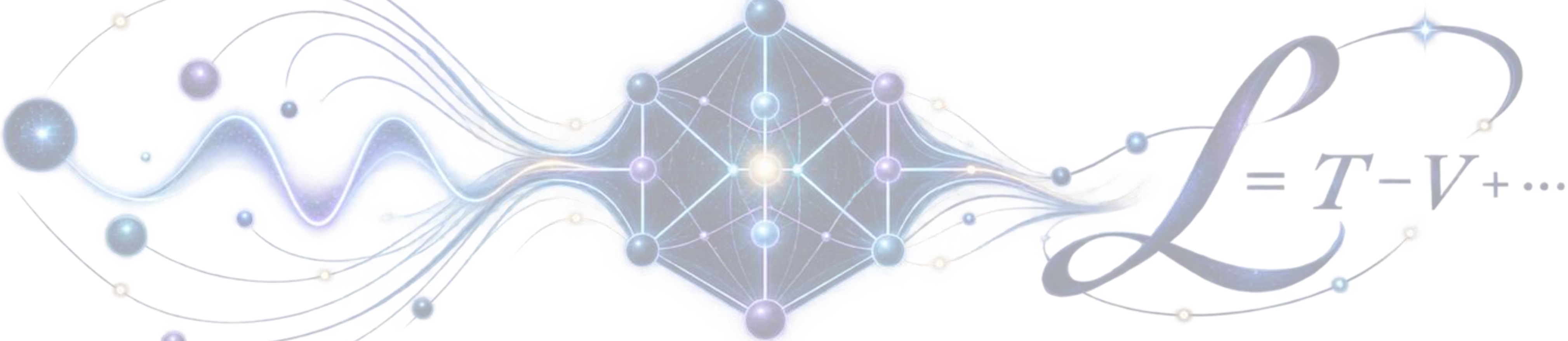
Fine-tuning Small Language Models for Particle-Physics Lagrangian Generation

Paul Schmidt

Supervised by:

Andreas Ipp,

Thomas Richard Ranner



Institute for Theoretical Physics

TU Wien, Austria

Motivation & Research Question

- LLMs show strong general capabilities
- Specialized physics tasks remain challenging
- Can a small pretrained LLM learn a strictly structured particle-physics task?
- Test case: particle-physics Lagrangian generation

Particle-physics Lagrangians

- **Previous work [1]**
  - Transformer-based Lagrangian generation
  - Structured field descriptions as input
    - Domain-Specific Language (DSL)
  - Publicly available datasets [2]

- **Our approach**
  - Pretrained 3B language model
  - Supervised fine-tuning
  - Evaluation of the output

[1] Y. S. Koay, R. Enberg, S. Moretti, and E. Camargo-Molina, “Generating particle physics Lagrangians with transformers,” arXiv:2501.09729 (2025).

[2] JoseEliel, “lagrangian_generation,” Hugging Face Dataset, https://huggingface.co/datasets/JoseEliel/lagrangian_generation

Particle-physics Lagrangians

- Lagrangian encodes the fields, dynamics and interactions

$$\mathcal{L} = \underbrace{\frac{1}{2}\,\partial_\mu\phi\,\partial^\mu\phi}_{\text{kinetic term}} - \underbrace{\frac{1}{2}\,m^2\phi^2}_{\text{mass term}} - \underbrace{\frac{\lambda}{4!}\,\phi^4}_{\text{interaction term}}$$

Simple example: one real scalar field ϕ

Particle-physics Lagrangians

- Field content and quantum numbers
  - Scalar, fermion (spinor), vector fields, tensor fields
    Spin and gauge representations
- Allowed field combinations
  - Symmetries determine which fields may interact
- Invariant contractions
  - Lorentz and gauge indices must be contracted correctly
- Each term must be a Lorentz scalar and a gauge singlet

Dataset

- **Input**
  - One to six matter fields
  - Scalars and fermions
  - Spin and helicity
  - Representations under $SU(3)_C \times SU(2)_L \times U(1)_Y$
- **Output**
  - Kinetic and mass terms
  - Allowed interaction terms
  - Corresponding invariant contractions

Domain-specific language (DSL)

