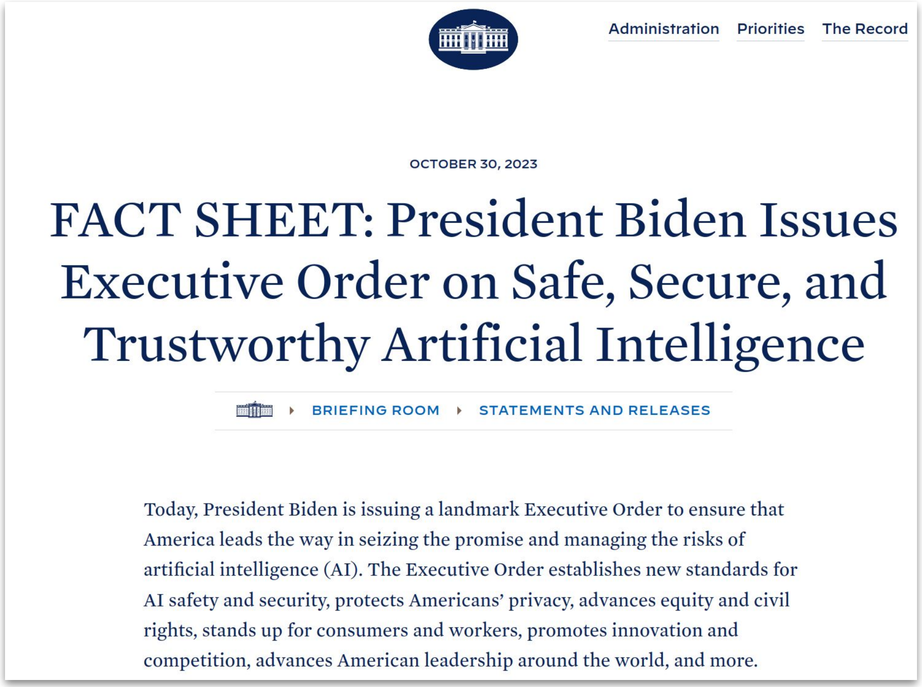

Misuse
The user instructs the AI system to cause harm
Key driver of risk: The user is an adversary
Misalignment
The AI system takes actions that it knows the developer didn't intend
Key driver of risk: The AI is an adversary
Mistakes
The AI system causes harm without realizing it
Key driver of risk: Real world is complex
Structural risks
Harms from multi-agent dynamics, where no single agent is at fault
Key driver of risk: Incentives, culture, etc.

Figure 1 | **Overview of risk areas.** We group risks based on factors that drive differences in mitigation approaches. For example, misuse and misalignment differ based on which actor has bad intent, because mitigations to handle bad human actors vary significantly from mitigations to handle bad AI actors.

source : https://deepmind.google/discover/blog/taking-a-responsible-path-t

International concerns and priorities



OCTOBER 30, 2023

FACT SHEET: President Biden Issues Executive Order on Safe, Secure, and Trustworthy Artificial Intelligence

BRIEFING ROOM ▸ STATEMENTS AND RELEASES

Today, President Biden is issuing a landmark Executive Order to ensure that America leads the way in seizing the promise and managing the risks of artificial intelligence (AI). The Executive Order establishes new standards for AI safety and security, protects Americans' privacy, advances equity and civil rights, stands up for consumers and workers, promotes innovation and competition, advances American leadership around the world, and more.

- New Standards for AI Safety and Security
- Protecting Americans' Privacy
- Advancing Equity and Civil Rights
- Standing Up for Consumers, Patients, and Students
- Supporting Workers
- Promoting Innovation and Competition
- Advancing American Leadership Abroad
- Ensuring Responsible and Effective Government Use of AI

https://www.whitehouse.gov/briefing-room/statements-releases/2023/10/30/fact-sheet-president-biden-issues-executive-order-on-safe-secure-and-trustworthy-artificial-intelligence/

Three pillars of DL

GPU/TPU/FPGA/...

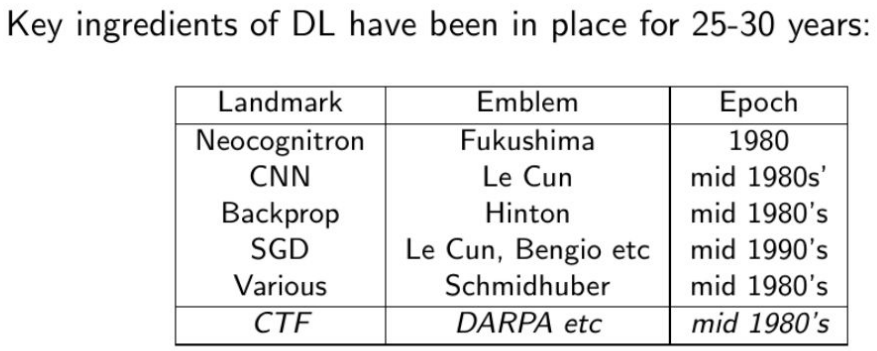
DATA

+

Specialized hardware

+

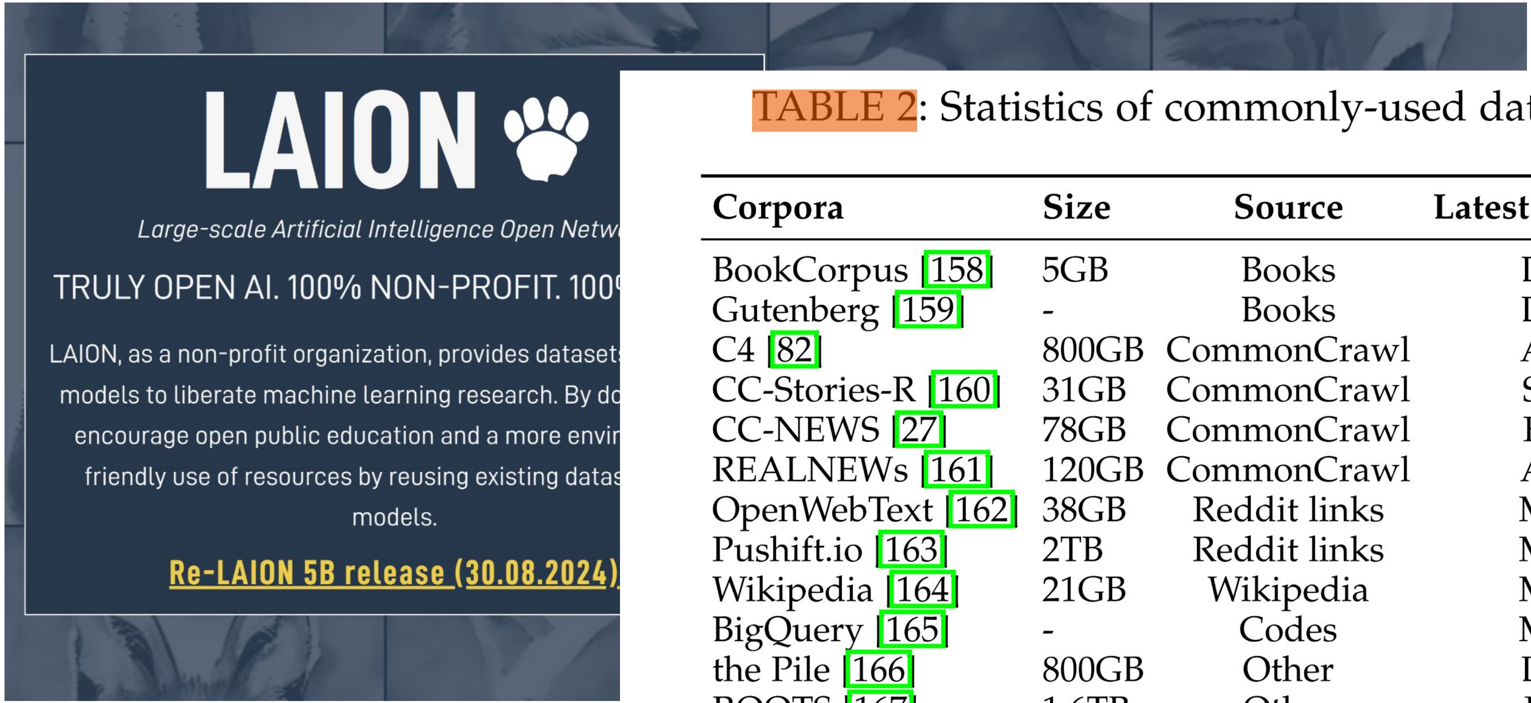
Specialized software



Key ingredients of DL have been in place for 25-30 years:

Landmark	Emblem	Epoch
Neocognitron	Fukushima	1980
CNN	Le Cun	mid 1980s'
Backprop	Hinton	mid 1980's
SGD	Le Cun, Bengio etc	mid 1990's
Various	Schmidhuber	mid 1980's
CTF	*DARPA etc*	*mid 1980's*

CV/NLP domains are lucky



LAION

Large-scale Artificial Intelligence Open Netw

TRULY OPEN AI. 100% NON-PROFIT. 100

LAION, as a non-profit organization, provides dataset
models to liberate machine learning research. By do
encourage open public education and a more envir
friendly use of resources by reusing existing datas
models.

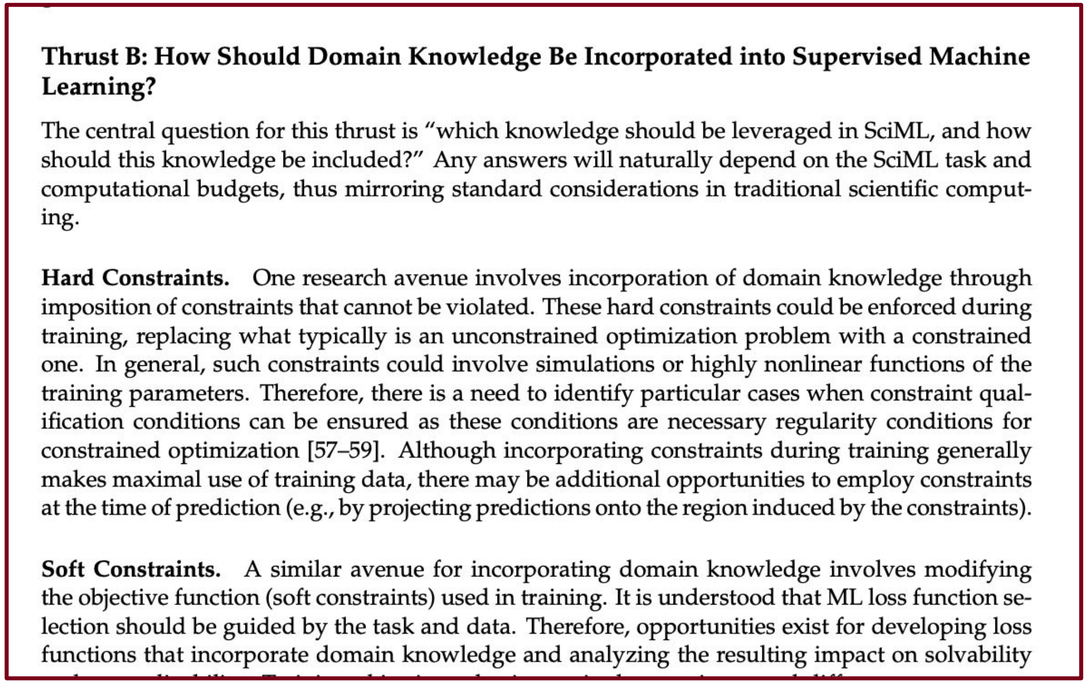
Re-LAION 5B release (30.08.2024)

TABLE 2: Statistics of commonly-used data sources.

Corpora	Size	Source	Latest Update Time
BookCorpus [158]	5GB	Books	Dec-2015
Gutenberg [159]	-	Books	Dec-2021
C4 [82]	800GB	CommonCrawl	Apr-2019
CC-Stories-R [160]	31GB	CommonCrawl	Sep-2019
CC-NEWS [27]	78GB	CommonCrawl	Feb-2019
REALNEWs [161]	120GB	CommonCrawl	Apr-2019
OpenWebText [162]	38GB	Reddit links	Mar-2023
Pushift.io [163]	2TB	Reddit links	Mar-2023
Wikipedia [164]	21GB	Wikipedia	Mar-2023
BigQuery [165]	-	Codes	Mar-2023
the Pile [166]	800GB	Other	Dec-2020
ROOTS [167]	1.6TB	Other	Jun-2022

source: https://arxiv.org/abs/2303.18223

Not all fields are as lucky

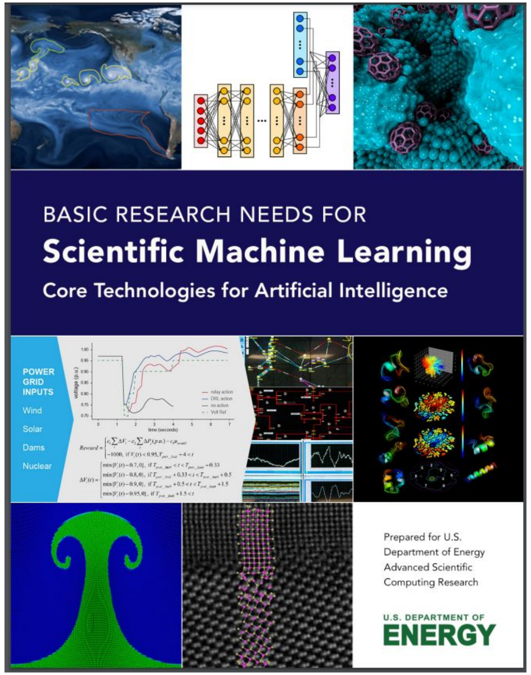


Thrust B: How Should Domain Knowledge Be Incorporated into Supervised Machine Learning?

The central question for this thrust is "which knowledge should be leveraged in SciML, and how should this knowledge be included?" Any answers will naturally depend on the SciML task and computational budgets, thus mirroring standard considerations in traditional scientific computing.

Hard Constraints. One research avenue involves incorporation of domain knowledge through imposition of constraints that cannot be violated. These hard constraints could be enforced during training, replacing what typically is an unconstrained optimization problem with a constrained one. In general, such constraints could involve simulations or highly nonlinear functions of the training parameters. Therefore, there is a need to identify particular cases when constraint qualification conditions can be ensured as these conditions are necessary regularity conditions for constrained optimization [57–59]. Although incorporating constraints during training generally makes maximal use of training data, there may be additional opportunities to employ constraints at the time of prediction (e.g., by projecting predictions onto the region induced by the constraints).

Soft Constraints. A similar avenue for incorporating domain knowledge involves modifying the objective function (soft constraints) used in training. It is understood that ML loss function selection should be guided by the task and data. Therefore, opportunities exist for developing loss functions that incorporate domain knowledge and analyzing the resulting impact on solvability

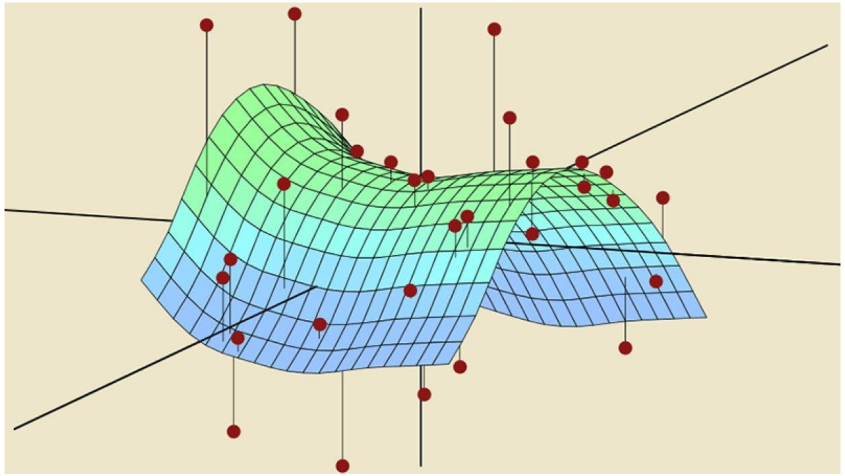

BASIC RESEARCH NEEDS FOR
Scientific Machine Learning
Core Technologies for Artificial Intelligence
POWER GRID INPUTS
Wind
Solar
Dams
Nuclear
Prepared for U.S. Department of Energy Advanced Scientific Computing Research
U.S. DEPARTMENT OF ENERGY

Ref https://www.osti.gov/servlets/purl/1478744

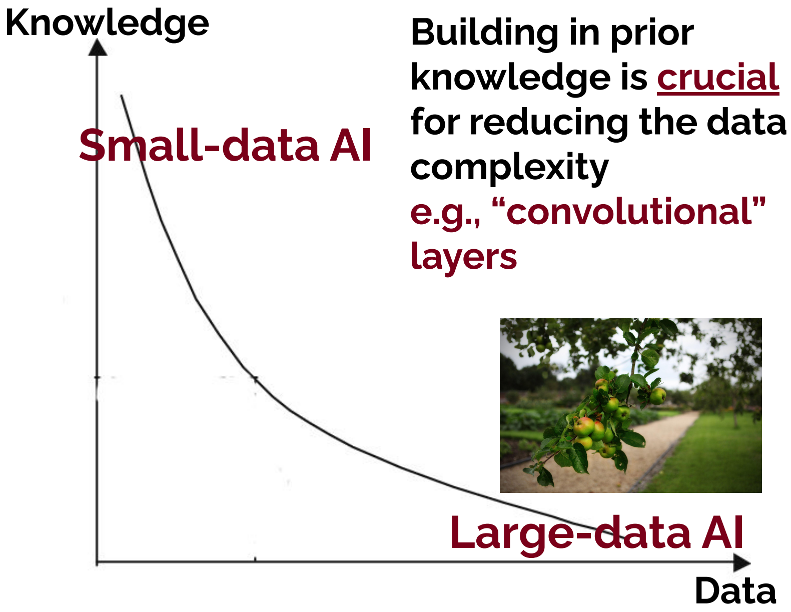
Domain-Aware Scientific Machine Learning

There's no free lunch!

Supervised learning as data fitting



Typically, #data points we need grow **exponentially** with respect to dimension (i.e., **curse of dimensionality**)



Building in prior knowledge is <u>crucial</u> for reducing the data complexity
e.g., "convolutional" layers

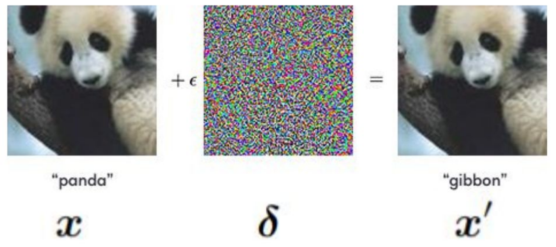
Today's talk: **toward trustworthy and efficient AI**

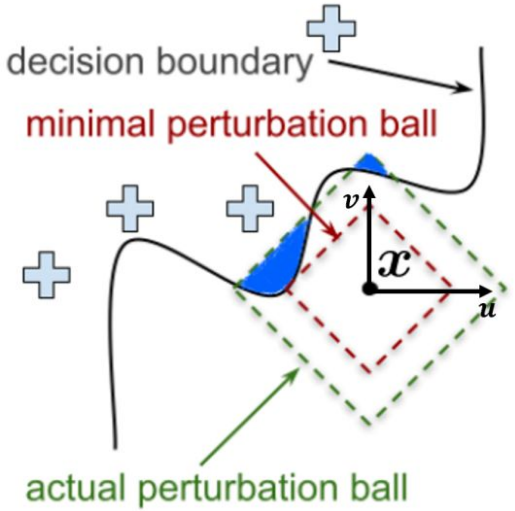
- **Robustness evaluation and selective classification**
- **Inverse problems with pretrained diffusion models**
- **AI for healthcare: handling data imbalance**

Today's talk: **toward trustworthy and efficient AI**

- **Robustness evaluation and selective classification**
- **Inverse problems with pretrained diffusion models**
- **AI for healthcare: handling data imbalance**

Robustness evaluation (RE)





Maximize loss/error function

$$\boxed{\max_{\boldsymbol{x}'} \ell(\boldsymbol{y}, f_{\boldsymbol{\theta}}(\boldsymbol{x}'))}$$

$$\text{s.t.} \ \boxed{d(\boldsymbol{x}, \boldsymbol{x}') \leq \varepsilon}, \quad \boxed{\boldsymbol{x}' \in [0,1]^n}$$

Allowable perturbation Valid image

Find robustness radius

$$\boxed{\min_{\boldsymbol{x}'} \ d(\boldsymbol{x}, \boldsymbol{x}')}$$

$$\text{s.t.} \ \boxed{\max_{i \neq y} f_{\boldsymbol{\theta}}^{i}(\boldsymbol{x}') \geq f_{\boldsymbol{\theta}}^{y}(\boldsymbol{x}')}, \ \boxed{\boldsymbol{x}' \in [0,1]^n}$$

On the decision boundary Valid image

Report robust accuracy over an evaluation set

Constrained optimization problems

$$\boxed{\max_{\boldsymbol{x}'} \ell\left(\boldsymbol{y}, f_{\boldsymbol{\theta}}(\boldsymbol{x}')\right)}$$
$$\text{s.t.} \boxed{d\left(\boldsymbol{x}, \boldsymbol{x}'\right) \leq \varepsilon}, \quad \boldsymbol{x}' \in [0,1]^n$$

$$\boxed{\min_{\boldsymbol{x}'} \ d\left(\boldsymbol{x}, \boldsymbol{x}'\right)}$$
$$\text{s.t.} \boxed{\max_{i \neq y} f_{\boldsymbol{\theta}}^{i}(\boldsymbol{x}') \geq f_{\boldsymbol{\theta}}^{y}(\boldsymbol{x}')}, \ \boldsymbol{x}' \in [0,1]^n$$

Both objective and constraint functions are **nonconvex** in general, e.g., when containing DL models

Projected gradient descent (PGD) for RE

$$\max_{\boldsymbol{x}'} \ell\left(\boldsymbol{y}, f_{\boldsymbol{\theta}}(\boldsymbol{x}')\right)$$

$$\text{s.t. } d\left(\boldsymbol{x}, \boldsymbol{x}'\right) \le \varepsilon\,, \quad \boldsymbol{x}' \in [0,1]^n$$

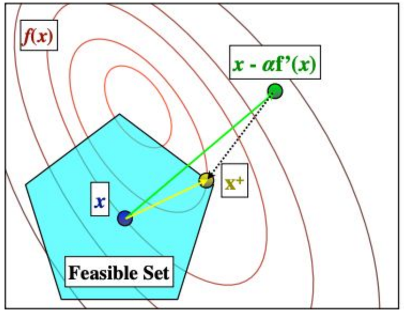
$$\min_{\mathbf{x}\in\mathcal{Q}} f(\mathbf{x})$$

Step size

$$\mathbf{x}_{k+1} = P_{\mathcal{Q}}\Big(\mathbf{x}_k - \alpha_k \nabla f(\mathbf{x}_k)\Big)$$

$$P_{\mathcal{Q}}(\mathbf{x}_0) = \arg\min_{\mathbf{x}\in\mathcal{Q}} \frac{1}{2}\|\mathbf{x}-\mathbf{x}_0\|_2^2$$

