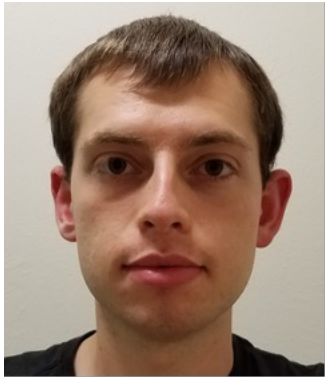
Meta-learning Adaptive Deep Kernel Gaussian Processes for Molecular Property Prediction and Optimization

Wenlin Chen
PhD Student at University of Cambridge & Max Planck Institute for Intelligent Systems

https://wenlin-chen.github.io/
wc337@cam.ac.uk

Collaborators

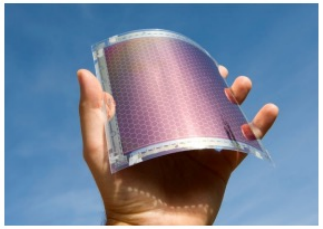


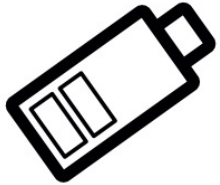
Austin Tripp
(Valence Labs)

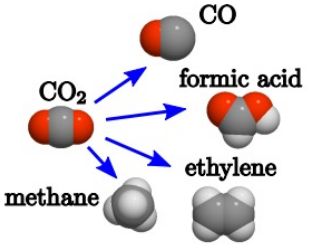
Miguel Hernández-Lobato
(University of Cambridge)

Motivation: Molecular Design

- Predict bio/physicochemical properties of given molecules
- Optimize properties to design novel molecules

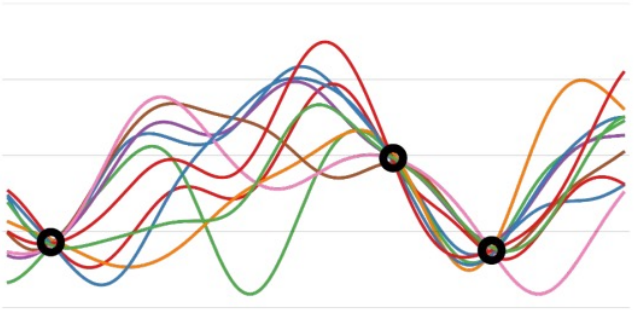




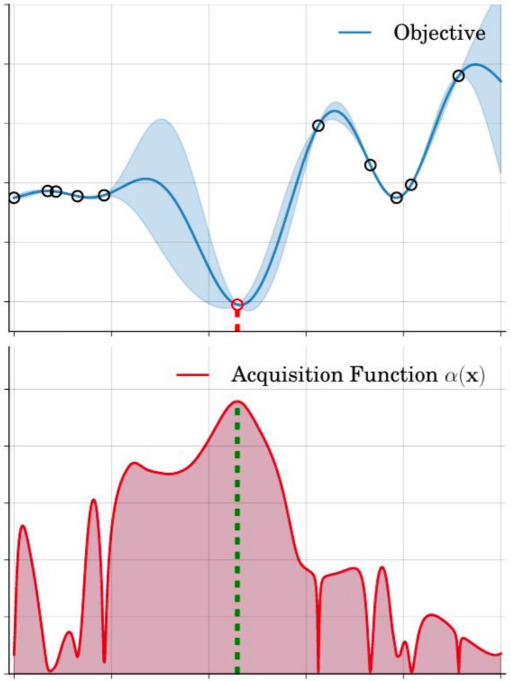


Small datasets are ubiquitous!

Gaussian Processes and Bayesian Optimization

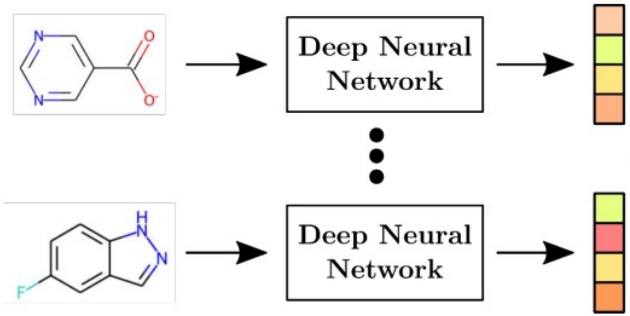


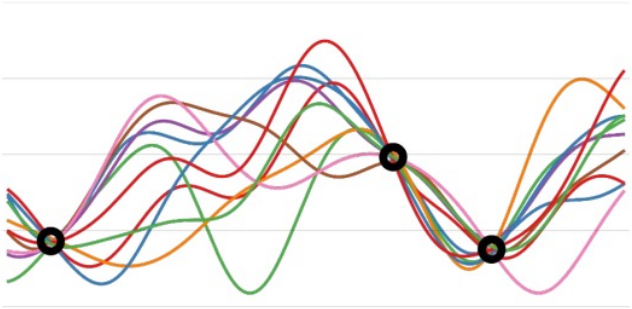
Gaussian Processes and Bayesian Optimization



