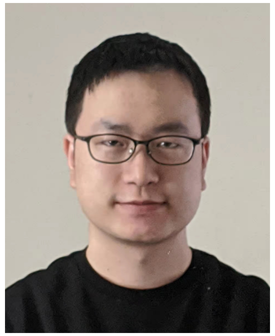
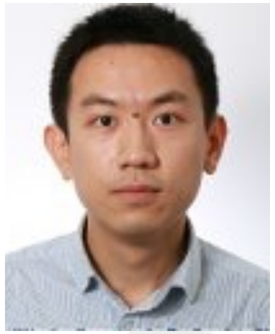
AIFARMS
Artificial Intelligence for Future Agricultural Resilience, Management, and Sustainability

Distribution-Informed Neural Networks for Domain Adaptation Regression





Jun Wu **Jingrui He** **Sheng Wang** **Kaiyu Guan** **Elizabeth Ainsworth**

University of Illinois Urbana-Champaign
{junwu3, jingrui, sheng12, kaiyug, ainswort}@illinois.edu

ILLINOIS

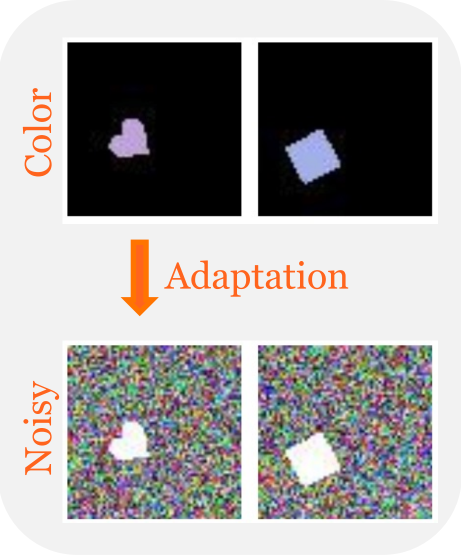
Domain Adaptation Regression

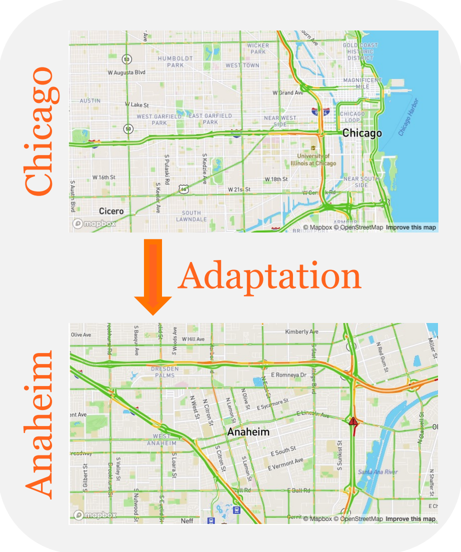
❑ Problem definition

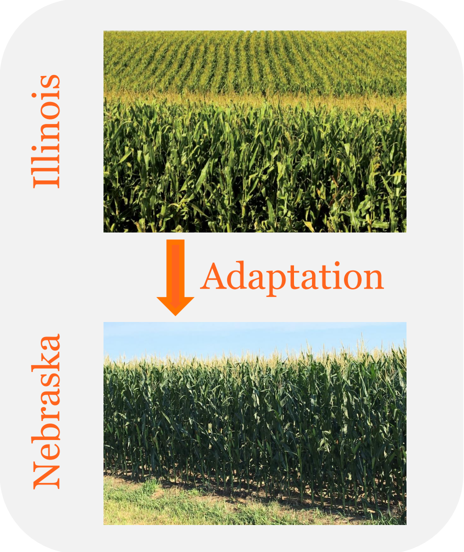
- Input: A source domain and a target domain
- Output: Prediction function on the target domain

❑ Applications

- **Computer vision**: Object localization
- **Natural language processing**: Sentiment analysis
- **Graph mining**: Traffic flow prediction
- **Agriculture analysis**: Plant phenotyping







Xinyang Chen, et al. "**Representation subspace distance for domain adaptation regression.**" In *ICML*. 2021.
Junteng Jia, et al. "**Residual correlation in graph neural network regression.**" In *KDD*. 2020.
Jun Wu et al. "**Adaptive transfer learning for plant phenotyping.**" In *MLCAS*. 2021

