WP4: High Performance Language Models

HPLT kickoff 2.9.2022 Prague

Overview

WP4 <u>optimizes, builds and evaluates language models</u> (LMs). (cf. WP5: machine translation models)

- Pretrain **BERT**-, **GPT**-, and **T5**-like models
- Cover **~80 languages** + multilingual LMs
- **Variations**: model sizes, efficient models, etc.
- **Evaluation**: perplexity + downstream tasks

<u>UTURKU</u>, UOSLO CUNI; spans 36 months; takes 78PM

Overview

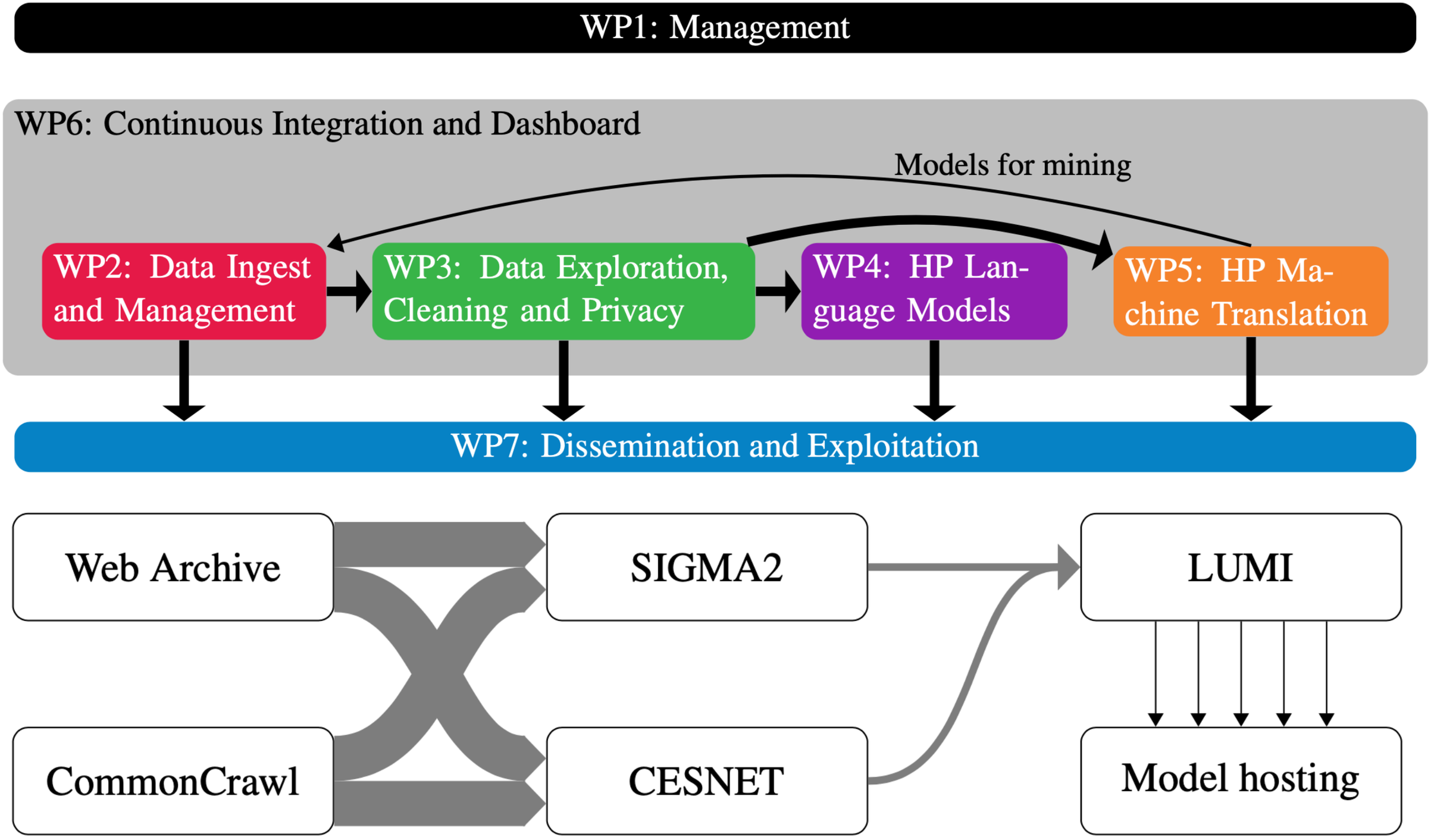
WP4 <u>optimizes, builds and evaluates language models</u> (LMs). (cf. WP5: machine translation models)

