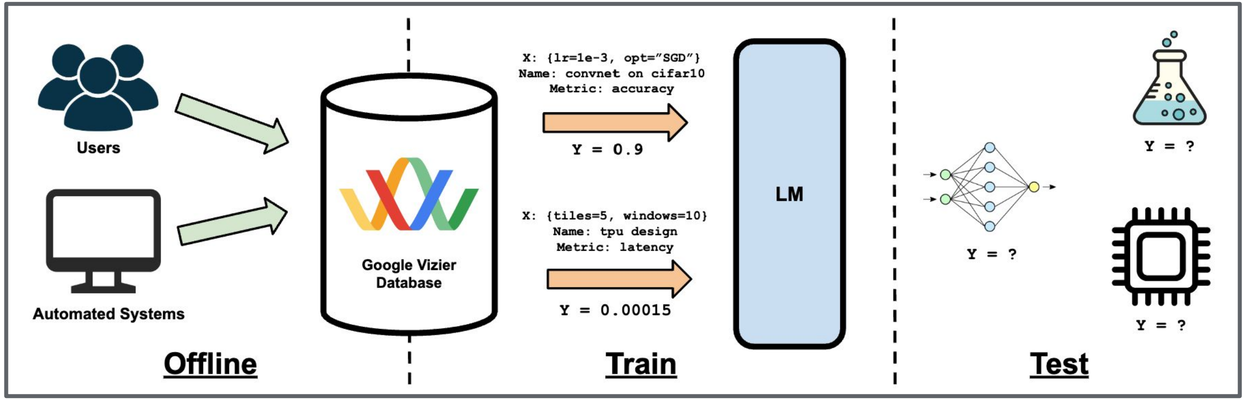
OmniPred

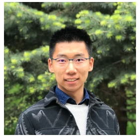
Regression with Language Models (?!)

Xingyou Song, Oscar Li, Chansoo Lee, Bangding Yang, Daiyi Peng, Sagi Perel, Yutian Chen
Google DeepMind, CMU, Google



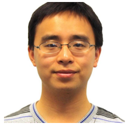
Credits

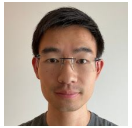
OmniPred Project


Oscar Li


Richard Song


Chansoo Lee


Yutian Chen


Jeffrey Yang

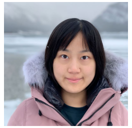

Daiyi Peng


Sagi Perel

Extended Collaborators + Support (Former + Current)


Richard Zhang


David Dohan


Zi Wang


Kazuya Kawakami

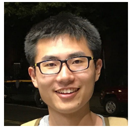

Greg Kochanski


Arnaud Doucet


Marc'aurelio Ranzato

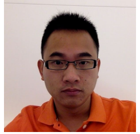

Lior Belenki


Nando de Freitas


Chengrun Yang


JD Co-Reyes


Aviral Kumar


Yingjie Miao

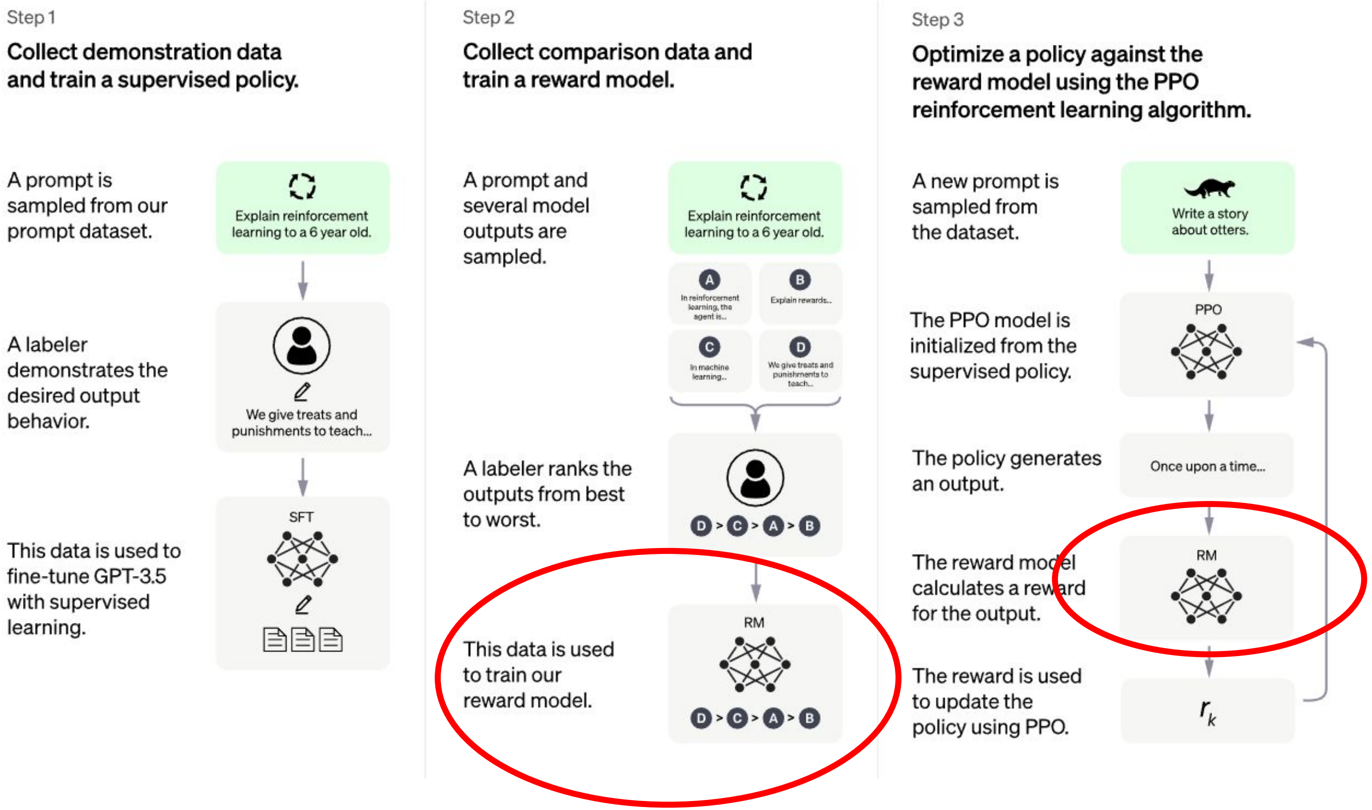

Timothy Chu


Daniel Golovin

Current State of ML

(AutoML Research Perspective)

RL Fine-Tuning



RL-HF: RL from *Human Feedback*

Humans dictate subjective ratings:

- Creativity + personality
- Safety
- (Human-verifiable) Factuality

$$y \in \{0, 1\} \text{ or } [0, 1]$$

Great for Human Interaction:

- Writing prose (poetry, essays)
- Conversations
- Likeability / Human Values

Poor For:

- Experiment Prediction
- Business Metrics
- Forecasting

Reward Models from the AutoML Lens

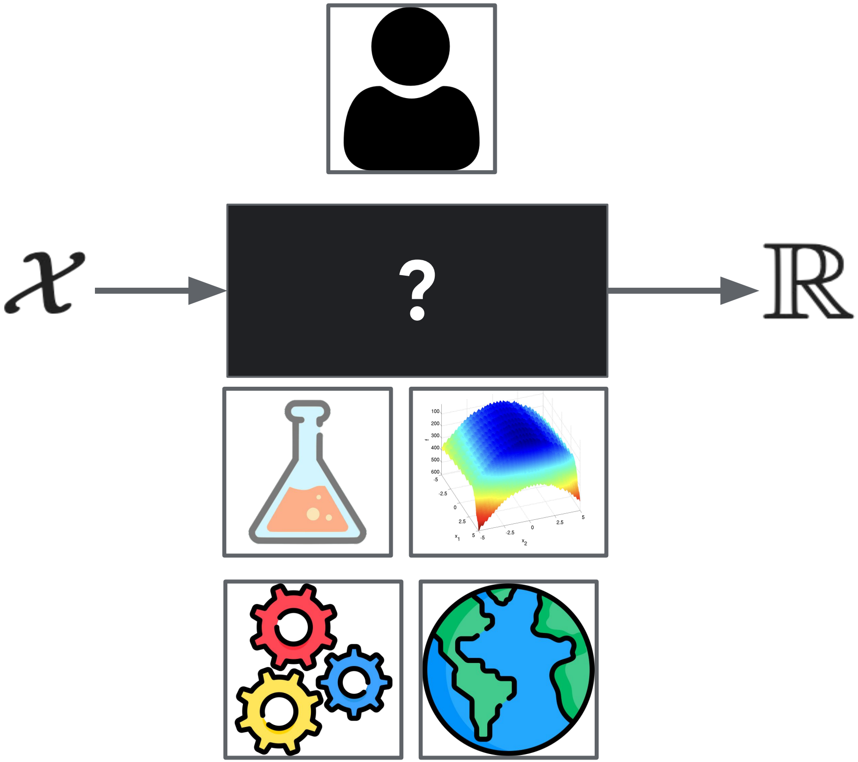
Reward Models are more broadly, LM regressors over blackbox objectives:

$$f : \mathcal{X} \to \mathbb{R}$$

Current focus on **human feedback**, but should be extended to **<u>any blackbox function.</u>**

Examples:

- Outcomes of expensive experiments
- Synthetic / Mathematical Expressions
- Metrics over production systems
- Environmental feedback



Different Names for Different Communities

LLM Lingo

- Reward Model
- AutoRater

AutoML Lingo

- Regression
- Performance Prediction
- Surrogate

Regression with Language Models

Description

Regression in experimental design.

x = Parameters to Blackbox Function

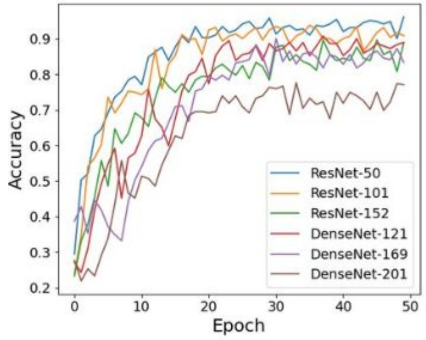
- Float (ex: Learning Rate)
- Category (ex: SGD or Adam)
- Integer (ex: Number of Layers)

y = Scalar Metric (ex: Accuracy)

$$y = f(x)$$

Prompt (X):
I'm training a ResNet-52 on CIFAR-10 with hyperparameters batch size=256 and learning_rate=0.01 with SGD, over 100 epochs. Predict accuracy?

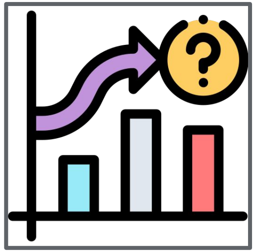
Target (Y):
0.912



Benefits

LLM Community

- Reward Models Simulating Systems / Nature
- More precise reward modeling



Stock Market

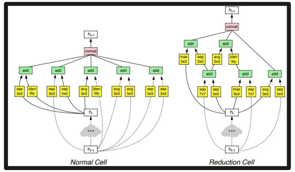
World News

AutoML Community

- Flexible text-based regression
- Multi-task + meta-learning
- Realistic surrogate-based benchmarks

```
"name": "gan1d 500 iters -
"2022-05-18"
"parameter": {
  "name": "learning_rate",
```

Hyperparameters



Architecture

```
import tensorflow as tf
mnist = tf.keras.datasets.mnist

(x_train, y_train),(x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0

model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(),
  tf.keras.layers.Dense(512, activation=tf.nn.relu),
  tf.keras.layers.Dropout(0.2),
  tf.keras.layers.Dense(10, activation=tf.nn.softmax)
])
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

model.fit(x_train, y_train, epochs=5)
model.evaluate(x_test, y_test)
```

Code

Ex: Could we speedup program search?

RESEARCH

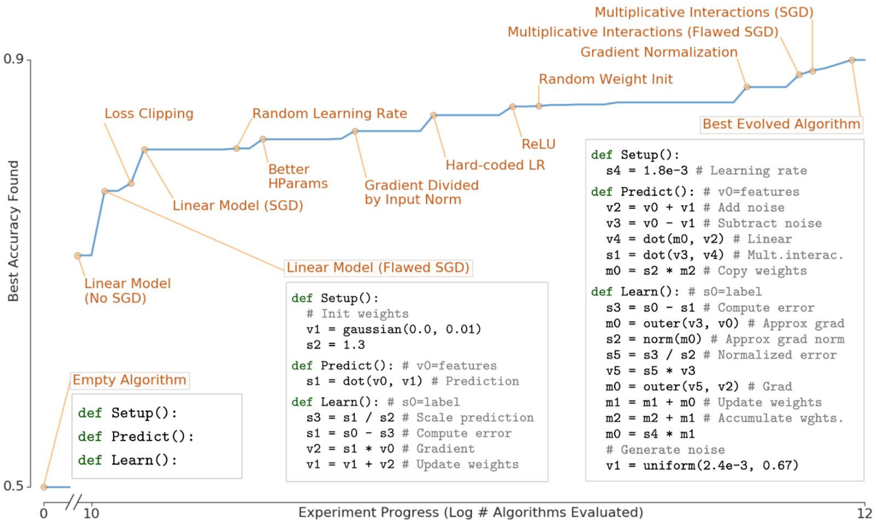
FunSearch: Making new discoveries in mathematical sciences using Large Language Models

14 DECEMBER 2023

Alhussein Fawzi and Bernardino Romera Paredes

Share

AutoML-Zero (Real, 2020)



OmniPred Details

Serialization

Standard “Prefix LM” Training: (**Prompt**, **Response**)

- X: Function input
- M: Metadata describing function
- Y: Objective value

	Language Model Textual Representation
x	batch_size:128,kernel:'rbf',learning_rate:0.5,model:'svm',optimizer:'sgd'
m	title:'classification',user:'some-person',description:'spam detection', objective:'accuracy'
y	<+><1><2><3><E-2>

Serialization (X)

- Natural Language / JSON
- Variable length / Parameter count
- Raw value, no normalization!

	Language Model Textual Representation
x	batch_size:128,kernel:'rbf',learning_rate:0.5,model:'svm',optimizer:'sgd'
m	title:'classification',user:'some person',description:'spam detection', objective:'accuracy'
y	<+><1><2><3><E-2>

Serialization (M)

- Natural Language
  - Shove in anything
- Conditions distribution
  - Important: Username, title, objective

	Language Model Textual Representation
x	batch_size:128,kernel:'rbf',learning_rate:0.5,model:'svm',optimizer:'sgd'
m	title:'classification',user:'some-person',description:'spam detection', objective:'accuracy'
y	<+><1><2><3><E-2>

Serialization (Y)

- Fixed-length custom tokens
  - (sign, mantissa, exponent)
- Restricted decoding (logit masking)
  - Always output correct tokens
- Raw value, no normalization!