Distribution-Informed Neural Networks

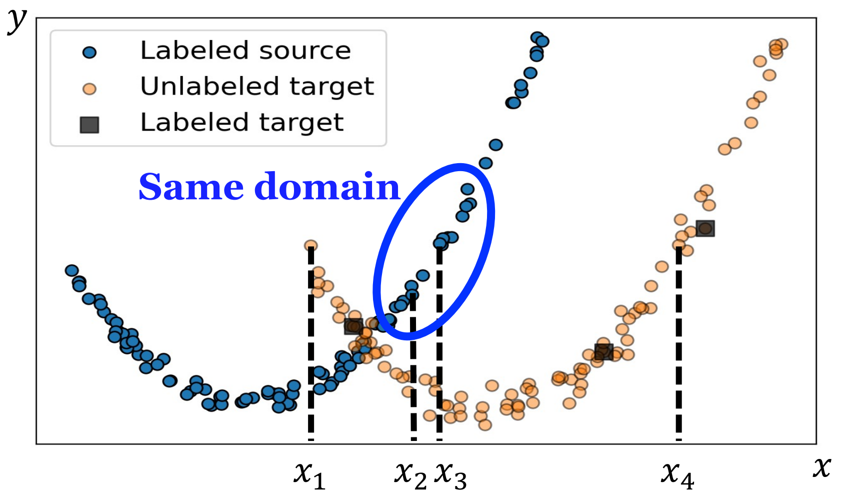
- **Motivation**
  - For standard neural network $f(\cdot)$ on a single domain

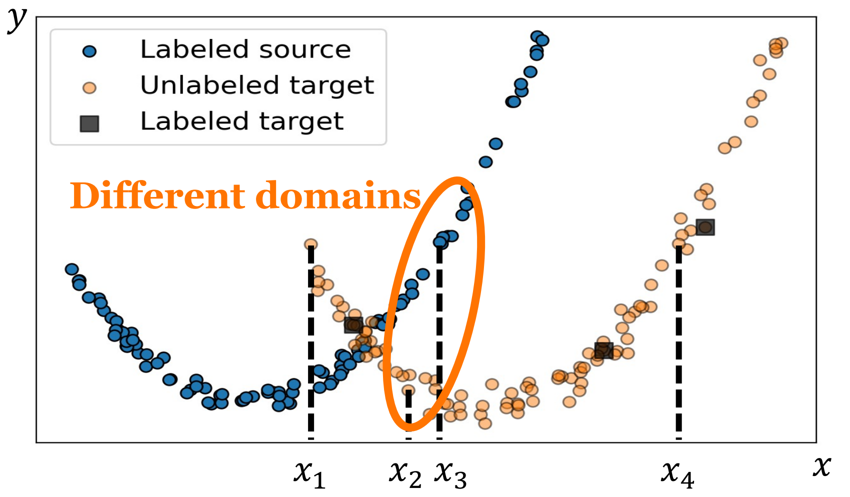
    $$f(x_2) \approx f(x_3) \text{ if } x_2 \approx x_3 \text{ (only source domain)}$$

  - Limitations on **heterogeneous domains**

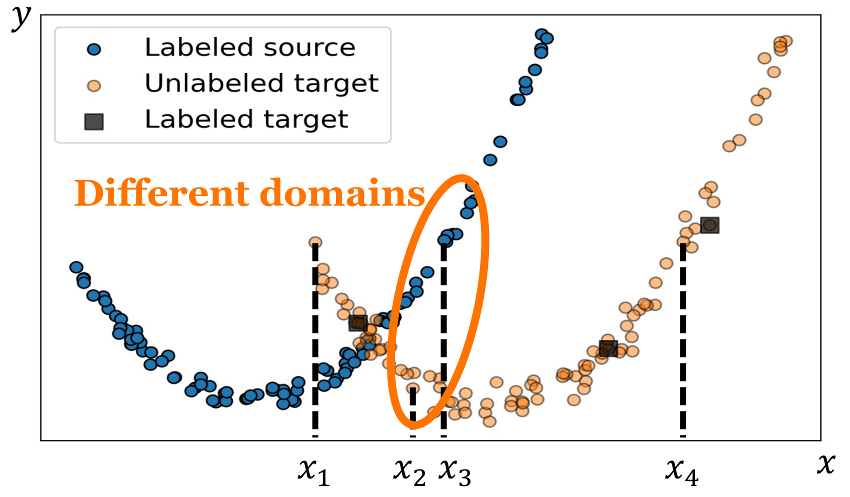
    $$f(x_2) \neq f(x_3) \text{ for } x_2 \approx x_3 \text{ (heterogeneous case)}$$