- Pretrain **BERT**-, **GPT**-, and **T5**-like models
- Cover **~80 languages** + multilingual LMs
- **Variations**: model sizes, efficient models, etc.
- **Evaluation**: perplexity + downstream tasks

Hundreds of models at minimum, potentially over 1000 (!)

<u>UTURKU</u>, UOSLO CUNI; spans 36 months; takes 78PM

Overview


WP1: Management
WP6: Continuous Integration and Dashboard
Models for mining
WP2: Data Ingest and Management
WP3: Data Exploration, Cleaning and Privacy
WP4: HP Language Models
WP5: HP Machine Translation
WP7: Dissemination and Exploitation
Web Archive
CommonCrawl
SIGMA2
CESNET
LUMI
Model hosting

Overview

Four tasks:

- T4.1: Building/Training Language Models (UTURKU, UOSLO)
- T4.2: Efficient Data Usage & HPC utilization (UOSLO)
- T4.3: Evaluating Large Language Models (UTURKU, UOSLO)
- T4.4: Ethical Considerations (UOSLO, CUNI)

Two deliverables:

- D4.1:Trained language models (UTURKU, M30)
- D4.2: Report on language model evaluation (UTURKU, M35)

T4.1: Building/Training Language Models

Starts M1, ends M36 (UTURKU, UOSLO)

Adapt and develop tools for training LMs, including

- **Bidirectional** (BERT-like)
- **Causal** (GPT-like)
- **Encoder-decoder** (T5-like)

Create **automated, unified and documented training process**; release tools and models openly

Target **76+ languages** and multilingual models

T4.2: Efficient Data Usage & HPC utilization

Starts M6, ends M30 (UOSLO)

Explore efficient use of data and compute

- **Alternative pre-training objectives** (e.g. w/annotation)
- **Efficient model variations** (e.g. ELECTRA)
- **Practical data requirements**

Systematically assess pretraining approaches, identify best practices

T4.3: Evaluating Large Language Models

Starts M1, ends M36 (UTURKU, UOSLO)

Systematically evaluate all created models, comparing with previously released models (incl. massively multilingual)

- **Intrinsic evaluation**: perplexity on held-out data
- **Extrinsic evaluation** on multilingual datasets for various downstream tasks (e.g. Universal Dependencies)

Need to assemble task-specific datasets and created automatic evaluation framework

T4.4: Ethical Considerations in Training and Deployment

Starts M1, ends M36 (UOSLO, CUNI)

Implements ethics plan for LM training

- Focus on exploring debiasing in an end-to-end fashion that was previously too costly to try.

Deliverables

D4.1: Trained language models (UTURKU, M30)

- Bidirectional, causal, and encoder-decoder LMs for 76+ languages and multilingual LMs, with variations on each (T4.1 + T4.2)