Projection operator

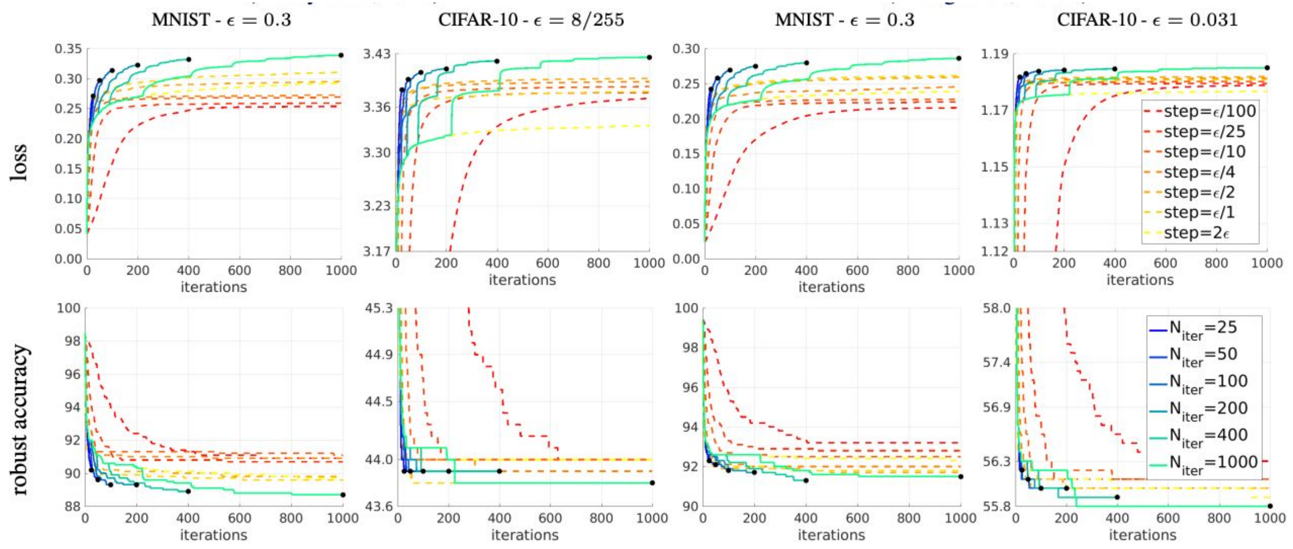


Key hyperparameters:
(1) step size
(2) iteration number

Algorithm 1 APGD

```
1: Input: $f$, $S$, $x^{(0)}$, $\eta$, $N_{\text{iter}}$, $W=\{w_0,\dots,w_n\}$
2: Output: $x_{\max}$, $f_{\max}$
3: $x^{(1)} \leftarrow P_S\left(x^{(0)} + \eta\nabla f(x^{(0)})\right)$
4: $f_{\max} \leftarrow \max\{f(x^{(0)}), f(x^{(1)})\}$
5: $x_{\max} \leftarrow x^{(0)}$ if $f_{\max} \equiv f(x^{(0)})$ else $x_{\max} \leftarrow x^{(1)}$
6: for $k = 1$ to $N_{\text{iter}}-1$ do
7:     $z^{(k+1)} \leftarrow P_S\left(x^{(k)} + \eta\nabla f(x^{(k)})\right)$
8:     $x^{(k+1)} \leftarrow P_S\Big(x^{(k)} + \alpha(z^{(k+1)} - x^{(k)})$
           $+(1-\alpha)(x^{(k)} - x^{(k-1)})\Big)$
9:     if $f(x^{(k+1)}) > f_{\max}$ then
10:        $x_{\max} \leftarrow x^{(k+1)}$ and $f_{\max} \leftarrow f(x^{(k+1)})$
11:    end if
12:    if $k \in W$ then
13:        if Condition 1 or Condition 2 then
14:            $\eta \leftarrow \eta/2$ and $x^{(k+1)} \leftarrow x_{\max}$
15:        end if
16:    end if
17: end for
```

Problem with projected gradient descent



$$\max_{\boldsymbol{x}'} \ell\left(\boldsymbol{y}, f_{\boldsymbol{\theta}}(\boldsymbol{x}')\right)$$

$$\text{s.t. } d\left(\boldsymbol{x}, \boldsymbol{x}'\right) \leq \varepsilon\,, \quad \boldsymbol{x}' \in [0,1]^n$$

Tricky to set:
iteration number & step size
i.e., tricky to decide where to stop

Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. Croce, F., Hein, M., ICML 2020
https://arxiv.org/pdf/2003.01690.pdf

Penalty methods for complicated d

$$\max_{\boldsymbol{x}'} \ell\left(\boldsymbol{y}, f_{\boldsymbol{\theta}}(\boldsymbol{x}')\right)$$

$$\text{s.t. } d\left(\boldsymbol{x}, \boldsymbol{x}'\right) \leq \varepsilon, \quad \boldsymbol{x}' \in [0,1]^n$$

$$d(\boldsymbol{x}, \boldsymbol{x}') \doteq \|\phi(\boldsymbol{x}) - \phi(\boldsymbol{x}')\|_2$$

$$\text{where} \quad \phi(\boldsymbol{x}) \doteq [\, \widehat{g}_1(\boldsymbol{x}), \ldots, \widehat{g}_L(\boldsymbol{x}) \,]$$

perceptual distance

Projection onto the constraint is complicated

Penalty methods

$$\max_{\widetilde{\mathbf{x}}} \quad \mathcal{L}(f(\widetilde{\mathbf{x}}), y) - \lambda \max\left(0, \|\phi(\widetilde{\mathbf{x}}) - \phi(\mathbf{x})\|_2 - \epsilon\right)$$

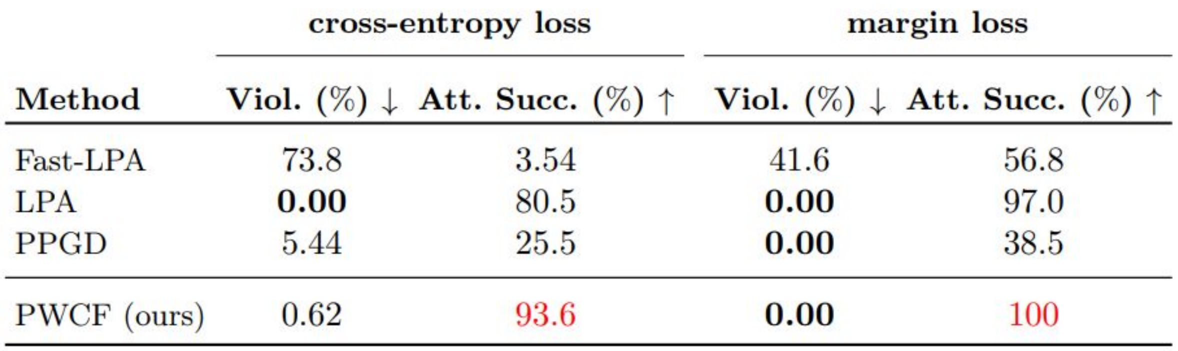
Solve it for each fixed λ and then increase λ

Algorithm 2 Lagrangian Perceptual Attack (LPA)

```
procedure LPA(classifier network $f(\cdot)$, LPIPS distance $d(\cdot,\cdot)$, input $\mathbf{x}$, label $y$, bound $\epsilon$)
    $\lambda \leftarrow 0.01$
    $\widetilde{\mathbf{x}} \leftarrow \mathbf{x} + 0.01 * \mathcal{N}(0,1)$    ▷ initialize perturbations with random Gaussian noise
    for $i$ in $1, \ldots, S$ do    ▷ we use $S = 5$ iterations to search for the best value of $\lambda$
        for $t$ in $1, \ldots, T$ do    ▷ $T$ is the number of steps
            $\Delta \leftarrow \nabla_{\widetilde{\mathbf{x}}} \left[\mathcal{L}(f(\widetilde{\mathbf{x}}), y) - \lambda \max\left(0, d(\widetilde{\mathbf{x}}, \mathbf{x}) - \epsilon\right)\right]$    ▷ take the gradient of (5)
            $\hat{\Delta} = \Delta / \|\Delta\|_2$    ▷ normalize the gradient
            $\eta = \epsilon * (0.1)^{t/T}$    ▷ the step size $\eta$ decays exponentially
            $m \leftarrow d(\widetilde{\mathbf{x}}, \widetilde{\mathbf{x}} + h\hat{\Delta})/h$    ▷ $m \approx$ derivative of $d(\widetilde{\mathbf{x}}, \cdot)$ in the direction of $\hat{\Delta}$; $h = 0.1$
            $\widetilde{\mathbf{x}} \leftarrow \widetilde{\mathbf{x}} + (\eta/m)\hat{\Delta}$    ▷ take a step of size $\eta$ in LPIPS distance
        end for
        if $d(\widetilde{\mathbf{x}}, \mathbf{x}) > \epsilon$ then
            $\lambda \leftarrow 10\lambda$    ▷ increase $\lambda$ if the attack goes outside the bound
        end if
    end for
    $\widetilde{\mathbf{x}} \leftarrow$ PROJECT$(d, \widetilde{\mathbf{x}}, \mathbf{x}, \epsilon)$
    return $\widetilde{\mathbf{x}}$
end procedure
```

Ref Perceptual adversarial robustness: Defense against unseen threat models. Laidlaw, C., Singla, S., & Feizi, S. https://arxiv.org/abs/2006.12655

Problem with penalty methods



Method	cross-entropy loss		margin loss	
	Viol. (%) ↓	Att. Succ. (%) ↑	Viol. (%) ↓	Att. Succ. (%) ↑
Fast-LPA	73.8	3.54	41.6	56.8
LPA	**0.00**	80.5	**0.00**	97.0
PPGD	5.44	25.5	**0.00**	38.5
PWCF (ours)	0.62	93.6	**0.00**	100

$$\max_{\boldsymbol{x}'} \ell\left(\boldsymbol{y}, f_{\boldsymbol{\theta}}(\boldsymbol{x}')\right)$$
$$\text{s. t. } d\left(\boldsymbol{x}, \boldsymbol{x}'\right) \leq \varepsilon\,, \quad \boldsymbol{x}' \in [0,1]^n$$

$$d(\boldsymbol{x}, \boldsymbol{x}') \doteq \|\phi(\boldsymbol{x}) - \phi(\boldsymbol{x}')\|_2$$
$$\text{where} \quad \phi(\boldsymbol{x}) \doteq [\, \widehat{g}_1(\boldsymbol{x}), \ldots, \widehat{g}_L(\boldsymbol{x}) \,]$$

LPA, Fast-LPA: penalty methods **PPGD**: Projected gradient descent

Penalty methods tend to encounter **large constraint violation** (i.e., infeasible solution, known in optimization theory) or **suboptimal solution**

PWCF, an optimizer with a principled stopping criterion on **stationarity & feasibility**

Ref Optimization and Optimizers for Adversarial Robustness. Liang, H., Liang, B., Peng, L., Cui, Y., Mitchell, T., & Sun, J. arXiv preprint arXiv:2303.13401.

Unreliable optimization
=Unreliable RE

Issues and answers

projected gradient descent

$$\min_{\mathbf{x}\in\mathcal{Q}} f(\mathbf{x})$$

$$\mathbf{x}_{k+1} = P_{\mathcal{Q}}\Big(\mathbf{x}_k - \alpha_k \nabla f(\mathbf{x}_k)\Big)$$

Issue: no principled stopping criterion /step size rules

penalty methods

$$\min_{\boldsymbol{x}} \ f(\boldsymbol{x}) \quad \text{s.t.} \ g(\boldsymbol{x}) \le \mathbf{0}$$

$$\min_{\boldsymbol{x}} \ f(\boldsymbol{x}) + \lambda \max(0, g(\boldsymbol{x}))$$

Solved with increasing λ : sequence

Issue: infeasible solution

- **Feasible & stationary solution Stationarity and feasibility check: KKT condition**
- **Reasonable speed Line search & 2nd order methods**
- **A hidden problem: nonsmoothness**

A principled solver for constrained, nonconvex, nonsmooth problems

Nonconvex, nonsmooth, constrained

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}), \quad \text{s.t. } c_i(\boldsymbol{x}) \le 0,\ \forall\, i \in \mathcal{I};\quad c_i(\boldsymbol{x}) = 0,\ \forall\, i \in \mathcal{E}.$$

Penalty sequential quadratic programming (P-SQP)

$$\boxed{\begin{aligned} \min_{d\in\mathbb{R}^n,\ s\in\mathbb{R}^p} \quad & \mu(f(x_k) + \nabla f(x_k)^\mathsf{T} d) + e^\mathsf{T} s + \frac{1}{2} d^\mathsf{T} H_k d \\ \text{s.t.} \quad & c(x_k) + \nabla c(x_k)^\mathsf{T} d \le s, \quad s \ge 0, \end{aligned}}$$

Advantage: 2nd order method (BFGS) $\rightarrow$ high-precision solution

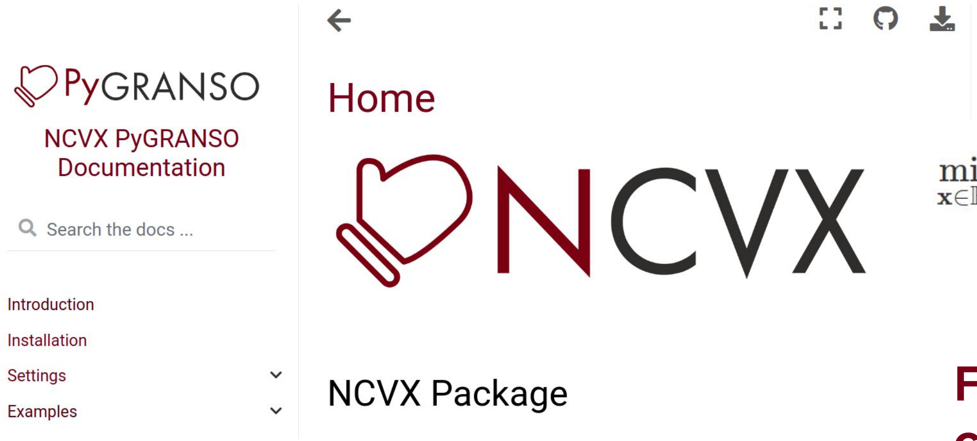
Principled line search, stationarity/feasibility check

Ref Curtis, Frank E., Tim Mitchell, and Michael L. Overton. "A BFGS-SQP method for nonsmooth, nonconvex, constrained optimization and its evaluation using relative minimization profiles." Optimization Methods and Software 32.1 (2017): 148-181.

Our PyGranso (and NCVX framework)

https://ncvx.org/

GRANSO + PyTorch


PyGRANSO

NCVX PyGRANSO Documentation

Search the docs ...

Introduction

Installation

Settings

Examples

Home

NCVX

NCVX Package

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}), \text{ s.t. } c_i(\mathbf{x}) \leq 0, \forall i \in \mathcal{I};\ c_i(\mathbf{x}) = 0, \forall i \in \mathcal{E}$$

First general-purpose solver for constrained DL problems

NCVX: A General-Purpose Optimization Solver for Constrained Machine and Deep Learning

Buyun Liang, Tim Mitchell, Ju Sun

Strategies to speed up PyGranso for RE

$$\max_{\boldsymbol{x}'} \ell\left(\boldsymbol{y}, f_{\boldsymbol{\theta}}(\boldsymbol{x}')\right)$$
$$\text{s.t. } d\left(\boldsymbol{x}, \boldsymbol{x}'\right) \le \varepsilon\,, \quad \boldsymbol{x}' \in [0,1]^n$$

$$\min_{\boldsymbol{x}'} d\left(\boldsymbol{x}, \boldsymbol{x}'\right)$$
$$\text{s.t. } \max_{i \neq y} f^i_{\boldsymbol{\theta}}(\boldsymbol{x}') \ge f^y_{\boldsymbol{\theta}}(\boldsymbol{x}')\,, \ \boldsymbol{x}' \in [0,1]^n$$

Constraint folding: many constraints into few

$$h_j(\boldsymbol{x}) = 0 \iff |h_j(\boldsymbol{x})| \le 0\,,$$
$$c_i(\boldsymbol{x}) \le 0 \iff \max\{c_i(\boldsymbol{x}), 0\} \le 0\,,$$

$$\mathcal{F}(|h_1(\boldsymbol{x})|, \cdots, |h_i(\boldsymbol{x})|, \max\{c_1(\boldsymbol{x}), 0\},$$
$$\cdots, \max\{c_j(\boldsymbol{x}), 0\}) \le 0,$$

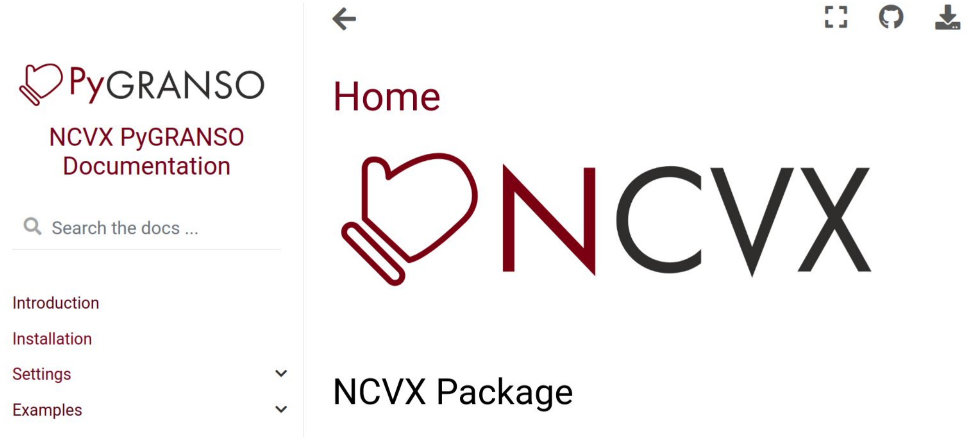
Two-stage optimization

1. **Stage 1 (selecting the best initialization):** Optimize the problems by PWCF with R different random initialization $\boldsymbol{x}^{(r,0)}$ for k iterations, where $r = 1, \ldots, R$, and collect the final first-stage solution $\boldsymbol{x}^{(r,k)}$ for each run. Determine the best intermediate result $\boldsymbol{x}^{(*,k)}$ following Algorithm 1.
2. **Stage 2 (optimization):** Warm start the optimization process with $\boldsymbol{x}^{*,k}$ until the stopping criterion is met[7] (i.e., reaching both the stationarity and feasibility tolerance, or reaching the MaxIter K).

Digression

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}),\ \text{s.t. } c_i(\mathbf{x}) \le 0, \forall i \in \mathcal{I};\ c_i(\mathbf{x}) = 0, \forall i \in \mathcal{E}$$

First general-purpose solver for constrained DL problems



PyGRANSO

NCVX PyGRANSO Documentation

Search the docs ...

Introduction

Installation

Settings

Examples

Home

NCVX

NCVX Package

NCVX: A General-Purpose Optimization Solver for Constrained Machine and Deep Learning

Buyun Liang, Tim Mitchell, Ju Sun

Enabling

Robustness evaluation

$$\max_{\boldsymbol{x}'} \ell(\boldsymbol{y}, f_{\boldsymbol{\theta}}(\boldsymbol{x}'))$$

$$\text{s.t. } \boldsymbol{x}' \in \Delta(\boldsymbol{x}) = \{\boldsymbol{x}' \in [0,1]^n : d(\boldsymbol{x}, \boldsymbol{x}') \le \varepsilon\}$$

$$\min_{\boldsymbol{x}'\in[0,1]^n} d(\boldsymbol{x}, \boldsymbol{x}') \quad \text{s.t. } \max_{\ell \ne y} f^{\ell}_{\boldsymbol{\theta}}(\boldsymbol{x}') \ge f^{y}_{\boldsymbol{\theta}}(\boldsymbol{x}')$$

Topology optimization

$$\min_{\boldsymbol{\theta},\boldsymbol{u}} \boldsymbol{u}^{\intercal}\boldsymbol{K}(\boldsymbol{G}_{\boldsymbol{\theta}}(\boldsymbol{\beta}))\boldsymbol{u} \quad \text{s.t. } \boldsymbol{K}(\boldsymbol{G}_{\boldsymbol{\theta}}(\boldsymbol{\beta}))\boldsymbol{u} = \boldsymbol{f}$$

$$\sum_{i\in\Omega} [\boldsymbol{G}_{\boldsymbol{\theta}}(\boldsymbol{\beta})]_i = v_0,\ \boldsymbol{G}_{\boldsymbol{\theta}}(\boldsymbol{\beta}) \in \{0,1\}^n$$

Buyun, Ryan, Hengyue

Imbalanced learning

$$\max_{\boldsymbol{\theta},t} \frac{\sum_{i=1}^{N} \mathbb{1}\{y_i = +1\}\, \mathbb{1}\{f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) > t\}}{\sum_{i=1}^{N} \mathbb{1}\{f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) > t\}}$$

$$\text{s.t. } \frac{\sum_{i=1}^{N} \mathbb{1}\{y_i = +1\}\, \mathbb{1}\{f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) > t\}}{\sum_{i=1}^{N} \mathbb{1}\{y_i = +1\}} \ge \alpha$$

Yash, Le, Zhong, Buyun

First general-purpose, reliable solver for RE

$$\max_{\boldsymbol{x}'} \ell\left(\boldsymbol{y}, f_{\boldsymbol{\theta}}(\boldsymbol{x}')\right) \quad \text{s.t. } d\left(\boldsymbol{x}, \boldsymbol{x}'\right) \le \varepsilon\,, \quad \boldsymbol{x}' \in [0,1]^n$$

$$\min_{\boldsymbol{x}'} \; d\left(\boldsymbol{x}, \boldsymbol{x}'\right) \quad \text{s.t. } \max_{i \neq y} f_{\boldsymbol{\theta}}^{i}(\boldsymbol{x}') \ge f_{\boldsymbol{\theta}}^{y}(\boldsymbol{x}')\,, \; \boldsymbol{x}' \in [0,1]^n$$

Reliability

- SOTA methods — RobustBench — A standardized benchmark for adversarial robustness
  No stopping criterion (only use maxit); step size scheduler
- PWCF (ours)
  Principled line-search criterion and termination condition

Generality

- SOTA methods
  Can mostly only handle several lp metrics (l1,l2,linf)
- PWCF (ours)
  Any differentiable metrics and both min and max forms
  E.g., perceptual distance

$$d(\boldsymbol{x}, \boldsymbol{x}') \doteq \|\phi(\boldsymbol{x}) - \phi(\boldsymbol{x}')\|_2$$

$$\text{where} \quad \phi(\boldsymbol{x}) \doteq [\; \widehat{g}_1(\boldsymbol{x}), \dots, \widehat{g}_L(\boldsymbol{x}) \;]$$

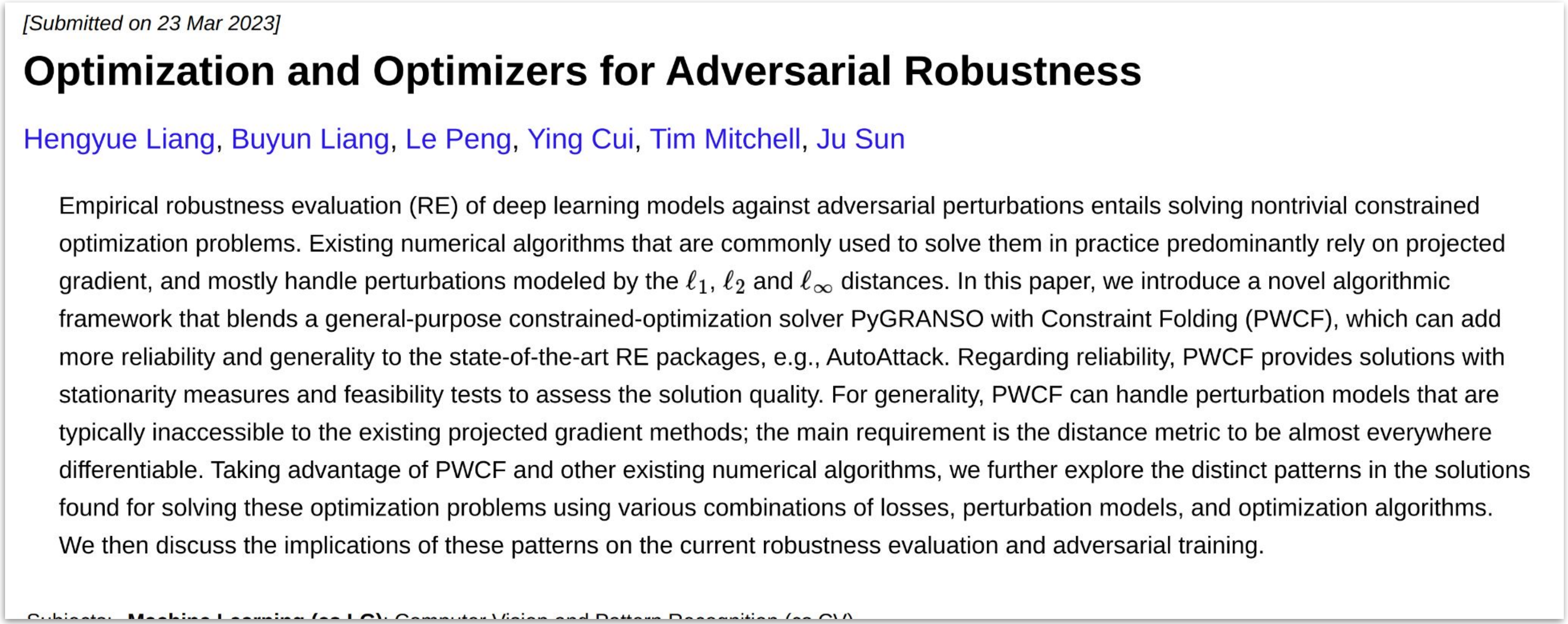
A quick example

$$\max_{\boldsymbol{x}'} \ell\left(\boldsymbol{y}, f_{\boldsymbol{\theta}}(\boldsymbol{x}')\right) \quad \text{s. t. } d\left(\boldsymbol{x}, \boldsymbol{x}'\right) \le \varepsilon\,, \quad \boldsymbol{x}' \in [0,1]^n$$

$$d(\boldsymbol{x}, \boldsymbol{x}') \doteq \|\phi(\boldsymbol{x}) - \phi(\boldsymbol{x}')\|_2 \quad \text{where} \quad \phi(\boldsymbol{x}) \doteq [\, \widehat{g}_1(\boldsymbol{x}), \dots, \widehat{g}_L(\boldsymbol{x})\,]$$

	cross-entropy loss		**margin loss**	
Method	**Viol. (%) ↓**	**Att. Succ. (%) ↑**	**Viol. (%) ↓**	**Att. Succ. (%) ↑**
Fast-LPA	73.8	3.54	41.6	56.8
LPA	**0.00**	80.5	**0.00**	97.0
PPGD	5.44	25.5	**0.00**	38.5
PWCF (ours)	0.62	93.6	**0.00**	100

More details



[Submitted on 23 Mar 2023]

Optimization and Optimizers for Adversarial Robustness

Hengyue Liang, Buyun Liang, Le Peng, Ying Cui, Tim Mitchell, Ju Sun

Empirical robustness evaluation (RE) of deep learning models against adversarial perturbations entails solving nontrivial constrained optimization problems. Existing numerical algorithms that are commonly used to solve them in practice predominantly rely on projected gradient, and mostly handle perturbations modeled by the ℓ_1, ℓ_2 and ℓ_∞ distances. In this paper, we introduce a novel algorithmic framework that blends a general-purpose constrained-optimization solver PyGRANSO with Constraint Folding (PWCF), which can add more reliability and generality to the state-of-the-art RE packages, e.g., AutoAttack. Regarding reliability, PWCF provides solutions with stationarity measures and feasibility tests to assess the solution quality. For generality, PWCF can handle perturbation models that are typically inaccessible to the existing projected gradient methods; the main requirement is the distance metric to be almost everywhere differentiable. Taking advantage of PWCF and other existing numerical algorithms, we further explore the distinct patterns in the solutions found for solving these optimization problems using various combinations of losses, perturbation models, and optimization algorithms. We then discuss the implications of these patterns on the current robustness evaluation and adversarial training.