    - x_2 from target domain, and x_3 from source domain
  - Implication:
    - Input-output relationship varies among different domains
    - It cannot be directly captured by a standard neural network





Distribution-Informed Neural Networks



❑ Formal definition

- Distribution-aware input-output relationship
  - ➢ $\tilde{f}: (x, \mathbb{P}) \to y$ where $x \in \mathbb{P}$
  - ➢ **$\tilde{f}(x_2, \mathbb{P}_{tgt}) \neq \tilde{f}(x_3, \mathbb{P}_{src})$ for $x_2 \approx x_3$ (heterogeneous case)**
- Distribution-informed neural network (DINO)

DINO:

$$\tilde{f}(x, \mathbb{P}) \coloneqq f_\theta(x) \cdot g_{w_g}(\mathbb{P}|x)$$
$$= \left(\phi_{\theta^{<L}}(x)^T w\right) \cdot \left(\Phi_x(\mathbb{P})^T w_g\right) = w^T\left(\phi_{\theta^{<L}}(x)\Phi_x(\mathbb{P})^T\right)w_g$$

Input representation learning

A fully-connected NN: $f_\theta(x) = \phi_{\theta^{<L}}(x)^T w$

$\theta^{<L}$: Parameters of the first $L-1$ layers
w: Parameters of the output layer

Input-oriented distribution representation learning

Infinitely-wide $f_\theta(\cdot)$ ➡ NNGP kernel space $K_\mathcal{X}$ ➡ $\Phi_x(\mathbb{P}) = \sum_{i=1}^{n} \beta_{x,\tilde{x}_i} \langle \cdot, \tilde{x}_i \rangle_{K_\mathcal{X}}$ ➡ $g_{w_g}(\mathbb{P}|x) = \Phi_x(\mathbb{P})^T w_g$

Jaehoon Lee, et al. "**Deep neural networks as gaussian processes.**" *In ICLR. 2018*

Proposed Algorithm: DINO-INIT

- Observation at model initialization
  - DINO is a Gaussian process with adaptive NNGP kernel

Under random initialization, when the network width goes to infinity, we have $\tilde{f}(\cdot) \sim \mathcal{N}(0, K^{DA})$ with

$$K^{DA}\big((x, \mathbb{P}), (x', \mathbb{P}')\big) = K_{\mathcal{X}}(x, x') \cdot K_{\mathcal{P}|\mathcal{X}}(\mathbb{P}, \mathbb{P}'|x, x')$$

where $K_{\mathcal{X}}(\cdot,\cdot)$ is the NNGP kernel, and $K_{\mathcal{P}|\mathcal{X}}(\cdot,\cdot)$ is a distribution kernel, i.e.,

$$K_{\mathcal{P}|\mathcal{X}}(\mathbb{P}, \mathbb{P}'|x, x') = \sum_{i=1}^{n} \sum_{j=1}^{n'} \beta_{x, \tilde{x}_i} \beta_{x', \tilde{x}_j} K_{\mathcal{X}}(\tilde{x}_i, \tilde{x}_j)$$

- Adaptive Gaussian process algorithm
  - Prior GP $\tilde{f}(\cdot) \sim \mathcal{N}(0, K^{DA})$
  - Prediction function $p\big(Y|X_*^{tgt}\big) = \mathcal{N}(\bar{\mu}, \bar{\Sigma})$

$$\bar{\mu} = K^{DA}\big(X_*^{tgt}, X\big)C^{-1}Y \qquad \bar{\Sigma} = K^{DA}\big(X_*^{tgt}, X_*^{tgt}\big) - K^{DA}\big(X_*^{tgt}, X\big)C^{-1}K^{DA}\big(X_*^{tgt}, X\big)^T$$

Jaehoon Lee, et al. "**Deep neural networks as gaussian processes.**" *In ICLR. 2018.*