Element	**DSL representation**
Field	`FIELD`
Spin	`SPIN 0`, `SPIN 1 / 2`
Symmetrie	`SU3 3`, `SU2 3`, `U1 2`, `U1 - 4 / 7`, ...
Helicity	`HEL 1 / 2`, `HEL - 1 / 2`
Conjugation	`DAGGER`
Field identifier	`ID0`, `ID1`, ..., `ID9`
Derivative	`DERIVATIVE`
Spinor structure	`SIGMA_BAR`
Contractions	`CONTRACTIONS SU3 ...`, `CONTRACTIONS SU2 ...`, `CONTRACTIONS LORENTZ ...`
Commutators	`COMMUTATOR_A`, `COMMUTATOR_B`
Mathematical symbols	`i`, `+`, `-`, `/`, `0`, ..., `9`

- Example

fields	Lagrangian		train/eval
FIELD SPIN 0	DERIVATIVE ID4 FIELD SPIN 0 DAGGER ID1 DERIVATIVE ID7 FIELD SPIN 0 ID5 CONTRACTIONS LORENTZ ID4 ID7	$\partial^\mu \phi^\dagger \, \partial_\mu \phi$	eval

Dataset

Input

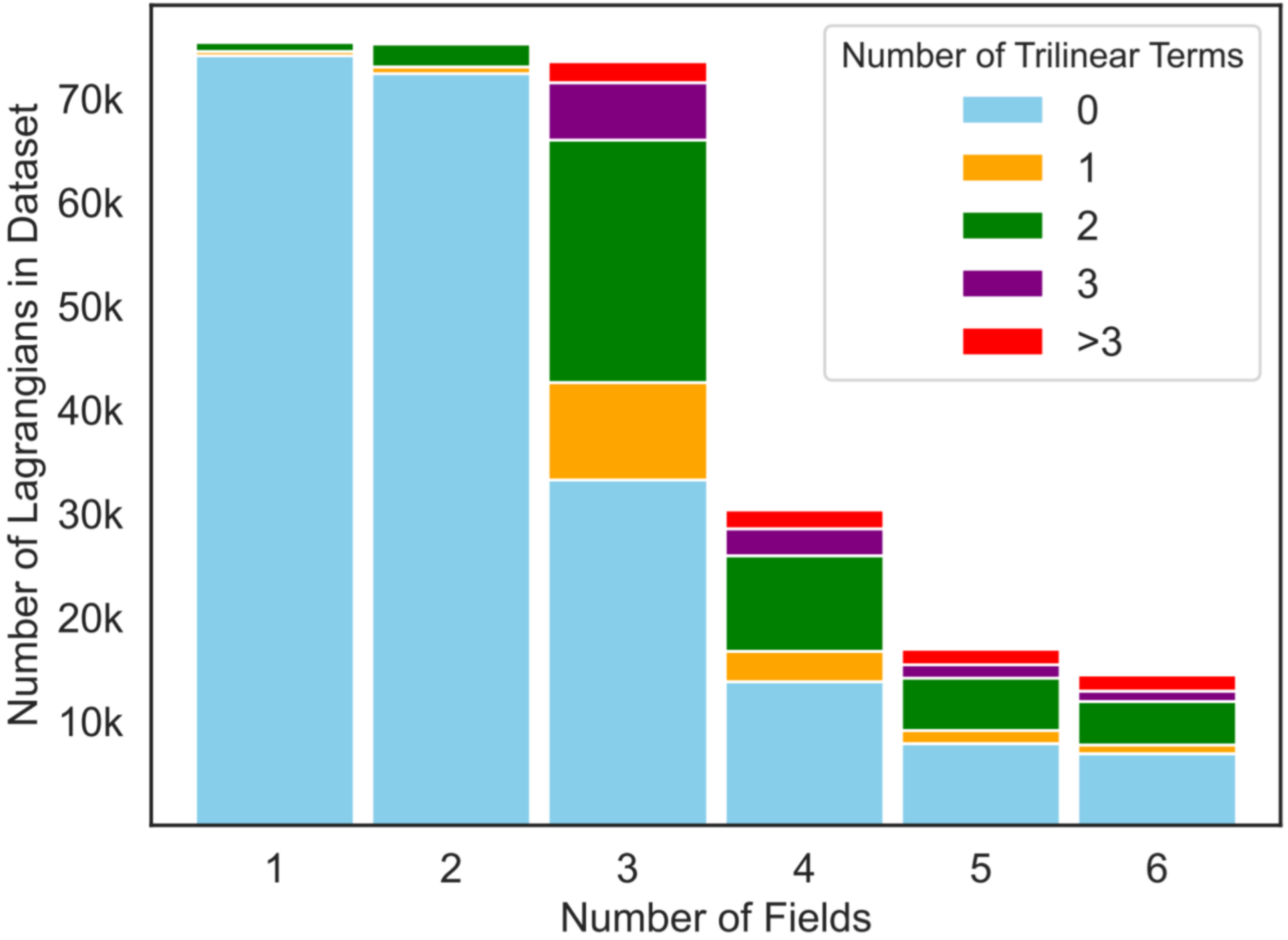
```
FIELD SPIN 1/2 SU2 2 U1 3 HEL 1/2 FIELD SPIN 0 SU2 2 U1 - 7 / 2 FIELD SPIN 0 SU3 - 3 U1 - 8 / 3
```