	Language Model Textual Representation
x	batch_size:128,kernel:'rbf',learning_rate:0.5,model:'svm',optimizer:'sgd'
m	title:'classification',user:'some-person',description:'spam detection', objective:'accuracy'
y	<+><1><2><3><E-2>

Regression Paradigm Change

Dynamic Input Spaces: (X) serialization ignores search space bounds

Multitask: Just look at (M)

No Tensorization: Avoid fixed-length embeddings

No Rescaling/Normalization: Avoid numerical instability issues

Regressor	Dynamic Input Spaces?	Can Multitask?	Tensorize?	Rescale?
MLP	No	Only fixed spaces	Yes	Yes
Tree-based	No	Only fixed spaces	Yes	Optional
Gaussian Process (GP)	No	Only fixed spaces	Yes	Yes
GNN / Transformer / RNN	No	Only fixed domains	Yes	Yes
OMNIPRED (Ours)	**Yes**	**Yes**	**No**	**No**

What about new data?

Required for regressor-guided search, new problems, etc.

Finetune from pretrained model on new data

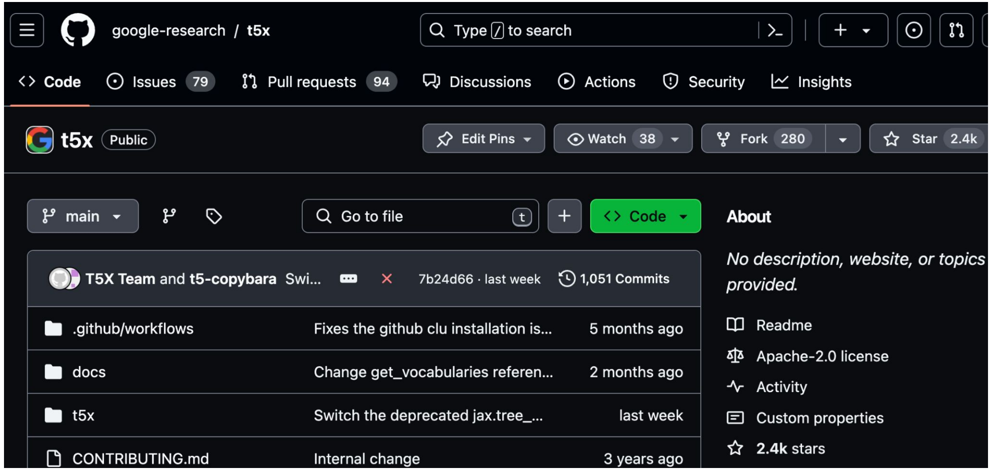
- (Optional) LoRA for efficiency

$$\theta_{ft} = \theta_{pt} - \nabla \ell_{(x,y)\sim\mathcal{D}_{train}}(\theta)$$

Custom LM

Basic 12-layer T5X Encoder-Decoder (200M Params)

- **<u>No English pretraining.</u>**
- **~8 GPU for Training, 1 GPU inference**



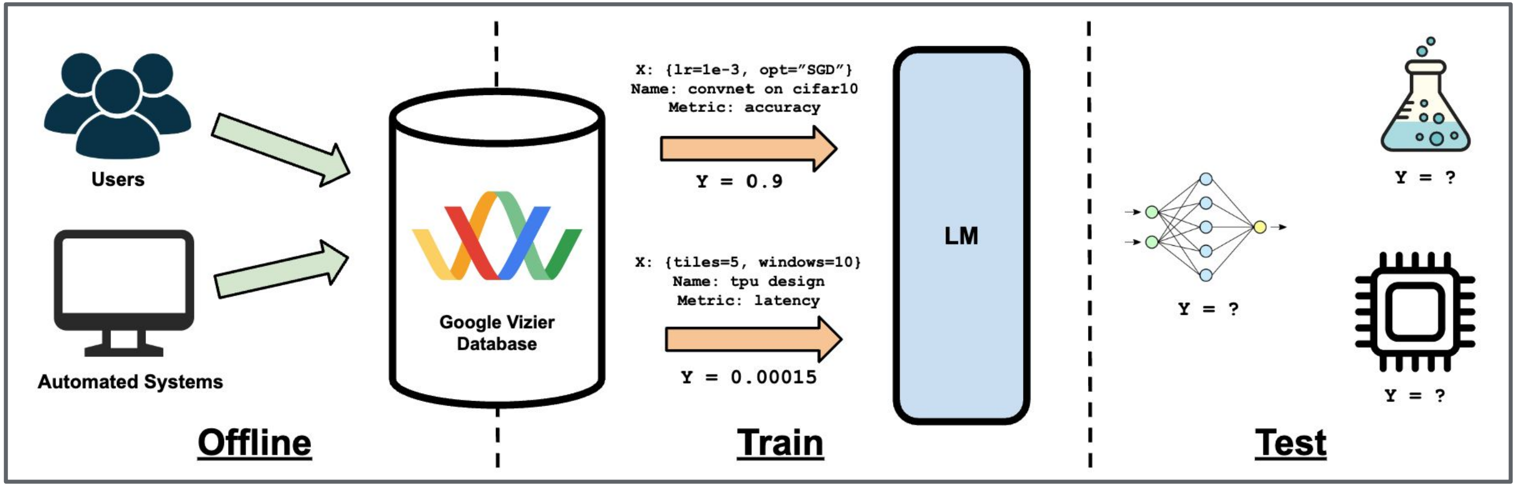
Training Data + Evaluation

Using Google Vizier Data

Convenient source of regression data from:

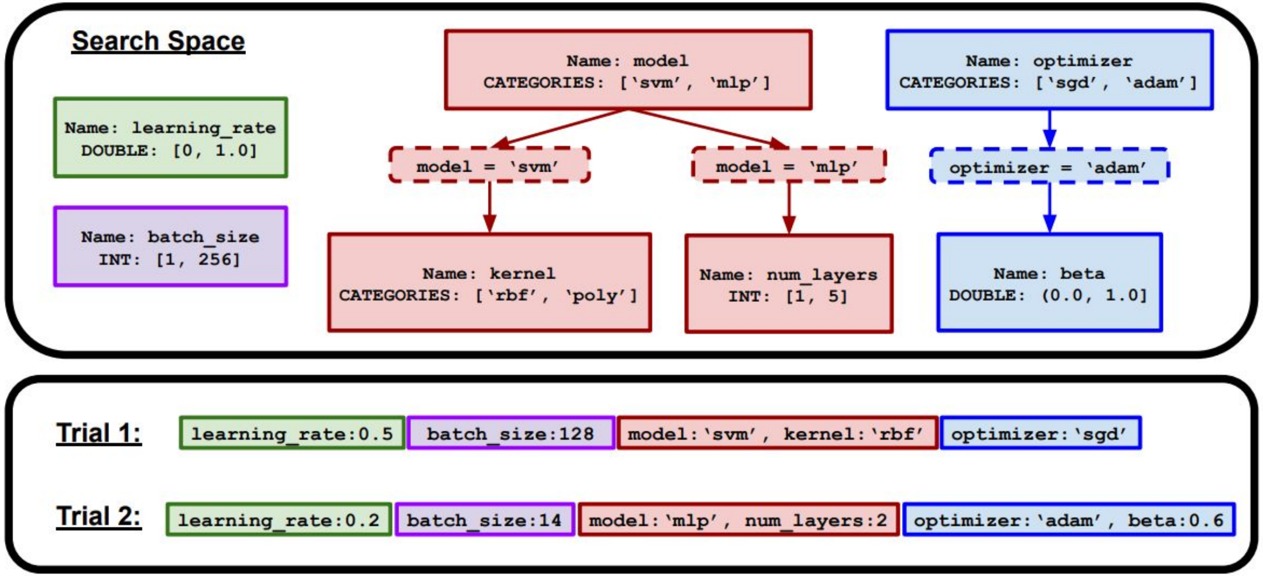
- AutoML (Hyperparameter tuning)
- Chemistry / Biological
- Production (Ads Bidding)