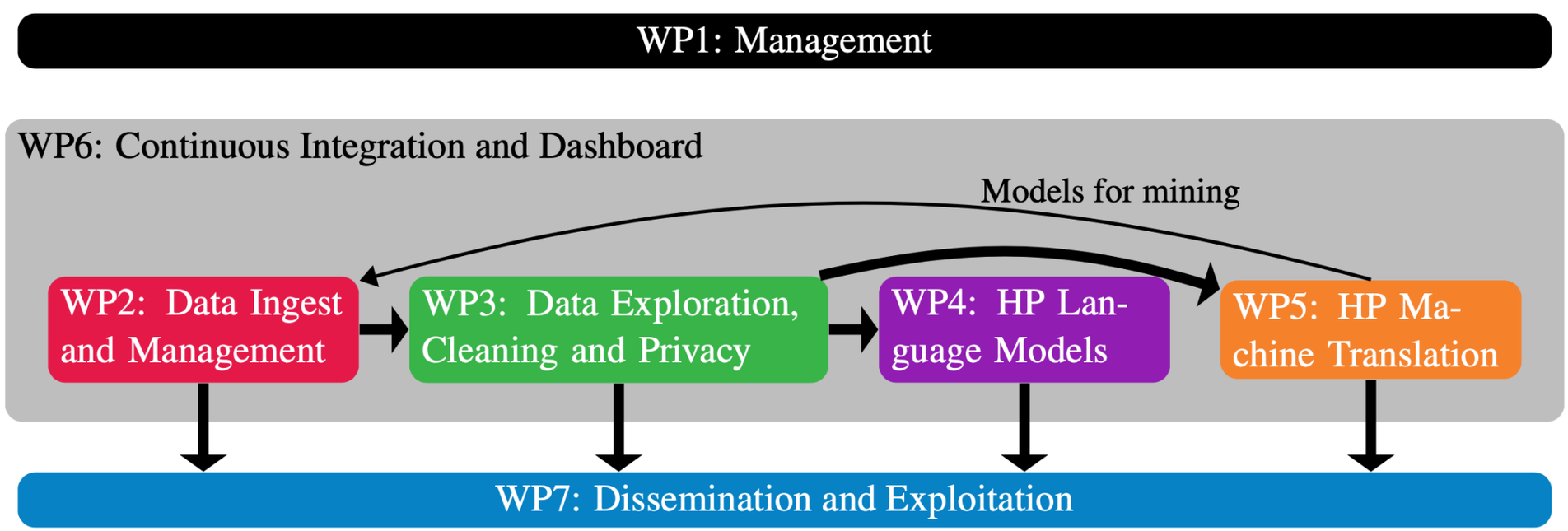
D4.2: Report on language model evaluation (UTURKU, M35)

- Results of LM evaluation (T4.3)

Implementation

Key components:

- Monolingual **datasets** (WP2 → WP3 → WP4)
- **Compute** (LUMI-G)
- **Model** and pre-training implementations



Compute

HPLT has **3M GPU-hours on LUMI**, the 3rd fastest supercomputer in the world (also 3rd-greenest, and fastest in Europe)

LUMI-G has 2560 nodes, each with 4 AMD MI250X devices (10240 GPUs / 20480 GCDs), 375 Pflop/s

The 3M GPU-hours will be primarily used in WP4 and WP5

Compute

As much as 3M GPU-hours may sound like, it is fairly limiting given that WP4 will at minimum train 100s of LMs:

- English GPT-3 model required **3640 PFLOPS-days**
- Assuming 40 TFLOPS performance on LUMI, training a single full GPT-3 model would require **~2M GPU-hours**

→ Even if the full 3M GPU-hour compute budget were used only on training GPT models, each model could only use a few % points of the compute to create GPT-3

Compute

LUMI-G pilot projects originally scheduled for Dec. 2021, but LUMI-G currently still **unavailable** (“pre-pilot phase”)

Pilots currently projected to start **late September** and general use **late October** 2022

→ HPLT LUMI-G allocation likely to become available for use in late October at the earliest

Before that, relevant work can start in **pre-pilot experiments and pilot projects**