Target Lagrangian

```
 + FIELD SPIN 0 SU2 2 U1 - 7 / 2 ID1 FIELD SPIN 0 SU2 2 U1 7 / 2 DAGGER ID8 FIELD SPIN 0 SU3 - 3 U1 - 8 / 3 ID3 FIELD SPIN 0 SU3 3 U1 8 / 3 DAGGER ID9 CONTRACTIONS
↪  SU3 ID3 ID3 ID9 SU2 ID1 ID8 + FIELD SPIN 0 SU2 2 U1 - 7 / 2 ID8 FIELD SPIN 0 SU2 2 U1 - 7 / 2 ID5 FIELD SPIN 0 SU2 2 U1 7 / 2 DAGGER ID4 FIELD SPIN 0 SU2 2 U1 7 /
↪  2 DAGGER ID9 CONTRACTIONS SU2 ID8 ID4 SU2 ID5 ID9 + FIELD SPIN 0 SU2 2 U1 - 7 / 2 ID2 FIELD SPIN 0 SU2 2 U1 - 7 / 2 ID7 FIELD SPIN 0 SU2 2 U1 7 / 2 DAGGER ID3
↪  FIELD SPIN 0 SU2 2 U1 7 / 2 DAGGER ID9 CONTRACTIONS SU2 ID2 ID7 SU2 ID3 ID9 + FIELD SPIN 0 SU3 - 3 U1 - 8 / 3 ID8 FIELD SPIN 0 SU3 - 3 U1 - 8 / 3 ID6 FIELD SPIN 0
↪  SU3 3 U1 8 / 3 DAGGER ID9 FIELD SPIN 0 SU3 3 U1 8 / 3 DAGGER ID5 CONTRACTIONS SU3 ID8 ID8 ID9 SU3 ID6 ID6 ID5 + FIELD SPIN 0 SU3 - 3 U1 - 8 / 3 ID8 FIELD SPIN 0
↪  SU3 - 3 U1 - 8 / 3 ID1 FIELD SPIN 0 SU3 3 U1 8 / 3 DAGGER ID7 FIELD SPIN 0 SU3 3 U1 8 / 3 DAGGER ID3 CONTRACTIONS SU3 ID8 ID8 ID1 SU3 ID1 ID7 ID3 + i FIELD SPIN 1
↪  / 2 SU2 2 U1 3 HEL 1 / 2 ID8 SIGMA_BAR ID4 DERIVATIVE SU2 U1 ID0 FIELD SPIN 1 / 2 SU2 2 U1 - 3 HEL - 1 / 2 DAGGER ID1 CONTRACTIONS LORENTZ ID4 ID0 LORENTZ ID4
↪  ID8 ID1 SU2 ID8 ID1 + DERIVATIVE SU2 U1 ID5 FIELD SPIN 0 SU2 2 U1 7 / 2 DAGGER ID0 DERIVATIVE SU2 U1 ID4 FIELD SPIN 0 SU2 2 U1 - 7 / 2 ID3 CONTRACTIONS LORENTZ
↪  ID5 ID4 SU2 ID0 ID3 + DERIVATIVE SU3 U1 ID6 FIELD SPIN 0 SU3 3 U1 8 / 3 DAGGER ID5 DERIVATIVE SU3 U1 ID7 FIELD SPIN 0 SU3 - 3 U1 - 8 / 3 ID1 CONTRACTIONS LORENTZ
↪  ID6 ID7 SU3 ID5 ID1 - COMMUTATOR_A DERIVATIVE SU2 ID9 COMMUTATOR_B DERIVATIVE SU2 ID8 COMMUTATOR_A DERIVATIVE SU2 ID5 COMMUTATOR_B DERIVATIVE SU2 ID4
↪  CONTRACTIONS LORENTZ ID9 ID5 LORENTZ ID8 ID4 - COMMUTATOR_A DERIVATIVE U1 ID3 COMMUTATOR_B DERIVATIVE U1 ID4 COMMUTATOR_A DERIVATIVE U1 ID8 COMMUTATOR_B
↪  DERIVATIVE U1 ID2 CONTRACTIONS LORENTZ ID3 ID8 LORENTZ ID4 ID2 - COMMUTATOR_A DERIVATIVE SU3 ID2 COMMUTATOR_B DERIVATIVE SU3 ID8 COMMUTATOR_A DERIVATIVE SU3 ID3
↪  COMMUTATOR_B DERIVATIVE SU3 ID4 CONTRACTIONS LORENTZ ID2 ID3 LORENTZ ID8 ID4 + FIELD SPIN 0 SU2 2 U1 7 / 2 DAGGER ID7 FIELD SPIN 0 SU2 2 U1 - 7 / 2 ID2
↪  CONTRACTIONS SU2 ID7 ID2 + FIELD SPIN 0 SU3 3 U1 8 / 3 DAGGER ID8 FIELD SPIN 0 SU3 - 3 U1 - 8 / 3 ID2 CONTRACTIONS SU3 ID8 ID2
```