Subjects: **Machine Learning (cs.LG)**; Computer Vision and Pattern Recognition (cs.CV)

ML models are not perfect

$$(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N) \sim_{iid} \mathcal{D}_{\mathcal{X}\times\mathcal{Y}} \text{ on } \mathcal{X} \times \mathcal{Y}.$$

$$\widehat{\mathsf{R}}_S(f) \doteq \frac{1}{N} \sum\nolimits_{i\in[N]} \mathbb{1}\{f(\boldsymbol{x}_i) \neq y_i\}$$

$$\mathsf{R}(f) \doteq \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{\mathcal{X}\times\mathcal{Y}}} \mathbb{1}\{f(\boldsymbol{x}) \neq y\}$$

$$h_* \in \arg\min\nolimits_{h \text{ ``reasonable''}} \mathsf{R}(h).$$

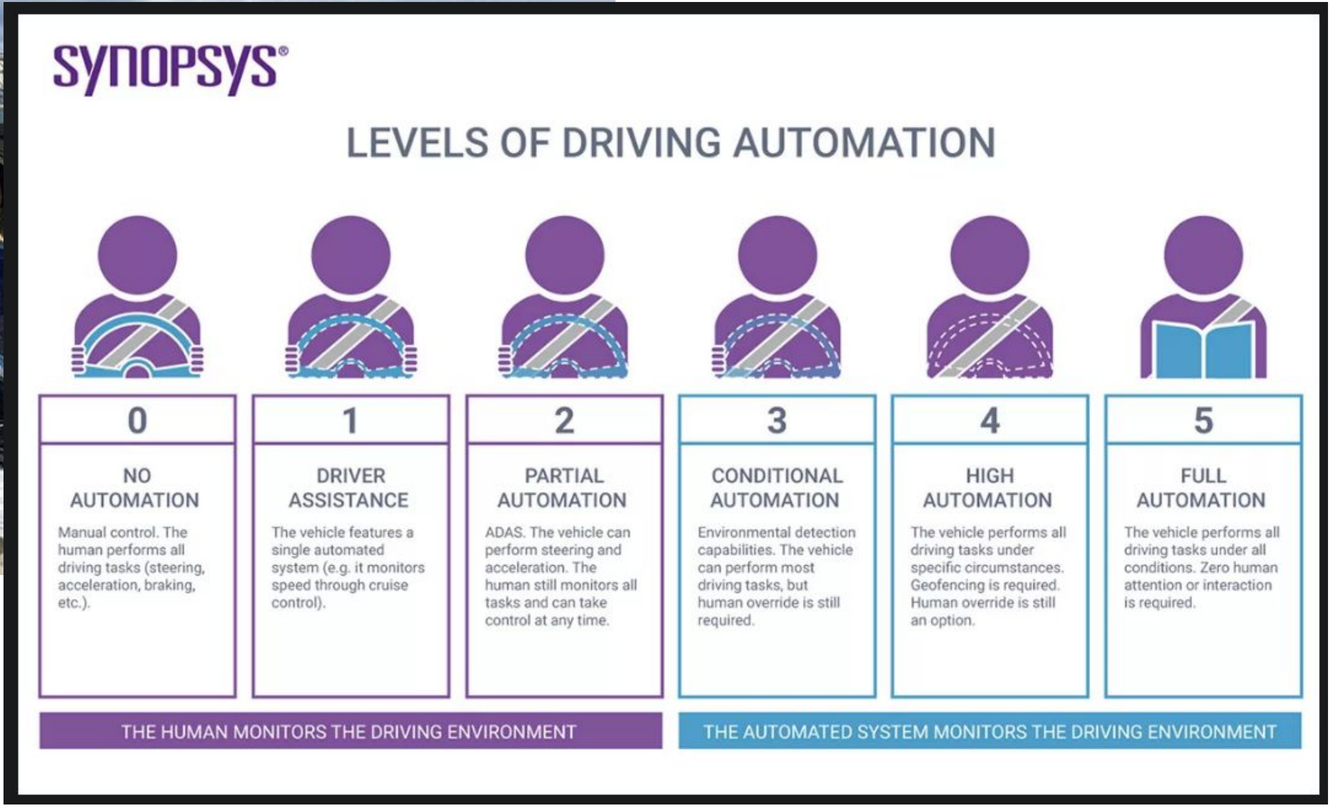
Bayes optimal classifier for binary classification

$$\arg\max\nolimits_{y\in\{+1,-1\}} \mathbb{P}[y|\boldsymbol{x}]$$

$$\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_{\mathcal{X}}} \min(\mathbb{P}[1|\boldsymbol{x}], \mathbb{P}[-1|\boldsymbol{x}]) \in [0, 1/2]$$

100% often not achievable even if with infinite amount of training data

Imperfect AI models can still be deployed



SYNOPSYS®

LEVELS OF DRIVING AUTOMATION

0	1	2	3	4	5
NO AUTOMATION	DRIVER ASSISTANCE	PARTIAL AUTOMATION	CONDITIONAL AUTOMATION	HIGH AUTOMATION	FULL AUTOMATION
Manual control. The human performs all driving tasks (steering, acceleration, braking, etc.).	The vehicle features a single automated system (e.g. it monitors speed through cruise control).	ADAS. The vehicle can perform steering and acceleration. The human still monitors all tasks and can take control at any time.	Environmental detection capabilities. The vehicle can perform most driving tasks, but human override is still required.	The vehicle performs all driving tasks under specific circumstances. Geofencing is required. Human override is still an option.	The vehicle performs all driving tasks under all conditions. Zero human attention or interaction is required.
THE HUMAN MONITORS THE DRIVING ENVIRONMENT			THE AUTOMATED SYSTEM MONITORS THE DRIVING ENVIRONMENT		

A crucial component: allowing AI to restrain itself

$$\text{predictor } f : \mathcal{X} \to \mathbb{R}^K \qquad \text{selector } g : \mathcal{X} \to \{0,1\}$$

$$(f,g)(\boldsymbol{x}) \triangleq \begin{cases} f(\boldsymbol{x}) & \text{if } g(\boldsymbol{x}) = 1; \\ \boxed{\text{abstain} \quad \text{if } g(\boldsymbol{x}) = 0}. \end{cases}$$

No prediction on uncertain samples and defer them to humans

$$g_\gamma(\boldsymbol{x}) = \mathbb{1}[s(\boldsymbol{x}) > \gamma]$$

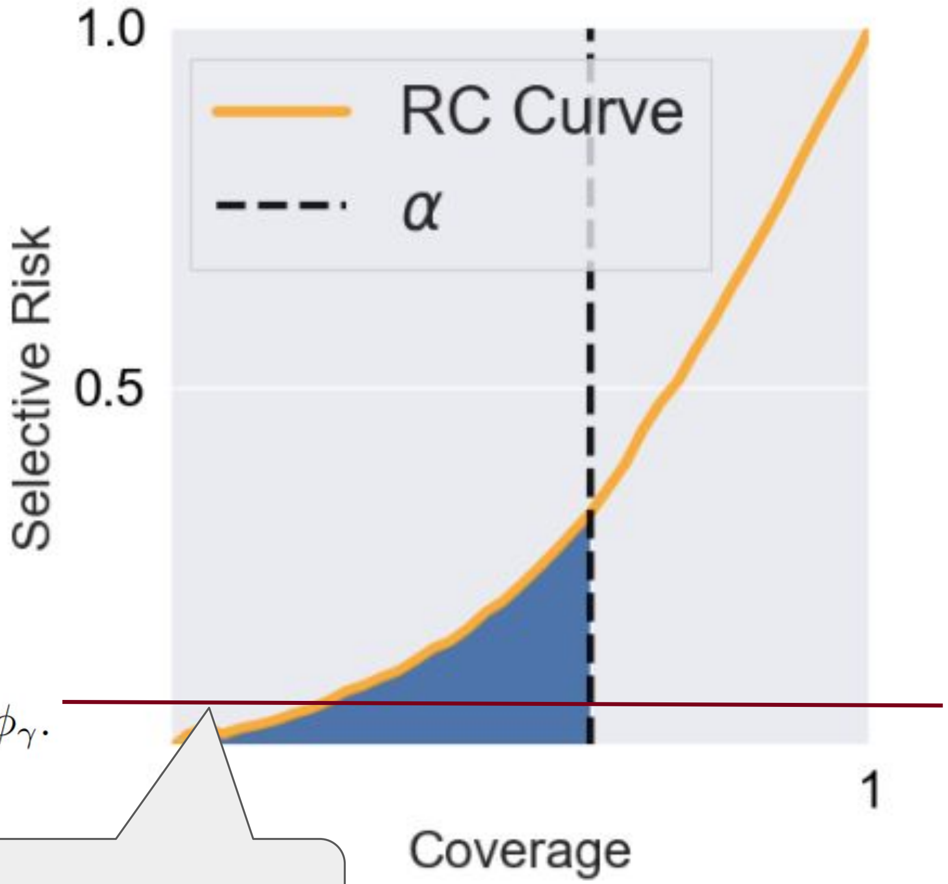
Typically, selection by thresholding prediction confidence

Risk-coverage tradeoff

$$(f,g)(\boldsymbol{x}) \triangleq \begin{cases} f(\boldsymbol{x}) & \text{if } g(\boldsymbol{x}) = 1; \\ \text{abstain} & \text{if } g(\boldsymbol{x}) = 0. \end{cases}$$

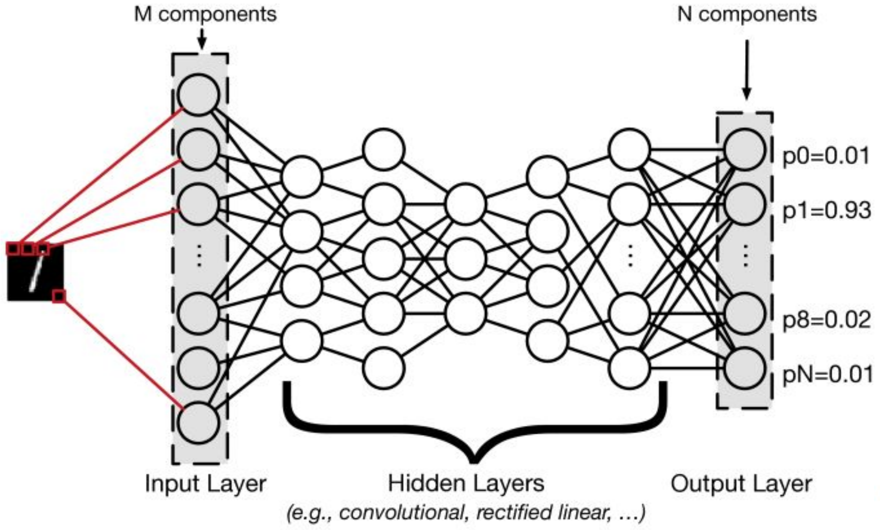
$$g_\gamma(\boldsymbol{x}) = \mathbb{1}[s(\boldsymbol{x}) > \gamma]$$

$$(\textbf{coverage})\ \phi_\gamma = \mathbb{E}_\mathcal{D}[g_\gamma(\boldsymbol{x})],$$
$$(\textbf{selection risk})\ R_\gamma = \mathbb{E}_\mathcal{D}[\ell(f(\boldsymbol{x}), y) g_\gamma(\boldsymbol{x})]/\phi_\gamma.$$



High-stakes corner

Which confidence score?



$$\approx p(y = 1|\boldsymbol{x})$$

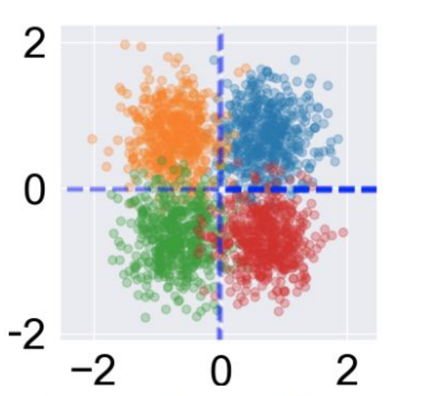
$\boldsymbol{z} \in \mathbb{R}^K$ contains the raw logits (RLs)

$$SR_{\text{max}} \triangleq \max_i \sigma(z^i),$$
$$SR_{\text{doctor}} \triangleq (\|\sigma(\boldsymbol{z})\|_2^2 - 1)/\|\sigma(\boldsymbol{z})\|_2^2 = 1 - \|\sigma(\boldsymbol{z})\|_1/\|\sigma(\boldsymbol{z})\|_2^2,$$
$$SR_{\text{ent}} \triangleq \sum_i \sigma(z^i) \log \sigma(z^i),$$

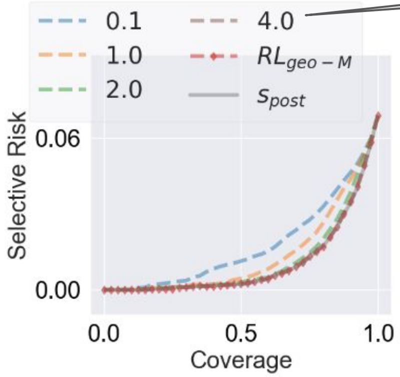
$z \in \mathbb{R}^K$ contains the raw logits (RLs)

But are they good scores?

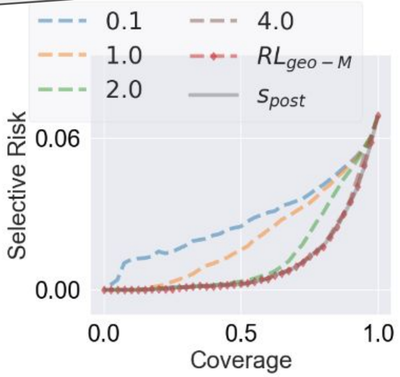
Scale factor applied to RLs



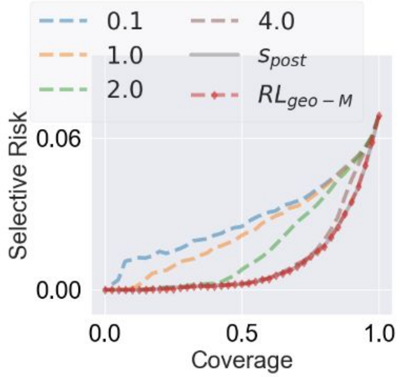
(a) Sample visualization



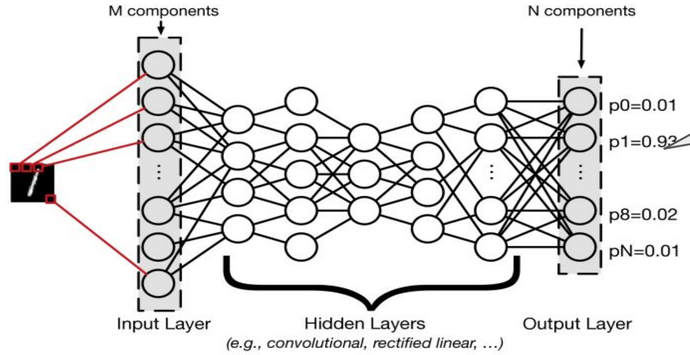
(b) $SR_{\max}$



(c) SR_{doctor}



(d) SR_{ent}



$\approx p(\cancel{\quad} 1|\boldsymbol{x})$

Calibration: align the outputs with the true posterior probs

Our margin-based scores

Signed dist to the separating hyperplane

Binary SVMs: $f(\boldsymbol{x}) = \boldsymbol{w}^\intercal \boldsymbol{x} + b$ **Geometric margin:** $y(\boldsymbol{w}^\intercal \boldsymbol{x} + b)/\|\boldsymbol{w}\|_2$

Multiclass SVMs: $f(\boldsymbol{x}) = \boldsymbol{W}^\intercal \boldsymbol{x} + \boldsymbol{b}$

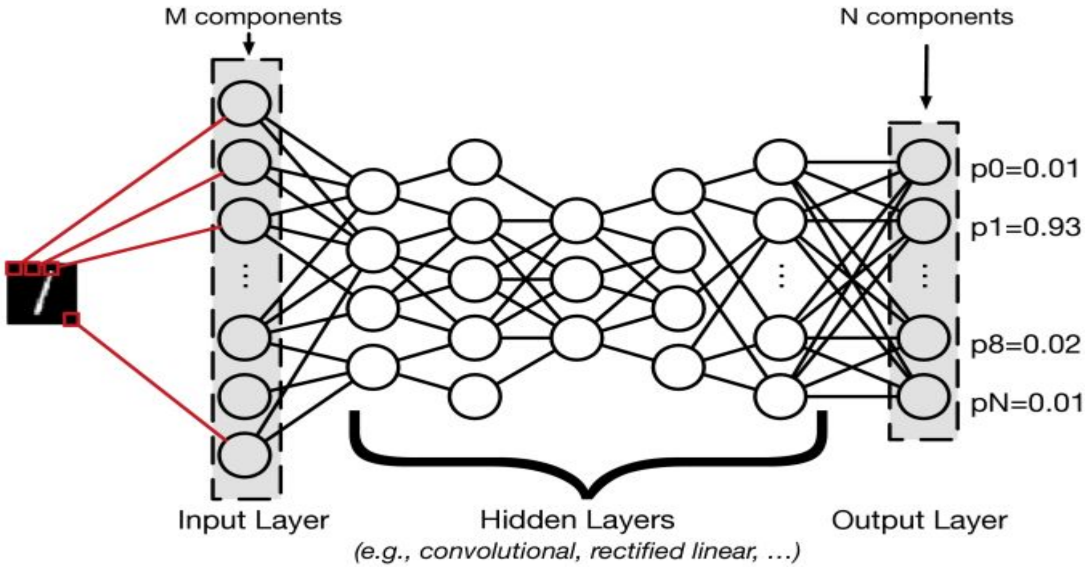
Geometric margin: $$\frac{\boldsymbol{w}_{y'}^\intercal \boldsymbol{x} + b_{y'}}{\|\boldsymbol{w}_{y'}\|_2} - \max_{j \in \{1,\ldots,K\} \setminus y'} \frac{\boldsymbol{w}_j^\intercal \boldsymbol{x} + b_j}{\|\boldsymbol{w}_j\|_2}$$

Confidence margin: $$(\boldsymbol{w}_{y'}^\intercal \boldsymbol{x} + b_{y'}) - \max_{i \in \{1,\ldots,K\} \setminus y'} (\boldsymbol{w}_i^\intercal \boldsymbol{x} + b_i)$$

These scores are not affected by the logit scaling

Difference of dists between the two nearest hyperplanes

Our margin-based scores



Geometric margin:

$$\frac{\boldsymbol{w}_{y'}^{\mathsf{T}}\boldsymbol{x} + b_{y'}}{\|\boldsymbol{w}_{y'}\|_2} - \max_{j\in\{1,\ldots,K\}\setminus y'} \frac{\boldsymbol{w}_j^{\mathsf{T}}\boldsymbol{x} + b_j}{\|\boldsymbol{w}_j\|_2}$$

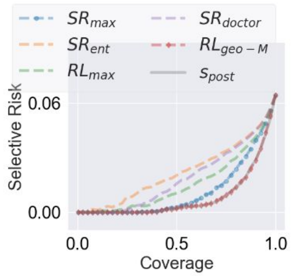
Confidence margin:

$$(\boldsymbol{w}_{y'}^{\mathsf{T}}\boldsymbol{x} + b_{y'}) - \max_{i\in\{1,\ldots,K\}\setminus y'} (\boldsymbol{w}_i^{\mathsf{T}}\boldsymbol{x} + b_i)$$

