Generative Foundation Models for Sciences

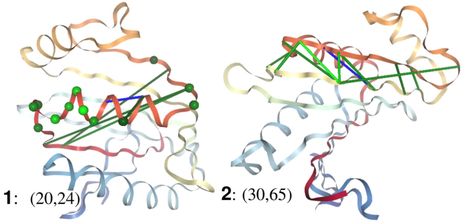
Tung Nguyen, CS261 Winter 2024

Science Advances by AI

AlphaFold: a solution to a 50-year-old grand challenge in biology

30 NOVEMBER 2020

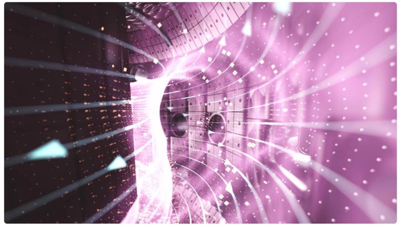
The AlphaFold team



Accelerating fusion science through learned plasma control

16 FEBRUARY 2022

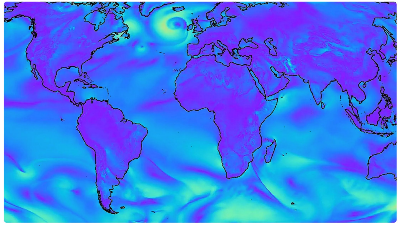
Pulsar Team, Swiss Plasma Center* (* External authors)



GraphCast: AI model for faster and more accurate global weather forecasting

14 NOVEMBER 2023

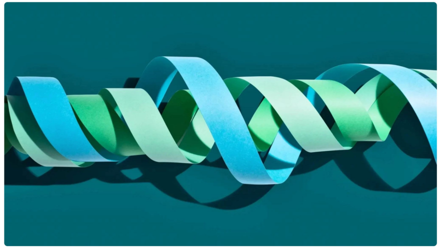
Remi Lam on behalf of the GraphCast team



Predicting gene expression with AI

4 OCTOBER 2021

Žiga Avsec

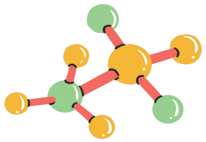


AI has the capability to discover and simulate complex patterns!

Jumper, John, et al., 2021; Degrave, Jonas, et al., 2022; Lam, Remi, et al., 2022; Avsec, Žiga, et al., 2021

General Problem

Learn a surrogate model f_θ to approximate $y = f(x)$



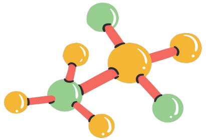
Chemical molecules

$\boxed{f(x)}$ Reactivity, Stability

$\boxed{x}$ Formula, structure

General Problem

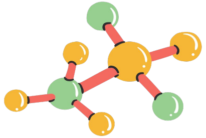
Learn a surrogate model f_θ to approximate $y = f(x)$

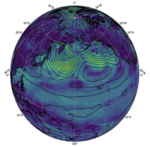


	Chemical molecules	Robotics
$\boxed{f(x)}$	Reactivity, Stability	Speed, Energy
$\boxed{x}$	Formula, structure	# limbs, mass

General Problem

Learn a surrogate model f_θ to approximate $y = f(x)$



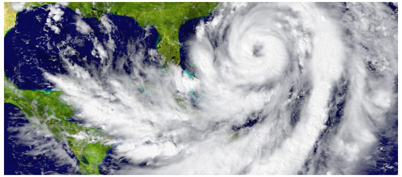


	Chemical molecules	Robotics	Climate Science
$f(x)$	Reactivity, Stability	Speed, Energy	Future weather
x	Formula, structure	# limbs, mass	Current weather

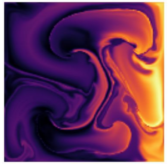
Surrogate Modeling Use Cases

Prediction

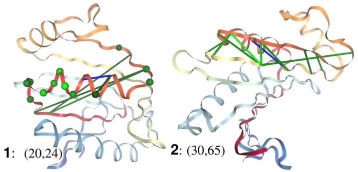
How to use $f_\theta(x)$ to **forecast/simulate** $f(x)$?



Weather forecasting



PDE
modeling



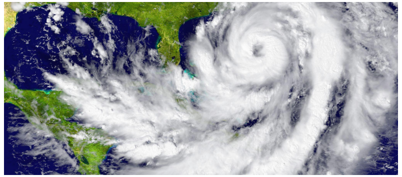
Protein structure
prediction

Image credit: Penn State University; Gupta, Jayesh K., et al., 2022; Jumper, John, et al., 2021

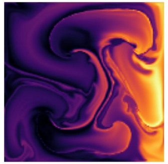
Surrogate Modeling Use Cases

Prediction

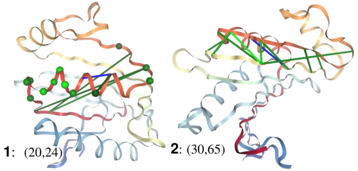
How to use $f_\theta(x)$ to **forecast/simulate** $f(x)$?



Weather forecasting



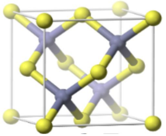
PDE modeling



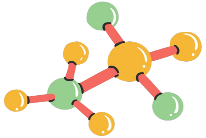
Protein structure prediction

Black-box Optimization

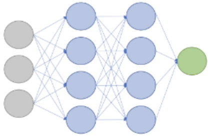
How to use $f_\theta(x)$ to **optimize** $f(x)$?



Material design



Molecular optimization



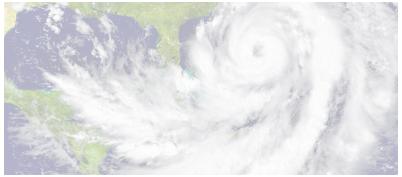
Hyperparameter optimization

Image credit: Penn State University; Gupta, Jayesh K., et al., 2022; Jumper, John, et al., 2021

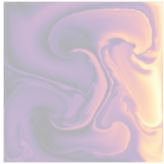
Today's Talk

Prediction

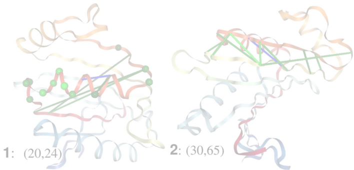
How to use $f_\theta(x)$ to **forecast/simulate** $f(x)$?



Weather forecasting



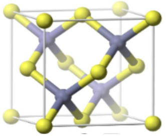
PDE modeling



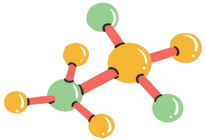
Protein structure prediction

Black-box Optimization

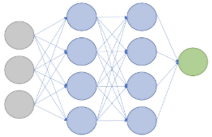
How to use $f_\theta(x)$ to **optimize** $f(x)$?



Material design



Molecular optimization



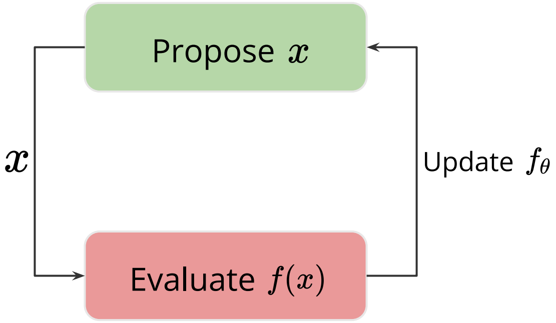
Hyperparameter optimization

Image credit: Penn State University; Gupta, Jayesh K., et al., 2022; Jumper, John, et al., 2021