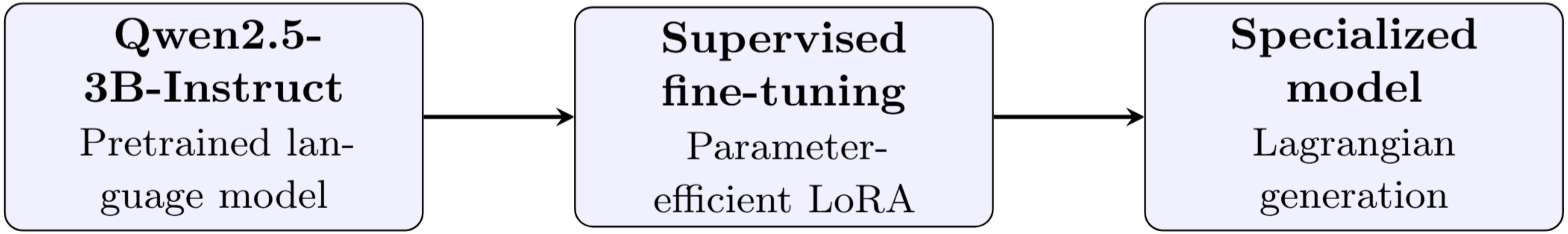
Sampled training dataset

- Training data are dominated by examples with one to three input fields
- Examples with trilinear interaction were sampled more frequently



[1] Y. S. Koay, R. Enberg, S. Moretti, and E. Camargo-Molina, “Generating particle physics Lagrangians with transformers,” arXiv:2501.09729 (2025).

Training



- **Base model:** Qwen2.5-3B-Instruct
- **Training method:** supervised fine-tuning with LoRA (Low Rank Adaptations
- **Input:** structured field description
- **Target: Target:** corresponding Lagrangian expressed as a sequence of DSL symbols

https://huggingface.co/Qwen/Qwen2.5-3B-Instruct

Scoring

$$S_{\text{Object}} = \frac{N_{\text{Correct Objects}}}{N_{\text{Expected}}}$$

Expected terms matched at the object level

$$S_{\text{Contraction}} = \frac{N_{\text{Correct Contractions}}}{N_{\text{Expected}}}$$

Correct objects and correct contraction structure

$$P_{\text{Length}} = \frac{N_{\text{Extra}}}{N_{\text{Expected}}}$$

Penalty for additional predicted terms

$$S_{\text{Lagrangian}} = S_{\text{Contraction}} - P_{\text{Length}}$$

Overall structural score after penalizing additional terms

Following Koay et al. [1], we use the same task-specific scoring scheme.