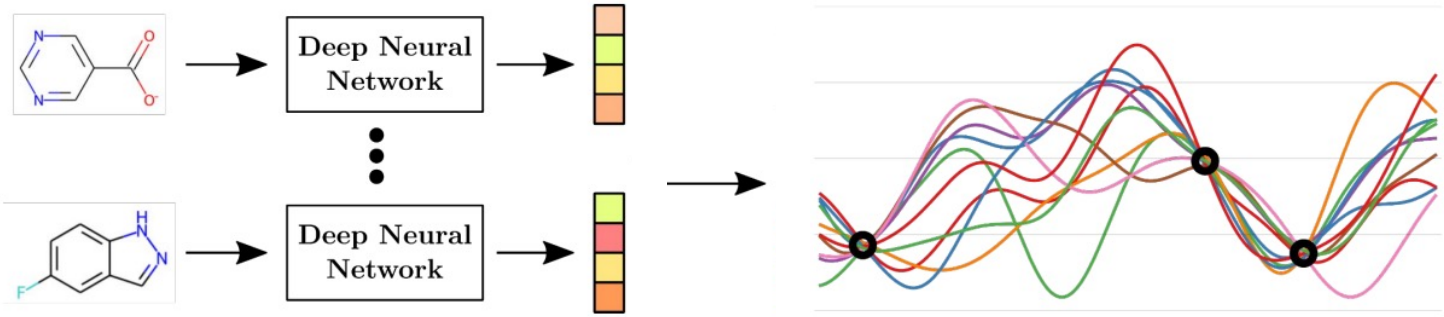
1. Get initial sample.
2. Fit a model to the data:
$$p(y|\mathrm{x}, \mathcal{D}_n)\,.$$
3. **Select data collection strategy:**
$$\alpha(\mathrm{x}) = \mathrm{E}_{p(y|\mathrm{x},\mathcal{D}_n)}[U(y|\mathrm{x}, \mathcal{D}_n)]\,.$$
4. **Optimize acquisition function $\alpha(\mathrm{x})$.**
5. **Collect data and update model.**
6. **Repeat!**

Gaussian Processes (GPs) and Neural Networks (NNs)





Deep Kernel Gaussian Processes



Deep kernel GPs operate on features learned by a NN.

$$k_{\theta,\phi}(x, x') = c_\theta(\mathbf{f}_\phi(x), \mathbf{f}_\phi(x'))$$

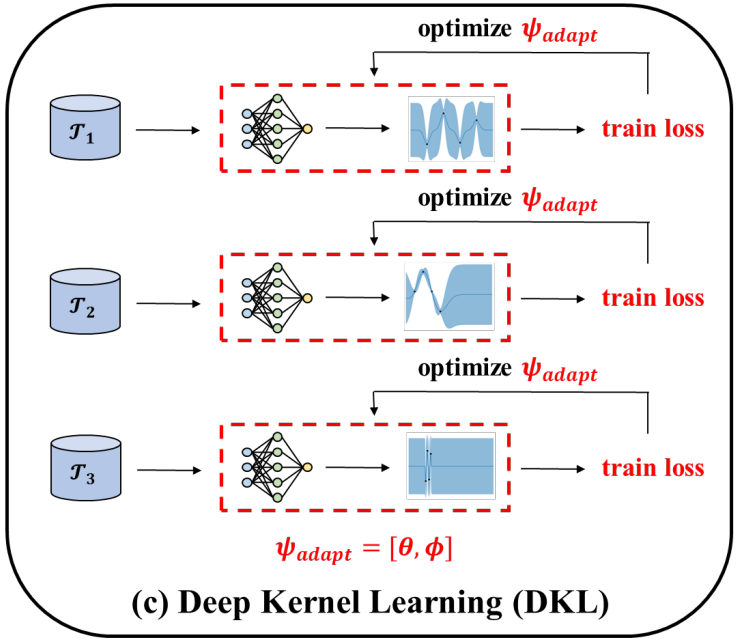
deep kernel

learnable parameters $\psi = (\theta, \phi)$

base kernel (e.g., RBF)

NN feature extractor

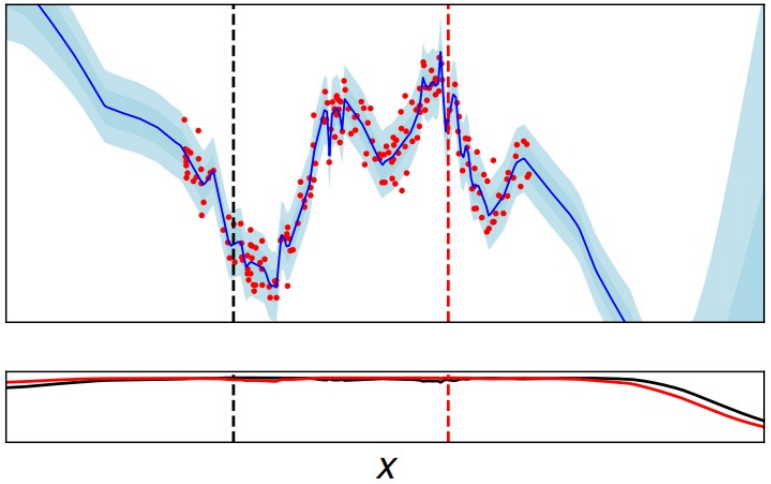
Prior Work: Deep Kernel Learning (DKL)



$$\psi^* = \arg\min_{\psi} \text{NLML}\left(\psi, \mathcal{S}_{\mathcal{T}}\right)$$

Single-task method:

Overfitting!