Property	Statistic
# Studies	$\mathcal{O}$(70M+)
# Trials	$\mathcal{O}$(120B+)
# Distinct Users	$\mathcal{O}$(14K)



Vizier Trials

- Flat Types: **`DOUBLE, INTEGER, DISCRETE, CATEGORICAL`**
- Conditional Parameters (non-fixed parameter count for an X)

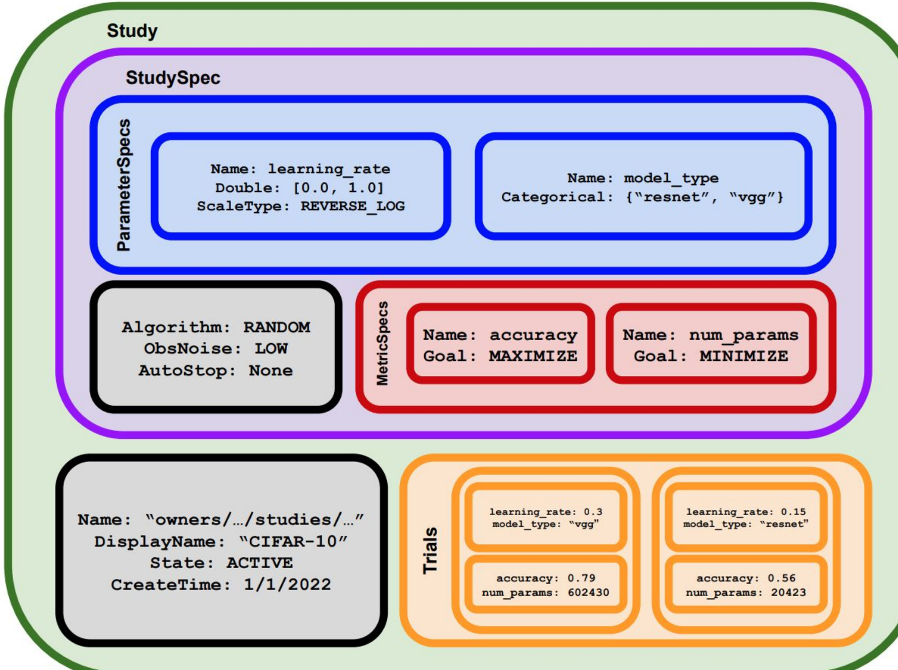


Vizier Metadata

Metadata: Title, Owner, Description, Objective Name, Free-form text

Transferability sources:

- Single user, similar experiments
- Different user, similar experiments
  - Ex: ResNets on CIFAR10
- Similar params, different experiments
  - Ex: “learning_rate”
- Description / Free-Form Metadata
  - Ex: Associated code / file locations



Actual Serialization Examples

Domain	X	M
Google AutoML	batch_size: 128 model_type: REDACTED activation_fn: "tanh" batch_norm: "True" dropout: 0.143 embedding_combiner: "mean" gradient_clip_norm: 1.63e+03 num_hidden_layers: 1 hidden_units[0]: 359 optimizer_type: "AdamOptimizer" beta1: 0.9 beta2: 0.999 learning_rate: 0.926 nlp_vocabulary_strategy: "adjusted_mutual_info" vocabulary_strategy: "adjusted_mutual_info"	title: "n-w597ng99i7lj0-q40zcboi1ea7l" user: REDACTED description: "" objective: "val_categorical_cross_entropy" amc_model_version: REDACTED task_type: "multi_class_classification"

Actual Serialization Examples

Domain	X	M
Init2Winit	dropout_rate: 0.6 decay_factor: 0.0379 label_smoothing: 0.378 lr_hparams.base_lr: 0.00285 lr_hparams.decay_steps_factor: 0.854 lr_hparams.power: 1.94 opt_hparams.one_minus_momentum: 0.0557	title: "d_sp1-lm1b_trfmr-b1024-2021aug20" user: REDACTED description: "" objective: "valid/ce_loss"

Actual Serialization Examples

Domain	X	M
Protein Design	p00000000:"9" p00000001:"16" p00000002:"1" p00000003:"11" p00000004:"16" p00000006:"9" p00000006:"0" p00000007:"14" ... p00000047:"13"	title: "871cac30956711eab5bef371aa1bb25a" user: REDACTED description:"" objective:""

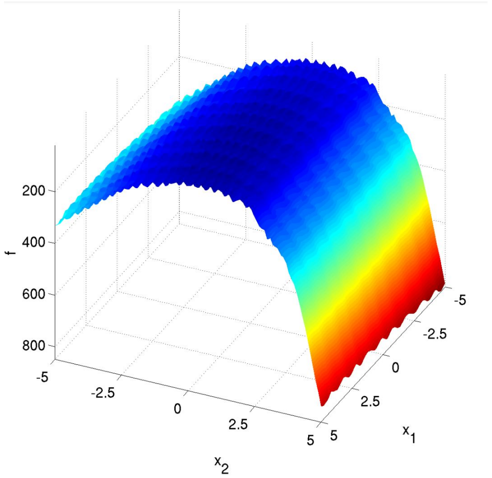
Actual Serialization Examples

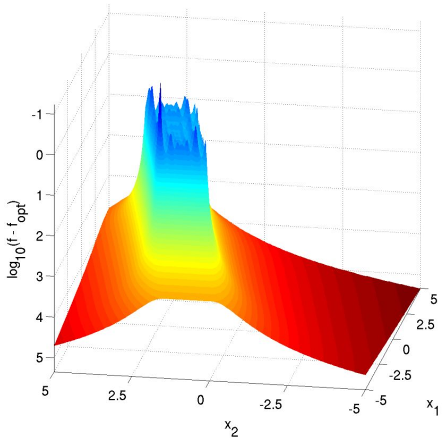
Domain	X	M
Vertex AI Text	universal_model_type: "single_dense_feature" token_model_type: "cnn" token_bow_combiner: "sqrtn" token_model_type: "bow" rand:0 batch_size: 4 convnet: "2:3:4*100pa" dropout_keep_prob: 1 hidden_layer_dims: 50 max_length: 1.54e+03 max_num_classes_for_per_class_metric: 0 max_token_vocab_size: 1e+05 merge_word_embeddings_vocab_size: 1e+05 token_freq_cutoff: 1 tokenizer_spec: "delimiter" word_embedding: REDACTED word_embedding_dim: 100	title: "20220228-621c9aea-0000-2c94" user: REDACTED description: REDACTED objective:"micro-auc-pr-label0_label" atc_study_dataset_tag:"" atc_study_notes:""

Additional Synthetic BBOB Data

Controlled experiments: 24 BBOB functions w/ multi-task augmentation

- Use random shifts f(x - shift)
- Metadata (M) shows (function name, dimension, shift)
    - Ex: "`(Sphere, dimension=3, shift=[0.1,-0.3, -2.1])`"