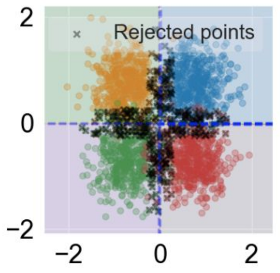
Apply them to the RLs z

Benefit: We don't need to worry about the scale of z

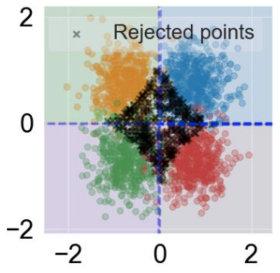
Additional benefit: robustness



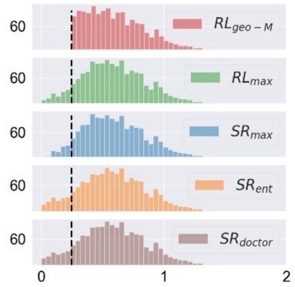
(a-1) RC curves



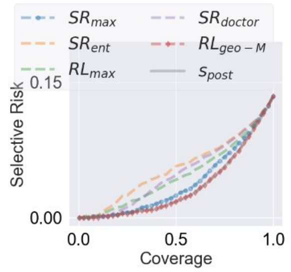
(b-1) $RL_{\text{geo-M}}$



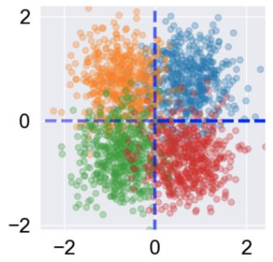
(c-1) $SR_{\max}$



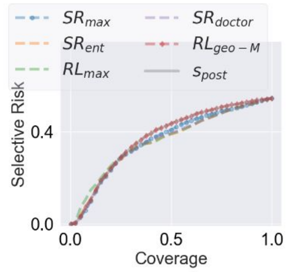
(d-1) Robustness radius



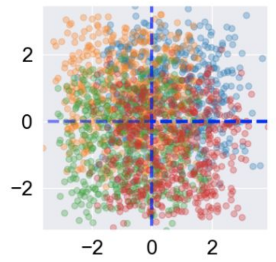
(a-2) RC curves



(b-2) Sample visualization



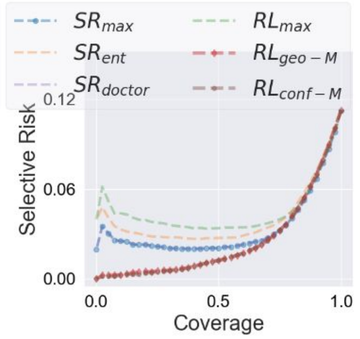
(a-3) RC curves



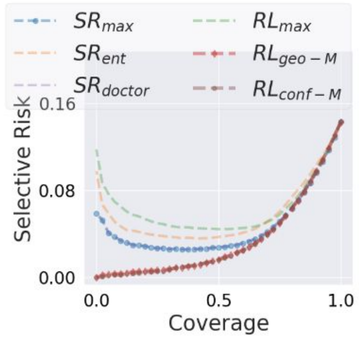
(b-3) Sample visualization

On real data

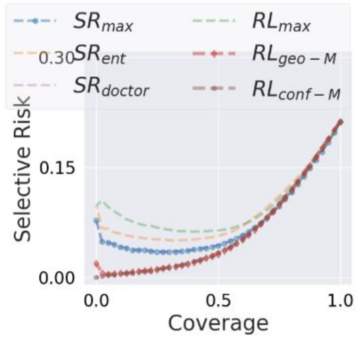
ImageNet vs ImageNet-C



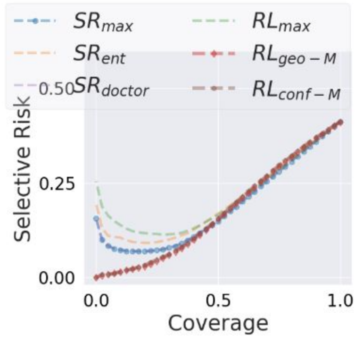
(a) IN (Clean)



(b) Gaussian blur Lv.1



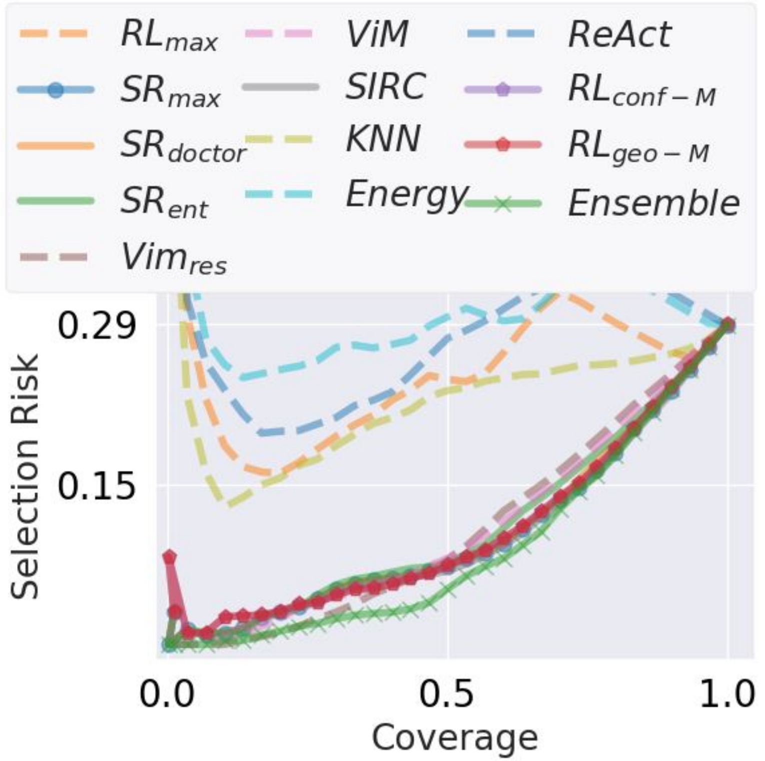
(c) Gaussian blur Lv.3

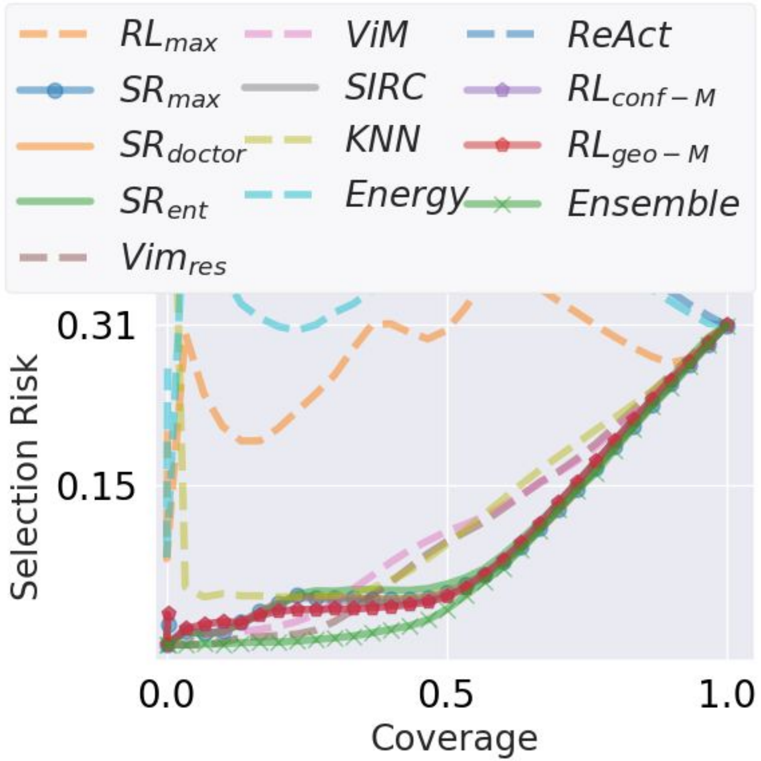


(d) Gaussian blur Lv.5

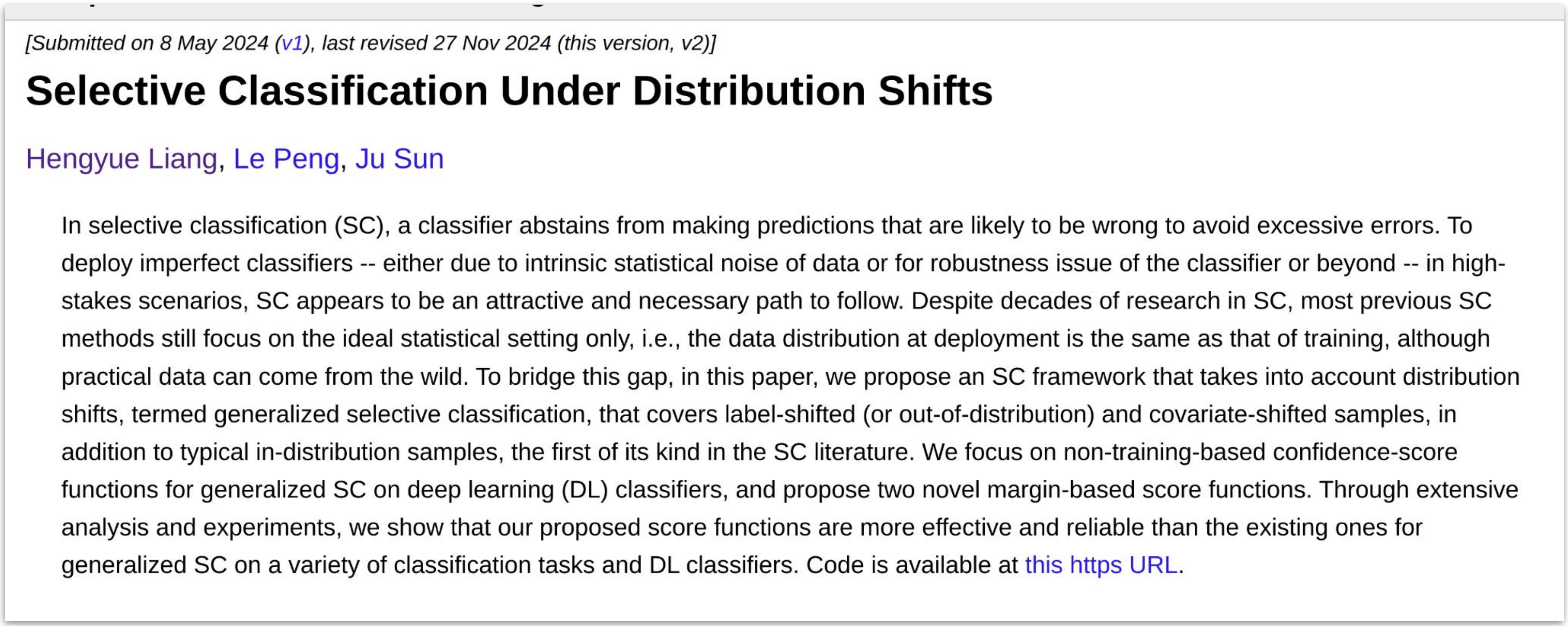
	IN (Clean)			Gaussian Blur			Brightness			Fog			Snow		
α	0.1	0.5	1	0.1	0.5	1	0.1	0.5	1	0.1	0.5	1	0.1	0.5	1
$RL_{\text{conf-M}}$	**0.16**	**0.53**	**2.39**	**0.37**	**1.31**	6.05	**0.21**	**0.72**	**3.35**	**0.14**	**0.79**	**4.21**	**0.17**	**0.95**	**4.80**
$RL_{\text{geo-M}}$	0.27	0.59	2.43	0.57	1.36	**6.04**	0.33	0.79	3.39	0.25	0.86	4.22	0.34	1.02	4.81
$RL_{\max}$	5.54	4.05	4.57	9.74	7.38	9.52	7.38	5.17	6.06	7.74	5.77	7.01	9.44	6.44	7.90
$SR_{\max}$	3.19	2.40	3.38	5.02	4.02	7.39	4.07	2.90	4.53	3.92	3.07	5.37	5.35	3.67	6.13
SR_{ent}	4.28	3.13	4.04	6.80	5.63	8.71	5.51	4.01	5.48	5.56	4.37	6.42	7.29	5.07	7.27
SR_{doctor}	3.21	2.38	3.40	5.05	4.05	7.47	4.10	2.93	4.58	3.95	3.10	5.42	5.39	3.71	6.20

Boosting the confidence?





More details



[Submitted on 8 May 2024 (v1), last revised 27 Nov 2024 (this version, v2)]

Selective Classification Under Distribution Shifts

Hengyue Liang, Le Peng, Ju Sun

In selective classification (SC), a classifier abstains from making predictions that are likely to be wrong to avoid excessive errors. To deploy imperfect classifiers -- either due to intrinsic statistical noise of data or for robustness issue of the classifier or beyond -- in high-stakes scenarios, SC appears to be an attractive and necessary path to follow. Despite decades of research in SC, most previous SC methods still focus on the ideal statistical setting only, i.e., the data distribution at deployment is the same as that of training, although practical data can come from the wild. To bridge this gap, in this paper, we propose an SC framework that takes into account distribution shifts, termed generalized selective classification, that covers label-shifted (or out-of-distribution) and covariate-shifted samples, in addition to typical in-distribution samples, the first of its kind in the SC literature. We focus on non-training-based confidence-score functions for generalized SC on deep learning (DL) classifiers, and propose two novel margin-based score functions. Through extensive analysis and experiments, we show that our proposed score functions are more effective and reliable than the existing ones for generalized SC on a variety of classification tasks and DL classifiers. Code is available at this https URL.

Today's talk: **toward trustworthy and efficient AI**

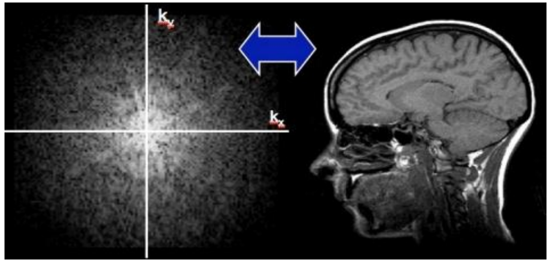
- **Robustness evaluation and selective classification**
- **Inverse problems with pretrained diffusion models**
- **AI for healthcare: handling data imbalance**

Inverse problems

$\boxed{\text{Inverse problem: given } \mathbf{y} = f(\mathbf{x}), \text{ recover } \mathbf{x}}$

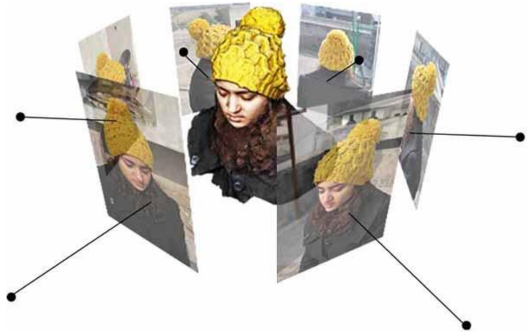
Image denoising

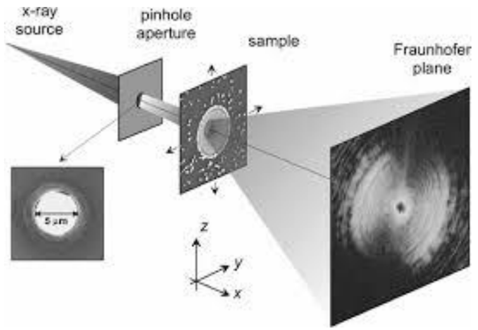
$$\begin{tikzcd} \boxed{\mathbf{x}} \arrow[r, bend left, "f"] & \boxed{\mathbf{y}} \arrow[l, dashed, bend left, "f^{-1}"] \end{tikzcd}$$



MRI reconstruction

Image super-resolution


3D reconstruction



Coherent diffraction imaging (CDI)

Traditional methods

Inverse problem: given $\mathbf{y} = f(\mathbf{x})$, recover $\mathbf{x}$

$$\min_{\mathbf{x}} \underbrace{\ell(\mathbf{y}, f(\mathbf{x}))}_{\text{data fitting}} + \lambda \underbrace{\mathrm{R}(\mathbf{x})}_{\text{regularizer}}$$

RegFit

Questions

- Which ℓ? (e.g., unknown/compound noise)
- Which R? (e.g., structures not amenable to math description)
- Speed

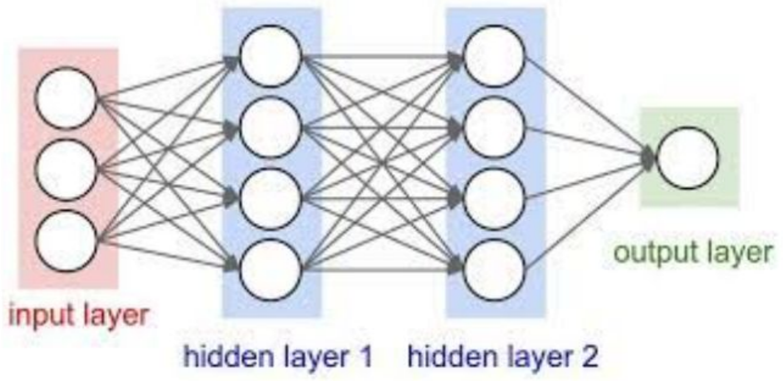
Deep learning has changed everything

With paired datasets $\{(\boldsymbol{y}_i, \boldsymbol{x}_i)\}_{i=1,\dots,N}$

Direct inversion

Learn f^{-1} from $\{(\boldsymbol{y}_i, \boldsymbol{x}_i)\}_{i=1,\dots,N}$

$\mathbf{y}_i$



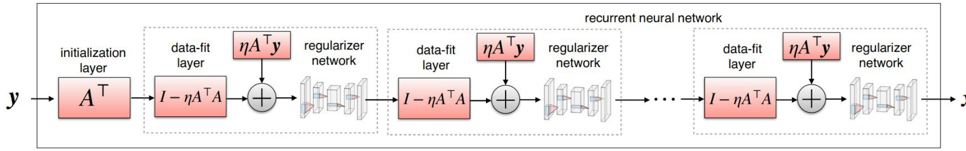
$\mathbf{x}_i$

Algorithm unrolling

$$\min_{\mathbf{x}} \ell(\mathbf{y}, f(\mathbf{x})) + \lambda \, \mathrm{R}(\mathbf{x})$$

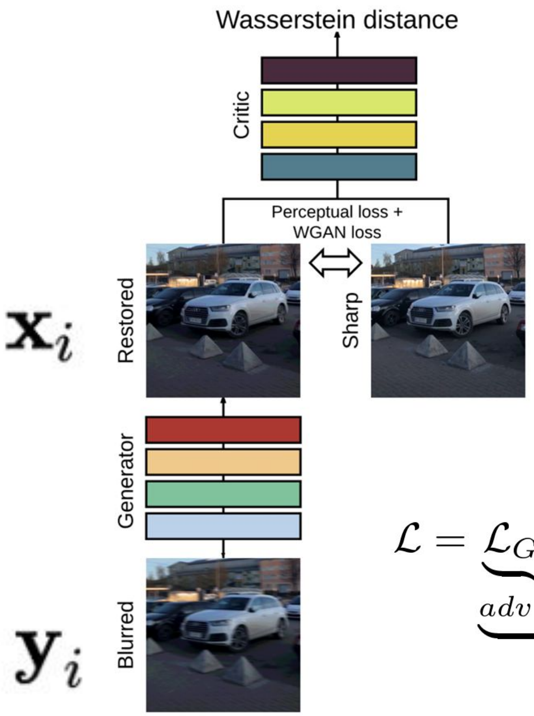
$$\mathbf{x}^{k+1} = \mathcal{P}_R\left(\mathbf{x}^k - \eta \nabla^\top f(\mathbf{x}^k) \ell'(\mathbf{y}, f(\mathbf{x}^k))\right)$$

Idea: make $\mathcal{P}_R$ trainable



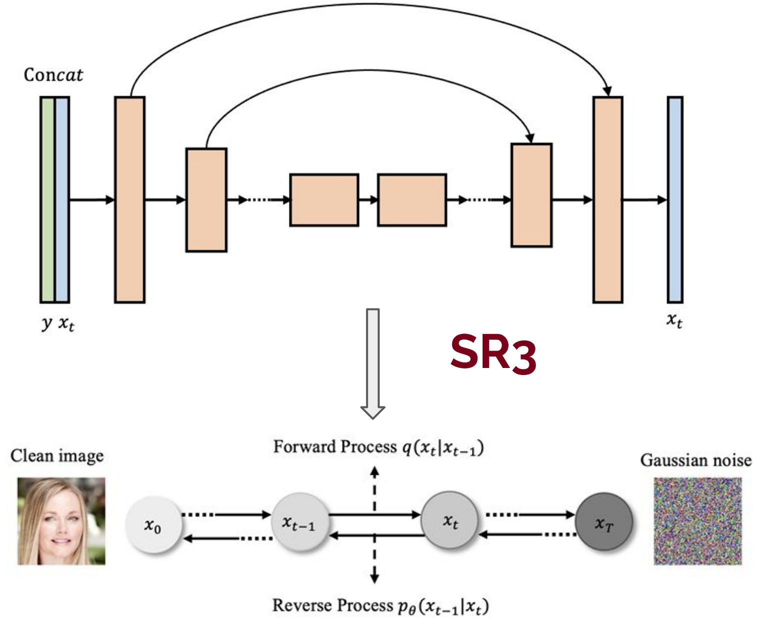
With paired datasets $\{(\boldsymbol{y}_i, \boldsymbol{x}_i)\}_{i=1,\ldots,N}$

Conditional generation & regularization



DeblurGAN

$$\mathcal{L} = \underbrace{\underbrace{\mathcal{L}_{GAN}}_{adv\ loss} + \underbrace{\lambda \cdot \mathcal{L}_X}_{content\ loss}}_{total\ loss}$$



Concat
y x_t
x_t
Clean image
Forward Process $q(x_t|x_{t-1})$
Gaussian noise
x_0
x_{t-1}
x_t
x_T
Reverse Process $p_\theta(x_{t-1}|x_t)$

SR3

Image credit: https://arxiv.org/abs/2308.09388

With object datasets only $\{\boldsymbol{x}_i\}_{i=1,\ldots,N}$

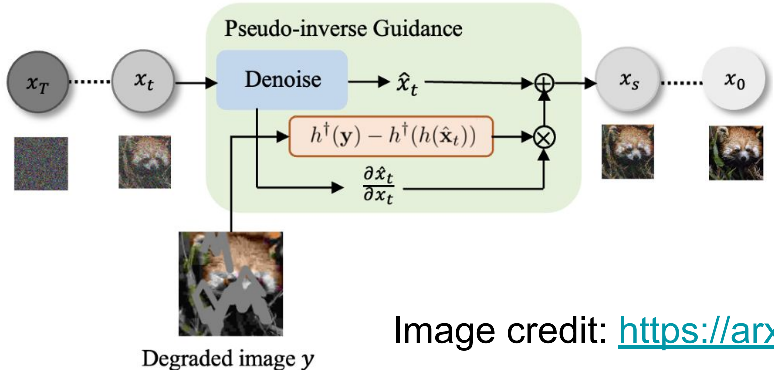
Model the distribution of the objects first, and then plug the prior in

GAN Inversion

Pretraining: $\mathbf{x}_i \approx G_\theta(\mathbf{z}_i) \ \forall i$

Deployment: $\min_{\mathbf{z}} \ \ell(\mathbf{y}, f \circ G_\theta(\mathbf{z})) + \lambda R \circ G_\theta(\mathbf{z})$

Interleaving pretrained diffusion models



Pseudo-inverse Guidance
x_T
x_t
Denoise
$\hat{x}_t$
$h^\dagger(\mathbf{y}) - h^\dagger(h(\hat{\mathbf{x}}_t))$
$\frac{\partial \hat{x}_t}{\partial x_t}$
x_s
x_0
Degraded image y

Image credit: https://arxiv.org/abs/2308.09388

Without datasets? Single-instance methods

Deep image prior (DIP) $\mathbf{x} \approx G_\theta(\mathbf{z})$ G_θ (and $\mathbf{z}$) trainable

$$\min_{\mathbf{x}} \underbrace{\ell(\mathbf{y},\, f(\mathbf{x}))}_{\text{data fitting}} + \lambda \underbrace{\mathrm{R}(\mathbf{x})}_{\text{regularizer}}$$

$$\Downarrow$$

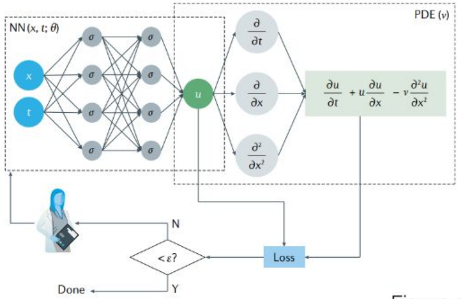
$$\min_{\theta} \ell(\mathbf{y},\, f \circ G_\theta(\mathbf{z})) + \lambda R \circ G_\theta(\mathbf{z})$$

No extra training data!