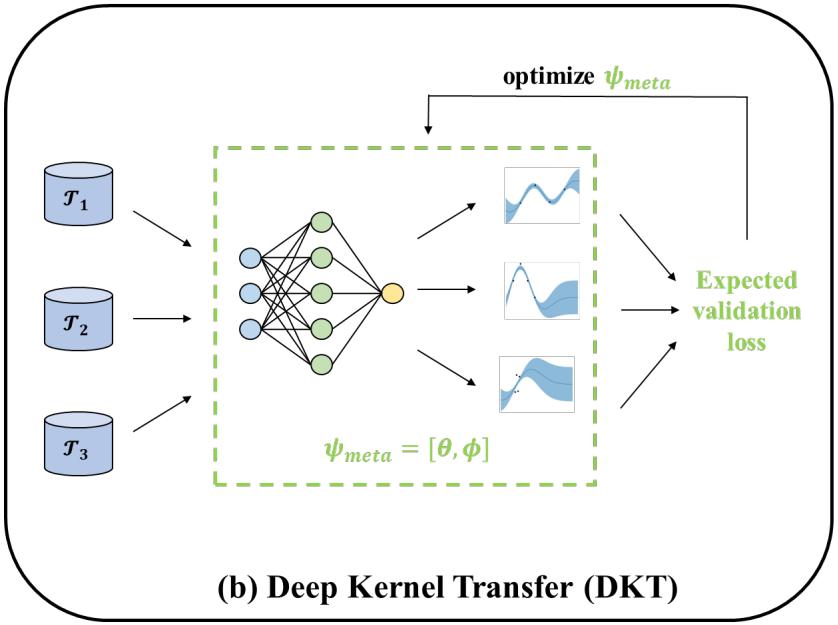


(b) Exact DKL kernel

Wilson, Andrew Gordon, et al. "Deep kernel learning." Artificial intelligence and statistics. PMLR, 2016.

Ober, Sebastian W., Carl E. Rasmussen, and Mark van der Wilk. "The promises and pitfalls of deep kernel learning." Uncertainty in Artificial Intelligence. PMLR, 2021.

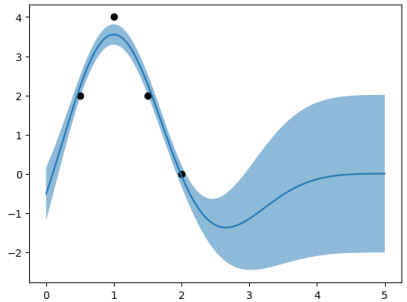
Prior Work: Deep Kernel Transfer (DKT)

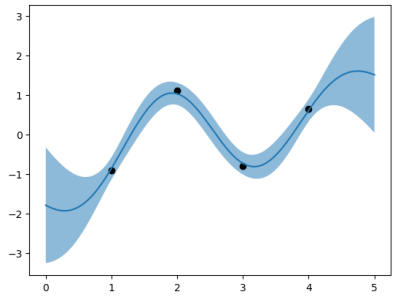


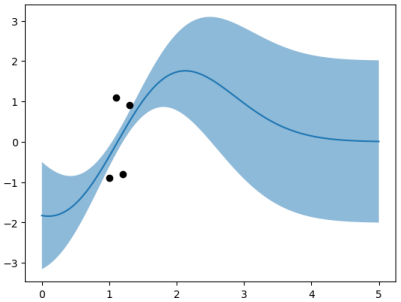
(b) Deep Kernel Transfer (DKT)

$$\psi^* = \arg\min_{\psi} \mathbb{E}_{p(\mathcal{T})}[\text{NLML}(\psi, \mathcal{T})]$$

Multi-task method:
Underfitting!





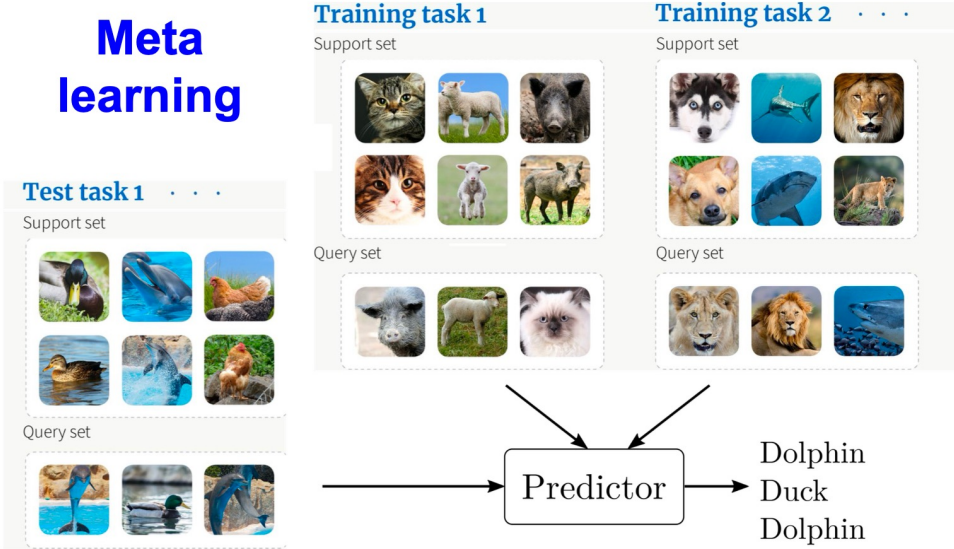


Patacchiola, Massimiliano, et al. "Bayesian meta-learning for the few-shot setting via deep kernels." Advances in Neural Information Processing Systems 33 (2020): 16108-16118.