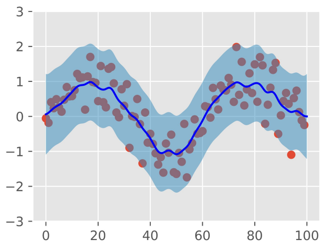


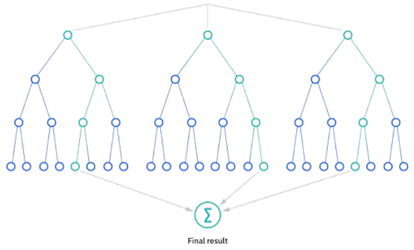


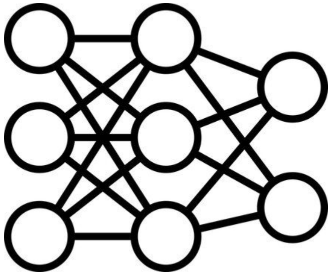
Single-Task Baselines

- Gaussian Process (Vizier Default)
- Random Forest + Trees (XGBoost)
- MLP (2-Layer ReLU, MSE Loss)

$$x \in \mathbb{R}^d$$







Caveat: Not trying to start a regression fight

Evaluation Protocol

Normalized Mean Average Error per Study

- Different y-scales (CIFAR10 is [0, 1], BBOB is [10^2, 10^9])
- **LM pointwise prediction is <u>median of 64 samples.</u>**

$$\frac{1}{y_{\max} - y_{\min}} \frac{1}{|\mathcal{D}^{\text{test}}|} \sum_{(x,y) \in \mathcal{D}^{\text{test}}} |f(x) - y|$$

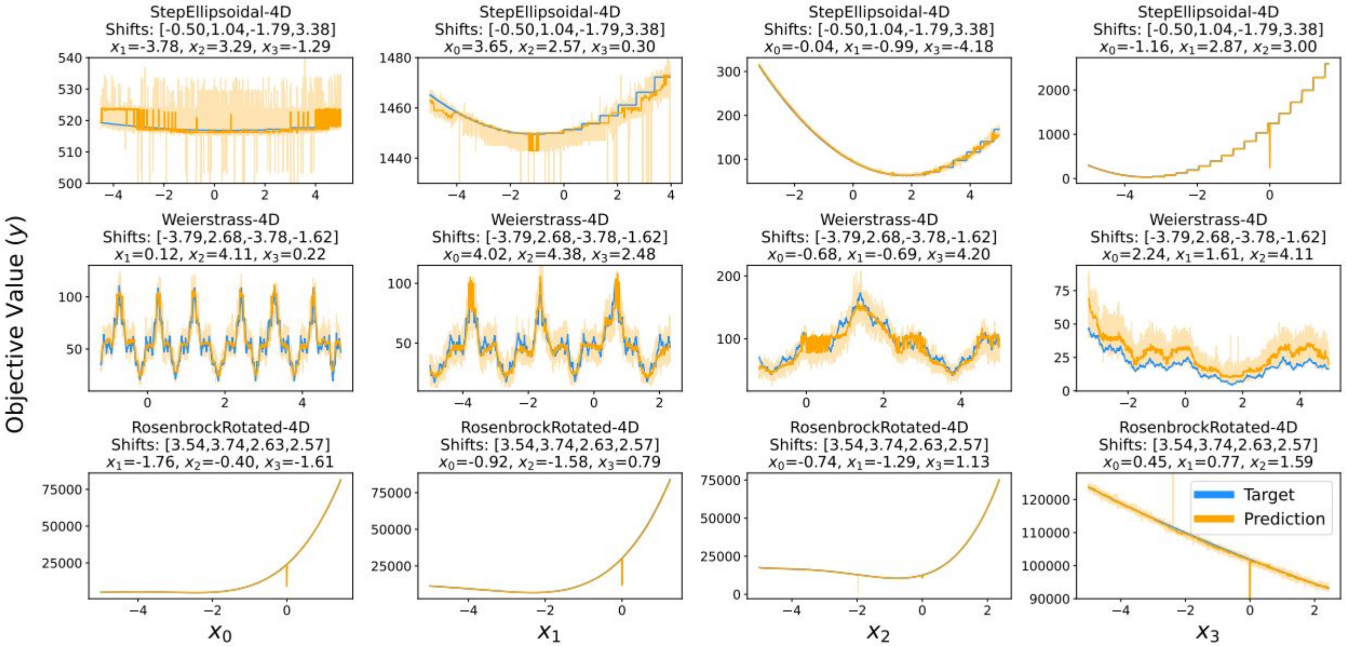
Experiments

Key Questions:

- Can LM simultaneously regress on multiple-tasks?
    - Over different spaces and numeric scales?
- How is multi-task training useful?
    - Why would we train on f’(x) if we’re eval-ing on f(x) only?
- Can fine-tuning deal with unseen tasks?
    - Does pretrained knowledge carry over?

Simultaneous Regression (BBOB)

Is LM capable of high-precision, simultaneous regression?

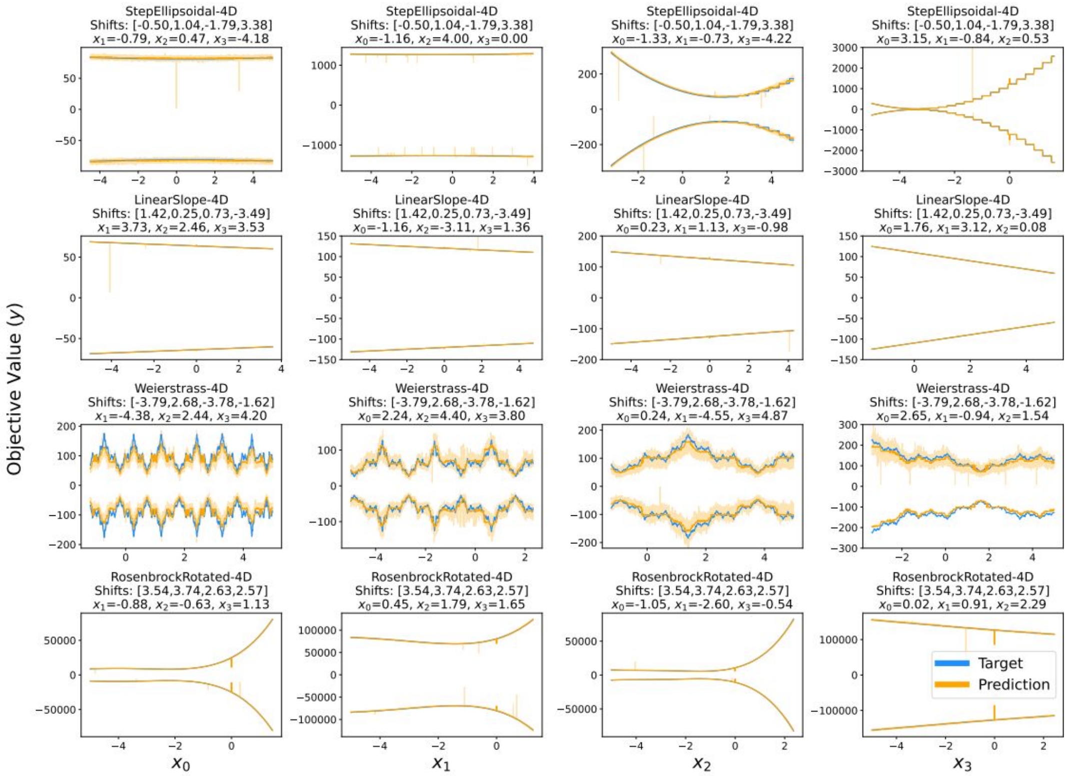


Uncertainty Estimation

Randomly flip y-sign. What happens?