Technology

UTURKU currently focusing on

- **ROCm**: AMD's CUDA-workalike (**mature**)
- **Pytorch** backend for model implementations (in **beta** for ROCm)
- **HF Transformers**: high-level LM implementations (**mature**, but not highly optimized)
- **DeepSpeed**: Microsoft library for large LMs (**beta**-level ROCm support)
- **Megatron**: NVIDIA's large LM implementation (**experimental** ROCm support)

UOSLO: also TF, JAX

Technology

Current status of technology stack on LUMI-G by model class (UTURKU):

- **Causal** (GPT-like): **fully functional**, scaled to 800 GPUs in preliminary experiments, **deconverge** issues for very large LMs
- **Bidirectional** (BERT-like): **fully functional**, but only tested on **comparatively small models** (BERT base/large)
- **Encoder-Decoder** (T5-like): so far **unable to run** on ROCm platform

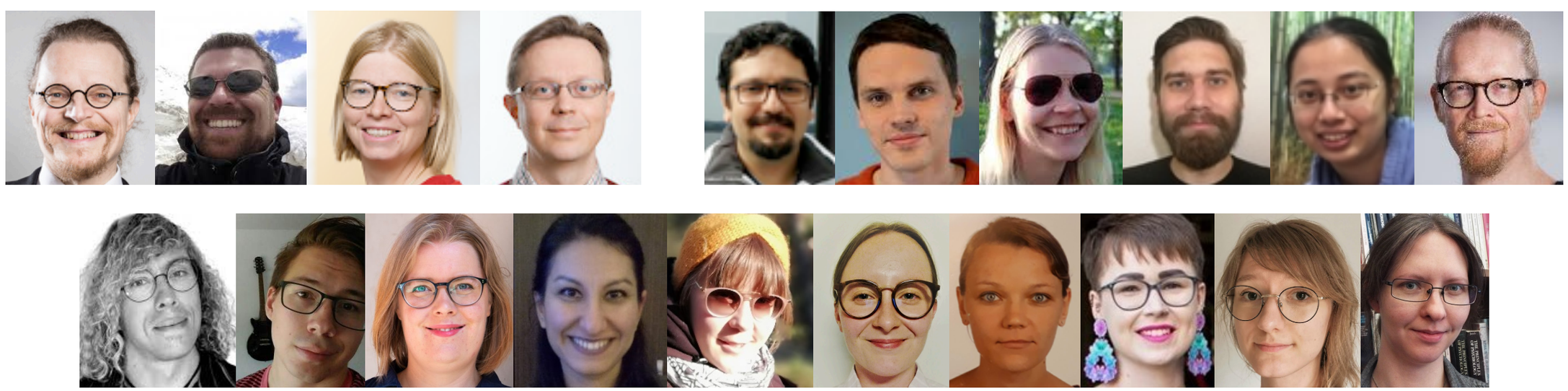
(Working through tech issues with LUST and AMD staff)

Group

TurkuNLP is 20+ years old and has more than 20 members

Substantial focus on **large LM training** and use in last ~4 years, with perhaps half the group working with large LMs

Two members **starting on HPLT now**; both have been working on large LM training on supercomputers



Discussion

- **First monolingual data delivery**: format, schedule, etc.
- **What languages to focus on first**?
- **Which multilingual model to train**? Balance between limited and massively multilingual?
- **What model sizes to train, and when**? Focus on largest feasible first, or work up from smaller models?
- **Which additional LMs to explore**? Interest in memory/ retrieval-augmented models?
- **Which downstream tasks to target in evaluation**?

Discussion

- **How to split compute budget**? WP4/WP5/others, project participants, GPT/BERT/T5/others?
- **Apply for additional compute**? (HPLT members already have several million GPU-h in separate projects!)
- **How generic should pretraining implementations be?** e.g. LUMI only / ROCm+Slurm platforms / supercomputers / any computer?
- **How generic should evaluation implementations be?**
- (Related: How to prioritize training efficiency vs. generality of implementation?)
- **How to coordinate technical work** on WP4/WP5 to minimize duplication of effort?