Neural implicit representation (NIR)

$\mathbf{x} \approx \mathcal{D} \circ \overline{\mathbf{x}}$ $\mathcal{D}$: discretization $\overline{\mathbf{x}}$: continuous function

Physics-informed neural networks (PINN)


NN (x, t; θ)
x
t
σ
u
PDE (ν)
$\frac{\partial}{\partial t}$
$\frac{\partial}{\partial x}$
$\frac{\partial^2}{\partial x^2}$
$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2}$
Loss
< ε?
N
Y
Done

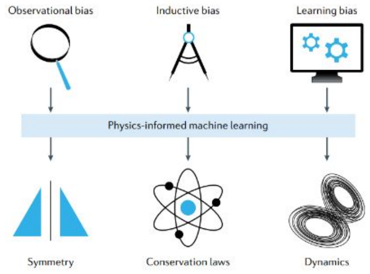


Figure credit: https://www.nature.com/articles/s42254-021-00314-5

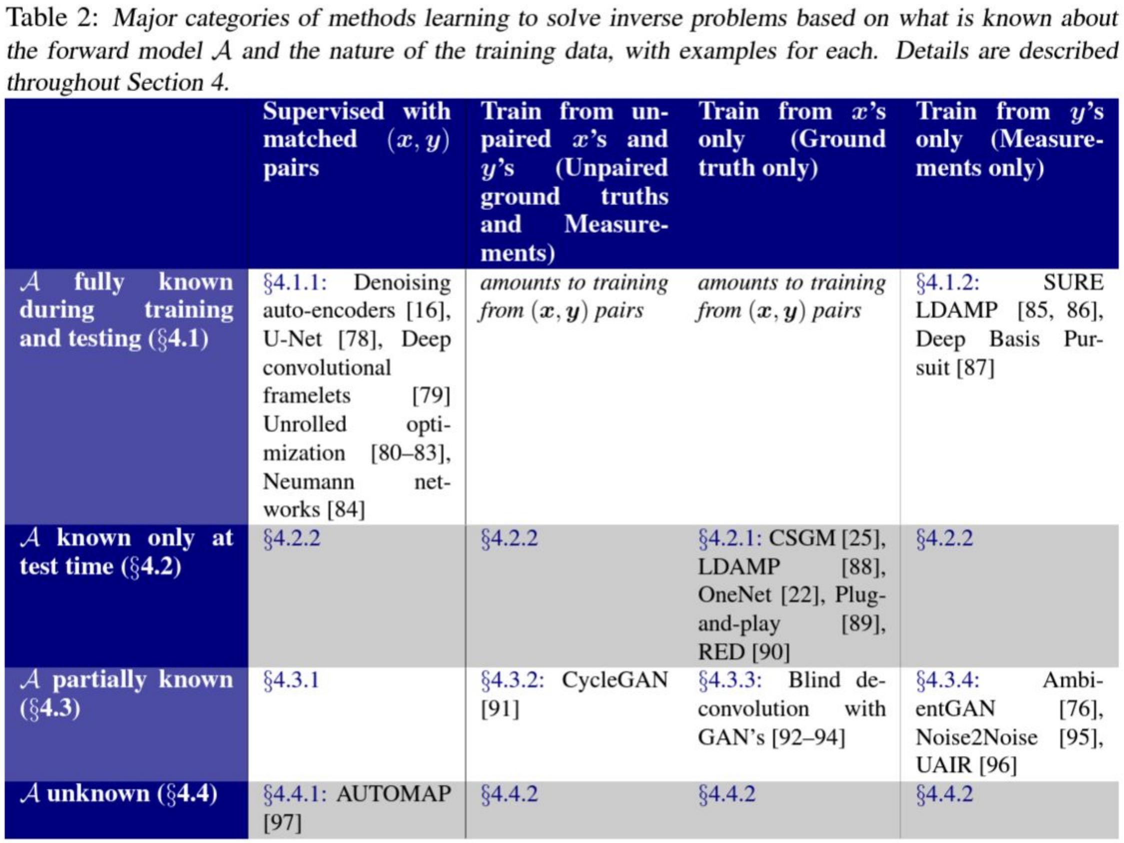
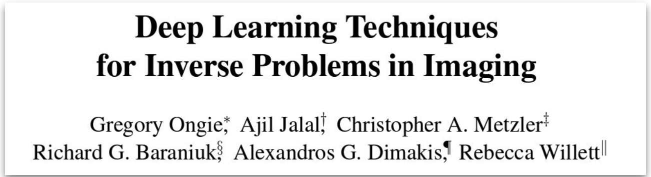


Table 2: *Major categories of methods learning to solve inverse problems based on what is known about the forward model $\mathcal{A}$ and the nature of the training data, with examples for each. Details are described throughout Section 4.*

	Supervised with matched $(\boldsymbol{x}, \boldsymbol{y})$ pairs	**Train from unpaired $\boldsymbol{x}$'s and $\boldsymbol{y}$'s (Unpaired ground truths and Measurements)**	**Train from $\boldsymbol{x}$'s only (Ground truth only)**	**Train from $\boldsymbol{y}$'s only (Measurements only)**
$\mathcal{A}$ fully known during training and testing (§4.1)	§4.1.1: Denoising auto-encoders [16], U-Net [78], Deep convolutional framelets [79] Unrolled optimization [80–83], Neumann networks [84]	*amounts to training from $(\boldsymbol{x}, \boldsymbol{y})$ pairs*	*amounts to training from $(\boldsymbol{x}, \boldsymbol{y})$ pairs*	§4.1.2: SURE LDAMP [85, 86], Deep Basis Pursuit [87]
$\mathcal{A}$ known only at test time (§4.2)	§4.2.2	§4.2.2	§4.2.1: CSGM [25], LDAMP [88], OneNet [22], Plug-and-play [89], RED [90]	§4.2.2
$\mathcal{A}$ partially known (§4.3)	§4.3.1	§4.3.2: CycleGAN [91]	§4.3.3: Blind deconvolution with GAN's [92–94]	§4.3.4: AmbientGAN [76], Noise2Noise [95], UAIR [96]
$\mathcal{A}$ unknown (§4.4)	§4.4.1: AUTOMAP [97]	§4.4.2	§4.4.2	§4.4.2



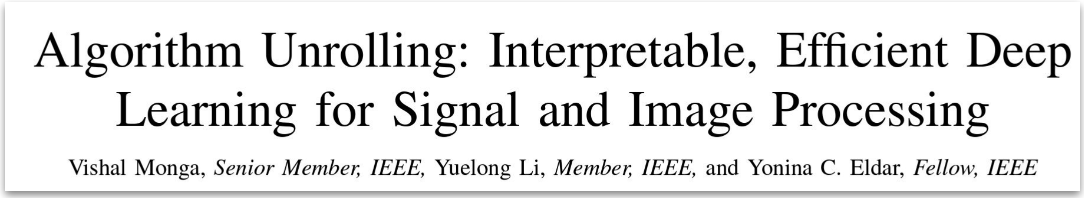
Deep Learning Techniques for Inverse Problems in Imaging

Gregory Ongie[*] Ajil Jalal[†] Christopher A. Metzler[‡]
Richard G. Baraniuk[§] Alexandros G. Dimakis[¶] Rebecca Willett[‖]

https://arxiv.org/abs/2005.06001

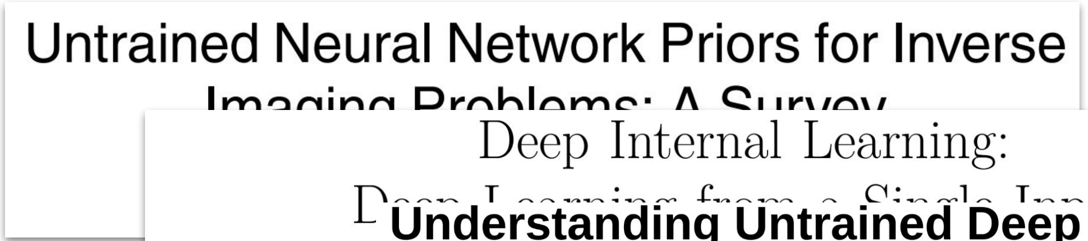
But focused on linear IPs

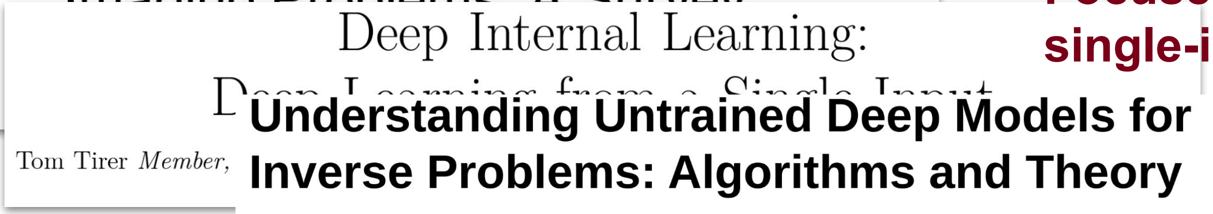
Other specialized surveys


Algorithm Unrolling: Interpretable, Efficient Deep Learning for Signal and Image Processing

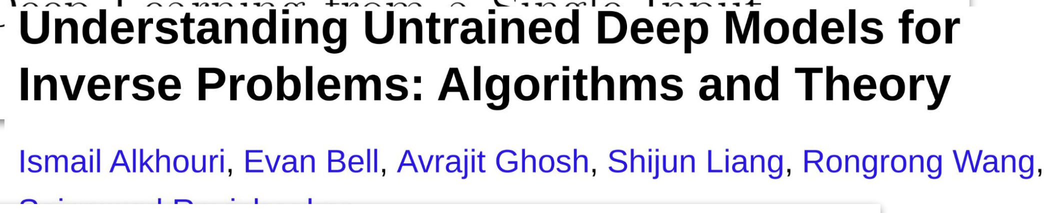
Vishal Monga, *Senior Member, IEEE,* Yuelong Li, *Member, IEEE,* and Yonina C. Eldar, *Fellow, IEEE*

Focused on alg. unrolling


Untrained Neural Network Priors for Inverse Imaging Problems: A Survey

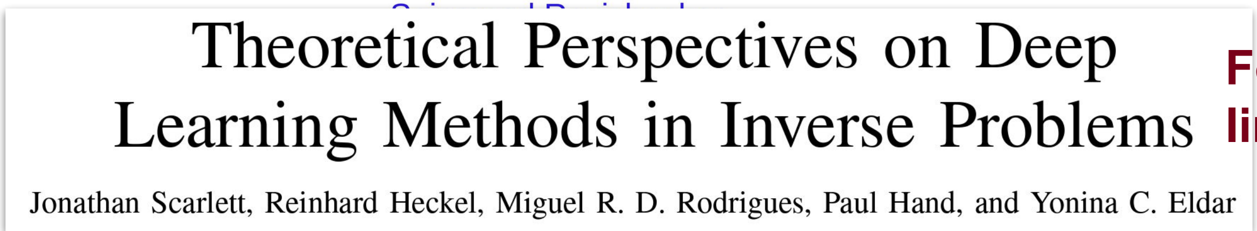

Deep Internal Learning: Deep Learning from a Single Input

Tom Tirer *Member,*


Understanding Untrained Deep Models for Inverse Problems: Algorithms and Theory

Ismail Alkhouri, Evan Bell, Avrajit Ghosh, Shijun Liang, Rongrong Wang, [illegible]

Focused on single-instance methods


Theoretical Perspectives on Deep Learning Methods in Inverse Problems

Jonathan Scarlett, Reinhard Heckel, Miguel R. D. Rodrigues, Paul Hand, and Yonina C. Eldar

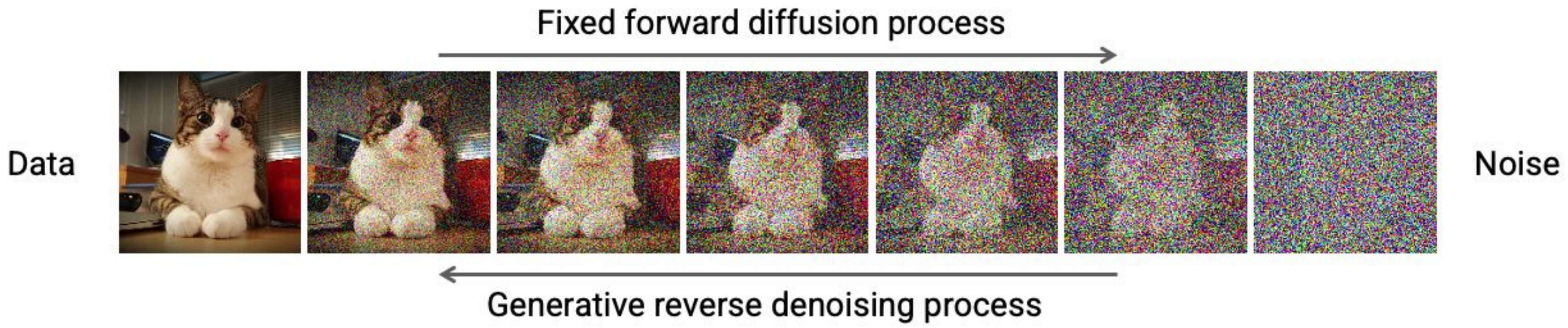
Focused on theories for linear IPs

Focus here:

Solving Inverse Problems (IPs)
Using Pretrained Diffusion Models

Diffusion models

$$d\boldsymbol{x} = -\beta_t/2 \cdot \boldsymbol{x}dt + \sqrt{\beta_t}d\boldsymbol{w},$$

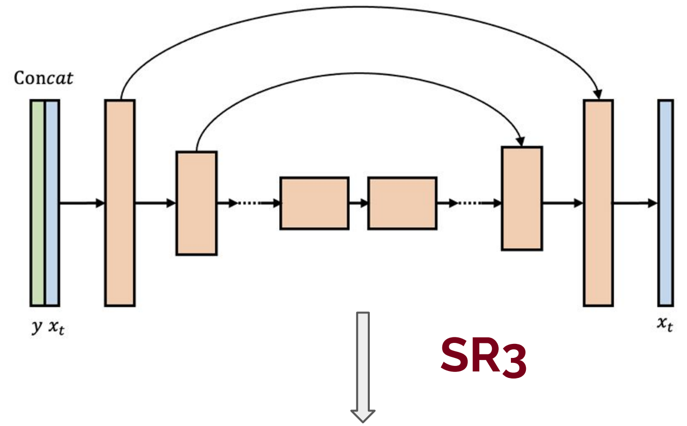

Fixed forward diffusion process
Data
Noise
Generative reverse denoising process

$$d\boldsymbol{x} = -\beta_t \left[\boldsymbol{x}/2 + \boxed{\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})}\right] dt + \sqrt{\beta_t}d\overline{\boldsymbol{w}}.$$

$$\cong \quad \boldsymbol{\varepsilon}_{\boldsymbol{\theta}}^{(t)}(\boldsymbol{x})$$

Diffusion models for inverse problems (IPs)

Supervised



SR3

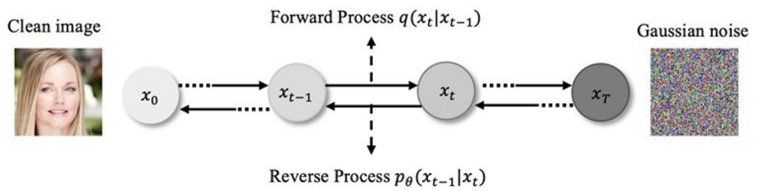
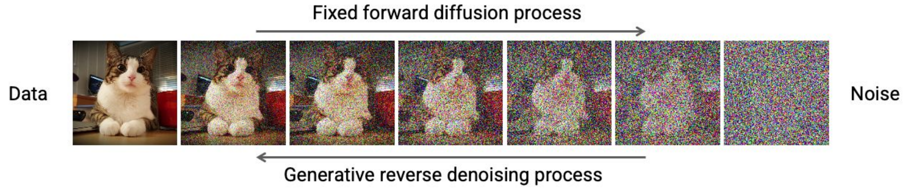
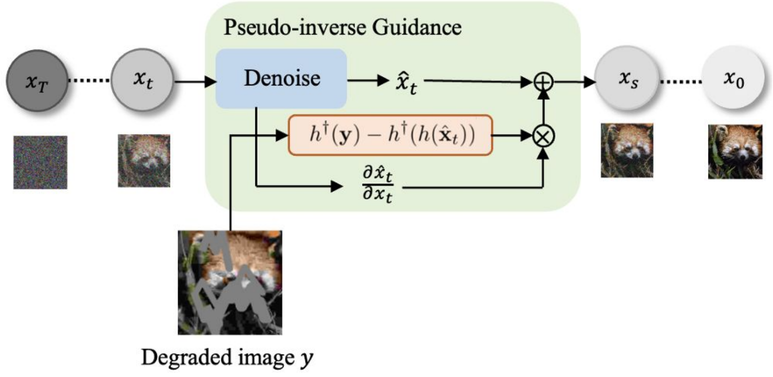

Clean image
Forward Process $q(x_t|x_{t-1})$
Gaussian noise
x_0
x_{t-1}
x_t
x_T
Reverse Process $p_\theta(x_{t-1}|x_t)$

Image credit: https://arxiv.org/abs/2308.09388

Zero-shot


Fixed forward diffusion process
Data
Noise
Generative reverse denoising process


Pseudo-inverse Guidance
x_T
x_t
Denoise
$\hat{x}_t$
$h^\dagger(\mathbf{y}) - h^\dagger(h(\hat{\mathbf{x}}_t))$
$\frac{\partial \hat{x}_t}{\partial x_t}$
x_s
x_0
Degraded image y

Focus: IPs with pretrained diffusion models

$$\text{(Reverse SDE for DDPM)}\ d\boldsymbol{x} = -\beta_t \left[\boldsymbol{x}/2 + \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})\right] dt + \sqrt{\beta_t} d\overline{\boldsymbol{w}}$$

$$\Downarrow$$

Think of **conditional score function**

$$\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}|\boldsymbol{y}) = \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}) + \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{y}|\boldsymbol{x})$$

$$\Downarrow$$

Conditional reverse SDE

$$d\boldsymbol{x} = \left[-\beta_t/2 \cdot \boldsymbol{x} - \beta_t(\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}) + \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{y}|\boldsymbol{x}))\right] dt + \sqrt{\beta_t} d\overline{\boldsymbol{w}}$$

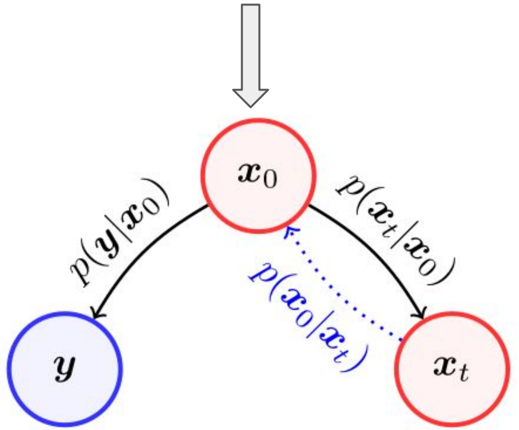
Coping with conditional score function

$$\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}|\boldsymbol{y}) = \boxed{\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})} + \boxed{\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{y}|\boldsymbol{x})}$$

$$\cong \boldsymbol{\varepsilon}_{\boldsymbol{\theta}}^{(t)}(\boldsymbol{x})$$

$$p_t(\boldsymbol{y}|\boldsymbol{x}(t))$$

$$\cong p_t(\boldsymbol{y}|\widehat{\boldsymbol{x}}(0)[\boldsymbol{x}(t)])$$



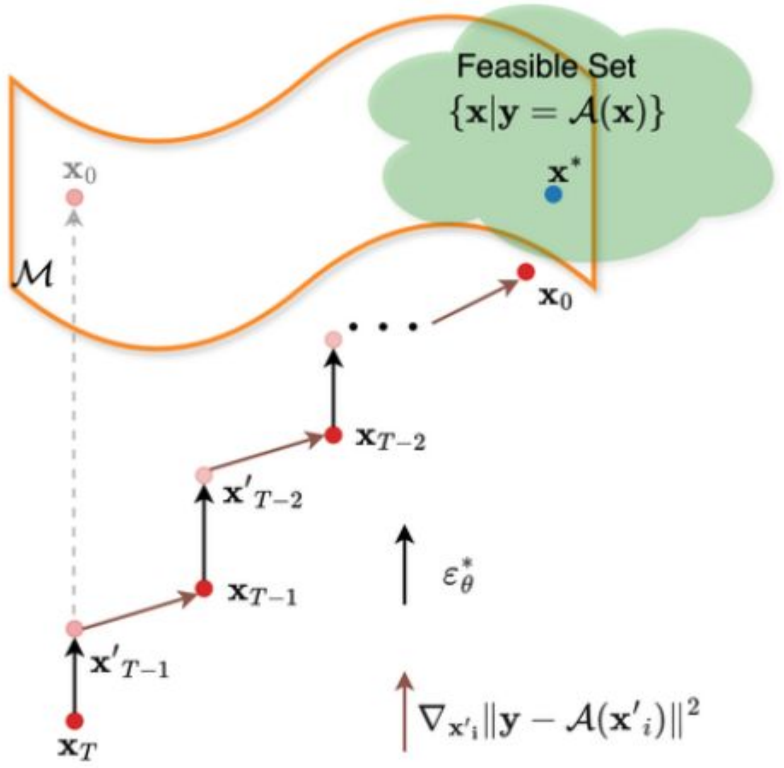
$\boldsymbol{x}_0$
$p(\boldsymbol{y}|\boldsymbol{x}_0)$
$p(\boldsymbol{x}_t|\boldsymbol{x}_0)$
$p(\boldsymbol{x}_0|\boldsymbol{x}_t)$
$\boldsymbol{y}$
$\boldsymbol{x}_t$

Figure 2: Probabilistic graph. Black solid line: tractable, blue dotted line: intractable in general.