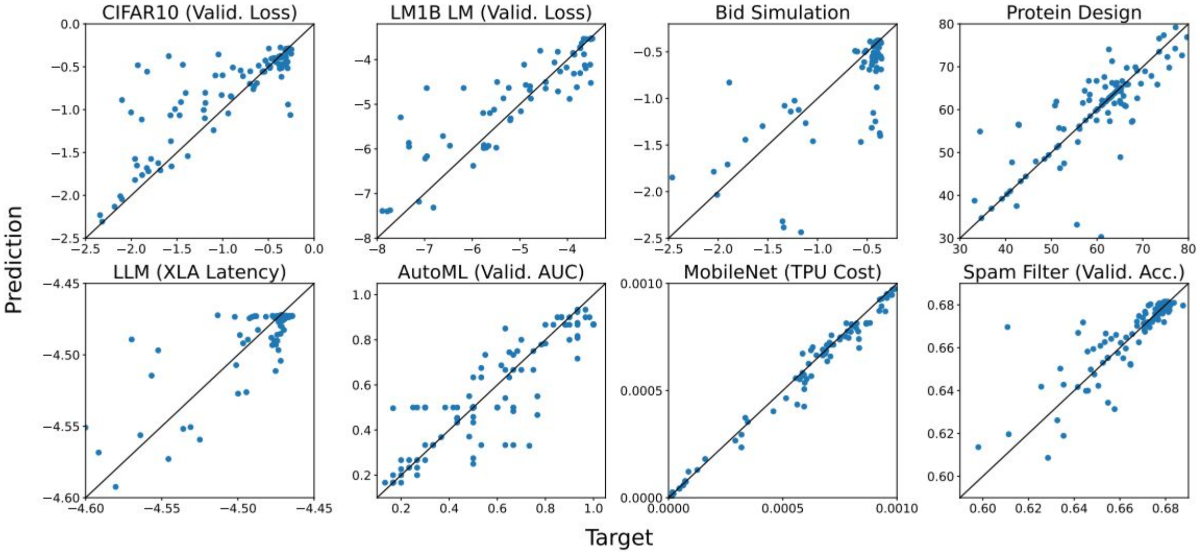
Capture uncertainty + bimodality

- No explicit ensemble count



Simultaneous Regression (Real World)

What about real-world objectives?



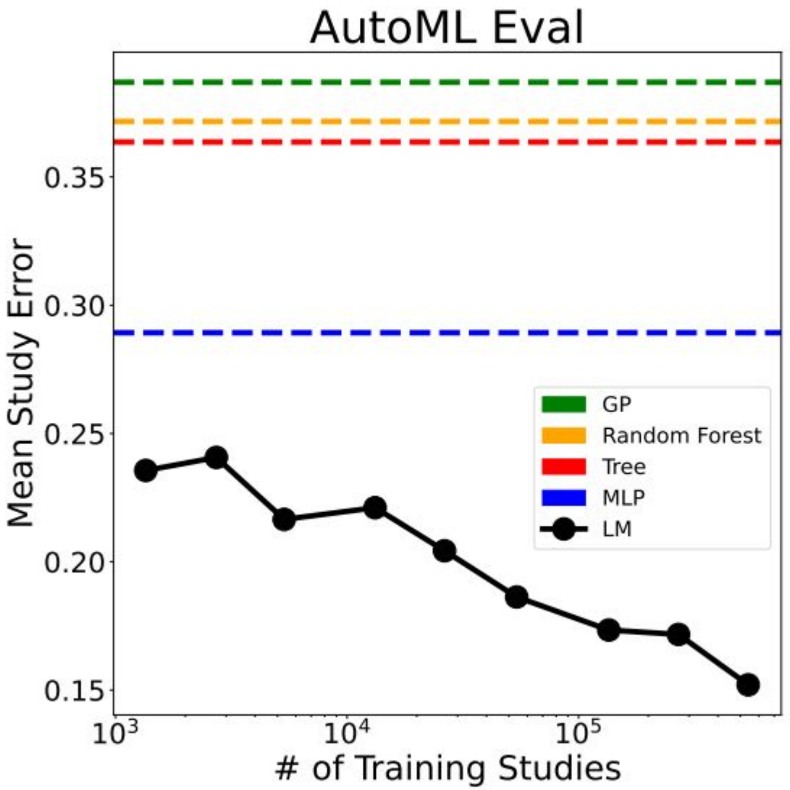
Name	Space
CIFAR10	4 Double
LM1B LM	4 Double
Bid Simulation	4 Double
Protein Design	60 Categories
LLM Latency	31 Hybrid
AutoML	3-H, 42-T
MobileNet	10 Discrete
Spam Filter	13-H, 15-T

Multi-task Training (AutoML Data)

(↓ Lower is better)

If I only eval on f(x)...

Does training on f'(x) help?

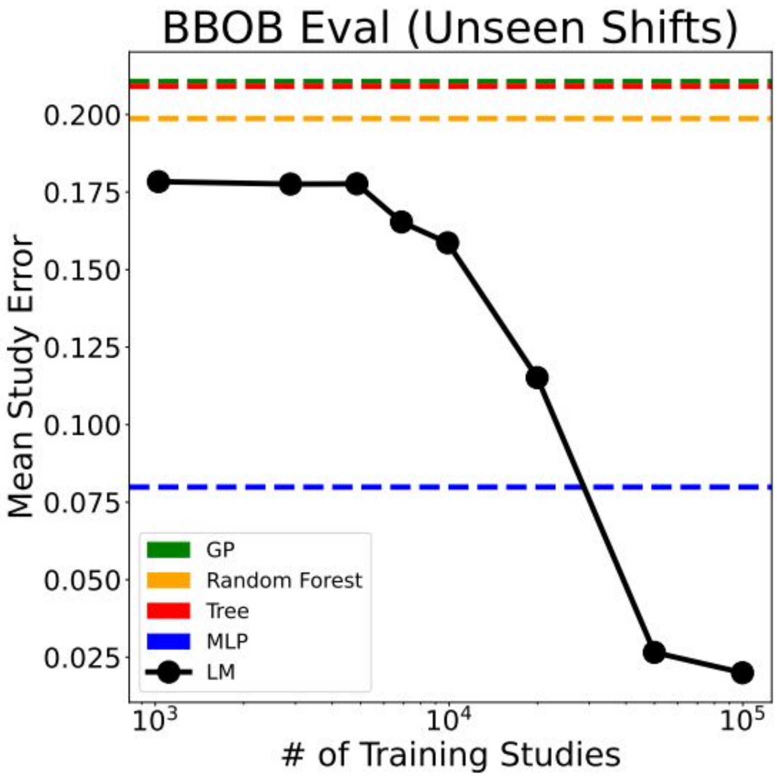


Multi-task Training (BBOB)

(↓ Lower is better)

BBOB seen as f(x,m).

What if we vary m (the shift)?