Proposed Algorithm: DINO-TRAIN

- DINO under gradient descent training
  - Objective function

$$\mathcal{L}(\theta) = \underbrace{\frac{\alpha}{2n_{src}} \sum_{i=1}^{n_{src}} \left(\tilde{f}(x_i^{src}, \mathbb{P}^{src}) - y_i^{src}\right)^2 + \frac{1-\alpha}{2n_{tgt}^l} \sum_{j=1}^{n_{tgt}^l} \left(\tilde{f}\left(x_j^{tgt}, \mathbb{P}^{tgt}\right) - y_j^{tgt}\right)^2}_{\textbf{Supervised loss over labeled examples}} + \underbrace{\frac{\mu}{2} \mathrm{MMD}_{\Theta_{DA}}^2(\mathbb{P}^{src}, \mathbb{P}^{tgt})}_{\textbf{Empirical MMD-NTK}}$$

  - Empirical Maximum Mean Discrepancy (MMD) over training dynamics
    - Measure the distribution shift during gradient descent training

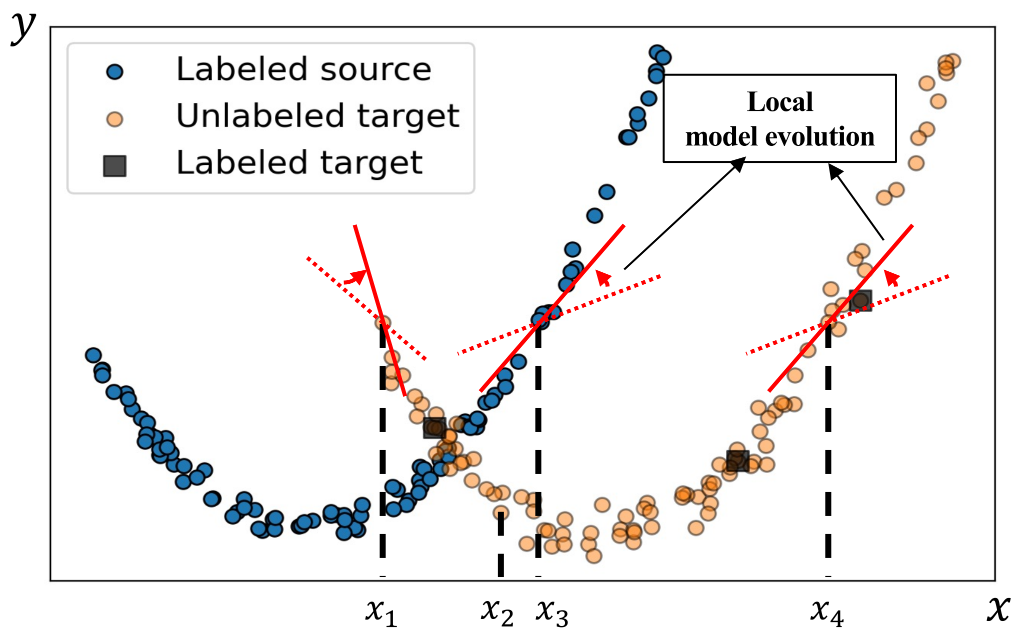
$$\mathrm{MMD}_{\Theta_{DA}}^2(\mathbb{P}^{src}, \mathbb{P}^{tgt}) = \left\| \frac{1}{n_{src}} \sum_{i=1}^{n_{src}} \nabla_\theta \tilde{f}(x_i^{src}, \mathbb{P}^{src}) - \frac{1}{n_{tgt}} \sum_{j=1}^{n_{tgt}} \nabla_\theta \tilde{f}\left(x_j^{tgt}, \mathbb{P}^{tgt}\right) \right\|_{\mathcal{H}_{DA}}^2$$

Arthur Jacot, et al. "**Neural tangent kernel: Convergence and generalization in neural networks**." In *NeurIPS*. 2018.
Arthur Gretton, et al. "**A kernel two-sample test**." *The Journal of Machine Learning Research*. 2012.