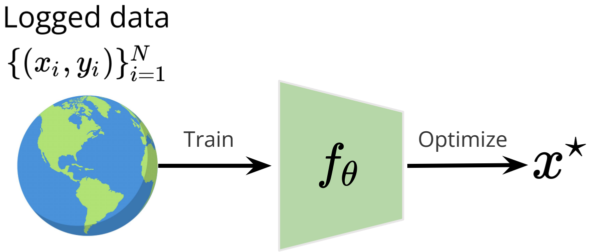
Black-box Optimization

- General Problem: **Optimize** a function **without** its functional form or **gradient** information

$$\boxed{x^\star = \operatorname{argmax} f(x)}$$



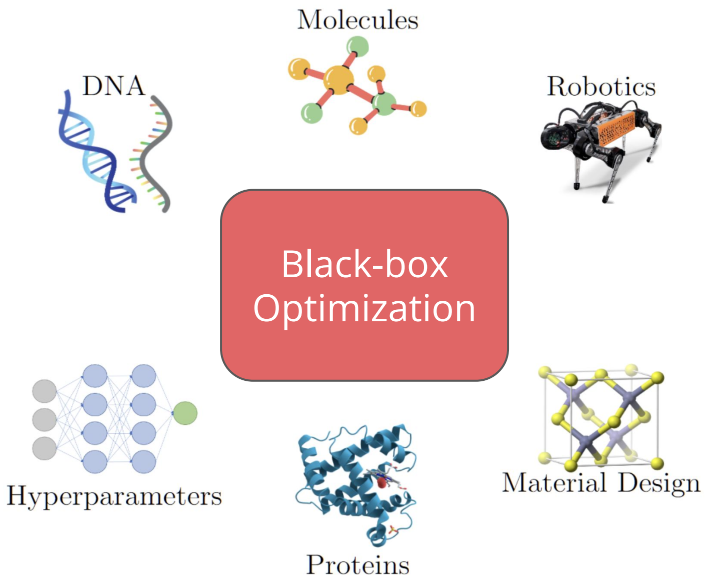
Online optimization

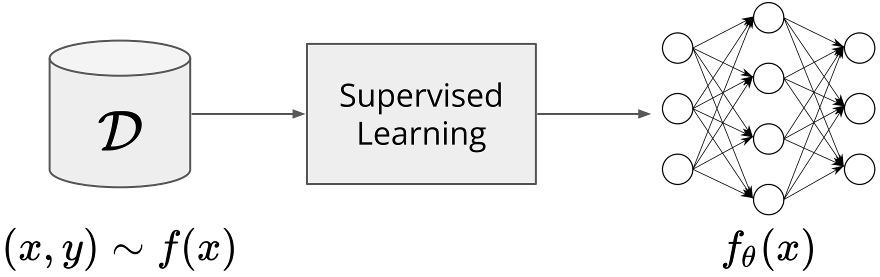


Offline optimization

Limited labeled data but plenty unlabeled data!

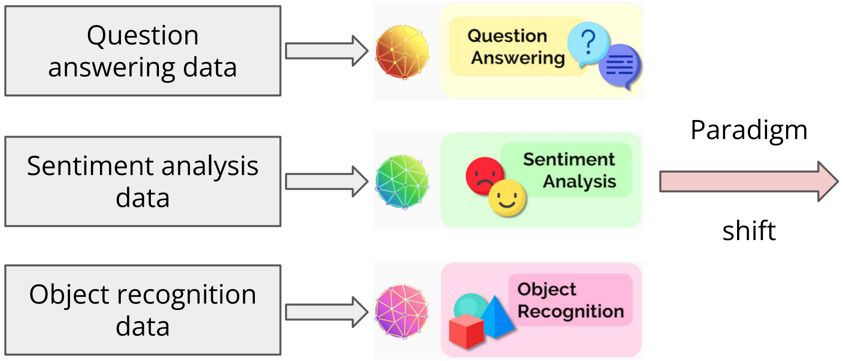
Task-specific Surrogate Models

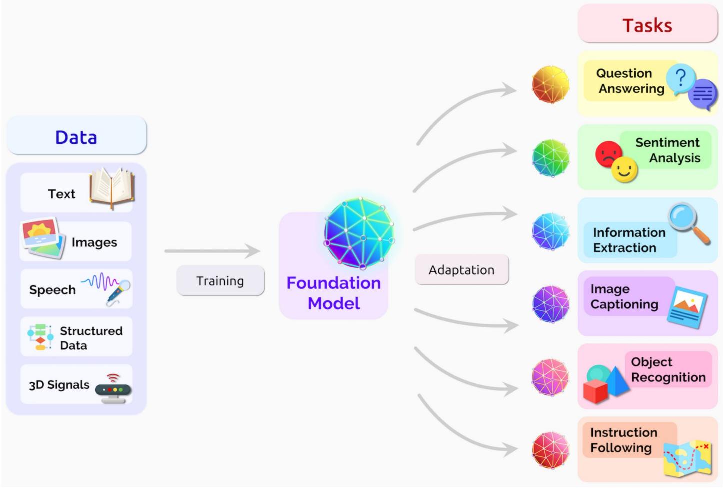


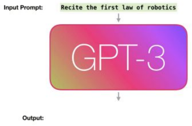


- ❏ Overfitting
- ❏ Train one network for each task
- ❏ Do not make use of unlabeled data

Foundation Models







language
[Vaswani et al., 2017; Radford et al., 2019; Brown et al., 2020; *inter alia*]

vision
[Radford et al., 2019; Ramesh et al., 2021; Rombach et al., 2021; *inter alia*]

- ❑ Unsupervised pretraining
- ❑ Efficiently generalize to unseen tasks

Image credit: Bommasani, Rishi, et al., 2021

GPT-3 Pretraining

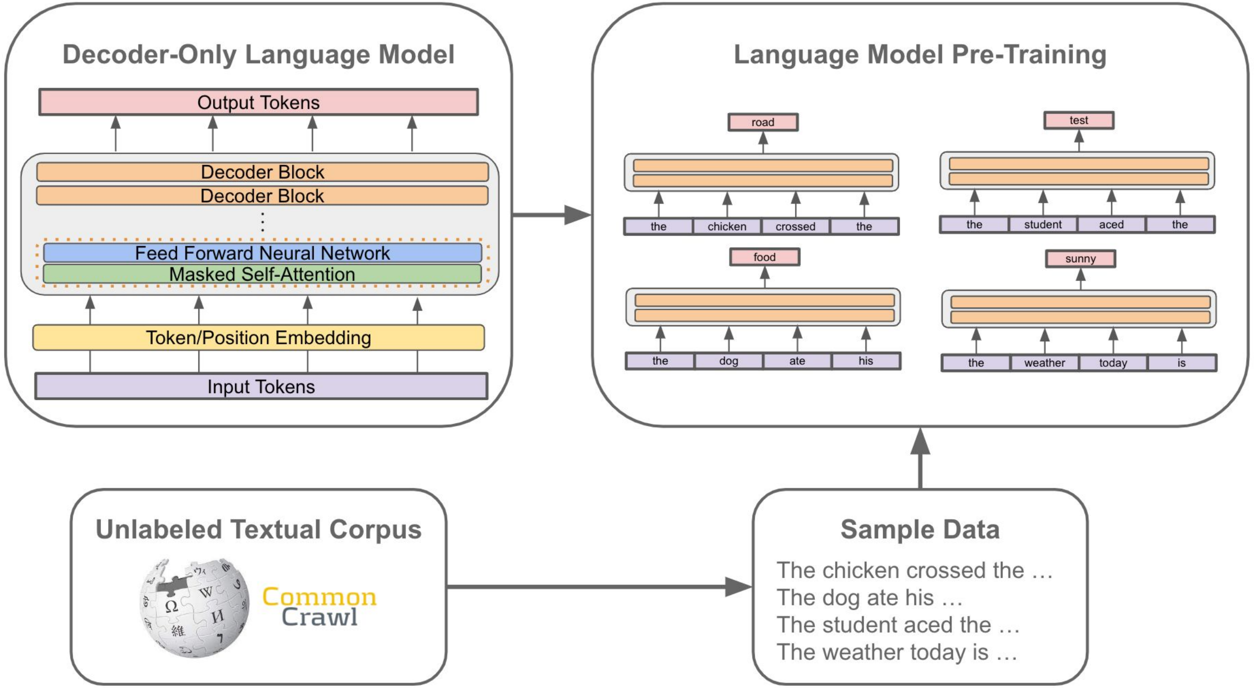
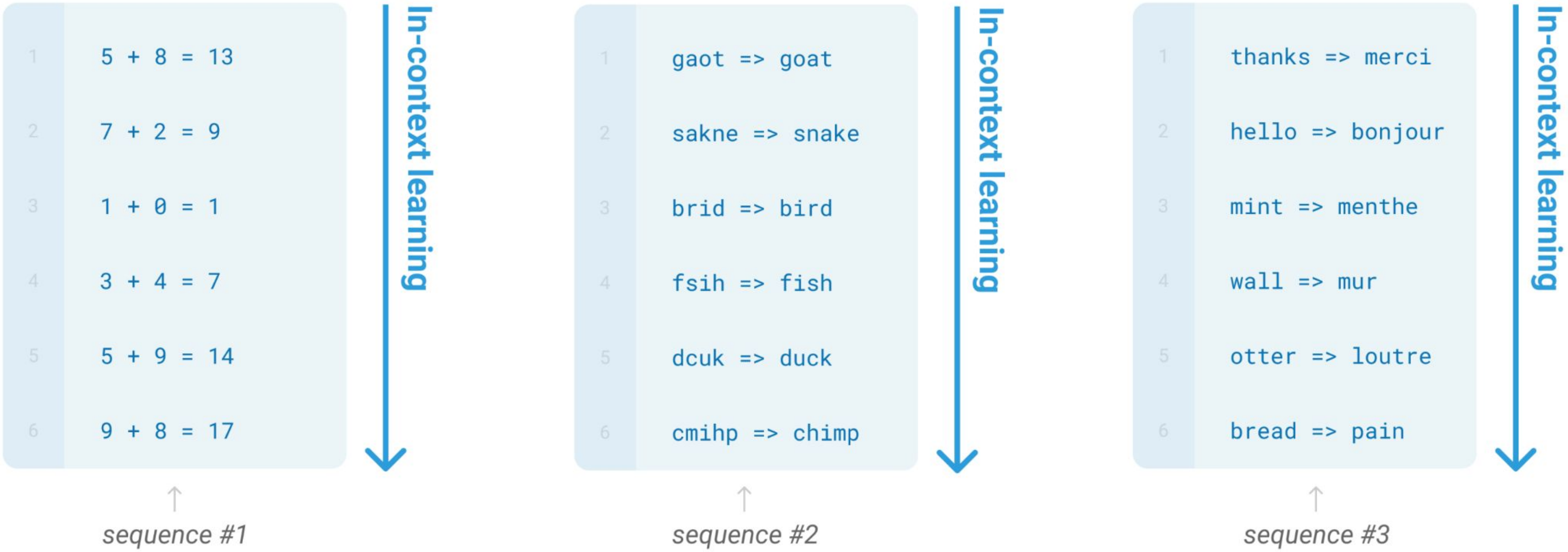