Predicted and reference Lagrangians are split into individual terms; randomly assigned IDs are canonicalized before comparison.

[1] Y. S. Koay, R. Enberg, S. Moretti, and E. Camargo-Molina, “Generating particle physics Lagrangians with transformers,” arXiv:2501.09729 (2025).

Overall performance

- Perfect Lagrangian Score

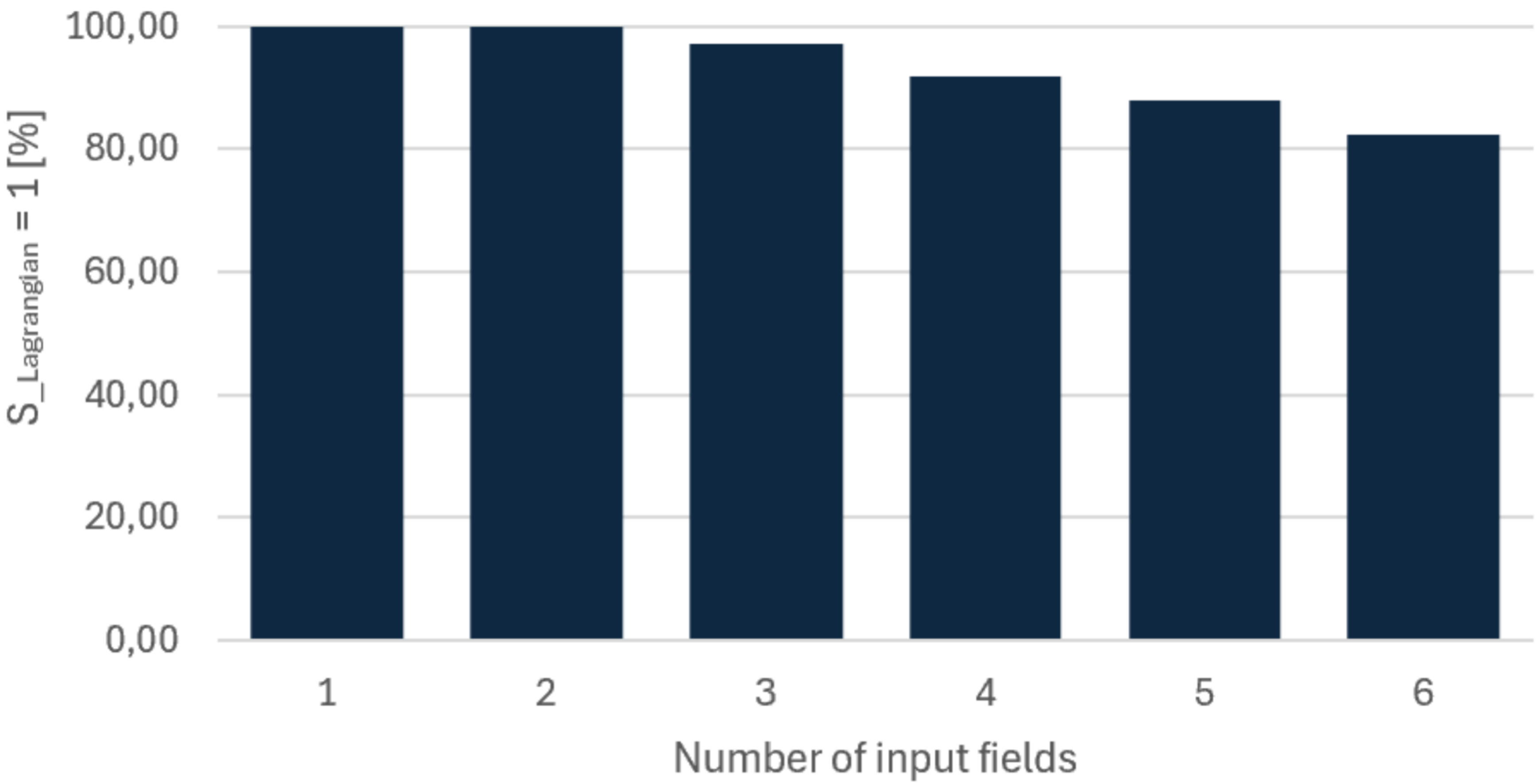