Proposed Algorithm: DINO-TRAIN

- Intuition behind MMD-NTK
  - Unified distribution discrepancy estimator
  - Local model evolvement

	Estimator	Characterization
MMD-RBF [Long et al. 2015]	▪ Two-stage discrepancy; ▪ Heuristic division: feature extractor and label predictor	▪ Simple model output
MMD-NTK	▪ Unified estimator over training dynamics of $\tilde{f}(\cdot)$	▪ Local model evolvement



Mingsheng Long, et al. "**Learning transferable features with deep adaptation networks.**" In *ICML*, 2015

Theoretical Analysis

❑ Convergence

There exists $\eta^* \in \mathbb{R}_+$ such that for the the infinitely-wide DINO $\tilde{f}(\cdot)$ trained under gradient flow with learning rate $\eta < \eta^*$, the prediction function **$\lim_{t\to\infty} \tilde{f}_{\theta_t}(X_*^{tgt})$ converges to a Gaussian process** with

$$\mu = \Theta_{\mathrm{DA}}(X_*^{tgt}, X)\Theta_{DA}(X,X)^{-1}Y$$

$$\Sigma = K^{DA}(X_*^{tgt}, X_*^{tgt}) + \Theta_{DA}(X_*^{tgt}, X)\Theta_{DA}(X,X)^{-1}K^{DA}\Theta_{DA}(X, X_*^{tgt}) - (\Theta_{DA}(X_*^{tgt}, X)\Theta_{DA}(X,X)^{-1}K^{DA}(X, X_*^{tgt}) + h.c.)$$

❑ Generalization

For any $\delta > 0$, with probability $1 - \delta$, the expected error in the target domain can be bounded by

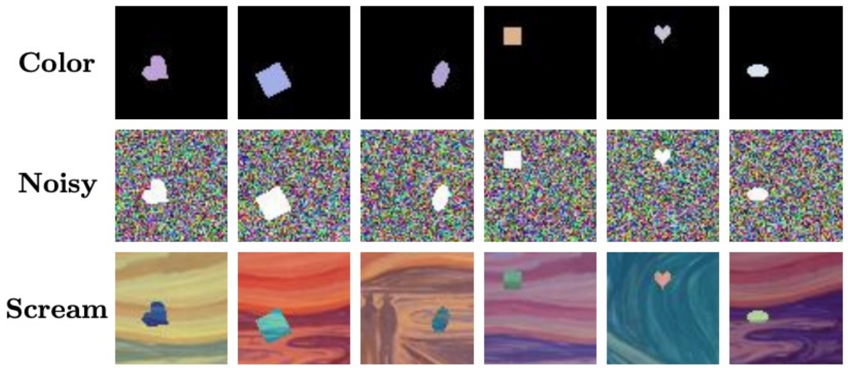
$$\epsilon_{tgt}(\tilde{f}) \le \underbrace{\frac{\alpha}{n_{src}} \sum_{i=1}^{n_{src}} \left(\tilde{f}(x_i^{src}, \mathbb{P}^{src}) - y_i^{src}\right)^2 + \frac{1-\alpha}{n_{tgt}^l} \sum_{j=1}^{n_{tgt}^l} \left(\tilde{f}\left(x_j^{tgt}, \mathbb{P}^{tgt}\right) - y_j^{tgt}\right)^2 + 8\alpha M_0 \cdot \mathrm{MMD}_{\Theta_{DA}}(\mathbb{P}^{src}, \mathbb{P}^{tgt})} + \Omega$$

DINO-TRAIN empirically minimizes the error bound

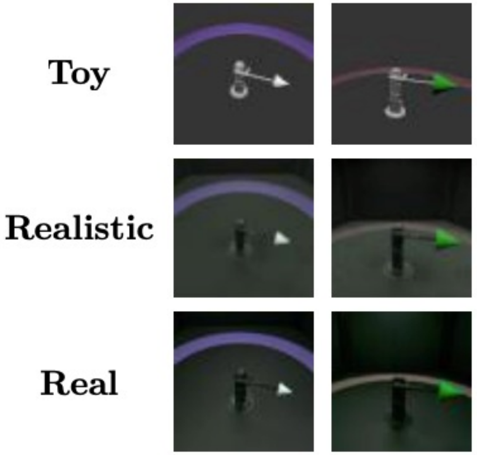
Arthur Jacot, et al. "**Neural tangent kernel: Convergence and generalization in neural networks**." In *NeurIPS*. 2018.
Jaehoon Lee, et al. "**Wide neural networks of any depth evolve as linear models under gradient descent**." In *NeurIPS*. 2019.