Conventional Learning vs Meta-learning

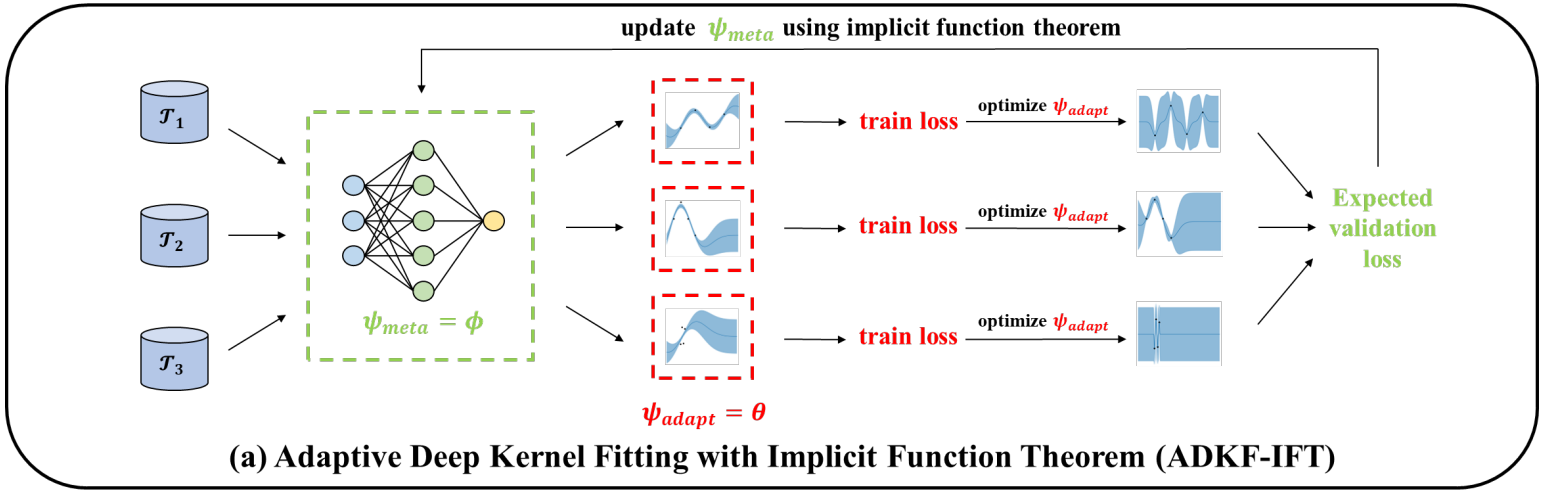
Conventional learning: optimize performance on training data. Evaluate on test data.

Meta learning: optimize performance on **query set** from training tasks **after having processed support set** from such tasks. Evaluate on query set from test task.



Figures source: Borealis AI.

Adaptive Deep Kernel Fitting with IFT (ADKF-IFT)



(a) Adaptive Deep Kernel Fitting with Implicit Function Theorem (ADKF-IFT)

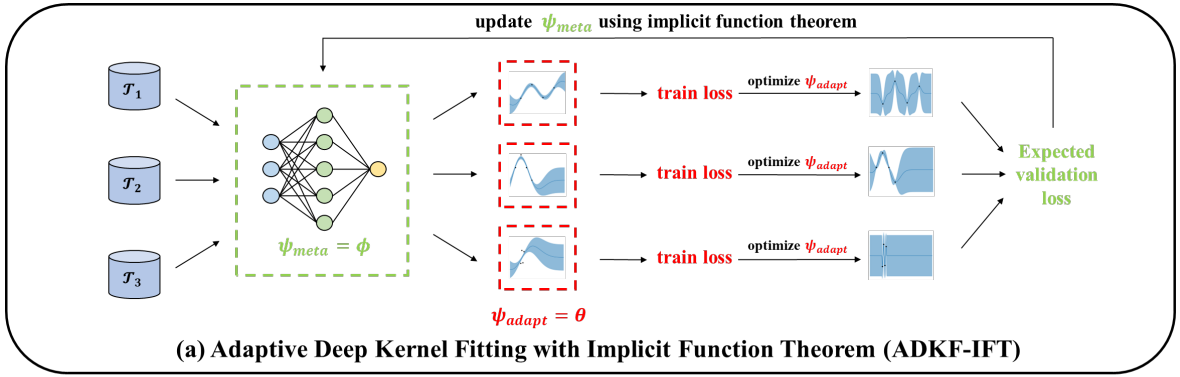
Bilevel Optimization

$$\psi^*_{\text{meta}} = \arg\min_{\psi_{\text{meta}}} \mathbb{E}_{p(\mathcal{T})}[\mathcal{L}_V(\psi_{\text{meta}}, \psi^*_{\text{adapt}}(\psi_{\text{meta}}, \mathcal{S}_\mathcal{T}), \mathcal{T})],$$

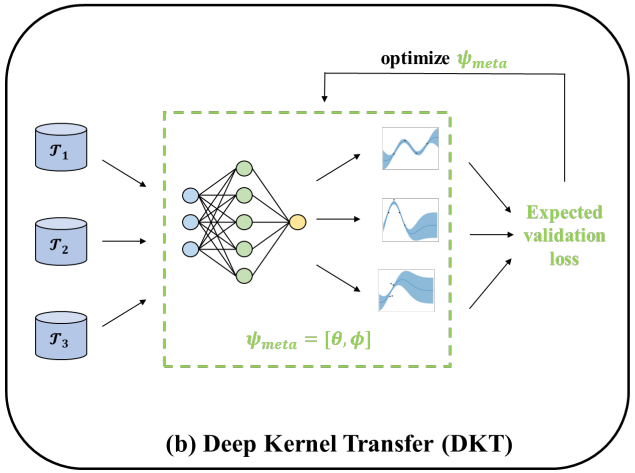
$$\text{s.t.} \quad \psi^*_{\text{adapt}}(\psi_{\text{meta}}, \mathcal{S}_\mathcal{T}) = \arg\min_{\psi_{\text{adapt}}} \mathcal{L}_T(\psi_{\text{meta}}, \psi_{\text{adapt}}, \mathcal{S}_\mathcal{T}).$$

