Hydro: Surrogate-based Hyperparameter Tuning Service in Datacenters

Qinghao Hu\textsuperscript{1,2}, Zhisheng Ye\textsuperscript{2,3}, Meng Zhang\textsuperscript{1,2}, Qiaoling Chen\textsuperscript{2,4}, Peng Sun\textsuperscript{2,5}, Yonggang Wen\textsuperscript{1}, Tianwei Zhang\textsuperscript{1}
Background
What is Hyperparameter Tuning?

General:
- learning_rate = 0.01
- batch_size = 256
- weight_decay = 0.01

Optimizer = SGD(momentum = 0.5)
/ Adam(betas = (0.9, 0.99))

LR_Scheduler = Step(gamma = 0.1)
/ CosineAnnealing(T_max = 10)

ResNet, GPT...
PyTorch v1.10 released an updated version of their official model weights. The new ResNet-50 + New Recipe model achieves an accuracy of 80.9% at ImageNet-1K, outperforming the previous models ResNet-50 and ResNet-152 by 4.8% and successfully competing with larger models.

Source: PyTorch Blog
Challenge 1: High Tuning Cost of Large Models

The cost of tuning large models is unacceptable
→ lead to subpar model quality
Challenge 2: Inefficient Resource Usage

Tuning jobs consume substantial resources from enterprise & institute clusters

**Microsoft**
90% of models require tuning, 75 trials in median \([1]\)

**Alibaba**
65% of jobs repeatedly run \(\geq 5\) times \([2]\)

**senseTime**
90% of jobs are repetitive for tuning or debugging \([3]\)

**VECTOR INSTITUTE**
46% of GPU hours contribute to single-GPU tuning jobs \([4]\)

Source: \([1]\) Themis (NSDI '20) \([2]\) MLaaS (NSDI '22) \([3]\) Lucid (ASPLOS '23) \([4]\) HFTA (MLSys '21)
Challenge 2: Inefficient Resource Usage

GPUs are significantly underutilized

**Job-level Hydro Tuner**

Automatically generate surrogate models for tuning by applying transfer theory and model fusion.

**Datacenter-level Hydro Coordinator**

Leverage idle bubble resources of pretraining jobs via interleaving training.
Key Mechanism: Surrogate-based Tuning

Existing Systems:
Search on target model directly

Search on surrogate model

Scale Down

Transfer HP Back

Surrogate Model
(Width-reduced model for tuning)

Target Model
Can Hyperparameters be Transferred?

**Toy Example:** 2-layer Transformer model over WikiText-2 dataset using Adam

![Graph showing the relationship between Scaling Ratio and Learning Rate with the best learning rate marked as 2^{-13} with Width=4096]
Can Hyperparameters be Transferred?

**Toy Example:** 2-layer Transformer model over WikiText-2 dataset using Adam

![Graph showing Scaling Ratio and Loss vs. Learning Rate]

Scaling Ratio: $S = 1$ vs. $S = 2$

- **Best LR of Width=2048**
Can Hyperparameters be Transferred?

**Toy Example:** 2-layer Transformer model over WikiText-2 dataset using Adam

![Graph showing the loss with different scaling ratios](image)

**Scaling Ratio:**
- $S = 1$
- $S = 2$
- $S = 4$

**Best LR of Width=1024**
Can Hyperparameters be Transferred?

**Toy Example**: 2-layer Transformer model over WikiText-2 dataset using Adam

Scaling Ratio: $S = 1$, $S = 2$, $S = 4$, $S = 8$

Best LR of Width=512
Can Hyperparameters be Transferred?