Experiments

- Data sets
  - dSprites
  - MPI3D
  - Plant Phenotyping
- Metric
  - MAE: Mean Absolute Error
- Baseline
  - Plain Gaussian process: NNGP and NTKGP
  - Adaptive Gaussian process: AT-GP and TL-NTK



Examples of dSprites



Examples of MPI3D

Jaehoon Lee, et al. "**Deep neural networks as gaussian processes**." In *ICLR*. 2018.
Bobby He, et al. "**Bayesian deep ensembles via the neural tangent kernel**." In *NeurIPS*. 2020.
Bin Cao, et al. "**Adaptive transfer learning**." In *AAAI*. 2010.
Wesley Maddox, et al. "**Fast adaptation with linearized neural networks**." In *AISTATS*. 2021.

Results

Methods	C → N	C → S	N → C	N → S	S → C	S → N	Avg.
NNGP [34]	$2.041_{\pm 0.001}$	$1.823_{\pm 0.001}$	$0.445_{\pm 0.002}$	$0.624_{\pm 0.001}$	$0.197_{\pm 0.002}$	$0.459_{\pm 0.002}$	0.932
NTKGP [25]	$1.345_{\pm 0.002}$	$1.227_{\pm 0.000}$	$0.323_{\pm 0.002}$	$0.529_{\pm 0.004}$	$0.248_{\pm 0.001}$	$0.425_{\pm 0.002}$	0.683
AT-GP [7]	$0.194_{\pm 0.005}$	$0.259_{\pm 0.002}$	$\mathbf{0.104}_{\pm \mathbf{0.001}}$	$0.252_{\pm 0.005}$	$0.118_{\pm 0.003}$	$0.189_{\pm 0.006}$	0.186
TL-NTK [38]	$0.164_{\pm 0.001}$	$\mathbf{0.231}_{\pm \mathbf{0.000}}$	$0.124_{\pm 0.005}$	$0.242_{\pm 0.002}$	$0.125_{\pm 0.001}$	$0.197_{\pm 0.004}$	0.181
DINO-INIT (ours)	$0.128_{\pm 0.001}$	$0.233_{\pm 0.003}$	$0.114_{\pm 0.002}$	$\mathbf{0.227}_{\pm \mathbf{0.002}}$	$\mathbf{0.112}_{\pm \mathbf{0.001}}$	$\mathbf{0.181}_{\pm \mathbf{0.005}}$	**0.166**
DINO-TRAIN (ours)	$\mathbf{0.127}_{\pm \mathbf{0.002}}$	$0.240_{\pm 0.003}$	$0.127_{\pm 0.000}$	$0.243_{\pm 0.000}$	$0.128_{\pm 0.001}$	$0.194_{\pm 0.001}$	0.177

Results on dSprites

Lower is better

Methods	M → MU	MU → M
NNGP [34]	$0.562_{\pm 0.001}$	$0.672_{\pm 0.010}$
NTKGP [25]	$0.562_{\pm 0.004}$	$0.702_{\pm 0.010}$
AT-GP [7]	$\mathbf{0.308}_{\pm \mathbf{0.006}}$	$0.593_{\pm 0.025}$
TL-NTK [38]	$0.316_{\pm 0.008}$	$0.488_{\pm 0.027}$
DINO-INIT (ours)	$0.316_{\pm 0.007}$	$0.645_{\pm 0.017}$
DINO-TRAIN (ours)	$0.314_{\pm 0.009}$	$\mathbf{0.443}_{\pm \mathbf{0.030}}$