best response function

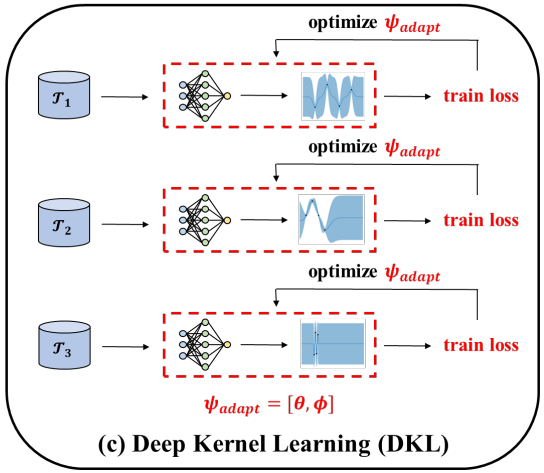
Contrast DKL, DKT and ADKF-IFT



(a) Adaptive Deep Kernel Fitting with Implicit Function Theorem (ADKF-IFT)



(b) Deep Kernel Transfer (DKT)



(c) Deep Kernel Learning (DKL)

ADKF-IFT resolves the overfitting and underfitting issues!

How to Solve Bilevel Optimization?

implicit function of $\boldsymbol{\psi}_{meta}$ alone

$$\boldsymbol{\psi}^*_{\text{meta}} = \arg\min_{\boldsymbol{\psi}_{\text{meta}}} \mathbb{E}_{p(\mathcal{T})}[\mathcal{L}_V(\boldsymbol{\psi}_{\text{meta}}, \boldsymbol{\psi}^*_{\text{adapt}}(\boldsymbol{\psi}_{\text{meta}}, \mathcal{S}_\mathcal{T}), \mathcal{T})],$$

$$\text{s.t.} \quad \boldsymbol{\psi}^*_{\text{adapt}}(\boldsymbol{\psi}_{\text{meta}}, \mathcal{S}_\mathcal{T}) = \arg\min_{\boldsymbol{\psi}_{\text{adapt}}} \mathcal{L}_T(\boldsymbol{\psi}_{\text{meta}}, \boldsymbol{\psi}_{\text{adapt}}, \mathcal{S}_\mathcal{T}).$$

best response function

- **Inner optimization:** L-BFGS with auto-diff.

- **Outer optimization:** how to calculate the gradient of the validation loss $\mathcal{L}_V$?

Hypergradient:

$$\frac{d\,\mathcal{L}_V}{d\,\boldsymbol{\psi}_{\text{meta}}} = \frac{\partial\,\mathcal{L}_V}{\partial\,\boldsymbol{\psi}_{\text{meta}}} + \frac{\partial\,\mathcal{L}_V}{\partial\,\boldsymbol{\psi}^*_{\text{adapt}}} \frac{\partial\,\boldsymbol{\psi}^*_{\text{adapt}}}{\partial\,\boldsymbol{\psi}_{\text{meta}}},$$

easy easy hard

(by chain rule)

Exact and Efficient Training with IFT

- **Outer optimization:** how to calculate the gradient of the validation loss $\mathcal{L}_V$?

Hypergradient:

$$\frac{d\,\mathcal{L}_V}{d\,\psi_{\text{meta}}} = \frac{\partial\,\mathcal{L}_V}{\partial\,\psi_{\text{meta}}} + \frac{\partial\,\mathcal{L}_V}{\partial\,\psi^*_{\text{adapt}}}\frac{\partial\,\psi^*_{\text{adapt}}}{\partial\,\psi_{\text{meta}}},$$

(by chain rule)

easy easy hard

- ψ^*_{adapt} is **a critical point of the train loss** $\mathcal{L}_T$

- Can apply the **Implicit Function Theorem** (**IFT**)!

IFT:

$$\left.\frac{\partial\,\psi^*_{\text{adapt}}}{\partial\,\psi_{\text{meta}}}\right|_{\psi'_{\text{meta}}} = -\left(\frac{\partial^2\,\mathcal{L}_T(\psi_{\text{meta}},\psi_{\text{adapt}},\mathcal{S}_{\mathcal{T}'})}{\partial\,\psi_{\text{adapt}}\,\partial\,\psi^T_{\text{adapt}}}\right)^{-1}\left.\frac{\partial^2\,\mathcal{L}_T(\psi_{\text{meta}},\psi_{\text{adapt}},\mathcal{S}_{\mathcal{T}'})}{\partial\,\psi_{\text{adapt}}\,\partial\,\psi^T_{\text{meta}}}\right|_{\psi'_{\text{meta}},\psi'_{\text{adapt}}},$$

(inverse Hessian) (mixed partial derivatives)

where $\psi'_{\text{adapt}} = \psi^*_{\text{adapt}}(\psi'_{\text{meta}}, \mathcal{S}_{\mathcal{T}'})$.

General Framework vs Specific Instantiations

ADKF-IFT can be formalized as a **bi-level optimization** problem:

$$\psi^*_{\text{meta}} = \arg\min_{\psi_{\text{meta}}} \mathbb{E}_{p(\mathcal{T})}[\mathcal{L}_V(\psi_{\text{meta}}, \psi^*_{\text{adapt}}(\psi_{\text{meta}}, \mathcal{S}_\mathcal{T}), \mathcal{T})],$$

$$\text{s.t.} \quad \psi^*_{\text{adapt}}(\psi_{\text{meta}}, \mathcal{S}_\mathcal{T}) = \arg\min_{\psi_{\text{adapt}}} \mathcal{L}_T(\psi_{\text{meta}}, \psi_{\text{adapt}}, \mathcal{S}_\mathcal{T}).$$

✓ **General framework:** $\boldsymbol{\psi}_{adapt}$, $\boldsymbol{\psi}_{meta}$, $\mathcal{L}_T$, $\mathcal{L}_V$ could be anything.