Evaluation Across Domains

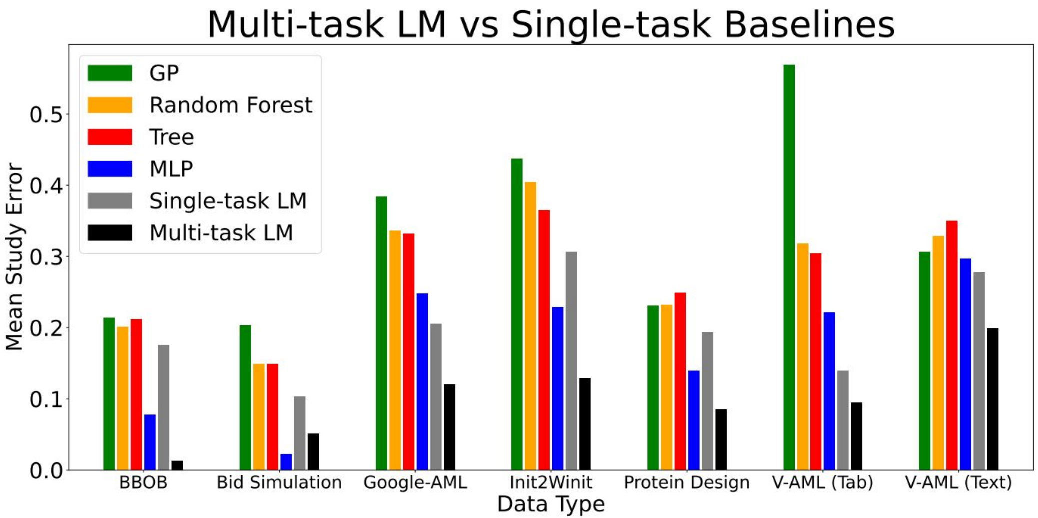
Single-task not bad

- Beats MLP on conditionals!

Multi-task beats **Single-task**

- Beats baselines most times

Name	# Studies	Avg. TpS	Avg. SS
BBOB	1M	30	4.0
Bid Simulation	22K	698	4.6
Google AutoML (Internal)	540K	250	(3.3, 29.9)
Init2winit	2K	176	3.6
Protein Design	54K	584	125.6
Vertex AI AutoML (Tabular)	1.4M	88	(4.6, 42.4)
Vertex AI AutoML (Text)	544K	118	56.0



(↓ Lower is better)

Does text help?

“Anonymize” prompts via study-dependent hashing:

- Parameter names “batch_size” -> “s710kdf9”
- Categorical values “adam” -> “129sd923”
- Metadata “shift: [-0.1, 0.2]” -> “dsnf9133”

De-correlates f(x) from f’(x).

Model still trains, but eval much worse.

Datasets (# Training Studies)	Mean Study Error (↓) Original	Mean Study Error (↓) Anonymized
BBOB (50K)	**0.03**	0.46
BBOB (Full 1M)	**0.01**	FAIL
AutoML (26.3K)	**0.19**	0.44
AutoML (Full 540K)	**0.15**	0.43

Local Finetuning Experiments

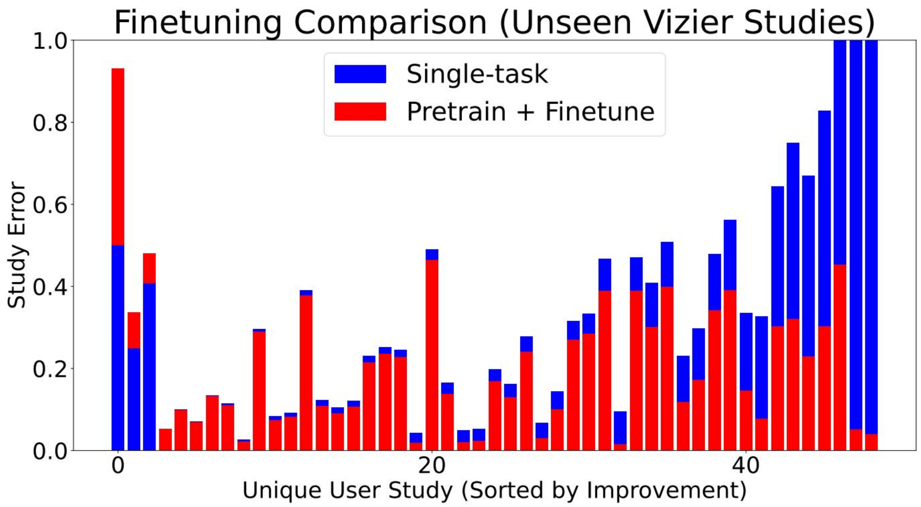
Classic GPT-Trick: Pretrain then Finetune

$$\theta_{ft} = \theta_{pt} - \nabla \ell_{(x,y)\sim\mathcal{D}_{train}}(\theta)$$

Eval on unseen Vizier studies

- Studies after March 31, 2023
- From distinct users

(↓ Lower is better)



Local Finetuning Experiments

Positive Transfer

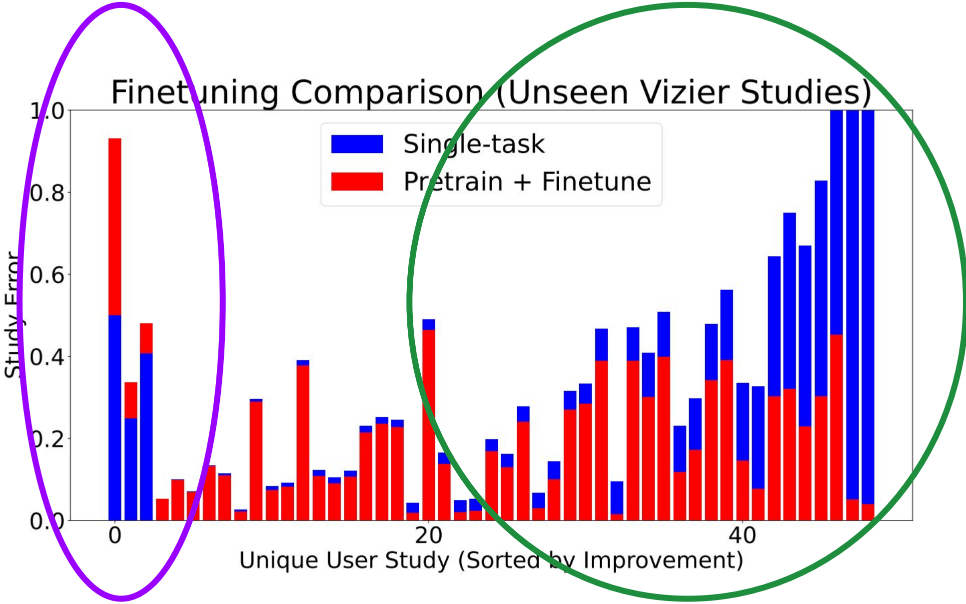
- Pretrained knowledge carries over to new studies

Negative Transfer…?

- Sometimes pretrained knowledge is worse…?

$$\theta_{ft} = \theta_{pt} - \nabla \ell_{(x,y)\sim\mathcal{D}_{train}}(\theta)$$

(↓ Lower is better)



Positive or Negative Fine-tuning Transfer?

Suppose we always eval on AutoML data.

Possible pretrained checkpoints

- None (Single-task)
- BBOB
- AutoML itself
- Entire Vizier

$$\theta_{ft} = \theta_{pt} - \nabla \ell_{(x,y)\sim\mathcal{D}_{train}}(\theta)$$

Pretraining Dataset	Mean Study Error (↓) on AutoML Before Finetuning	Mean Study Error (↓) on AutoML After Finetuning
None (Single-Task)	0.98	0.20
BBOB	0.98	0.45
AutoML	**0.15**	**0.15**
Entire Vizier	0.31	**0.15**

(Optional) Ablations

Best y-tokenization?