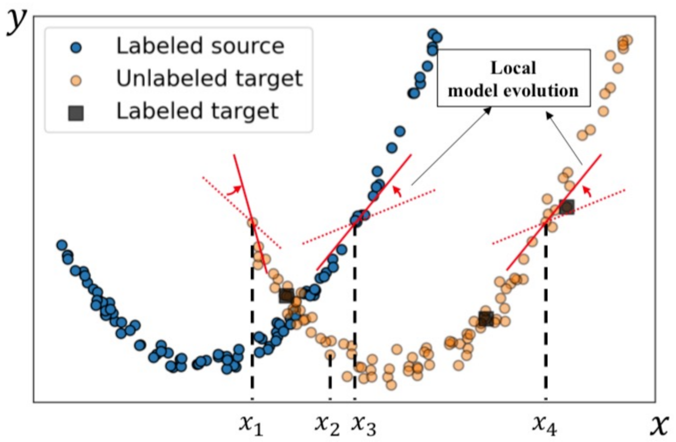
Results on Plant Phenotyping

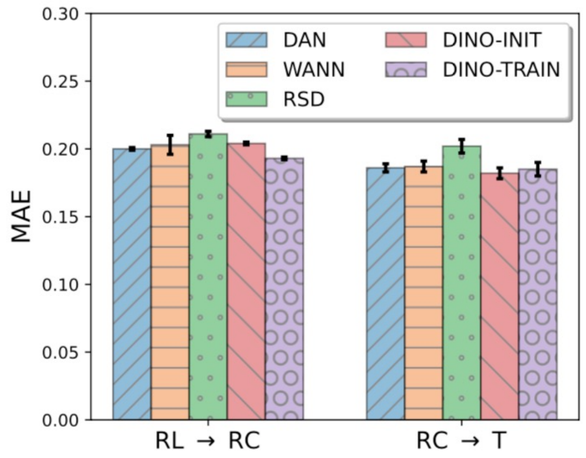
Methods	RL → RC	RL → T	RC → RL	RC → T	T → RL	T → RC	Avg.
NNGP [34]	$0.313_{\pm 0.001}$	$0.438_{\pm 0.004}$	$0.356_{\pm 0.005}$	$0.515_{\pm 0.008}$	$0.367_{\pm 0.001}$	$0.324_{\pm 0.004}$	0.386
NTKGP [25]	$0.396_{\pm 0.001}$	$0.365_{\pm 0.001}$	$0.200_{\pm 0.007}$	$0.390_{\pm 0.003}$	$0.390_{\pm 0.000}$	$0.354_{\pm 0.003}$	0.349
AT-GP [7]	$0.214_{\pm 0.011}$	$0.209_{\pm 0.002}$	$0.227_{\pm 0.010}$	$0.198_{\pm 0.002}$	$0.236_{\pm 0.000}$	$0.249_{\pm 0.000}$	0.222
TL-NTK [38]	$0.206_{\pm 0.004}$	$0.200_{\pm 0.002}$	$0.213_{\pm 0.000}$	$0.197_{\pm 0.000}$	$0.226_{\pm 0.001}$	$0.218_{\pm 0.000}$	0.210
DINO-INIT (ours)	$0.204_{\pm 0.001}$	$\mathbf{0.185}_{\pm \mathbf{0.006}}$	$\mathbf{0.207}_{\pm \mathbf{0.003}}$	$\mathbf{0.182}_{\pm \mathbf{0.004}}$	$\mathbf{0.218}_{\pm \mathbf{0.001}}$	$\mathbf{0.212}_{\pm \mathbf{0.001}}$	**0.201**
DINO-TRAIN (ours)	$\mathbf{0.193}_{\pm \mathbf{0.001}}$	$0.194_{\pm 0.003}$	$\mathbf{0.207}_{\pm \mathbf{0.003}}$	$0.188_{\pm 0.002}$	$0.226_{\pm 0.001}$	$0.218_{\pm 0.001}$	0.204

Results on MPI3D

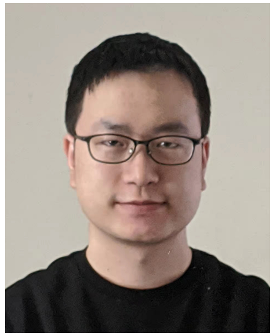
Conclusion

- **Problem**: Domain adaptation regression
- **Algorithms**: Distribution-informed neural networks
  - **DINO-INIT**: Gaussian process with adaptive NNGP kernel
  - **DINO-TRAIN**: Discrepancy minimization using MMD in the NTK-induced RKHS
- **Evaluation**: Effectiveness on adaptive regression tasks
  - Object localization
  - Plant phenotyping

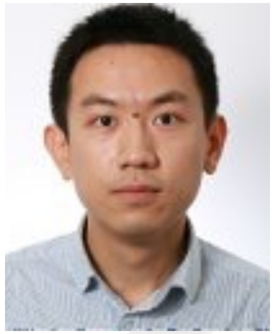




Distribution-Informed Neural Networks for Domain Adaptation Regression


Jun Wu


Jingrui He


Sheng Wang


Kaiyu Guan


Elizabeth Ainsworth

University of Illinois Urbana-Champaign
{junwu3, jingrui, sheng12, kaiyug, ainswort}@illinois.edu

AIFARMS
Artificial Intelligence for Future Agricultural Resilience, Management, and Sustainability

ILLINOIS