$$S\_{\text{Lagrangian}} = 1$$

$$\boxed{95.40\%}$$

$$N\_{\text{Eval}} = 40751$$

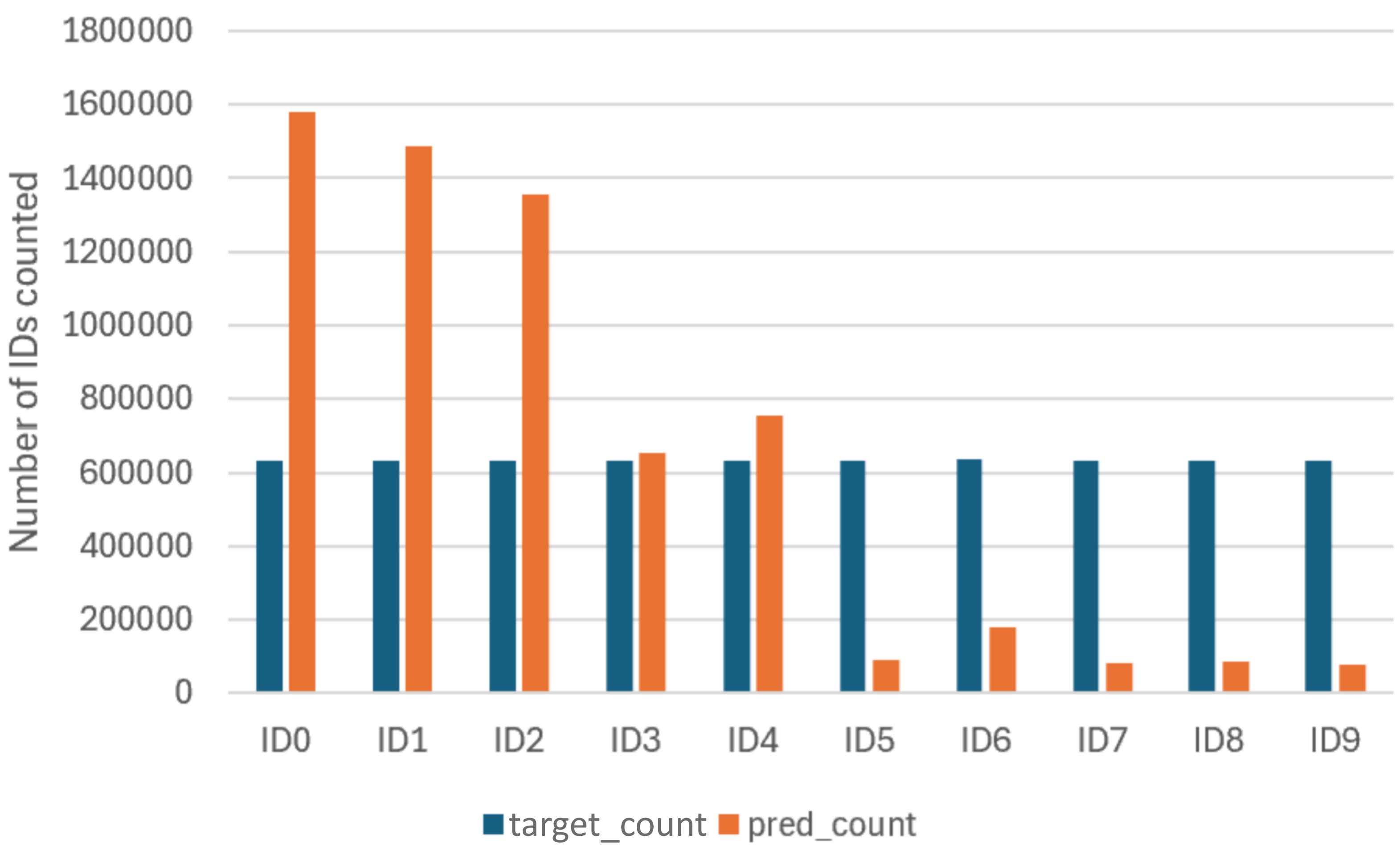
- Diagnostic results
  - **96.75%** object-perfect
  - **96.58%** contraction-perfect
  - **98.77%** no excess terms

Performance decreases with increasing field count



Sample sizes: n = 4 for one field; n = 2,899 - 14441 for two to six fields.