✓ Any particular choice of $\boldsymbol{\psi}_{adapt}$, $\boldsymbol{\psi}_{meta}$, $\mathcal{L}_T$, $\mathcal{L}_V$ is an **instantiation**.

✓ DKL and DKT are special instantiations **(extreme cases)** of the general framework!

General Framework vs Specific Instantiations

Adaptive Deep Kernel Fitting with Implicit Function Theorem (ADKF-IFT)

$$\psi^*_{\text{meta}} = \arg\min_{\psi_{\text{meta}}} \mathbb{E}_{p(\mathcal{T})}[\mathcal{L}_V(\psi_{\text{meta}}, \psi^*_{\text{adapt}}(\psi_{\text{meta}}, \mathcal{S}_\mathcal{T}), \mathcal{T})],$$

$$\text{s.t.} \quad \psi^*_{\text{adapt}}(\psi_{\text{meta}}, \mathcal{S}_\mathcal{T}) = \arg\min_{\psi_{\text{adapt}}} \mathcal{L}_T(\psi_{\text{meta}}, \psi_{\text{adapt}}, \mathcal{S}_\mathcal{T}).$$

$\boldsymbol{\psi_{adapt} = \psi = [\theta, \phi]}$

$\boldsymbol{\psi_{meta} = \emptyset}$

$\mathcal{L}_T = \text{NLML}$

$\mathcal{L}_V = \text{NLML}$

$\boldsymbol{\psi_{meta} = \psi = [\theta, \phi]}$

$\boldsymbol{\psi_{adapt} = \emptyset}$

$$\psi^* = \arg\min_{\psi} \text{NLML}(\psi, \mathcal{S}_\mathcal{T})$$

Deep Kernel Learning (DKL)

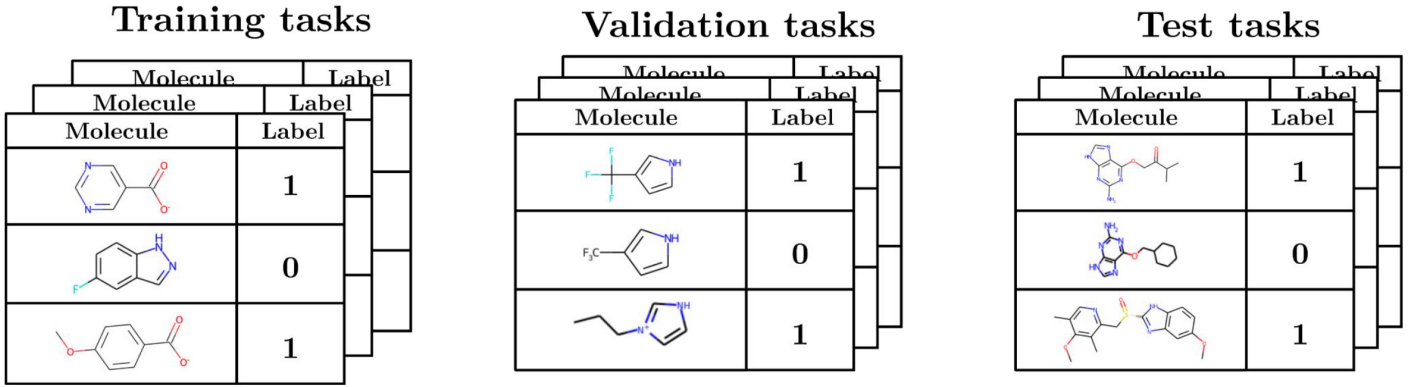
$$\psi^* = \arg\min_{\psi} \mathbb{E}_{p(\mathcal{T})}[\text{NLML}(\psi, \mathcal{T})]$$

Deep Kernel Transfer (DKT)

Few-shot Molecular Property Prediction

Stanley et al. FS-Mol: A Few-Shot Learning Dataset of Molecules, NeurIPS D. & B. track 2021.

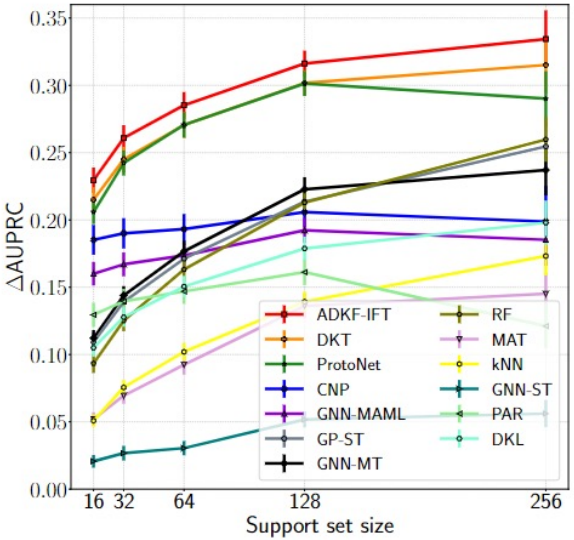
- Data extracted from **ChEMBL**: a database of molecules with drug-like properties.
- **4,938** training, **40** validation and **157** test tasks.
- Each task is associated with a **target protein** and has **32** to **500** data points (compounds).
- **Binary classification**: IC50 or EC50 value larger than median for the task.