Image credit: https://cameronrwolfe.substack.com/

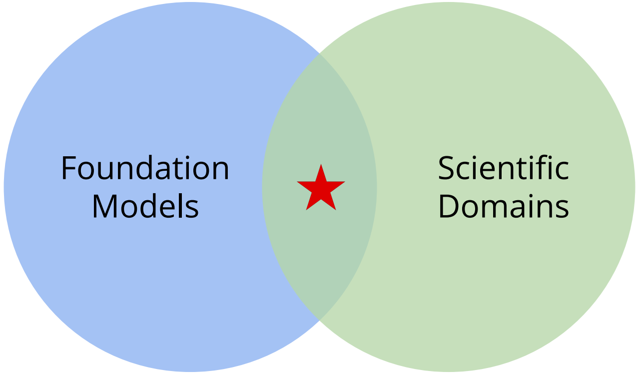
GPT-3 In-context Learning



Emergent in-context learning!

Mann, Ben, et al. "Language models are few-shot learners." NeurIPS 2020.

Today's Talk



Scalable & Generalizable Models

Optimization in Scientific domains

★ Today's talk

Foundation Models for Sciences

Today's Talk

- ❑ ***Part 1:*** Learning to Learn In-context

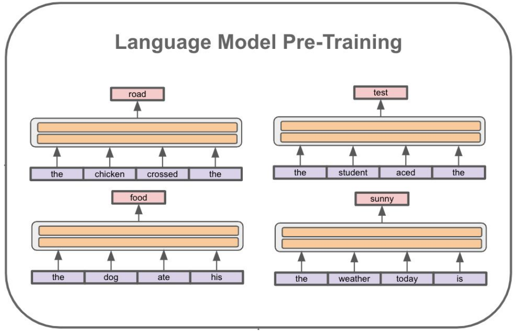
  How to train a model that ***generalizes*** *to* ***unseen functions*** *via* ***in-context conditioning****?*

- ❑ ***Part 2:*** Cross-domain In-context Learning

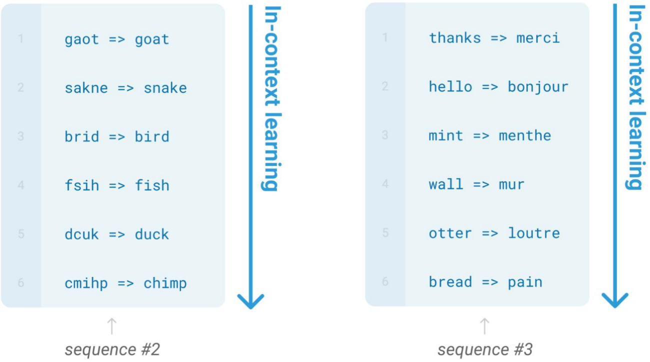
  How to leverage a pretrained LLM for in-context learning in an unseen domain?

Today's Talk

- **Question**: How to train a model to **generalize** to unseen **functions/tasks?**
  - In-context learning is an *emergent capability* of GPT-3 at a certain scale



Large-scale Unsupervised Pretraining



Emergent In-context Learning

Can we *explicitly* train a model to generalize to unseen functions?

Mann, Ben, et al. "Language models are few-shot learners." NeurIPS 2020.

Generalizing over Functions

- **Question**: How to train a model to **generalize** to unseen **functions/tasks?**

Data Points Generalization

- Learning over data points drawn from the same function

$$\mathbb{E}_{x \sim p_{\text{train}}(x), y=f(x)}[\log p(y \mid x)]$$

- The model generalizes to unseen data points

$$x \sim p_{\text{test}}(x), y = f(x)$$

Generalizing over Functions

- **Question**: How to train a model to **generalize** to unseen **functions/tasks?**

Data Points Generalization

- Learning over data points drawn from the same function

$$\mathbb{E}_{x \sim p_{\text{train}}(x), y = f(x)} [\log p(y \mid x)]$$

- The model generalizes to unseen data points

$$x \sim p_{\text{test}}(x), y = f(x)$$

Functional Generalization

- Learning over functions, given finite context of labeled points

$$\boxed{\mathbb{E}_{f \sim \mathcal{F}_{\text{train}}, (x,y), C \sim f} [\log p(y \mid x, C)]}$$

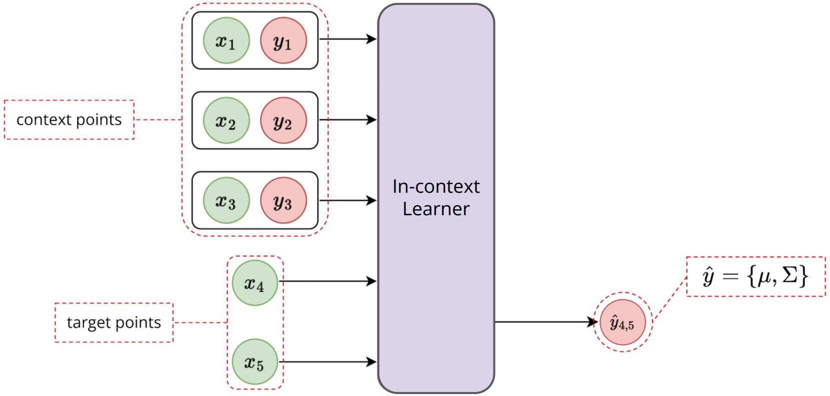
- The model generalizes to unseen functions given the context

$$f \sim \mathcal{F}_{\text{test}}, (x, y), C \sim f$$

Learning to learn in-context

Learning to Learn In-context

- Sample $f \sim \mathcal{F}_{\text{train}}$ and $x_{1:N}, y_{1:N} \sim f$
- Observe context points $\{x_i, y_i\}_{i=1}^{m}$ and make predictions for a set of target points $\{x_j\}_{j=m+1}^{N}$.