**Toy Example:** 2-layer Transformer model over WikiText-2 dataset using Adam

- **Scaling Ratio:** $S = 1$, $S = 2$, $S = 4$, $S = 8$, $S = 16$

  - **Best LR shifts** across model scales
  - **Best LR of Width=256**

  ![Graph showing learning rate vs. loss for different scaling ratios](image-url)
Hydro Makes Hyperparameters Transferable

Applying Hydro on the same Transformer model

Scaling Ratio: $S = 1$  $S = 2$  $S = 4$  $S = 8$  $S = 16$

Consistent Best LR

Lower Loss

Best LR $\sim 2^{-14}$

Best LR $\sim 2^{-8}$

+ Hydro
Underlying Theory: Maximum Update (MU) Parametrization[1]

Theoretically enabling maximal feature learning for infinite-width neural networks

1-hidden-layer MLP:

```
Input → Hidden Layer U ∈ \( \mathbb{R}^{w \times 1} \) (\( w \to \infty \)) → ...
```

```
Output Layer V ∈ \( \mathbb{R}^{w \times 1} \)
```

Optimizer: SGD with \( lr = 1 \)

Common Practice: Initialization: \( U \sim \mathcal{N}(0, 1) \), \( V \sim \mathcal{N}(0, 1/w) \)

Learning rates: \( \eta_U = 1 \), \( \eta_V = 1 \)

MU Parametrization: Initialization: \( U \sim \mathcal{N}(0, 1) \), \( V \sim \mathcal{N}(0, 1/w^2) \)

Learning rates: \( \eta_U = w \), \( \eta_V = 1/w \).

Avoid Output Layer Blow-up

Source: [1] Feature Learning in Infinite-Width Neural Networks (ICML '21)
MU Parametrization: Intuitive Insights[1]

**Theoretical**: Maximal feature learning for infinite-width neural networks

**Empirical**: Hyperparameter transfer across model scales (in terms of width)

**Impact in Practice**

Correspond models with different scales to their ∞ limits

**Benefits:**
- Solve the unbalanced training issue (e.g., output layer update much faster) via layer-wise $lr$ adjustment
- Ensure consistent magnitude updates for each layer during training regardless of its width

**Problem**: Manually implementing MU parametrization is burdensome and error-prone

Source: [1] Tuning Large Neural Networks via Zero-Shot Hyperparameter Transfer (NeurIPS ’21)
Hydro Tuner

MU parametrization theory + system support to jointly accelerate tuning

**Target Model**

1. Trace & Scale

**Surrogate Model**

1. Parametrize

\( S \): User-defined Model Scaling Ratio

**Input Layer:**
1. Init Variance Multiply \( S \)
2. If SGD Optimizer, Layer LR Divide \( S \)

**Hidden Layer:**
1. Init Variance Multiply \( S \)
2. SGD & Adam Optimizer LR Multiply \( S \)

**Output Layer:**
1. Zero-Variance Initialization
2. Layer Input Multiply \( S \)
3. If SGD Optimizer, Layer LR Divide \( S \)
Hydro Tuner: Scaling Effect

Example: WideResNet-50

GFLOPs $\sim \frac{6}{S^2}$

1/64 Computation

Memory $\sim \frac{70}{S} + 4$
Hydro Tuner: Scaling Effect

However... For Small Model: ResNet-18

- **Epoch Time (s)**
  - S=1: 8.2
  - S=8: 7.6

- **GPU Utilization (%)**
  - S=1: 32%
  - S=8: 12%

Exacerbate Underutilization
Hydro further enables **inter-** and **intra-trial fusion** to improve hardware efficiency.
Hydro Tuner: Inter-trial Fusion

Hydro extends the application scope of HFTA\(^1\) & automizes the fusion process

Hydro Tuner: Inter-trial Fusion

Hydro extends the application scope of HFTA\cite{1} & automizes the fusion process

Data  Model  Optimizer  Loss

Fused Trial

Effect of Scaling + Inter-trial Fusion

Example: ResNet-18 (Scaling=8, CIFAR-10 Batch Size=256) on A100 80GB

Normalized Throughput

20x speedup

600+ fusion
**Hydro**

**Job-level Hydro Tuner**
Automatically generate surrogate models for tuning by applying transfer theory and model fusion.