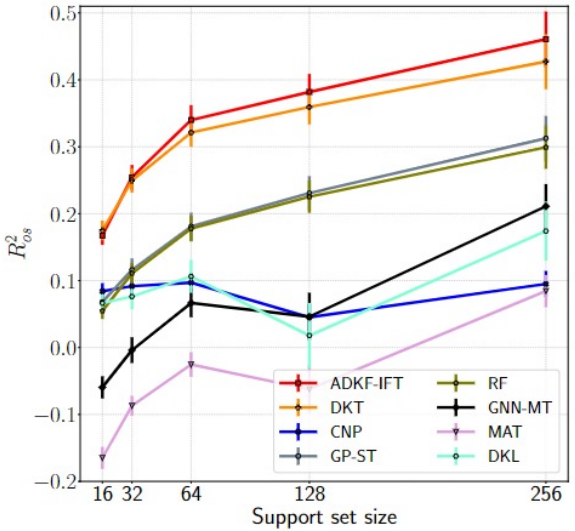
Stanley, Megan, et al. "Fs-mol: A few-shot learning dataset of molecules." NeurIPS 2021 (Datasets and Benchmarks Track).

Few-shot Molecular Property Prediction

FS-Mol: 4,938 training tasks, 40 validation tasks, 157 test tasks; 233,786 unique compounds.



(a) Classification (157 tasks).

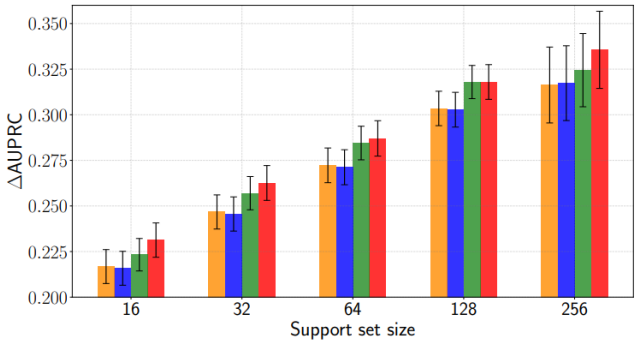


(b) Regression (111 tasks).

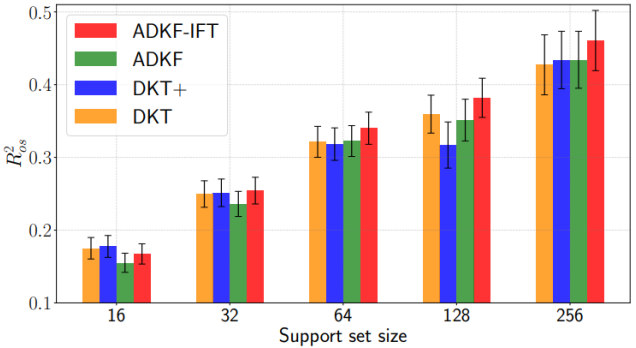
✓ The improvements of ADKF-IFT over other methods are **statistically significant!**

Stanley, Megan, et al. "Fs-mol: A few-shot learning dataset of molecules." NeurIPS 2021 (Datasets and Benchmarks Track).

Ablation Study on FS-Mol



(a) Classification (157 tasks).



(b) Regression (111 tasks).

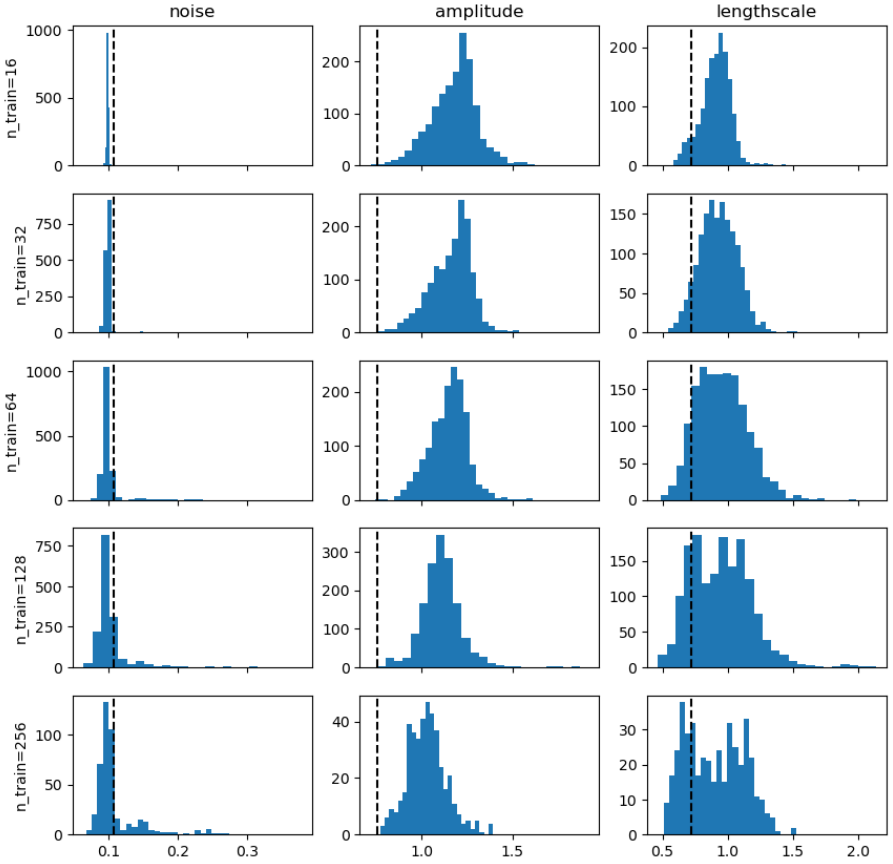
DKT ≈ DKT+ ≤ ADKF < ADKF-IFT

- **DKT+** is like **DKT** but tuning the base kernel parameters $\boldsymbol{\theta}$ during meta-testing.
- **ADKF** is like **ADKF-IFT** but ignoring the indirect gradient (second term below).