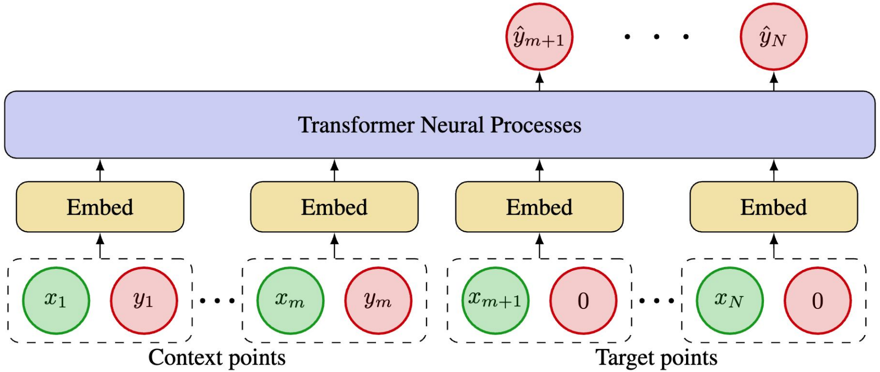


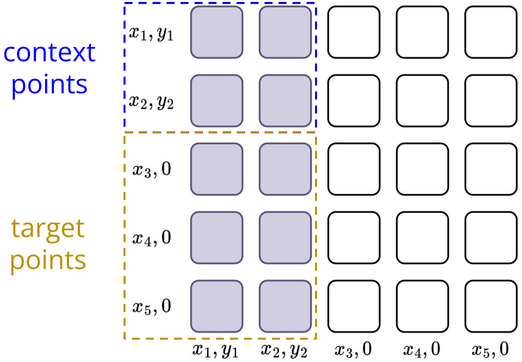
$$\mathcal{L}(\theta) = \mathbb{E}_{f \sim \mathcal{F}_{\text{train}}, x_{1:N}, y_{1:N} \sim f, m}\left[\log p_\theta\left(\underbrace{y_{m+1:N}}_{\text{target outputs}} \mid \underbrace{x_{m+1:N}}_{\text{target inputs}}, \underbrace{x_{1:m}, y_{1:m}}_{\text{context points}}\right)\right]$$

Nguyen, Tung, and Aditya Grover. "Transformer neural processes: Uncertainty-aware meta learning via sequence modeling." ICML 2022.

Model Architecture

❑ We instantiate the framework with a **transformers** architecture called **TNP**





An example mask with *N=5* and *m=2*

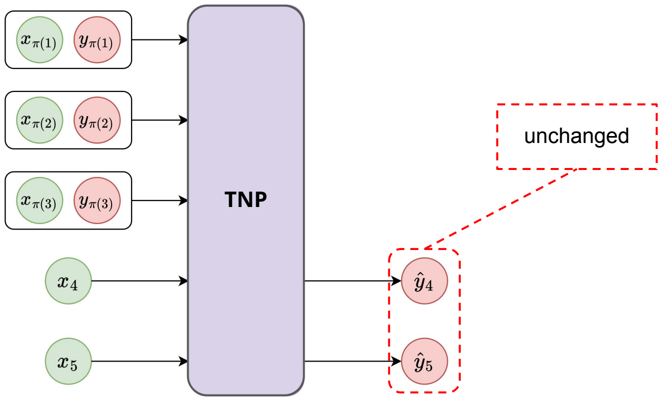
$$\mathcal{L}(\theta) = \mathbb{E}_{f\sim\mathcal{F}_{\text{train}}, x_{1:N}, y_{1:N}\sim f, m}\left[\log p_\theta\left(y_{m+1:N} \mid x_{m+1:N}, x_{1:m}, y_{1:m}\right)\right]$$

$$= \mathbb{E}_{f\sim\mathcal{F}_{\text{train}}, x_{1:N}, y_{1:N}\sim f, m}\left[\sum_{i=m+1}^{N} \log p(y_i \mid x_i, x_{1:m}, y_{1:m})\right]$$

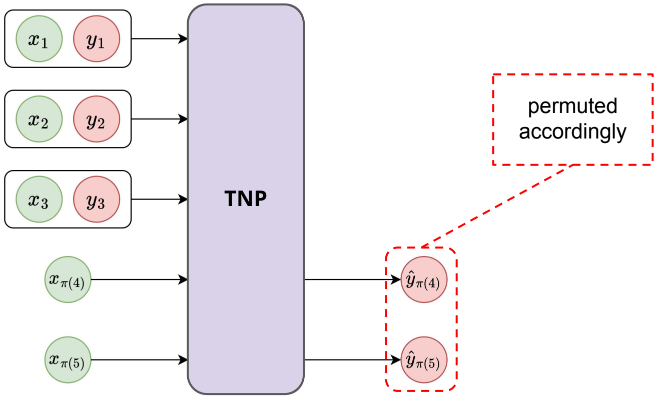
Nguyen, Tung, and Aditya Grover. "Transformer neural processes: Uncertainty-aware meta learning via sequence modeling." ICML 2022.

Properties of TNP

Property 1. *Context invariance.*
The model's predictions do not depend on the permutation of the context points.



Property 2. *Target equivariance.*
Whenever we permute the target inputs, the predictions are permuted accordingly.



Nguyen, Tung, and Aditya Grover. "Transformer neural processes: Uncertainty-aware meta learning via sequence modeling." ICML 2022.

Variants of TNPs

We introduce TNP-A and TNP-ND to improve expressivity

Autoregressive TNP (TNP-A)

Predict the target jointly with an autoregressive factorization:

$$p_\theta\left(y_{m+1:N} \mid x_{1:N}, y_{1:m}\right) \\ = \prod_{i=m+1}^{N} p_\theta\left(y_i \mid x_{1:i}, y_{1:i-1}\right)$$

Non-Diagonal TNP (TNP-ND)

Predict the target jointly using a multivariate Gaussian distribution with a non-diagonal covariance matrix:

$$p_\theta\left(y_{m+1:N} \mid x_{1:N}, y_{1:m}\right) \\ = \mathcal{N}\left(y_{m+1:N} \mid \mu_\theta\left(x_{1:N}, y_{1:m}\right), \Sigma_\theta\left(x_{1:N}, y_{1:m}\right)\right)$$

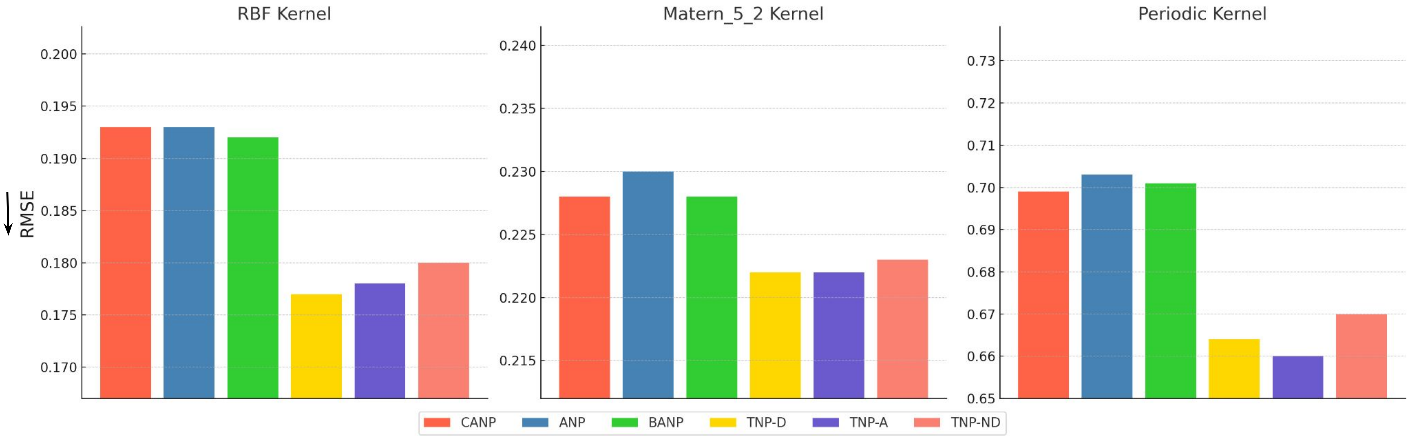
Nguyen, Tung, and Aditya Grover. "Transformer neural processes: Uncertainty-aware meta learning via sequence modeling." ICML 2022.