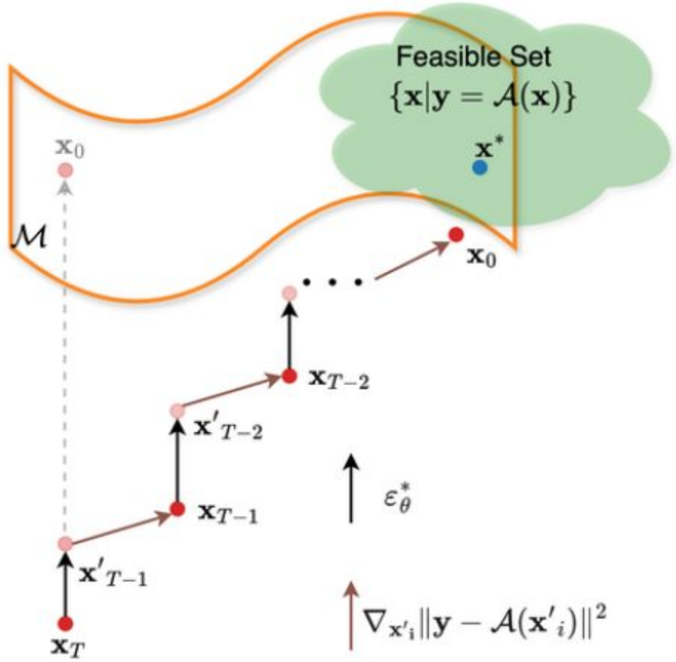
Interleaving methods

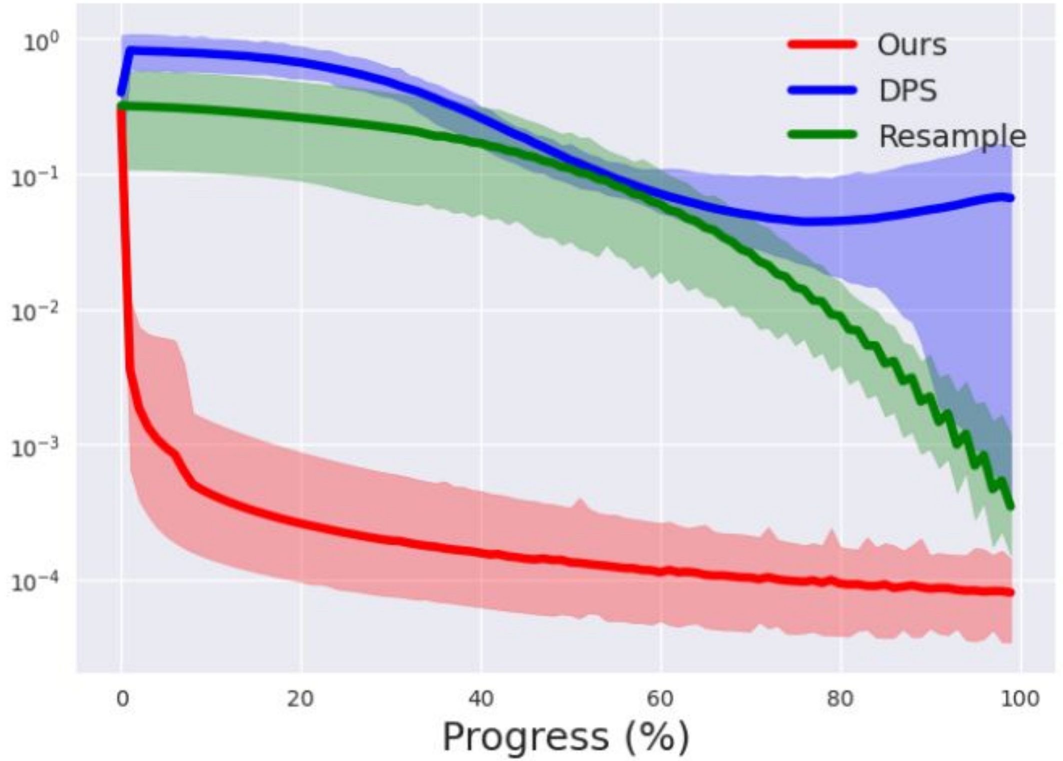
Algorithm 1 Template for interleaving methods

```
Input: # Diffusion steps $T$, measurement $\boldsymbol{y}$
1: $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
2: for $i = T - 1$ to $0$ do
3:     $\hat{\boldsymbol{s}} \leftarrow \boldsymbol{\varepsilon}_{\boldsymbol{\theta}}^{(i)}(\boldsymbol{x}_i)$
4:     $\hat{\boldsymbol{x}}_0 \leftarrow \frac{1}{\sqrt{\bar{\alpha}_i}}\left(\boldsymbol{x}_i - \sqrt{1-\bar{\alpha}_i}\,\hat{\boldsymbol{s}}\right)$
5:     $\boldsymbol{x}'_{i-1} \leftarrow$ DDIM reverse with $\hat{\boldsymbol{x}}_0$ and $\hat{\boldsymbol{s}}$
6:     $\boldsymbol{x}_{i-1} \leftarrow$ (Approximately) Projection [39, 30, 33, 32, 40, 41, 34] or gradient update [20, 28, 19, 21, 29, 27, 26] with $\hat{\boldsymbol{x}}_0$ and $\boldsymbol{x}'_{i-1}$ to get closer to $\{\boldsymbol{x} | \boldsymbol{y} = \mathcal{A}(\boldsymbol{x})\}$
7: end for
Output: Recovered object $\boldsymbol{x}_0$
```

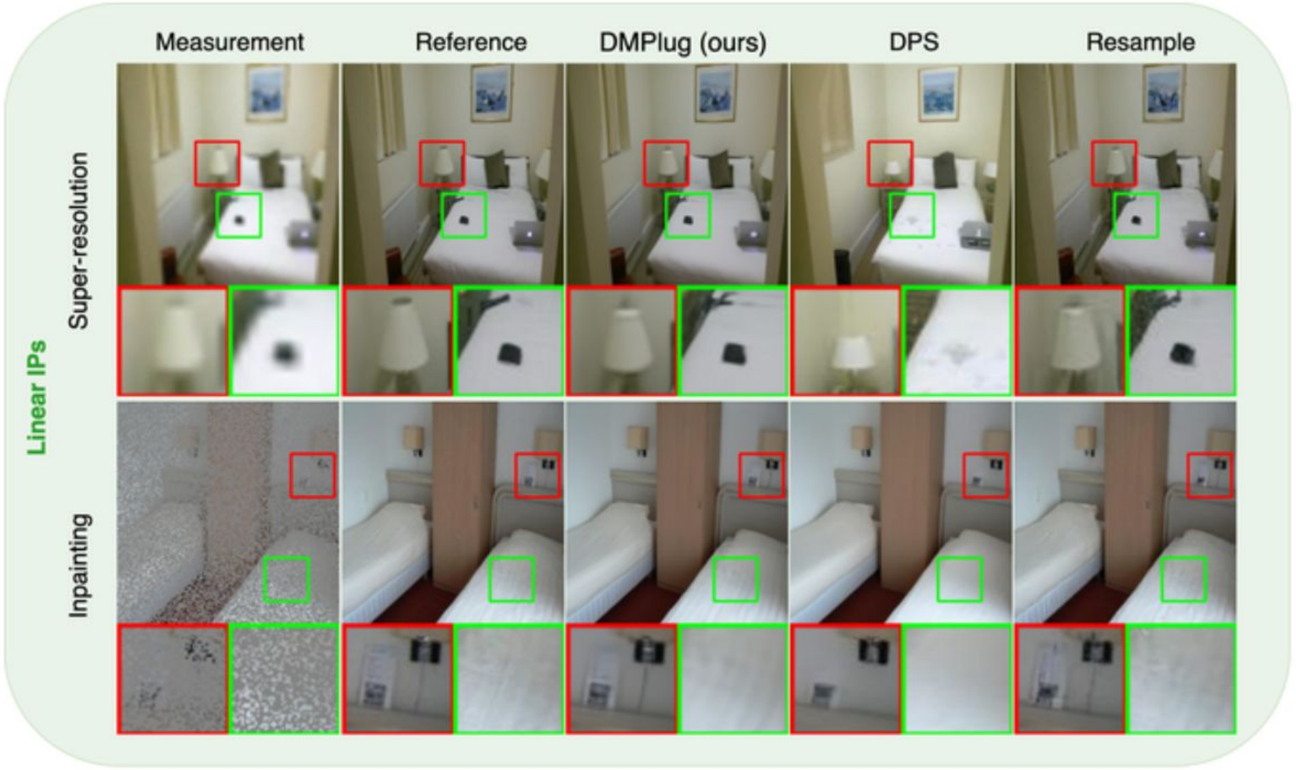


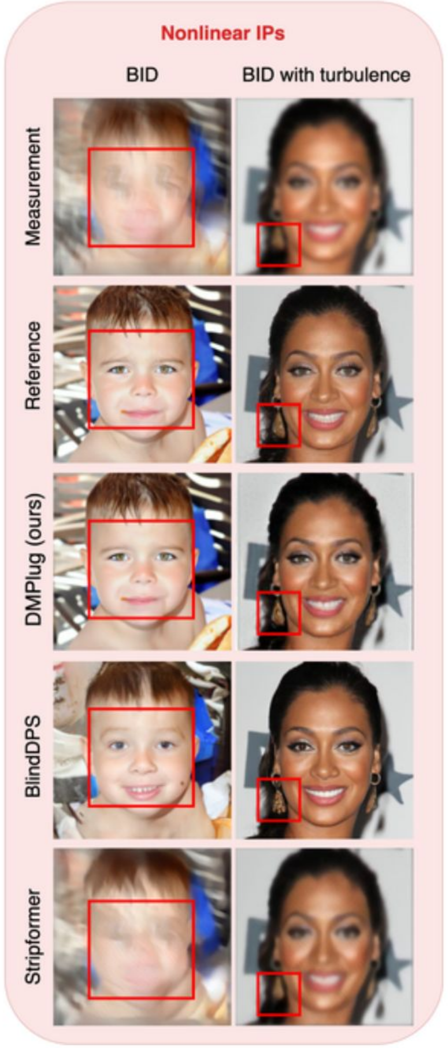
Issue I: Measurement feasibility



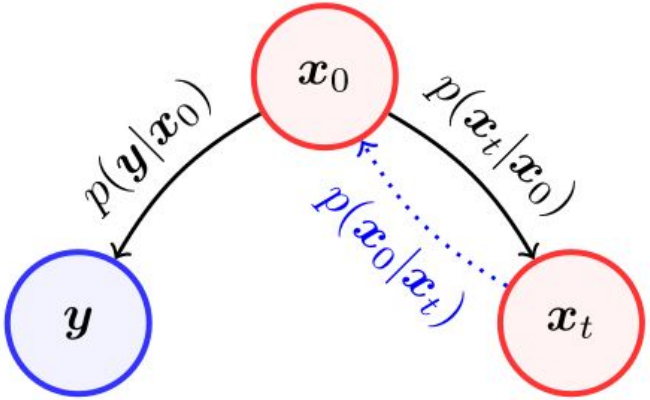


Issue 2: Manifold feasibility





Issue 3: Robustness to unknown noise



$\boldsymbol{x}_0$
$\boldsymbol{y}$
$\boldsymbol{x}_t$
$p(\boldsymbol{y}|\boldsymbol{x}_0)$
$p(\boldsymbol{x}_t|\boldsymbol{x}_0)$
$p(\boldsymbol{x}_0|\boldsymbol{x}_t)$

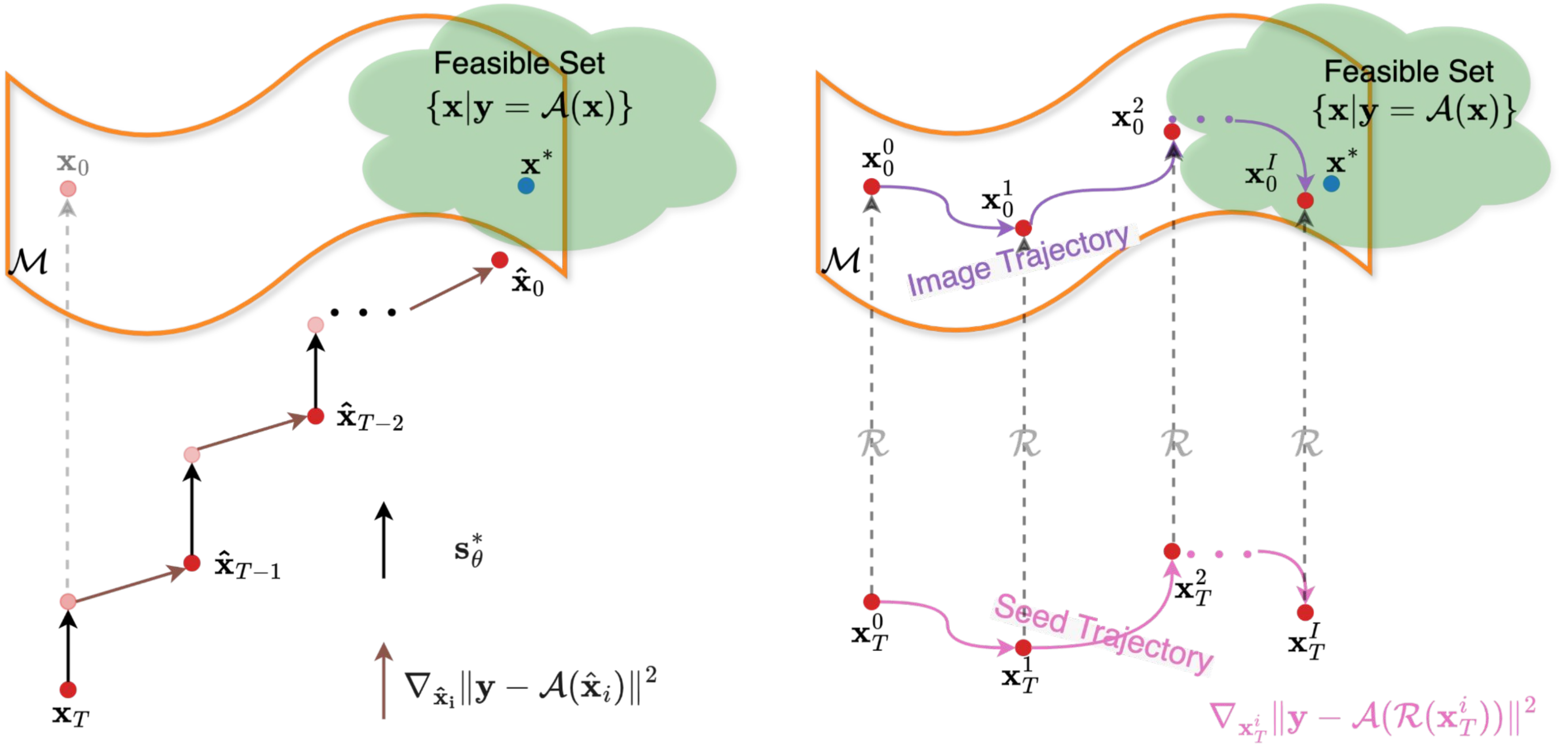
Figure 2: Probabilistic graph. Black solid line: tractable, blue dotted line: intractable in general.

Algorithm 1 DPS - Gaussian

```
Require: $N, \boldsymbol{y}, \{\zeta_i\}_{i=1}^N, \{\tilde{\sigma}_i\}_{i=1}^N$
1: $\boldsymbol{x}_N \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
2: for $i = N-1$ to $0$ do
3:     $\hat{\boldsymbol{s}} \leftarrow \boldsymbol{s}_\theta(\boldsymbol{x}_i, i)$
4:     $\hat{\boldsymbol{x}}_0 \leftarrow \frac{1}{\sqrt{\bar{\alpha}_i}}(\boldsymbol{x}_i + (1-\bar{\alpha}_i)\hat{\boldsymbol{s}})$
5:     $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
6:     $\boldsymbol{x}'_{i-1} \leftarrow \frac{\sqrt{\alpha_i}(1-\bar{\alpha}_{i-1})}{1-\bar{\alpha}_i}\boldsymbol{x}_i + \frac{\sqrt{\bar{\alpha}_{i-1}}\beta_i}{1-\bar{\alpha}_i}\hat{\boldsymbol{x}}_0 + \tilde{\sigma}_i\boldsymbol{z}$
7:     $\boldsymbol{x}_{i-1} \leftarrow \boldsymbol{x}'_{i-1} - \zeta_i \nabla_{\boldsymbol{x}_i}\|\boldsymbol{y} - \mathcal{A}(\hat{\boldsymbol{x}}_0)\|_2^2$
8: end for
9: return $\hat{\mathbf{x}}_0$
```

depending on noise level

Our solution: DMPlug



Our solution: DMPlug

Viewing the reverse process as a function $\mathcal{R}$

$$\mathcal{R} = g_{\boldsymbol{\varepsilon}_{\boldsymbol{\theta}}^{(0)}} \circ g_{\boldsymbol{\varepsilon}_{\boldsymbol{\theta}}^{(1)}} \circ \cdots \circ g_{\boldsymbol{\varepsilon}_{\boldsymbol{\theta}}^{(T-2)}} \circ g_{\boldsymbol{\varepsilon}_{\boldsymbol{\theta}}^{(T-1)}}. \qquad (\circ \text{ means function composition})$$

$$(\textbf{DMPlug}) \ \boldsymbol{z}^* \in \arg\min_{\boldsymbol{z}} \boxed{\ell(\boldsymbol{y}, \mathcal{A}(\boxed{\mathcal{R}(\boldsymbol{z})}))} + \Omega(\mathcal{R}(\boldsymbol{z})), \qquad \boldsymbol{x}^* = \mathcal{R}(\boldsymbol{z}^*).$$

Measurement feasibility

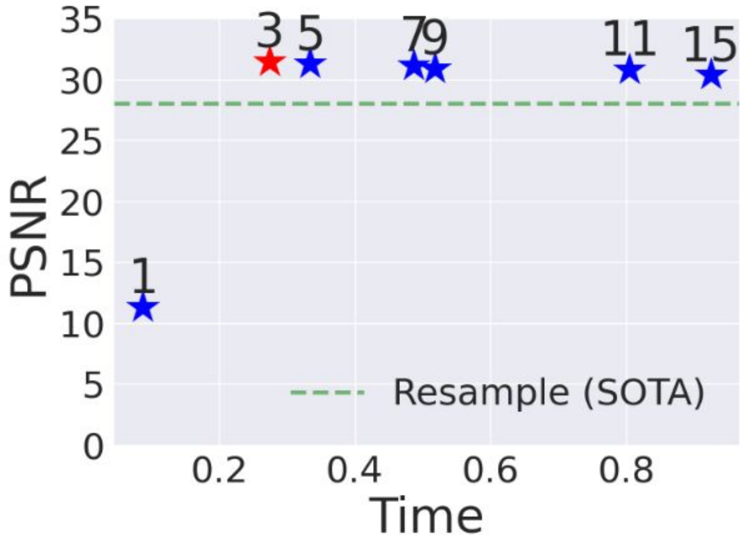
Manifold feasibility

Overcoming the computational bottleneck

$$\mathcal{R} = g_{\boldsymbol{\varepsilon}_{\boldsymbol{\theta}}^{(0)}} \circ g_{\boldsymbol{\varepsilon}_{\boldsymbol{\theta}}^{(1)}} \circ \cdots \circ g_{\boldsymbol{\varepsilon}_{\boldsymbol{\theta}}^{(T-2)}} \circ g_{\boldsymbol{\varepsilon}_{\boldsymbol{\theta}}^{(T-1)}}. \qquad (\circ \text{ means function composition})$$

$$(\textbf{DMPlug})\ \boldsymbol{z}^* \in \arg\min_{\boldsymbol{z}}\ \ell(\boldsymbol{y}, \mathcal{A}(\mathcal{R}(\boldsymbol{z}))) + \Omega(\mathcal{R}(\boldsymbol{z})), \qquad \boldsymbol{x}^* = \mathcal{R}(\boldsymbol{z}^*).$$

Issue: T blocks of DNNs involved, and we have to back-propagate through it



On linear IPs

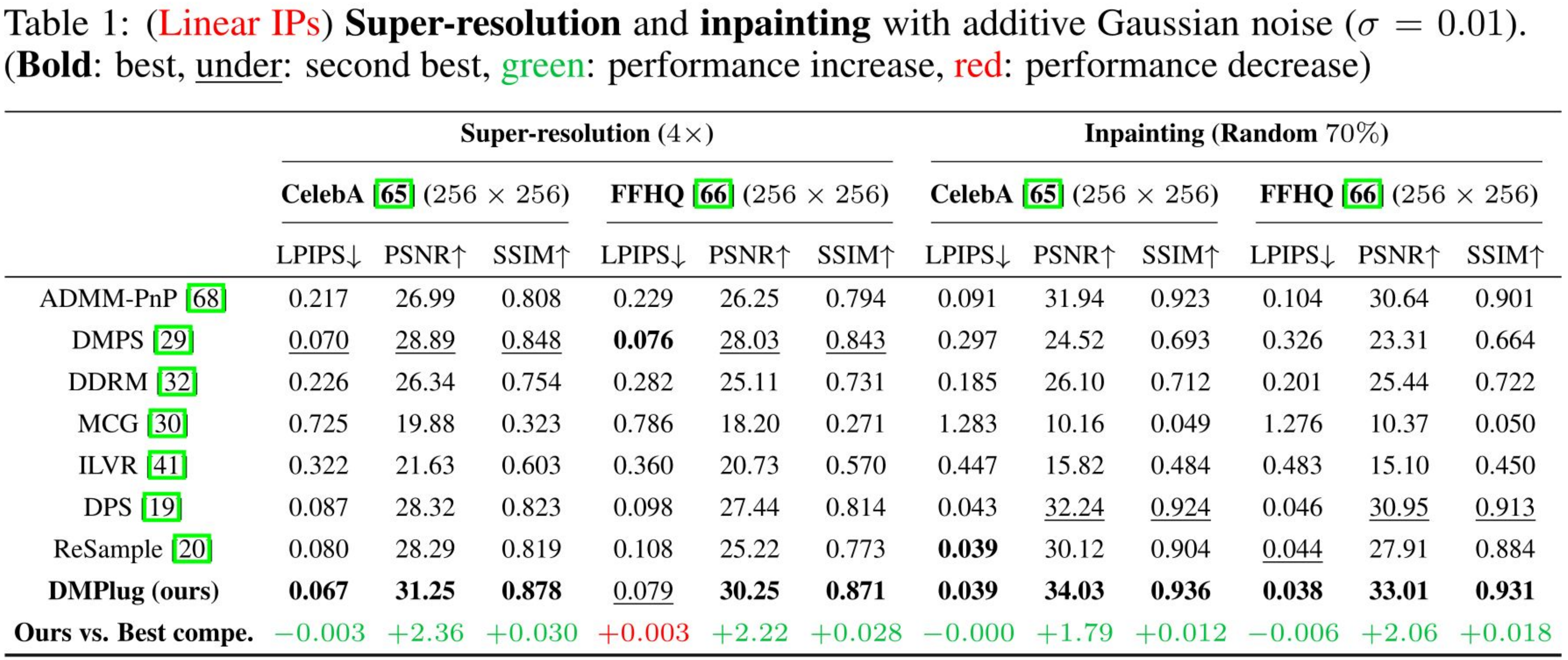


Table 1: (Linear IPs) **Super-resolution** and **inpainting** with additive Gaussian noise ($\sigma = 0.01$). (**Bold**: best, under: second best, green: performance increase, red: performance decrease)

	Super-resolution (4×)						**Inpainting (Random** 70%)					
	CelebA [65] (256 × 256)			**FFHQ** [66] (256 × 256)			**CelebA** [65] (256 × 256)			**FFHQ** [66] (256 × 256)		
	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑
ADMM-PnP [68]	0.217	26.99	0.808	0.229	26.25	0.794	0.091	31.94	0.923	0.104	30.64	0.901
DMPS [29]	0.070	28.89	0.848	**0.076**	28.03	0.843	0.297	24.52	0.693	0.326	23.31	0.664
DDRM [32]	0.226	26.34	0.754	0.282	25.11	0.731	0.185	26.10	0.712	0.201	25.44	0.722
MCG [30]	0.725	19.88	0.323	0.786	18.20	0.271	1.283	10.16	0.049	1.276	10.37	0.050
ILVR [41]	0.322	21.63	0.603	0.360	20.73	0.570	0.447	15.82	0.484	0.483	15.10	0.450
DPS [19]	0.087	28.32	0.823	0.098	27.44	0.814	0.043	32.24	0.924	0.046	30.95	0.913
ReSample [20]	0.080	28.29	0.819	0.108	25.22	0.773	**0.039**	30.12	0.904	0.044	27.91	0.884
DMPlug (ours)	**0.067**	**31.25**	**0.878**	0.079	**30.25**	**0.871**	**0.039**	**34.03**	**0.936**	**0.038**	**33.01**	**0.931**
Ours vs. Best compe.	−0.003	+2.36	+0.030	+0.003	+2.22	+0.028	−0.000	+1.79	+0.012	−0.006	+2.06	+0.018

On nonlinear IPs

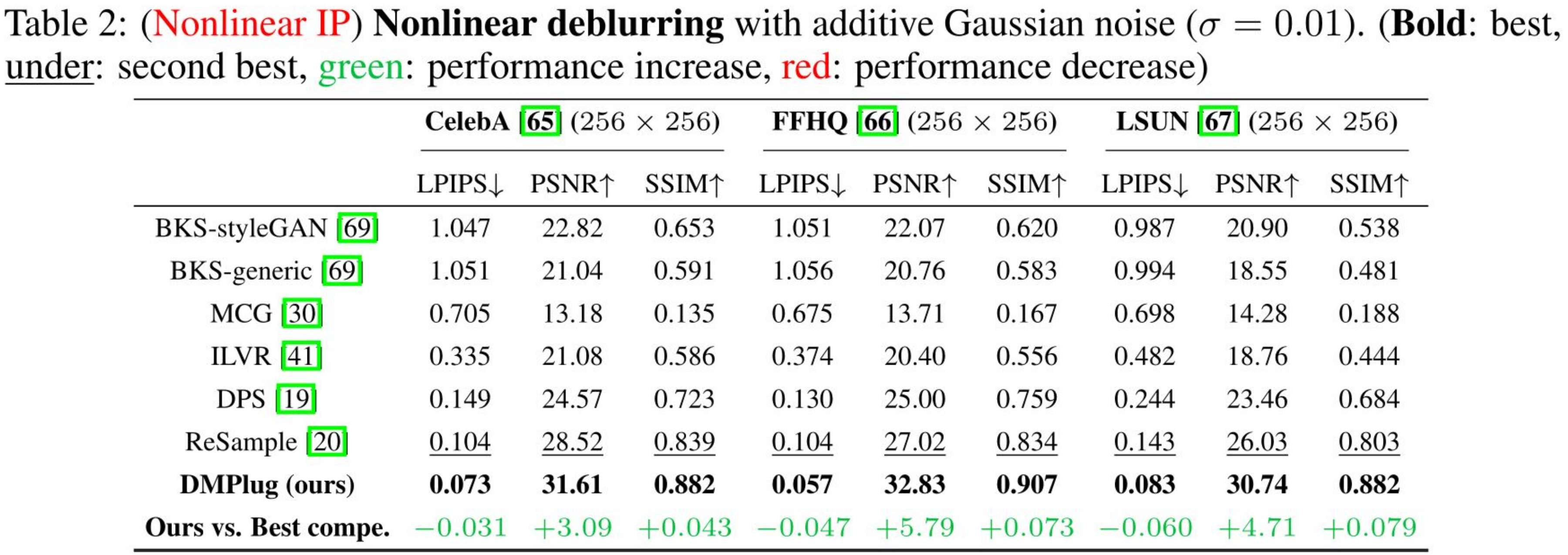


Table 2: (Nonlinear IP) **Nonlinear deblurring** with additive Gaussian noise ($\sigma = 0.01$). (**Bold**: best, under: second best, green: performance increase, red: performance decrease)