Perfect predictions still exhibit an ID bias



Predictions strongly favor lower IDs

Exploratory model and prompt comparison

- Same 200 examples for each setting
- Zero-shot
  - Task instructions + evaluation input ⟶ model output
- Few-shot
  - Task instructions + 6 input-output examples + evaluation input ⟶ model output

Qwen 3B
https://huggingface.co/Qwen/Qwen2.5-3B-Instruct

Qwen 3B+SFT

Qwen 7B
https://huggingface.co/Qwen/Qwen2.5-7B-Instruct

Qwen 32B
https://huggingface.co/Qwen/Qwen2.5-32B-Instruct

Prompt-based comparison ZERO_SHOT

Model	DSL vocabulary	Excact matches	Qualitative observation
Qwen 3B	128/200	0/200	Frequent DSL fragments, but no complete target terms found
Qwen 7B	15/200	0/200	Often alternative notation such as PHI CHI, repetitions
Qwen 32B	189/200	0/200	Output improves, ID and order issues, is it really better?
Qwen 3B SFT	148/200	5/200	Often structurally analyzable and partially consistent with the target solution,

Prompt-based comparison FEW_SHOT

- Complete terms compared after ID canonicalization
- Term order ignored
- Only term-comparable outputs included

$$R_{term} = \frac{N\_matched}{N\_target}$$

Term recall: fraction of target terms recovered

$$P_{term} = \frac{N\_matched}{N\_generated}$$

Term precision: fraction of generated terms matching the target

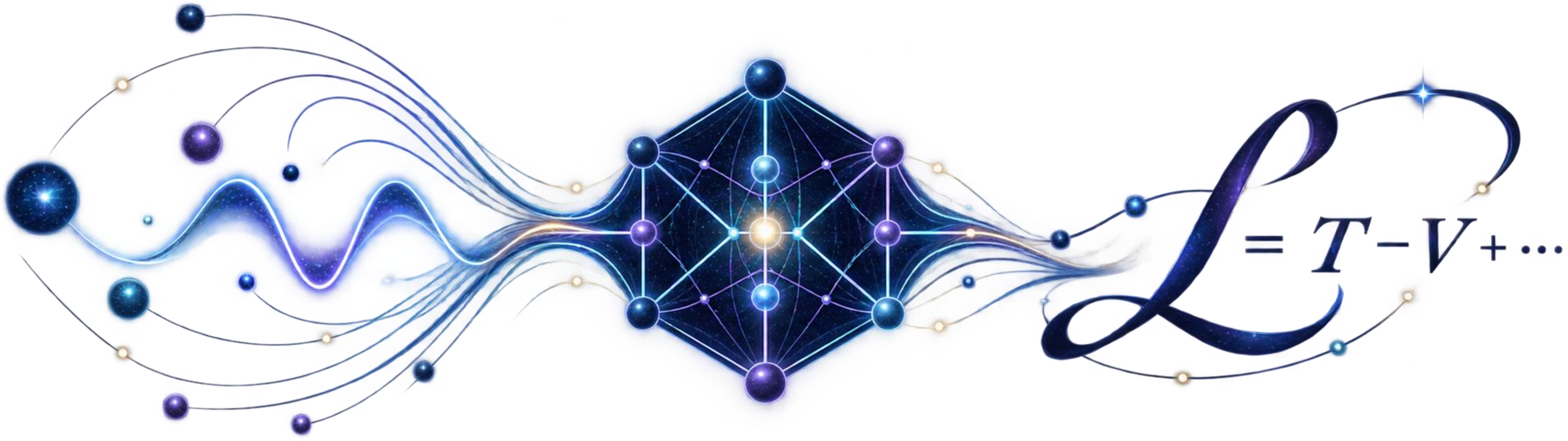
Prompt-based comparison FEW_SHOT

Model	DSL Output	R_Term	P_Term	Excact matches
Qwen 3B	97.0%	10.4%	12.5%	0/200
Qwen 7B	98.5%	13.8%	20.9%	0/200
Qwen 32B	99.5%	17.0%	23.8%	1/200
Qwen 3B SFT	100.0%	46.3%	86.0%	11/200

Conclusions and outlook

- **Main conclusions**
  - Task-specific SFT outperformed scaling the raw model size in this setup
  - Performance was sensitive to the match between training and inference formats
  - Successful DSL generation demonstrates learned structural patterns, but not necessarily physical understanding
- **Future directions**
  - Systematic prompt design: explicit zero-shot instructions versus few-shot examples
  - Prompt-aware fine-tuning using the intended inference format
  - Analysis of generalization and output biases

Thank You for your Attention!