TNPs for 1D Regression

- Train on functions generated from **GPs** with an **RBF kernel**

$$f \sim \mathcal{GP}(0, \mathcal{K}), \quad \mathcal{K}(x, x') = \sigma^2 \exp\left(-\frac{(x-x')^2}{2\ell^2}\right)$$

- Test on functions generated from **GPs** with **different kernels**



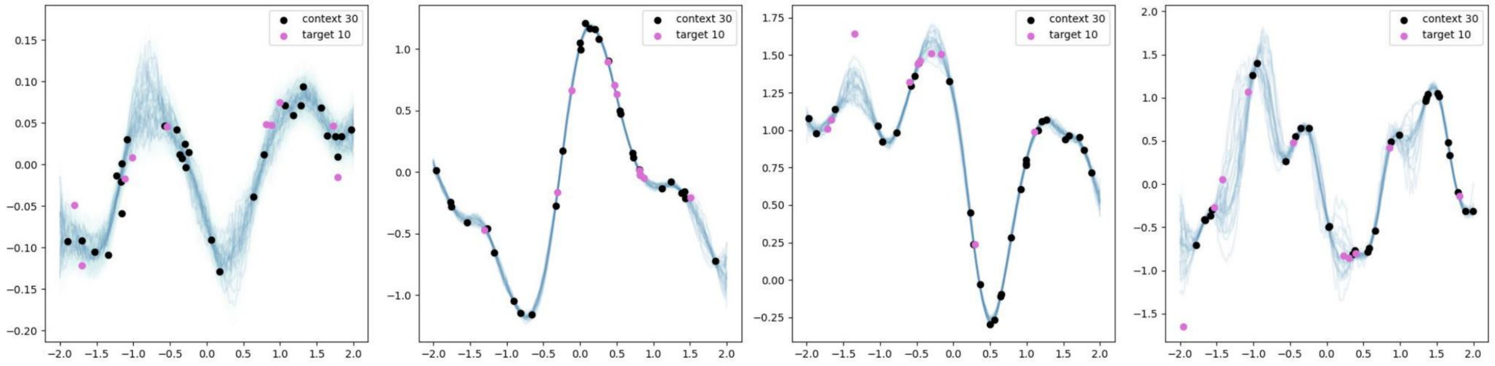
Nguyen, Tung, and Aditya Grover. "Transformer neural processes: Uncertainty-aware meta learning via sequence modeling." ICML 2022.

TNPs for 1D Regression

- Train on functions generated from **GPs** with an **RBF kernel**

$$f \sim \mathcal{GP}(0, \mathcal{K}), \quad \mathcal{K}(x, x') = \sigma^2 \exp\left(-\frac{(x-x')^2}{2\ell^2}\right)$$

- Test on functions generated from **GPs** with **different kernels**



TNP generalizes well to unseen functions!

Nguyen, Tung, and Aditya Grover. "Transformer neural processes: Uncertainty-aware meta learning via sequence modeling." ICML 2022.

TNPs for Image Completion

- ❑ Train to map from pixel coordinates → pixel values
- ❑ Each image is a 2-dimensional function

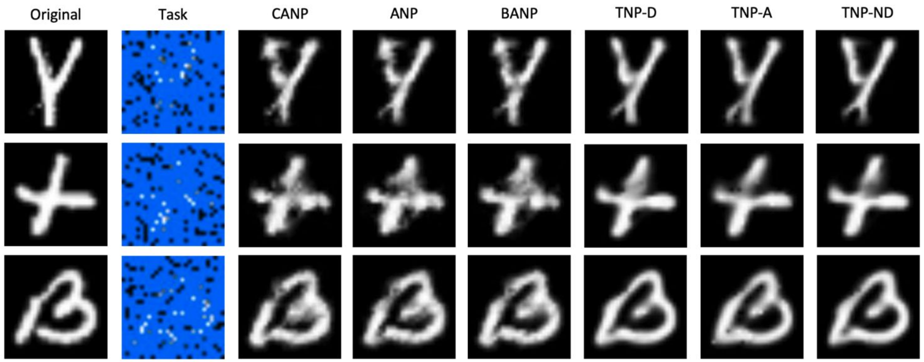
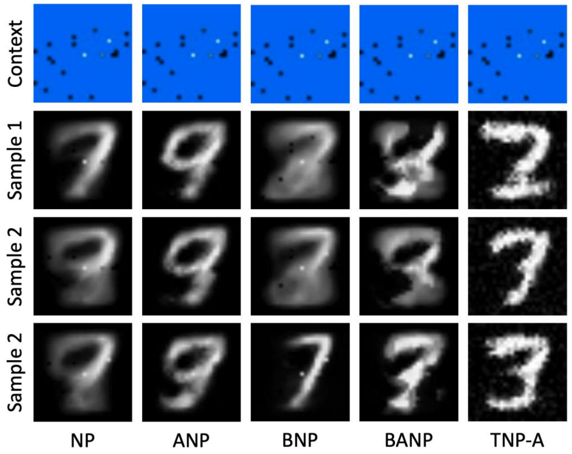


Image completion on unseen classes from 100 pixels



Samples from 20 context points.

Discussion

- ❏ TNP learns to learn in-context **from scratch**
  - ❏ It’s often desired in deep learning to start from a pretrained model
- ❏ There exists powerful in-context models in other domains, e.g., large language models.

*Can we **transfer** in-context learning from **LLMs** to **new domains**?*

Today's Talk

- ❏ ***Part 1:*** Learning to Learn In-context ✓

  *How to train a model that **generalizes** to **unseen functions** via **in-context conditioning**?*

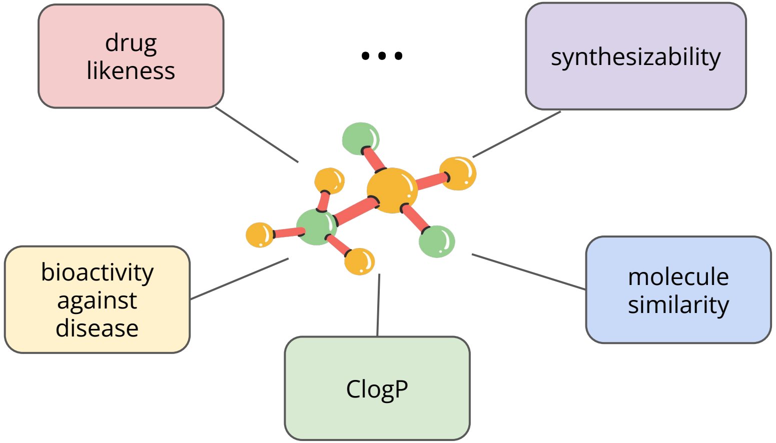
- ❏ ***Part 2:*** Cross-domain In-context Learning

  How to leverage a pretrained LLM for in-context learning in an unseen domain?

Molecular Optimization

- Problem: find the molecule that optimizes a certain property.

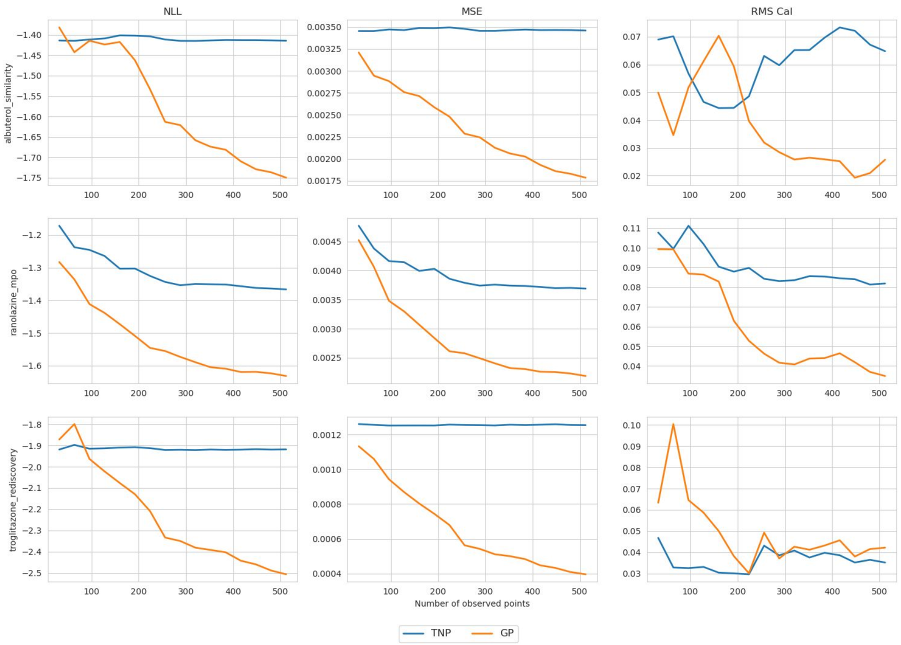
$$x^* = \arg\min_{x \in \mathcal{X}} f(x)$$



- Important domain and distinguished from language.