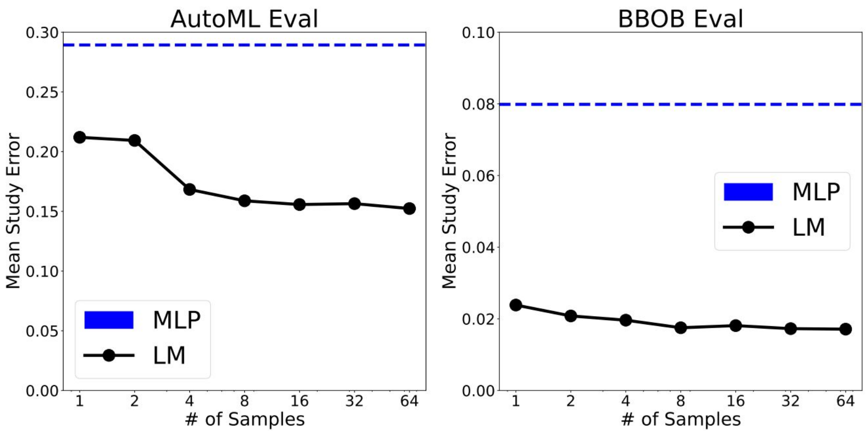
- (Default) Separate Sign and Digit-by-Digit: <+><1><2><3><4><E-2>
- Merged Mantissa: <+1234><E-2>
- Exponent Before Mantissa: <+><E-2><1><2><3><4>

	AutoML		BBOB	
Tokenization Method	Single-Task	Multi-Task	Single-Task	Multi-Task
Default	0.21	0.15	0.17	0.01
Merged Mantissa	0.73	0.15	0.41	0.01
Exponent Before Mantissa	0.24	0.15	0.17	0.01

Table 6: Mean Study Error (↓) comparisons between different tokenization methods.

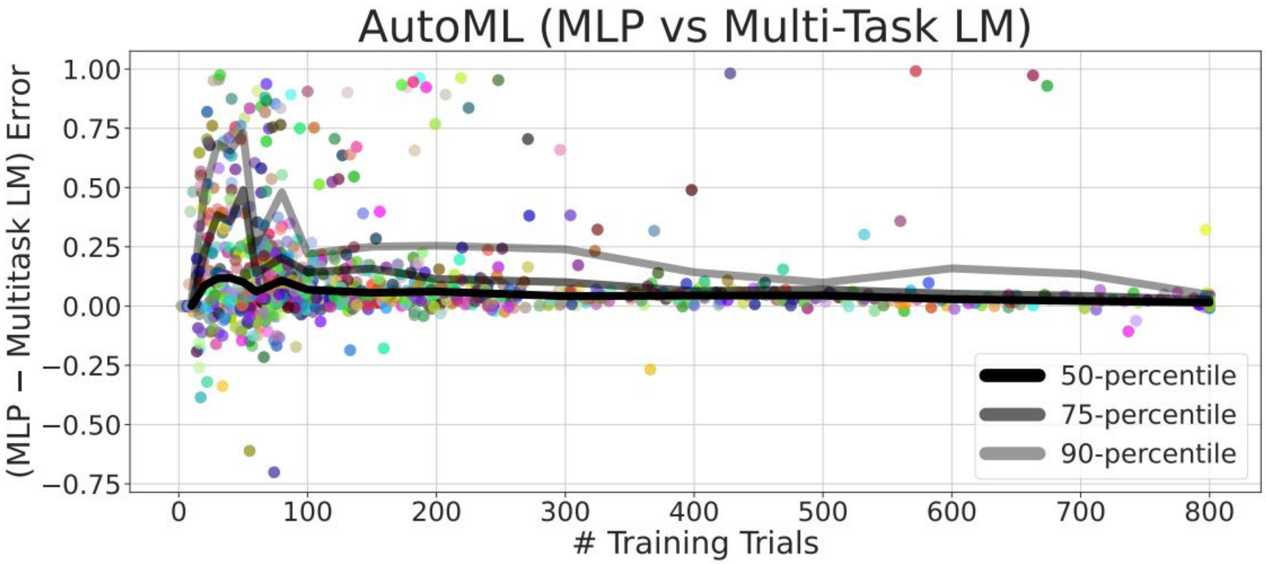
Importance of sampling aggregation?

Empirical Aggregation Method	Mean Study Error (↓) AutoML (Full 540K)	BBOB (Full 1M)
Median (default)	**0.15**	0.01
Max-likelihood	0.22	0.01
Mean	0.23	0.01



When does Multi-task training help?

Intuitively when eval task has low training data.



Conclusions and Future Work

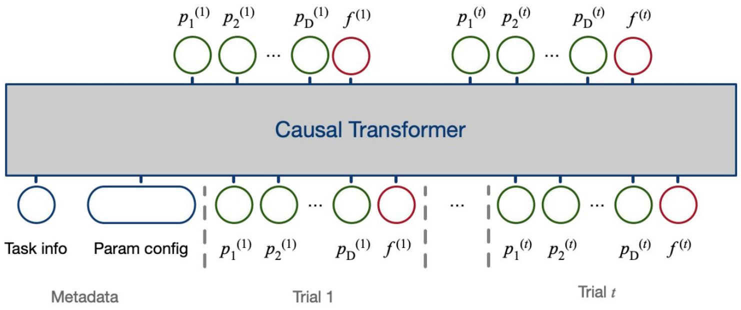
Gradients vs In-Context

ICL not only way to absorb (x,y).

Gradients:

- Unbounded limit for absorbing (x,y) pairs
- Easy JSON formatting, no (X) compression

Analogous to MLP vs GP



VS

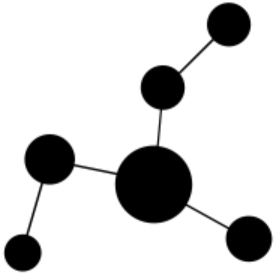
Many improvements

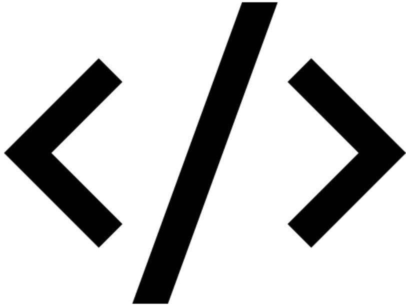
- Weighted cross-entropy loss
  - Sign + exponent most important
- Better x-tokenizations
  - SentencePiece: ‘1234.5’ -> [‘12’, ‘3’, ‘4.5’]
- Better warm-start
  - English pretraining
  - Pretraining to decode X’s

New Applications

(X,M) can be anything:

- Configuration Files
- Graphs
- Combinatorics
- Language
- Code





New Benchmarks

Reward Models simulate human ratings

LM Surrogates simulate realistic objectives

HPO-B: A Large-Scale Reproducible Benchmark for Black-Box HPO based on OpenML

Sebastian Pineda Arango*
University of Freiburg
pineda@cs.uni-freiburg.de

Hadi S. Jomaa*
University of Hildesheim
hsjomaa@ismll.uni-hildesheim.de

Martin Wistuba†
Amazon Research
marwistu@amazon.com

Josif Grabocka
University of Freiburg
grabocka@cs.uni-freiburg.de

Published as a conference paper at ICLR 2022

SURROGATE NAS BENCHMARKS: GOING BEYOND THE LIMITED SEARCH SPACES OF TABULAR NAS BENCHMARKS

Arber Zela[1,*], Julien Siems[1,*], Lukas Zimmer[1,*], Jovita Lukasik[2], Margret Keuper[2], Frank Hutter[1,3]
[1] University of Freiburg, [2] University of Mannheim, [3] Bosch Center for AI

More Links

- Paper
- Code
- Poster

Thank you!