	CelebA [65] (256×256)			**FFHQ** [66] (256×256)			**LSUN** [67] (256×256)		
	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑
BKS-styleGAN [69]	1.047	22.82	0.653	1.051	22.07	0.620	0.987	20.90	0.538
BKS-generic [69]	1.051	21.04	0.591	1.056	20.76	0.583	0.994	18.55	0.481
MCG [30]	0.705	13.18	0.135	0.675	13.71	0.167	0.698	14.28	0.188
ILVR [41]	0.335	21.08	0.586	0.374	20.40	0.556	0.482	18.76	0.444
DPS [19]	0.149	24.57	0.723	0.130	25.00	0.759	0.244	23.46	0.684
ReSample [20]	0.104	28.52	0.839	0.104	27.02	0.834	0.143	26.03	0.803
DMPlug (ours)	**0.073**	**31.61**	**0.882**	**0.057**	**32.83**	**0.907**	**0.083**	**30.74**	**0.882**
Ours vs. Best compe.	-0.031	$+3.09$	$+0.043$	-0.047	$+5.79$	$+0.073$	-0.060	$+4.71$	$+0.079$

More on nonlinear IPs

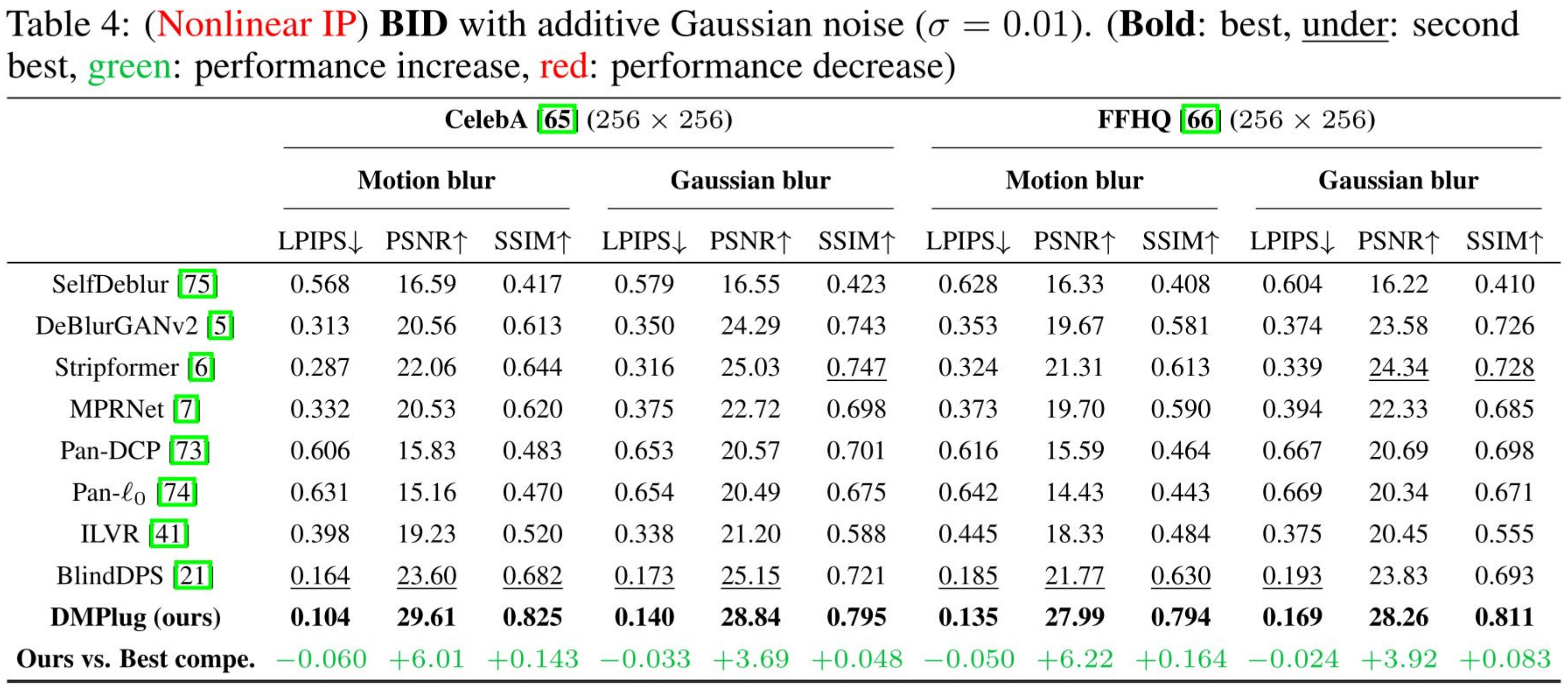
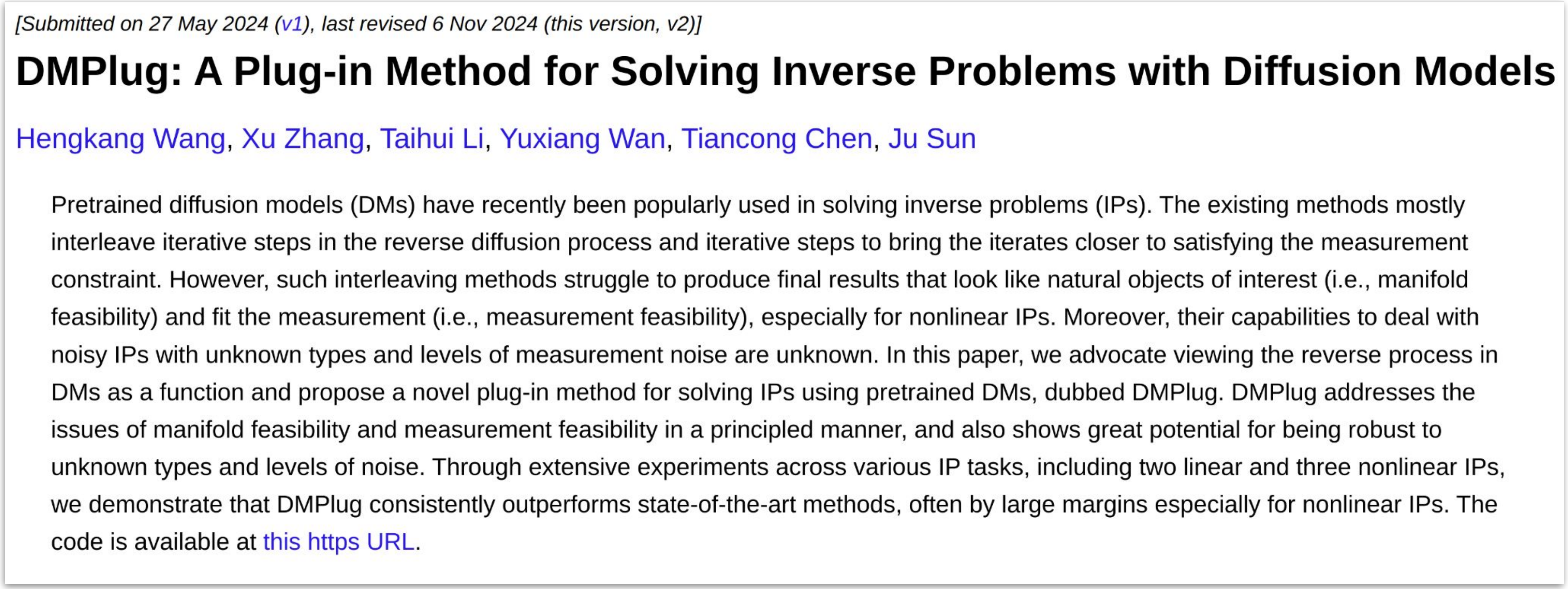


Table 4: (Nonlinear IP) **BID** with additive Gaussian noise ($\sigma = 0.01$). (**Bold**: best, under: second best, green: performance increase, red: performance decrease)

	CelebA [65] (256×256)						**FFHQ** [66] (256×256)					
	Motion blur			**Gaussian blur**			**Motion blur**			**Gaussian blur**		
	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑
SelfDeblur [75]	0.568	16.59	0.417	0.579	16.55	0.423	0.628	16.33	0.408	0.604	16.22	0.410
DeBlurGANv2 [5]	0.313	20.56	0.613	0.350	24.29	0.743	0.353	19.67	0.581	0.374	23.58	0.726
Stripformer [6]	0.287	22.06	0.644	0.316	25.03	0.747	0.324	21.31	0.613	0.339	24.34	0.728
MPRNet [7]	0.332	20.53	0.620	0.375	22.72	0.698	0.373	19.70	0.590	0.394	22.33	0.685
Pan-DCP [73]	0.606	15.83	0.483	0.653	20.57	0.701	0.616	15.59	0.464	0.667	20.69	0.698
Pan-ℓ_0 [74]	0.631	15.16	0.470	0.654	20.49	0.675	0.642	14.43	0.443	0.669	20.34	0.671
ILVR [41]	0.398	19.23	0.520	0.338	21.20	0.588	0.445	18.33	0.484	0.375	20.45	0.555
BlindDPS [21]	0.164	23.60	0.682	0.173	25.15	0.721	0.185	21.77	0.630	0.193	23.83	0.693
DMPlug (ours)	**0.104**	**29.61**	**0.825**	**0.140**	**28.84**	**0.795**	**0.135**	**27.99**	**0.794**	**0.169**	**28.26**	**0.811**
Ours vs. Best compe.	−0.060	+6.01	+0.143	−0.033	+3.69	+0.048	−0.050	+6.22	+0.164	−0.024	+3.92	+0.083

More details



[Submitted on 27 May 2024 (v1), last revised 6 Nov 2024 (this version, v2)]

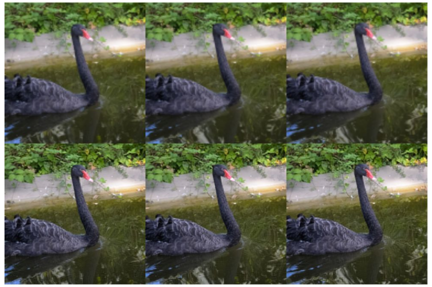
DMPlug: A Plug-in Method for Solving Inverse Problems with Diffusion Models

Hengkang Wang, Xu Zhang, Taihui Li, Yuxiang Wan, Tiancong Chen, Ju Sun

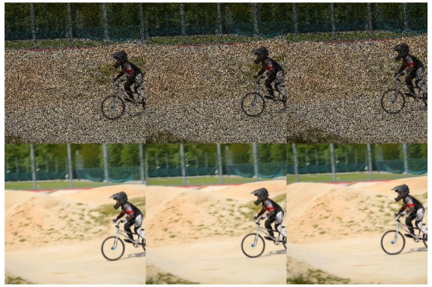
Pretrained diffusion models (DMs) have recently been popularly used in solving inverse problems (IPs). The existing methods mostly interleave iterative steps in the reverse diffusion process and iterative steps to bring the iterates closer to satisfying the measurement constraint. However, such interleaving methods struggle to produce final results that look like natural objects of interest (i.e., manifold feasibility) and fit the measurement (i.e., measurement feasibility), especially for nonlinear IPs. Moreover, their capabilities to deal with noisy IPs with unknown types and levels of measurement noise are unknown. In this paper, we advocate viewing the reverse process in DMs as a function and propose a novel plug-in method for solving IPs using pretrained DMs, dubbed DMPlug. DMPlug addresses the issues of manifold feasibility and measurement feasibility in a principled manner, and also shows great potential for being robust to unknown types and levels of noise. Through extensive experiments across various IP tasks, including two linear and three nonlinear IPs, we demonstrate that DMPlug consistently outperforms state-of-the-art methods, often by large margins especially for nonlinear IPs. The code is available at this https URL.

DMPlug for video restoration

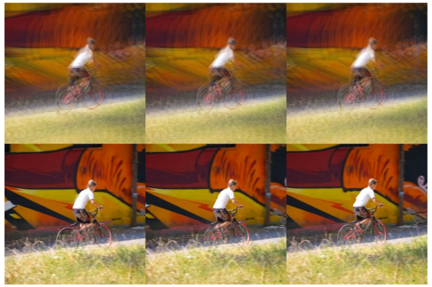
Super-resolution



Inpainting (random)



Temporal degradation



Temporal degradation + Motion deblur

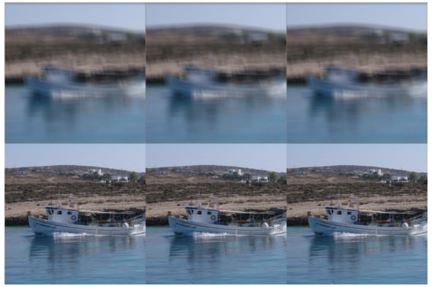
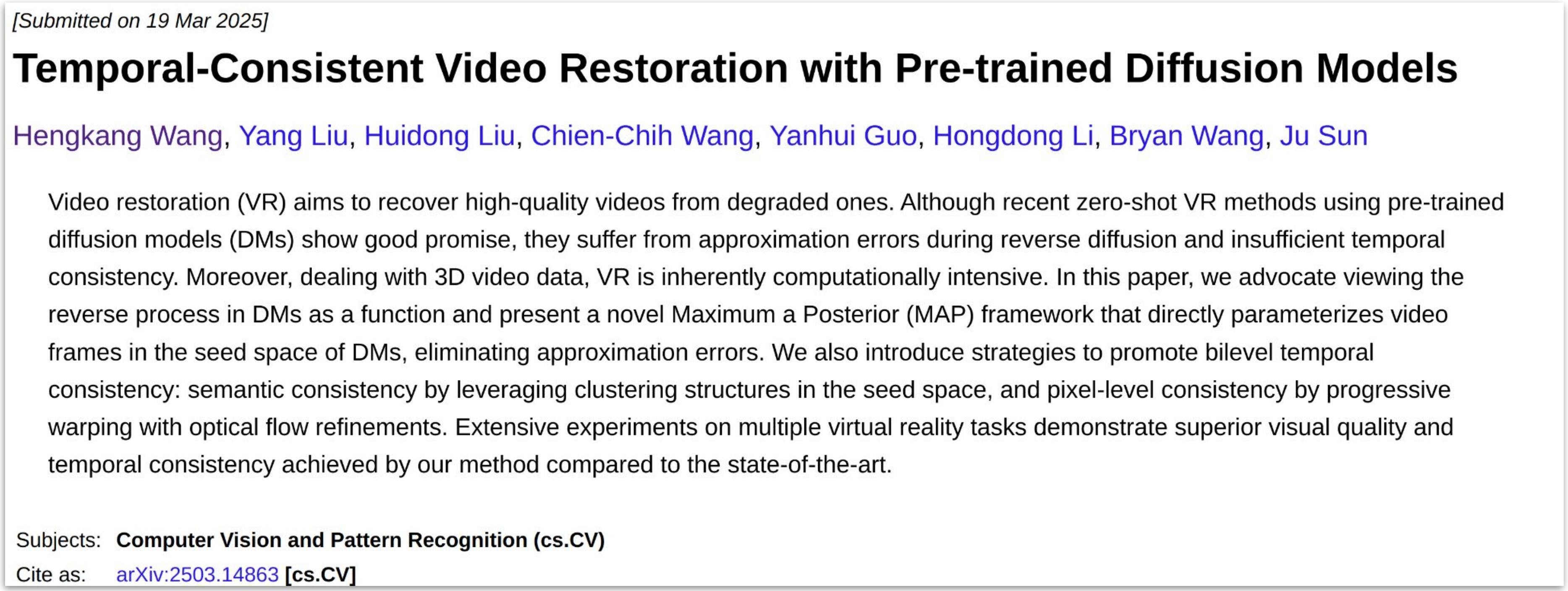


Table 7. Ablation study on two essential components for multi-level temporal consistency, performed on DAVIS dataset for video super-resolution $\times 4$. (**Bold**: best, under: second best)

Method	PSNR↑	SSIM↑	LPIPS↓	WE(10^{-2})↓
SOTA [9]	26.037	0.717	0.339	1.411
Base	24.701	0.612	0.366	1.398
Base + Semantic	26.098	0.703	0.410	1.057
Base + Pixel	27.141	0.736	**0.301**	0.943
Base + Semantic + Pixel	**27.959**	**0.790**	0.321	**0.725**

More details


[Submitted on 19 Mar 2025]

Temporal-Consistent Video Restoration with Pre-trained Diffusion Models

Hengkang Wang, Yang Liu, Huidong Liu, Chien-Chih Wang, Yanhui Guo, Hongdong Li, Bryan Wang, Ju Sun

Video restoration (VR) aims to recover high-quality videos from degraded ones. Although recent zero-shot VR methods using pre-trained diffusion models (DMs) show good promise, they suffer from approximation errors during reverse diffusion and insufficient temporal consistency. Moreover, dealing with 3D video data, VR is inherently computationally intensive. In this paper, we advocate viewing the reverse process in DMs as a function and present a novel Maximum a Posterior (MAP) framework that directly parameterizes video frames in the seed space of DMs, eliminating approximation errors. We also introduce strategies to promote bilevel temporal consistency: semantic consistency by leveraging clustering structures in the seed space, and pixel-level consistency by progressive warping with optical flow refinements. Extensive experiments on multiple virtual reality tasks demonstrate superior visual quality and temporal consistency achieved by our method compared to the state-of-the-art.

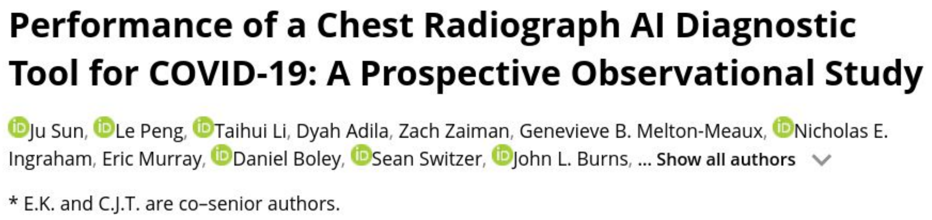
Subjects: **Computer Vision and Pattern Recognition (cs.CV)**
Cite as: arXiv:2503.14863 **[cs.CV]**

Today's talk: **toward trustworthy and efficient AI**

- **Robustness evaluation and selective classification**
- **Inverse problems with pretrained diffusion models**
- **AI for healthcare: handling data imbalance**

AI for healthcare

A COVID-19 diagnostic model deployed in 12 real-world hospitals

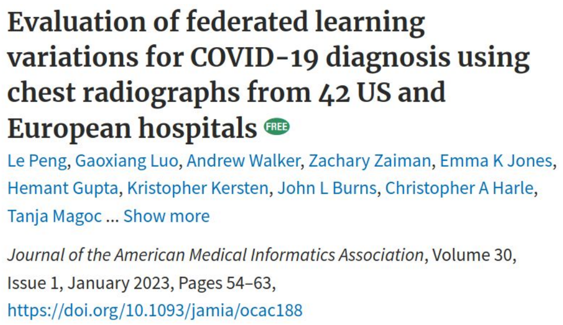


Performance of a Chest Radiograph AI Diagnostic Tool for COVID-19: A Prospective Observational Study

Ju Sun, Le Peng, Taihui Li, Dyah Adila, Zach Zaiman, Genevieve B. Melton-Meaux, Nicholas E. Ingraham, Eric Murray, Daniel Boley, Sean Switzer, John L. Burns, ... **Show all authors**

* E.K. and C.J.T. are co-senior authors.

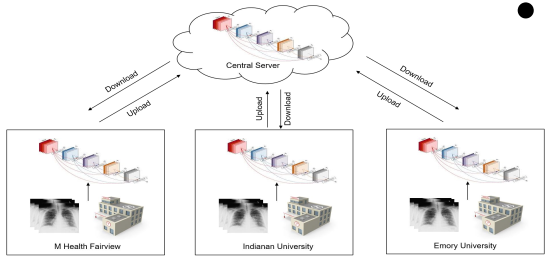
Among the first real-world federated learning implementation and validation



Evaluation of federated learning variations for COVID-19 diagnosis using chest radiographs from 42 US and European hospitals FREE

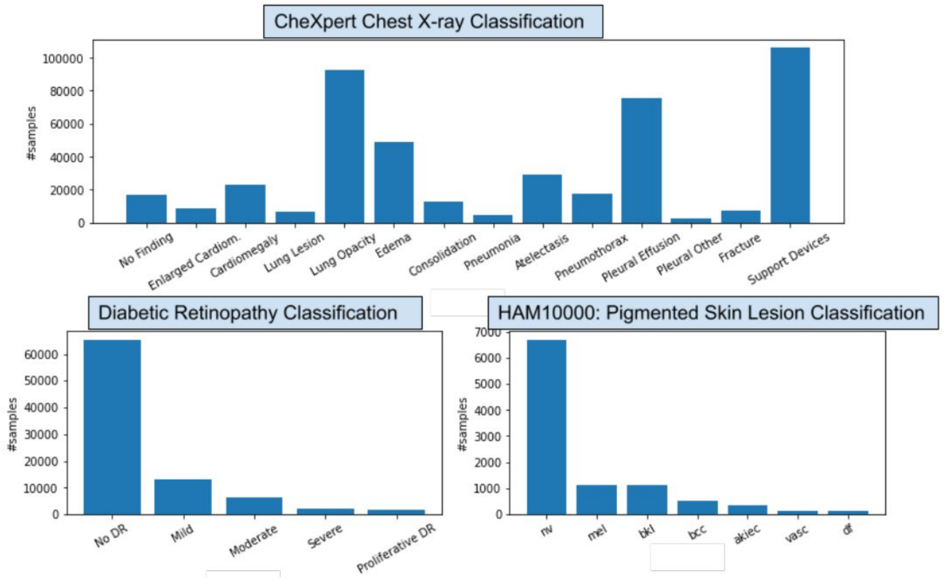
Le Peng, Gaoxiang Luo, Andrew Walker, Zachary Zaiman, Emma K Jones, Hemant Gupta, Kristopher Kersten, John L Burns, Christopher A Harle, Tanja Magoc ... Show more

Journal of the American Medical Informatics Association, Volume 30, Issue 1, January 2023, Pages 54–63, https://doi.org/10.1093/jamia/ocac188

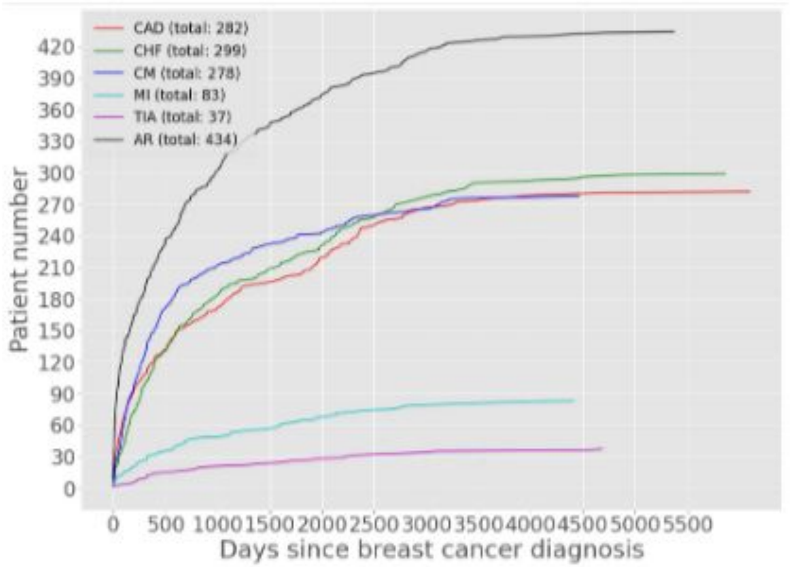


- **FL for medical NLP**
- **Liver cancer detection from ultrasound**
- **Tic detection from videos**
- **Rib fracture detection from x-rays**
- **Swallow study**
- **Hip fracture detection**
- **Audiogram image recognition**
- Etc

Addressing data inequality—imbalanced learning

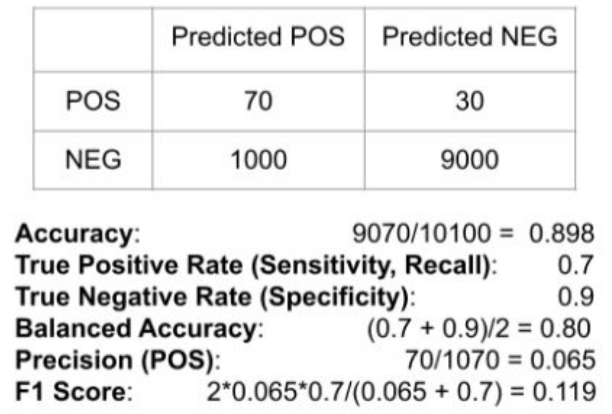


Imbalanced classification (IC)



Imbalanced regression (IR)

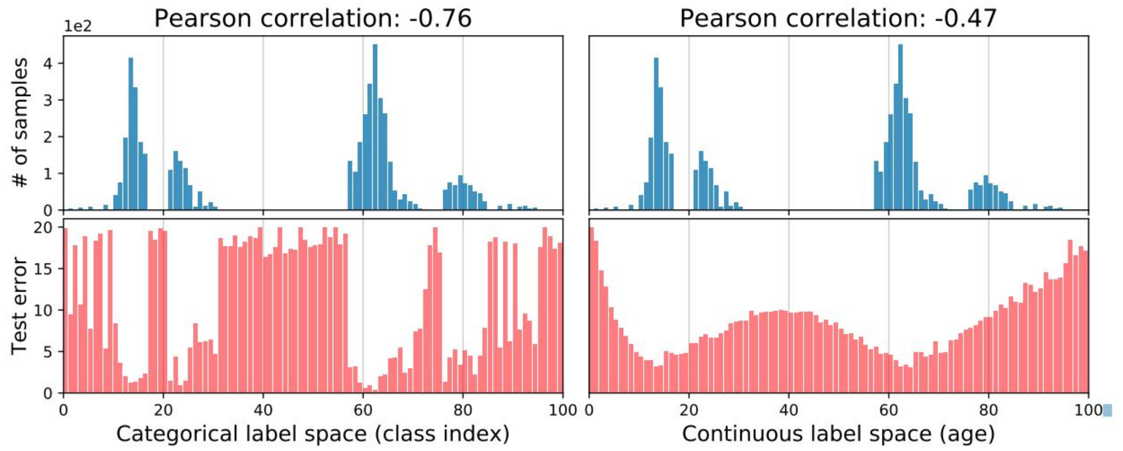
Why imbalance learning is challenging?