TNPs for Molecular Optimization

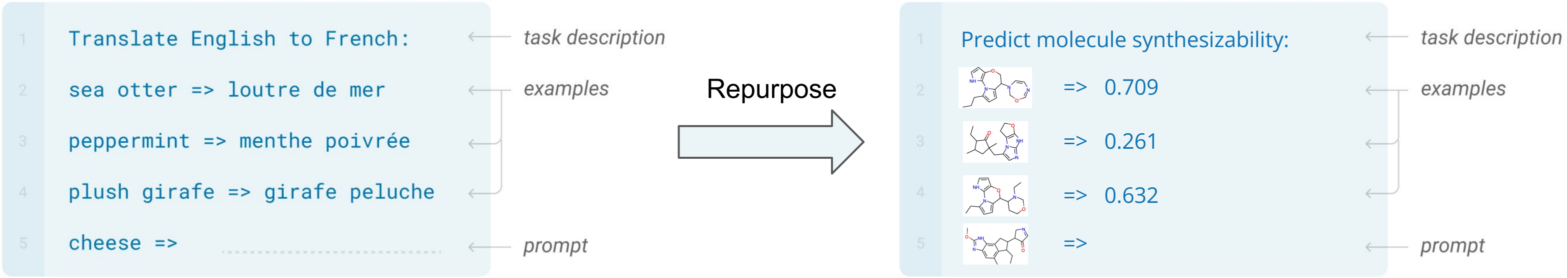
- First attempt: TNPs for in-context molecular property prediction



Solution: Use a more capable model, i.e., a pretrained LLM

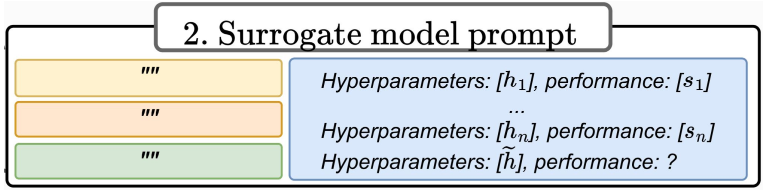
LLMs for Molecular Optimization

- ❏ Motivation: Repurpose a pretrained LLM for in-context learning in a new domain.



LLMs for Optimization

- Existing works directly prompt a pretrained LLM in the text space.
  - Not applicable to non-textual domains ❌
  - Do not generalize to underrepresented domains ❌



Liu, Tennison, et al. "Large Language Models to Enhance Bayesian Optimization." arXiv preprint arXiv:2402.03921 (2024).

Your task is to generate the instruction <INS>. Below are some previous instructions with their scores. The score ranges from 0 to 100.

text:
Let's figure it out!
score:
61

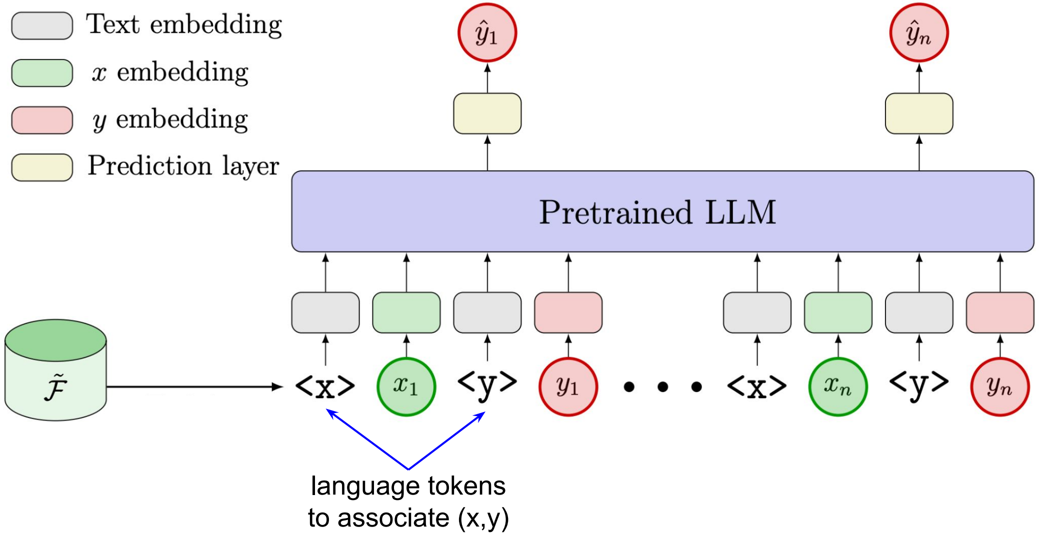
text:
Let's solve the problem.
score:
63

Generate an instruction that is different from all the instructions <INS> above, and has a higher score than all the instructions <INS> above. The instruction should begin with <INS> and end with </INS>. The instruction should be concise, effective, and generally applicable to all problems above.

Yang, Chengrun, et al. "Large language models as optimizers." arXiv preprint arXiv:2309.03409 (2023).

BOLM Architecture

- Learn separate embedding and prediction layers for the new domain
  - Domain-agnostic ✓
  - Computationally efficient ✓

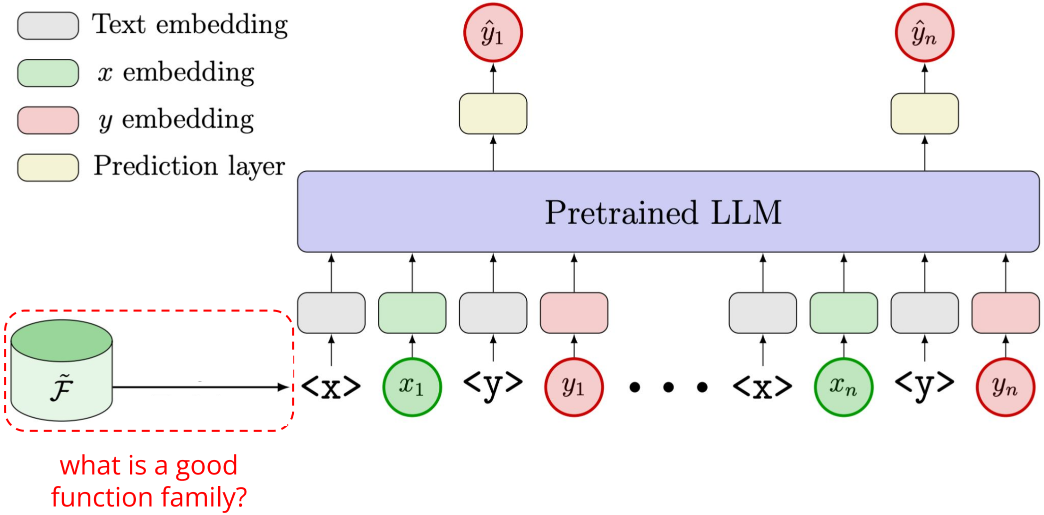


Nguyen, Tung, and Aditya Grover. "BOLM: Leveraging Large Language Models for Black-box Optimization." preprint 2024.