**Datacenter-level Hydro Coordinator**
Leverage idle bubble resources of pretraining jobs via interleaving training.
Resource Contention between LLM Pretraining and Tuning Jobs

Large Language Model (LLM) pretraining jobs occupy massive resources

→ Long queuing delay of tuning jobs
Opportunity: Co-exist LLM Pretraining Jobs

Massive Resources: Long-term occupy hundreds ~ thousands of GPUs

Pipeline Parallelism is commonly applied but introduces **bubbles**

Example: 1F1B\(^1\) Pipeline Schedule

![Pipeline Schedule Diagram]

**Example:**

- **Forward Pass:** Blue
- **Backward Pass:** Green
- **Bubble:** Blank
- **Flush:** Downward triangle

**GPU Memory:**

- **Model & Framework Memory:** Grey
- **Activation Memory:** Pink

Wasted spare resources

Unbalanced memory footprint

Source: [1] PipeDream (SOSP ‘19)
Hydro Coordinator: Leverage Bubble Resources

Solution: **Interleaving** Hydro trials with a LLM pretraining job

**Hydro Trials +1F1B Workload**

- **Forward Pass**
- **Backward Pass**
- **Bubble**
- **Hydro Trial**

**GPU Memory:**
- Model & Framework Memory
- Activation Memory
- Hydro Memory

- **Flush**
- **Resume**
- **Pause**

- **Trial 1**
- **Trial 2**
- **Trial 3**
- **Trial 4**

- **W1**
- **W2**
- **W3**
- **W4**

- **No interference**
- **Resilient trial sizes**

28
Hydro Coordinator: Leverage Bubble Resources

Solution: **Interleaving** Hydro trials with a LLM pretraining job

*Why Hydro Trials are suitable for interleaving?*

1. **Throughput Insensitive**
   Tuning jobs are more *tolerant* of partial trials *slowdown*

2. **Deterministic and Scaled Memory Footprint**
   Memory is *profiled* and greatly *reduced* via model scaling

3. **Elastic and Opportunistic Trial Placement**
   Trials can *adjust the fusion number* to fit the remaining memory

**GPU Memory:**

- Full
Effect of Hydro Coordinator

SM Activity of a GPT model with 4 pipeline stages (over 4x8 A100 GPUs) + Hydro Trials (fuse 16x ResNet-18 models) with interleaving training

![Diagram showing SM Activity over time with Hydro Trials and Forward/Backward phases.]
Evaluation
Evaluation: Intuitive Study of Surrogate-Based Tuning

10 trials of ResNet-18 on CIFAR-10: A-J: [batch_size, learning_rate, momentum]

Hyperparameter ranking transfers well across different scaling ratios
Evaluation: End-to-End Experiments

Testbed: A100 GPU cluster of Shanghai AI Laboratory

Baseline: Ray Tune\textsuperscript{[1]} system + FIFO algorithm

<table>
<thead>
<tr>
<th>Task</th>
<th>Search Space</th>
<th>Model</th>
<th>#GPUs</th>
<th>#Trial</th>
<th>Acceleration</th>
<th>Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language Modeling</td>
<td>lr: ((10^{-5}, 10^{-1})) gamma: ((0.01, 0.9))</td>
<td>GPT-3 XL</td>
<td>128</td>
<td>100</td>
<td>78.5 \times</td>
<td>-0.48 ppl*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transformer</td>
<td>8</td>
<td>200</td>
<td>8.7 \times</td>
<td>-0.15 ppl</td>
</tr>
<tr>
<td>Image Classification</td>
<td>lr: ((10^{-4}, 10^{0})) momentum: ((0.5, 0.999)) batchsize: ([128, 256, 512]) gamma: ((0.01, 0.9))</td>
<td>WideResNet-50</td>
<td>32</td>
<td>200</td>
<td>20.3 \times</td>
<td>+1.18% acc*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MobileNetV3-L</td>
<td>16</td>
<td>500</td>
<td>12.3 \times</td>
<td>+0.05% acc</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VGG-11</td>
<td>8</td>
<td>500</td>
<td>10.8 \times</td>
<td>+0.09% acc</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ResNet-18</td>
<td>8</td>
<td>1000</td>
<td>16.2 \times</td>
<td>+0.02% acc</td>
</tr>
</tbody>
</table>

* Compared with the official hyperparameter setting as the model quality baseline

Source: \textsuperscript{[1]} Tune: A Research Platform for Distributed Model Selection and Training (ICML AutoML ‘18)
Job-level
Joint optimization of theory and system techniques

Datacenter-level
Leverage idle bubble resources of pretraining jobs

https://github.com/S-Lab-System-Group/Hydro
qinghao.hu@ntu.edu.sg