	Predicted POS	Predicted NEG
POS	70	30
NEG	1000	9000

Accuracy:	9070/10100 = 0.898
True Positive Rate (Sensitivity, Recall):	0.7
True Negative Rate (Specificity):	0.9
Balanced Accuracy:	(0.7 + 0.9)/2 = 0.80
Precision (POS):	70/1070 = 0.065
F1 Score:	2*0.065*0.7/(0.065 + 0.7) = 0.119

Figure 2: An example confusion table for binary classification, and the various associated performance metrics. POS: positive; NEG: negative.



(a) CIFAR-100 (subsampled) (b) IMDB-WIKI (subsampled)

Evaluation metrics ⇒ Learning goals matter!

Principled learning goals

fix precision, optimize recall (FPOR): $\max_{\boldsymbol{\theta},t} \ \mathrm{recall}(f_{\boldsymbol{\theta}}, t) \quad \mathrm{s.\,t.} \ \ \mathrm{precision}(f_{\boldsymbol{\theta}}, t) \geq \alpha,$

fix recall, optimize precision (FROP): $\max_{\boldsymbol{\theta},t} \ \mathrm{precision}_t \quad \mathrm{s.\,t.}\, \mathrm{recall}(f_{\boldsymbol{\theta}}, t) \geq \alpha,$

optimize F_β score (OFBS): $\max_{\boldsymbol{\theta},t} \ F_\beta(f_{\boldsymbol{\theta}}, t),$

optimize AP (OAP): $\max_{\boldsymbol{\theta}} \ \mathrm{AP}(f_{\boldsymbol{\theta}}).$

optimize multiclass performance (OMCP): $\max_{\boldsymbol{\theta},\boldsymbol{t}} \ \text{multiclass-metric}(f_{\boldsymbol{\theta}}, \boldsymbol{t}).$

optimize regression performance (OREGP): $\max_{\boldsymbol{\theta}} \ \text{regression-metric}(f_{\boldsymbol{\theta}});$

Brand-new ideas for dealing with indicator functions

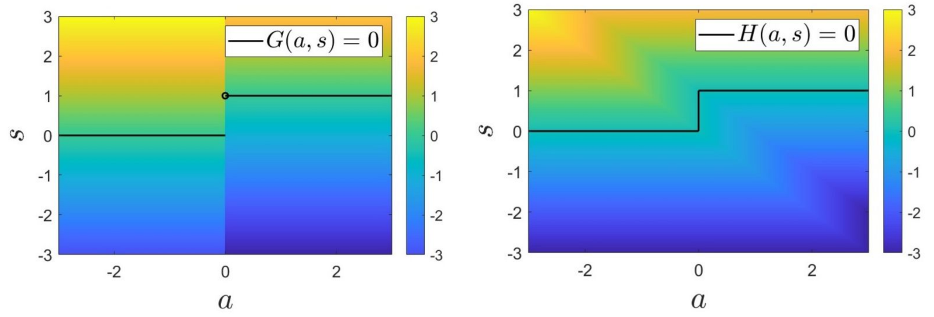
$$\max_{\boldsymbol{\theta},t} \; \frac{1}{N_+} \sum_{i \in \mathcal{P}} \mathbf{1}\{f_{\boldsymbol{\theta}}($$

Exact continuous reformulation of indicator function. Consider the following key observation:

$$s - \mathbf{1}\{a > 0\} \; = 0 \quad \overset{a \neq 0}{\Longleftrightarrow} \quad s + [s + a - 1]_+ - [s + a]_+ = 0, \tag{3.3}$$

where $s \in \mathbb{R}$, $a \in \mathbb{R} \setminus \{0\}$, $[\cdot]_+ \doteq \max\{\cdot, 0\}$. To verify the validity of (3.3), we present in Figure 3 visualizations of the function values of $G(a,s)$ and $H(a,s)$, along with their level sets at 0, where

$$G(a,s) \doteq s - \mathbf{1}\{a > 0\}, \quad H(a,s) \doteq s + [s + a - 1]_+ - [s + a]_+. \tag{3.4}$$



Consistently (substantially) better results

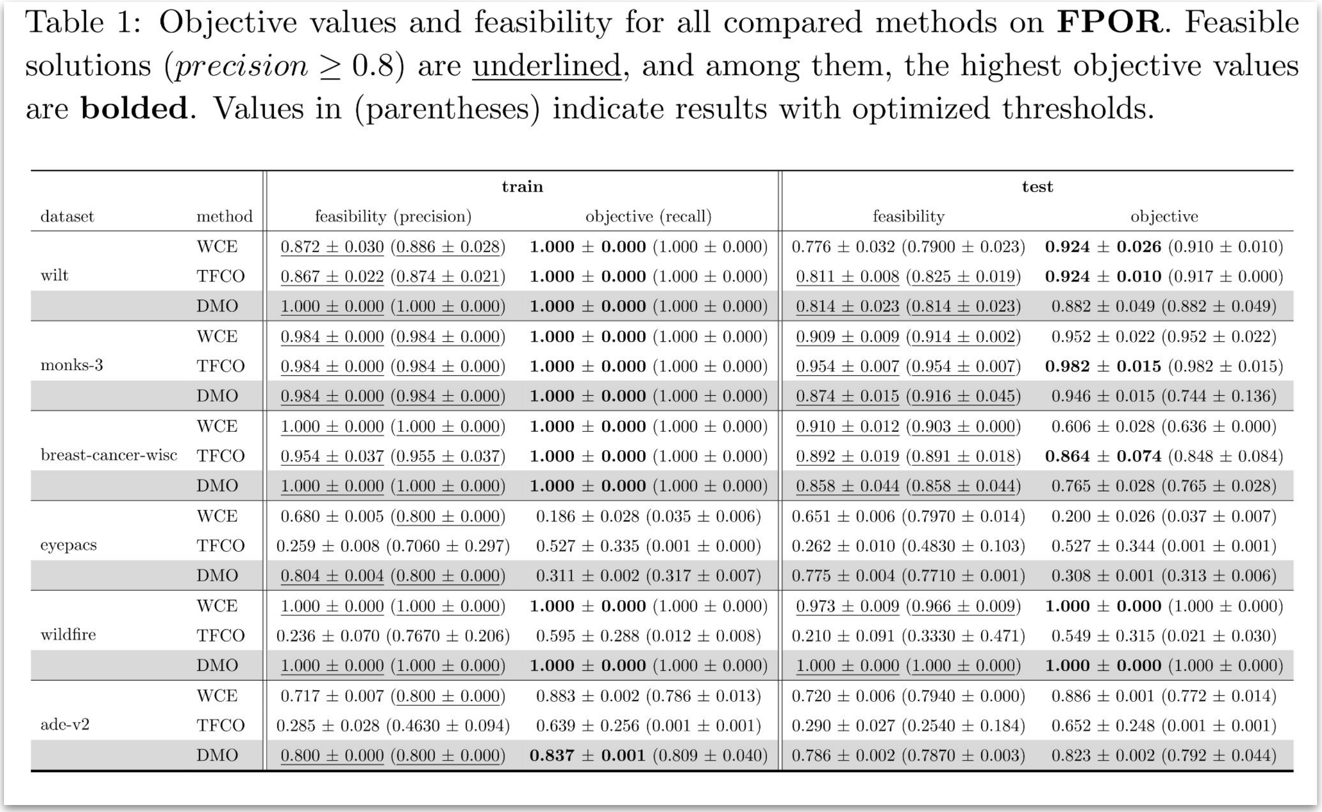
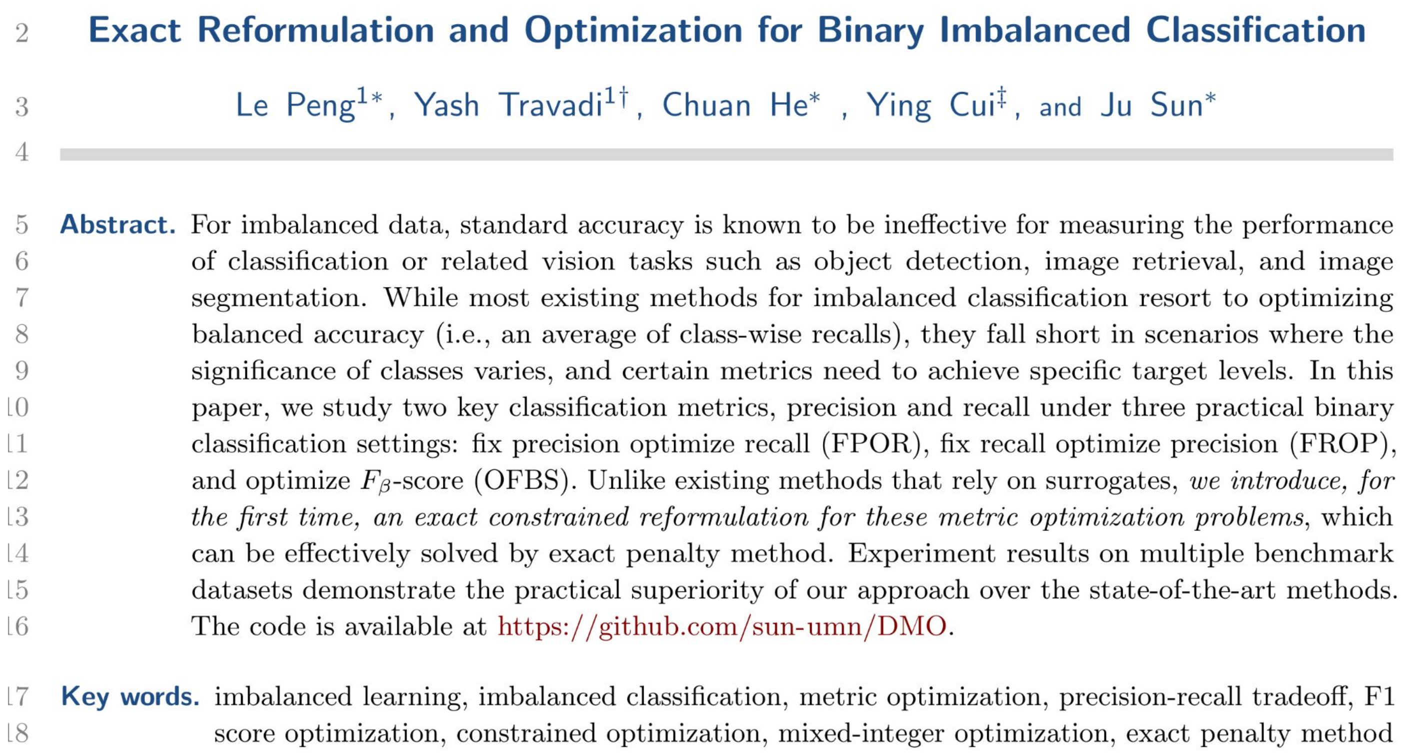


Table 1: Objective values and feasibility for all compared methods on **FPOR**. Feasible solutions (*precision* ≥ 0.8) are underlined, and among them, the highest objective values are **bolded**. Values in (parentheses) indicate results with optimized thresholds.

		train		test	
dataset	method	feasibility (precision)	objective (recall)	feasibility	objective
	WCE	0.872 ± 0.030 (0.886 ± 0.028)	**1.000** ± **0.000** (1.000 ± 0.000)	0.776 ± 0.032 (0.7900 ± 0.023)	**0.924** ± **0.026** (0.910 ± 0.010)
wilt	TFCO	0.867 ± 0.022 (0.874 ± 0.021)	**1.000** ± **0.000** (1.000 ± 0.000)	0.811 ± 0.008 (0.825 ± 0.019)	**0.924** ± **0.010** (0.917 ± 0.000)
	DMO	1.000 ± 0.000 (1.000 ± 0.000)	**1.000** ± **0.000** (1.000 ± 0.000)	0.814 ± 0.023 (0.814 ± 0.023)	0.882 ± 0.049 (0.882 ± 0.049)
	WCE	0.984 ± 0.000 (0.984 ± 0.000)	**1.000** ± **0.000** (1.000 ± 0.000)	0.909 ± 0.009 (0.914 ± 0.002)	0.952 ± 0.022 (0.952 ± 0.022)
monks-3	TFCO	0.984 ± 0.000 (0.984 ± 0.000)	**1.000** ± **0.000** (1.000 ± 0.000)	0.954 ± 0.007 (0.954 ± 0.007)	**0.982** ± **0.015** (0.982 ± 0.015)
	DMO	0.984 ± 0.000 (0.984 ± 0.000)	**1.000** ± **0.000** (1.000 ± 0.000)	0.874 ± 0.015 (0.916 ± 0.045)	0.946 ± 0.015 (0.744 ± 0.136)
	WCE	1.000 ± 0.000 (1.000 ± 0.000)	**1.000** ± **0.000** (1.000 ± 0.000)	0.910 ± 0.012 (0.903 ± 0.000)	0.606 ± 0.028 (0.636 ± 0.000)
breast-cancer-wisc	TFCO	0.954 ± 0.037 (0.955 ± 0.037)	**1.000** ± **0.000** (1.000 ± 0.000)	0.892 ± 0.019 (0.891 ± 0.018)	**0.864** ± **0.074** (0.848 ± 0.084)
	DMO	1.000 ± 0.000 (1.000 ± 0.000)	**1.000** ± **0.000** (1.000 ± 0.000)	0.858 ± 0.044 (0.858 ± 0.044)	0.765 ± 0.028 (0.765 ± 0.028)
	WCE	0.680 ± 0.005 (0.800 ± 0.000)	0.186 ± 0.028 (0.035 ± 0.006)	0.651 ± 0.006 (0.7970 ± 0.014)	0.200 ± 0.026 (0.037 ± 0.007)
eyepacs	TFCO	0.259 ± 0.008 (0.7060 ± 0.297)	0.527 ± 0.335 (0.001 ± 0.000)	0.262 ± 0.010 (0.4830 ± 0.103)	0.527 ± 0.344 (0.001 ± 0.001)
	DMO	0.804 ± 0.004 (0.800 ± 0.000)	0.311 ± 0.002 (0.317 ± 0.007)	0.775 ± 0.004 (0.7710 ± 0.001)	0.308 ± 0.001 (0.313 ± 0.006)
	WCE	1.000 ± 0.000 (1.000 ± 0.000)	**1.000** ± **0.000** (1.000 ± 0.000)	0.973 ± 0.009 (0.966 ± 0.009)	**1.000** ± **0.000** (1.000 ± 0.000)
wildfire	TFCO	0.236 ± 0.070 (0.7670 ± 0.206)	0.595 ± 0.288 (0.012 ± 0.008)	0.210 ± 0.091 (0.3330 ± 0.471)	0.549 ± 0.315 (0.021 ± 0.030)
	DMO	1.000 ± 0.000 (1.000 ± 0.000)	**1.000** ± **0.000** (1.000 ± 0.000)	1.000 ± 0.000 (1.000 ± 0.000)	**1.000** ± **0.000** (1.000 ± 0.000)
	WCE	0.717 ± 0.007 (0.800 ± 0.000)	0.883 ± 0.002 (0.786 ± 0.013)	0.720 ± 0.006 (0.7940 ± 0.000)	0.886 ± 0.001 (0.772 ± 0.014)
adc-v2	TFCO	0.285 ± 0.028 (0.4630 ± 0.094)	0.639 ± 0.256 (0.001 ± 0.001)	0.290 ± 0.027 (0.2540 ± 0.184)	0.652 ± 0.248 (0.001 ± 0.001)
	DMO	0.800 ± 0.000 (0.800 ± 0.000)	**0.837** ± **0.001** (0.809 ± 0.040)	0.786 ± 0.002 (0.7870 ± 0.003)	0.823 ± 0.002 (0.792 ± 0.044)

More details



Exact Reformulation and Optimization for Binary Imbalanced Classification

Le Peng[1*], Yash Travadi[1†], Chuan He[*], Ying Cui[‡], and Ju Sun[*]

Abstract. For imbalanced data, standard accuracy is known to be ineffective for measuring the performance of classification or related vision tasks such as object detection, image retrieval, and image segmentation. While most existing methods for imbalanced classification resort to optimizing balanced accuracy (i.e., an average of class-wise recalls), they fall short in scenarios where the significance of classes varies, and certain metrics need to achieve specific target levels. In this paper, we study two key classification metrics, precision and recall under three practical binary classification settings: fix precision optimize recall (FPOR), fix recall optimize precision (FROP), and optimize F_β-score (OFBS). Unlike existing methods that rely on surrogates, *we introduce, for the first time, an exact constrained reformulation for these metric optimization problems*, which can be effectively solved by exact penalty method. Experiment results on multiple benchmark datasets demonstrate the practical superiority of our approach over the state-of-the-art methods. The code is available at https://github.com/sun-umn/DMO.

Key words. imbalanced learning, imbalanced classification, metric optimization, precision-recall tradeoff, F1 score optimization, constrained optimization, mixed-integer optimization, exact penalty method

Today's talk: **toward trustworthy and efficient AI**

- General and Reliable **Robustness evaluation and** Improved and Robust **selective classification**

- Effective **Inverse problems with pretrained diffusion models**

- **AI for healthcare: handling data imbalance via** direct metric optimization

Thanks to

NSF

NATIONAL INSTITUTES OF HEALTH
NIH
UNITED STATES OF AMERICA

DEPARTMENT OF ENERGY
UNITED STATES OF AMERICA

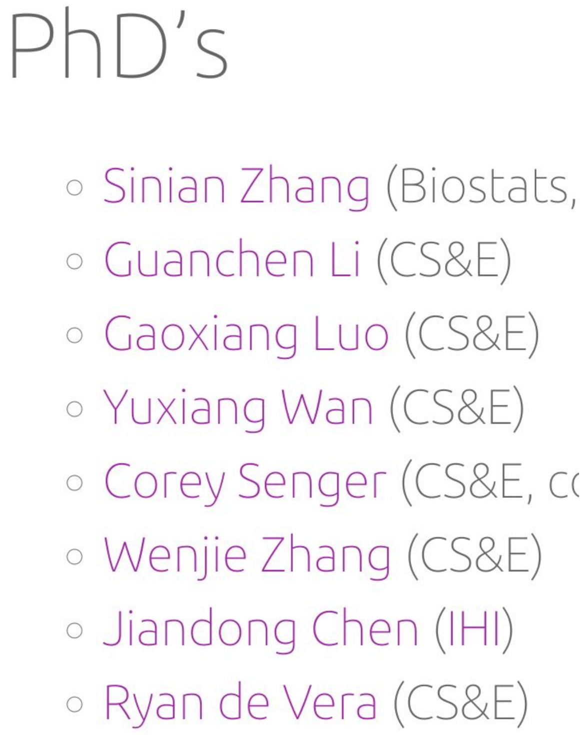
CISCO

DARPA

USDA
NIFA

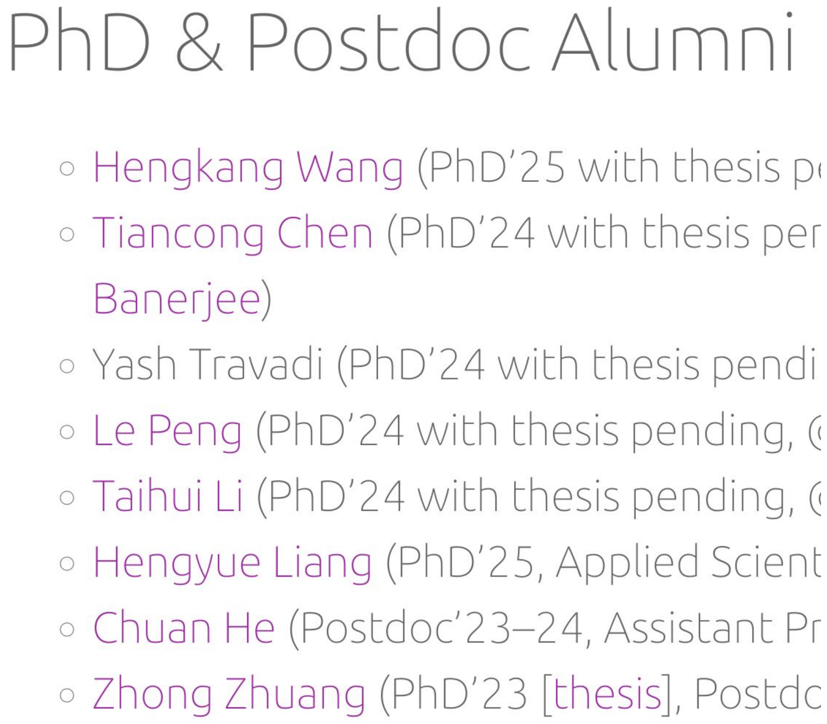
NVIDIA.

Thanks to



PhD’s

- Sinian Zhang (Biostats,
- Guanchen Li (CS&E)
- Gaoxiang Luo (CS&E)
- Yuxiang Wan (CS&E)
- Corey Senger (CS&E, c
- Wenjie Zhang (CS&E)
- Jiandong Chen (IHI)
- Ryan de Vera (CS&E)



PhD & Postdoc Alumni

- Hengkang Wang (PhD’25 with thesis p
- Tiancong Chen (PhD’24 with thesis pe Banerjee)
- Yash Travadi (PhD’24 with thesis pendi
- Le Peng (PhD’24 with thesis pending,
- Taihui Li (PhD’24 with thesis pending,
- Hengyue Liang (PhD’25, Applied Scient
- Chuan He (Postdoc’23–24, Assistant P
- Zhong Zhuang (PhD’23 [thesis], Postd