BOLM Training

- Train BOLM to perform in-context learning prediction

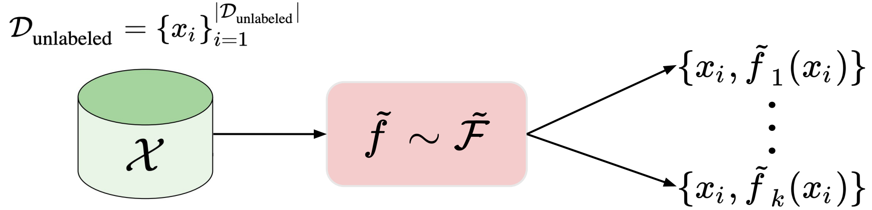
$$\mathcal{L}(\theta) = \mathbb{E}_{\tilde{f} \sim \tilde{F}, x_{1:N}, y_{1:N}} \left[\sum_{i=1}^{N} \log p_\theta \left(y_i \mid x_i, x_{<i}, y_{<i} \right) \right]$$



Nguyen, Tung, and Aditya Grover. "BOLM: Leveraging Large Language Models for Black-box Optimization." preprint 2024.

Training Data Generation

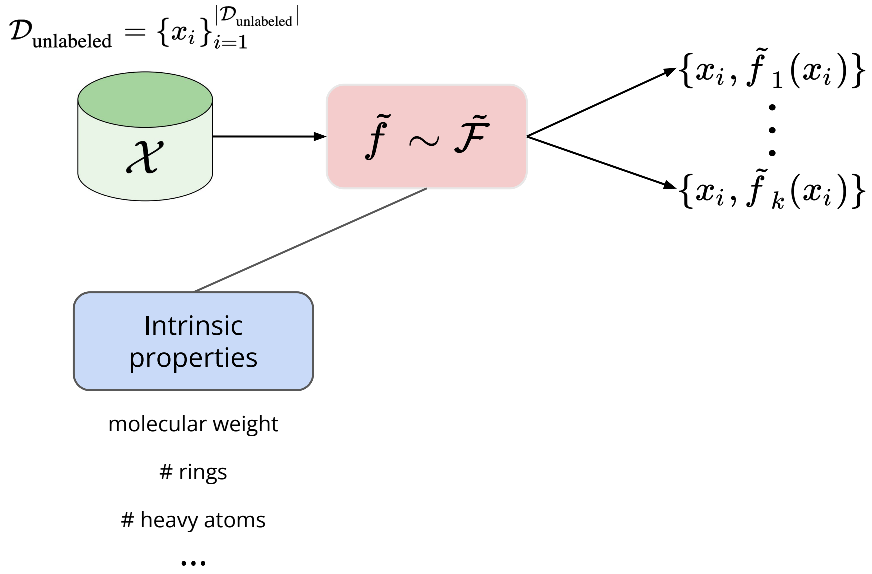
- Ideal family of functions $\tilde{\mathcal{F}}$
  - Close to downstream functions
  - Diverse



Nguyen, Tung, and Aditya Grover. "BOLM: Leveraging Large Language Models for Black-box Optimization." preprint 2024.

Training Data Generation

- Ideal family of functions $\tilde{\mathcal{F}}$
  - Close to downstream functions ✓
  - Diverse

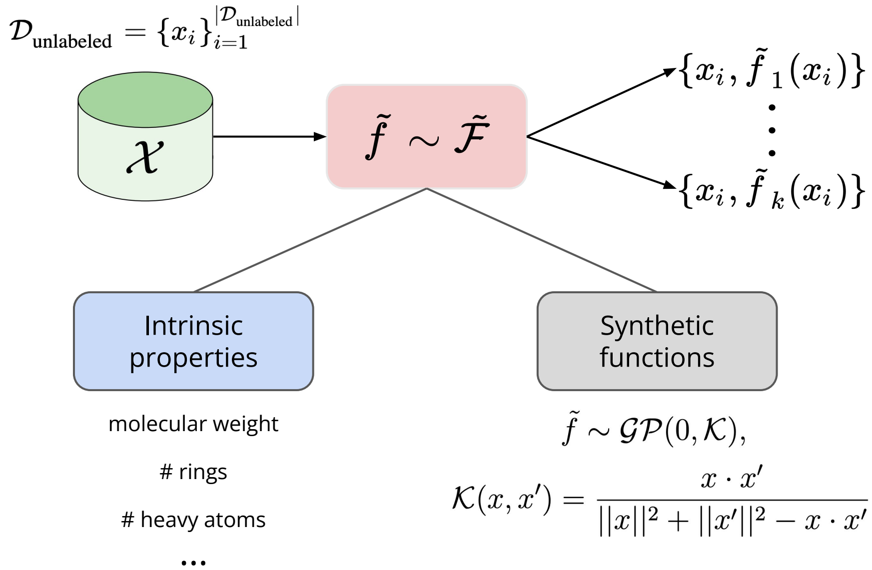


Nguyen, Tung, and Aditya Grover. "BOLM: Leveraging Large Language Models for Black-box Optimization." preprint 2024.

Training Data Generation

- Ideal family of functions $\tilde{\mathcal{F}}$
  - Close to downstream functions ✓
  - Diverse ✓

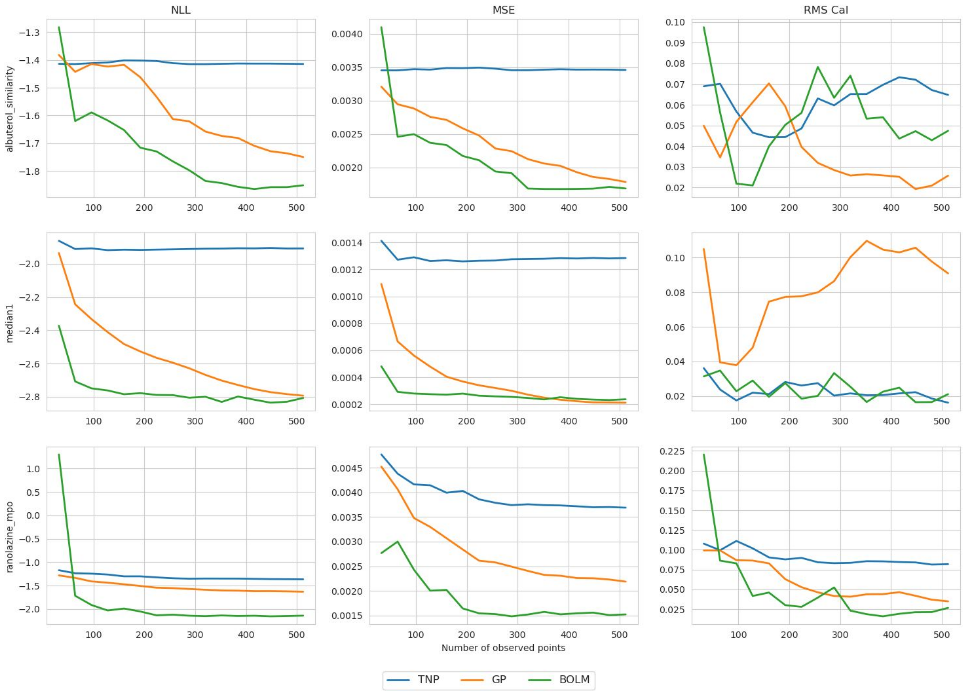
Semi-synthetic training!



Nguyen, Tung, and Aditya Grover. "BOLM: Leveraging Large Language Models for Black-box Optimization." preprint 2024.

Prediction Results

- Test BOLM on in-context molecular property prediction



Nguyen, Tung, and Aditya Grover. "BOLM: Leveraging Large Language Models for Black-box Optimization." preprint 2024.

BOLM for Black-box Optimization

- We ultimately care about optimization
- We consider the online setting

Algorithm 1 Black-box optimization with BOLM

Require: objective f, fine-tuned LLM surrogate f_θ, budget B, candidate pool size C, acquisition function α, batch size k

Initialize $\mathcal{D}_{\text{obs}} = \{\}$

while $|\mathcal{D}_{\text{obs}}| < B$ **do**

  Generate a set of candidates $\{x_i\}_{i=1}^{C}$ — Evolutionary

  for each candidate x_i **do**

    Predict $\mu_i, \sigma_i = f_\theta(x_i, \mathcal{D}_{\text{obs}})$ — BOLM

    Compute utility score $u_i = \alpha(\mu_i, \sigma_i)$ — UCB

  end for

  Select k candidates with the highest utility scores

  for each selected candidate x_j **do**

    Evaluate x_j using the actual objective $y_j = f(x_j)$

    Add (x_j, y_j) to the observation dataset $\mathcal{D}_{\text{obs}}$

  end for

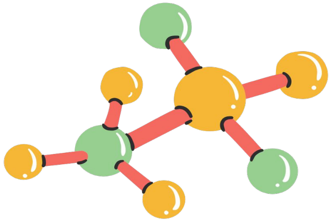
end while

Nguyen, Tung, and Aditya Grover. "BOLM: Leveraging Large Language Models for Black-box Optimization." preprint 2024.