$$\frac{d\,\mathcal{L}_V}{d\,\boldsymbol{\psi}_{\text{meta}}} = \frac{\partial\,\mathcal{L}_V}{\partial\,\boldsymbol{\psi}_{\text{meta}}} + \frac{\partial\,\mathcal{L}_V}{\partial\,\boldsymbol{\psi}^*_{\text{adapt}}} \frac{\partial\,\boldsymbol{\psi}^*_{\text{adapt}}}{\partial\,\boldsymbol{\psi}_{\text{meta}}},$$

0

Additional Analysis on FS-Mol

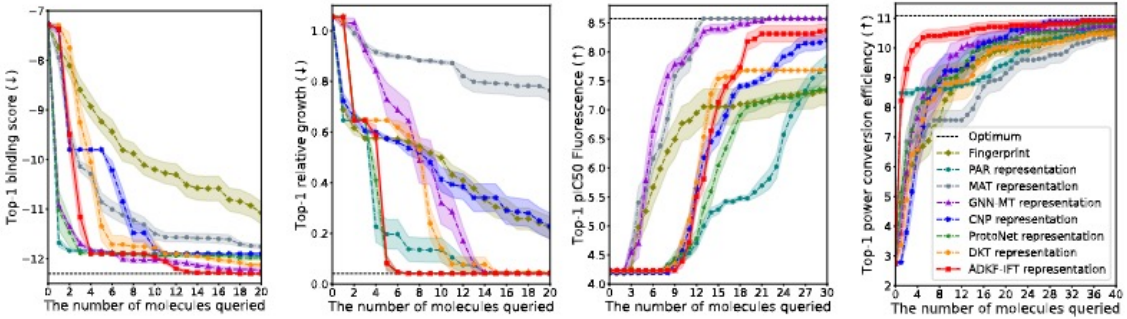


✓ The base kernel parameters $\boldsymbol{\theta}$ **vary across tasks!**

✓ ADKF-IFT achieves **better signal-to-noise ratio!**

OOD Molecular Property Prediction and Optimization

- Bayesian optimization (BO)



(a) Molecular docking. (b) Antibiotic discovery. (c) Antiviral drug design. (d) Material design.

- Test predictive negative log likelihood (NLL)

Feature representation	Out-of-domain molecular design task			
	Molecular docking	Antibiotic discovery	Antiviral drug design	Material design
Fingerprint	1.138 ± 0.014	1.669 ± 0.075	$\mathbf{4.601 \pm 0.086}$	1.091 ± 0.011
PAR	1.270 ± 0.019	2.185 ± 0.115	4.840 ± 0.086	1.283 ± 0.017
MAT	1.528 ± 0.028	2.390 ± 0.104	4.797 ± 0.088	2.198 ± 0.063
GNN-MT	1.994 ± 0.050	3.692 ± 0.225	6.399 ± 0.181	7.254 ± 0.217
CNP	1.493 ± 0.028	2.537 ± 0.162	5.005 ± 0.086	1.741 ± 0.043
ProtoNet	1.147 ± 0.013	1.615 ± 0.094	5.060 ± 0.086	1.032 ± 0.009
DKT	1.167 ± 0.012	1.602 ± 0.073	4.975 ± 0.092	1.026 ± 0.009
ADKF-IFT	$\mathbf{1.137 \pm 0.011}$	$\mathbf{1.496 \pm 0.043}$	4.781 ± 0.087	$\mathbf{0.996 \pm 0.007}$

- **Surrogate model:** GP operating on top of the features extracted by DNNs meta-trained on FS-Mol by different methods.

- **Evaluation:** four OOD molecular design tasks outside of FS-Mol.

✓ ADKF-IFT enables **fastest discovery of top performing molecules!**

✓ ADKF-IFT achieves **competitive test predictive performance!**

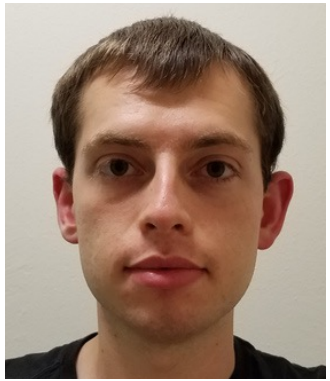
Summary

Our proposed **Adaptive Deep Kernel Fitting with Implicit Function Theorem (ADKF-IFT)** approach:

- ✓ Meta-learns feature representations that facilitate the adaptation of task-specific GP models.
- ✓ Generalizes DKL and DKT for training deep kernel GPs using a bilevel optimization framework.
- ✓ Efficiently solve the bilevel optimization problem by implicit function theorem.
- ✓ Produces state-of-the-art results on few-shot molecular property prediction benchmarks.
- ✓ Achieves great performance on OOD molecular property prediction and optimization tasks.
- ✓ Produces well-calibrated models for fully-automated high-throughput experimentation.

Reference and Collabroators

- **Meta-learning Adaptive Deep Kernel Gaussian Processes for Molecular Property Prediction**
  Wenlin Chen, Austin Tripp, José Miguel Hernández-Lobato
  International Conference on Learning Representations (ICLR), 2023.



Austin Tripp
(Valence Labs)

Miguel Hernández-Lobato
(University of Cambridge)