PMO Benchmark

- We test (a single) BOLM on 21 benchmark optimization tasks from PMO

Task
albuterol_similarity
amlodipine_mpo
celecoxib_rediscovery
deco_hop
drd2
fexofenadine_mpo
isomers_c7h8n2o2
isomers_c9h10n2o2pf2cl
median1
median2

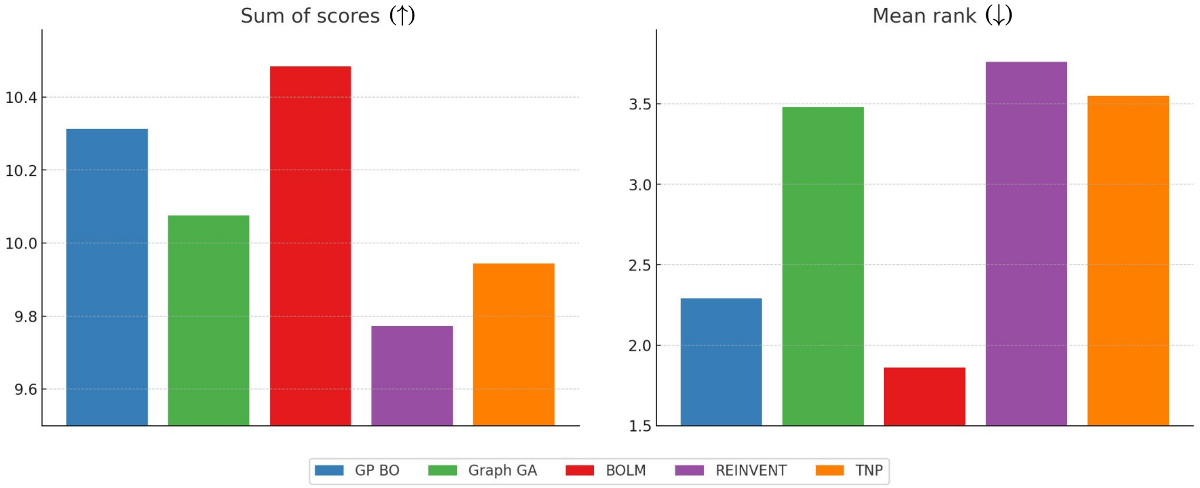


Task
mestranol_similarity
osimertinib_mpo
perindopril_mpo
qed
ranolazine_mpo
scaffold_hop
sitagliptin_mpo
thiothixene_rediscovery
troglitazone_rediscovery
valsartan_smarts
zaleplon_mpo

Nguyen, Tung, and Aditya Grover. "BOLM: Leveraging Large Language Models for Black-box Optimization." preprint 2024.

Experiments

- We test (a single) BOLM on 21 benchmark optimization tasks

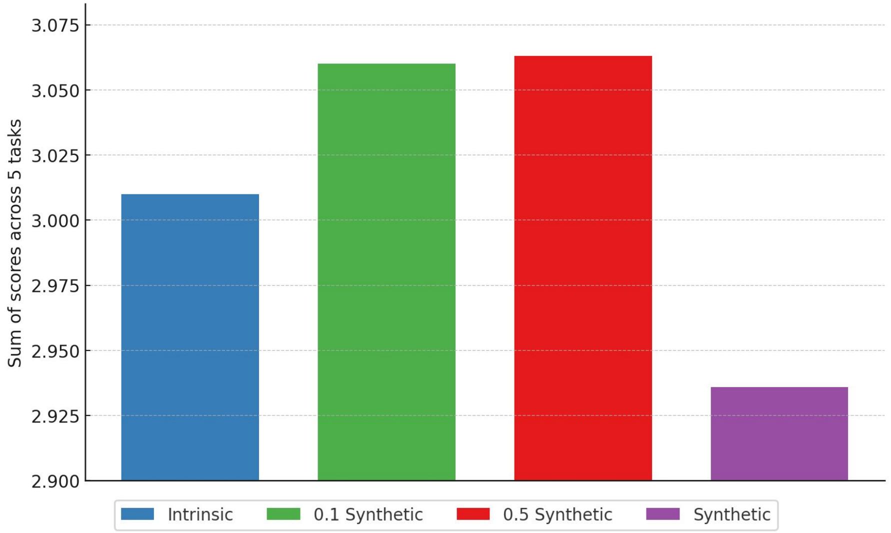


BOLM achieves the best overall scores and ranking!

Nguyen, Tung, and Aditya Grover. "BOLM: Leveraging Large Language Models for Black-box Optimization." preprint 2024.

Ablation Studies

- Importance of semi-synthetic training

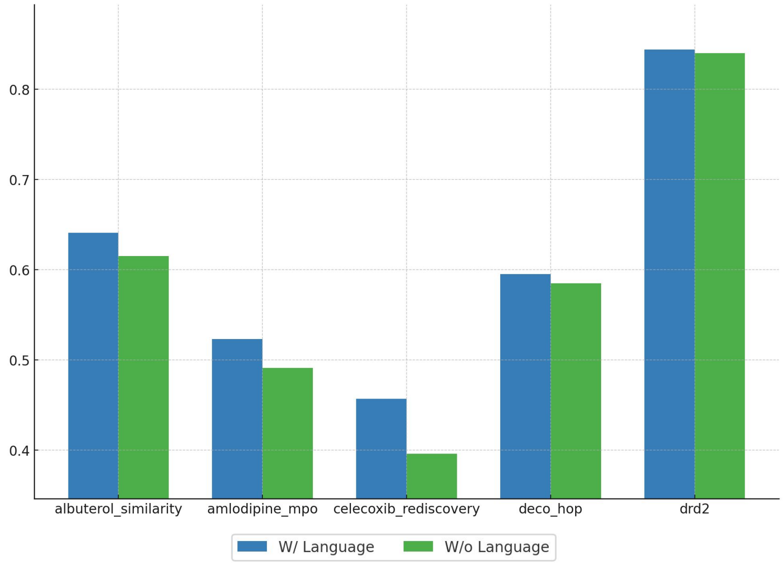


Semi-synthetic is better than intrinsic or synthetic alone

Nguyen, Tung, and Aditya Grover. "BOLM: Leveraging Large Language Models for Black-box Optimization." preprint 2024.

Ablation Studies

- ❑ Importance of language instruction

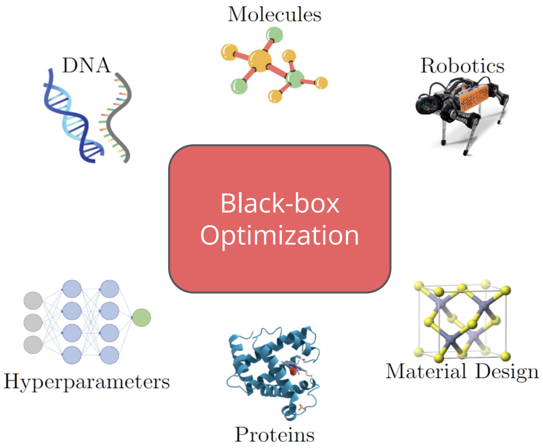


Language instruction is crucial to BOLM

Nguyen, Tung, and Aditya Grover. "BOLM: Leveraging Large Language Models for Black-box Optimization." preprint 2024.

Summary

- ❑ We can train a model to learn in-context
  - ❑ In-context learning can **generalize** from **synthetic to real** functions
  - ❑ In-context learning can be **transferred across domains**
- ❑ End goal: have a GPT-like model that scientists can use to optimize **arbitrary objectives**



Future Work

- Improve generality
  - Foundation models that can perform cross-domain optimization
- Improve performance
  - Scale up data and model size
  - Parameter-efficient finetuning
- Foundation models for end-to-end optimization
  - Candidate generation, exploration, surrogate modeling, etc.

Thank You!
Q&A