Setup

HPC system	MUSICA
GPU	1× NVIDIA H100 SXM5
GPU memory	94 GB
CPU / RAM	8 cores / 40 GB
Precision	BF16
Batch size / GPU	2
Gradient accumulation	8 steps
Effective batch size	16
Max. sequence length	4096 tokens

Training: Qwen2.5-3B-Instruct + LoRA
Frameworks: Hugging Face Transformers, TRL, PEFT

https://docs.asc.ac.at/systems/musica.html

Training

Training information	Value
Training examples	228,866
Final training step	9,210
Training time	~20 h
Checkpoint 1,000	87% perfect
Selected checkpoint 9,000	97% perfect

SFT Trainer / Tokenizer

```python
MODEL_ID = "Qwen/Qwen2.5-3B-Instruct"

MAX_LENGTH = 4096
NUM_TRAIN_EPOCHS = 3
LEARNING_RATE = 2e-4
WARMUP_RATIO = 0.03

PER_DEVICE_TRAIN_BATCH_SIZE = 2
PER_DEVICE_EVAL_BATCH_SIZE = 2
GRADIENT_ACCUMULATION_STEPS = 8

lora_config = LoraConfig(
    r=16,
    lora_alpha=32,
    lora_dropout=0.05,
    bias="none",
    task_type="CAUSAL_LM",
    target_modules=[
        "q_proj", "k_proj", "v_proj", "o_proj",
        "gate_proj", "up_proj", "down_proj",
    ],
)

sft_args = SFTConfig(
    max_length=MAX_LENGTH,
    completion_only_loss=True,
    num_train_epochs=NUM_TRAIN_EPOCHS,
    learning_rate=LEARNING_RATE,
    warmup_ratio=WARMUP_RATIO,
    lr_scheduler_type="cosine",
    gradient_checkpointing=True,
    bf16=True,
)
```

DSL Token	Qwen2.5 Tokens	Token IDs
FIELD	FIELD	28747
SPIN	SP · IN	4592, 687
SU3	SU · 3	59782, 18
U1	U · 1	52, 16
HEL	HEL	50712
DAGGER	D · AG · GER	35, 1890, 8738
SIGMA_BAR	SIG · MA · _BAR	50631, 4835, 47405
DERIVATIVE	DER · IV · ATIVE	11391, 3090, 23215
COMMUTATOR_A	COMM · UT · ATOR · _A	55554, 1381, 11857, 1566
LORENTZ	LO · RE · NT · Z	1593, 787, 6408, 57
CONTRACTIONS	CON · TRA · CTIONS	5790, 20252, 36768
ID0	ID · 0	915, 15

Supervised Fine-Tuning

- **Goal:** Adapt a pre-trained language model to a specific task using input–target pairs.
  - Field description $\longrightarrow$ Target Lagrangian
  - Model predicts the target token by token
  - Prediction is compared with the known target
  - Training minimizes the prediction loss
  - Learns the task-specific DSL structure and mapping

Low-Rank Adaptation (LoRA)

- **Goal:** Fine-tune efficiently without updating all model parameters.
  - $W = W_0 + \Delta W, \Delta W = BA, r \ll d$
  - Pre-trained weights W_0 remain **frozen**
  - Only small low-rank matrices A and B are trained
  - Far fewer trainable parameters → lower memory and compute requirements

SFT defines what the model learns — LoRA defines how the model is efficiently adapted.

Canonicalization

Prediction: + FIELD SPIN 0 ID12 FIELD SPIN 0 ID7 CONTRACTION ID7 ID12 + FIELD …

Target: + FIELD SPIN 0 ID3 FIELD SPIN 0 ID8 CONTRACTION ID8 ID3 + FIELD …

After canonicalization

Both: + FIELD SPIN 0 ID0 FIELD SPIN 0 ID1 CONTRACTION ID1 ID0

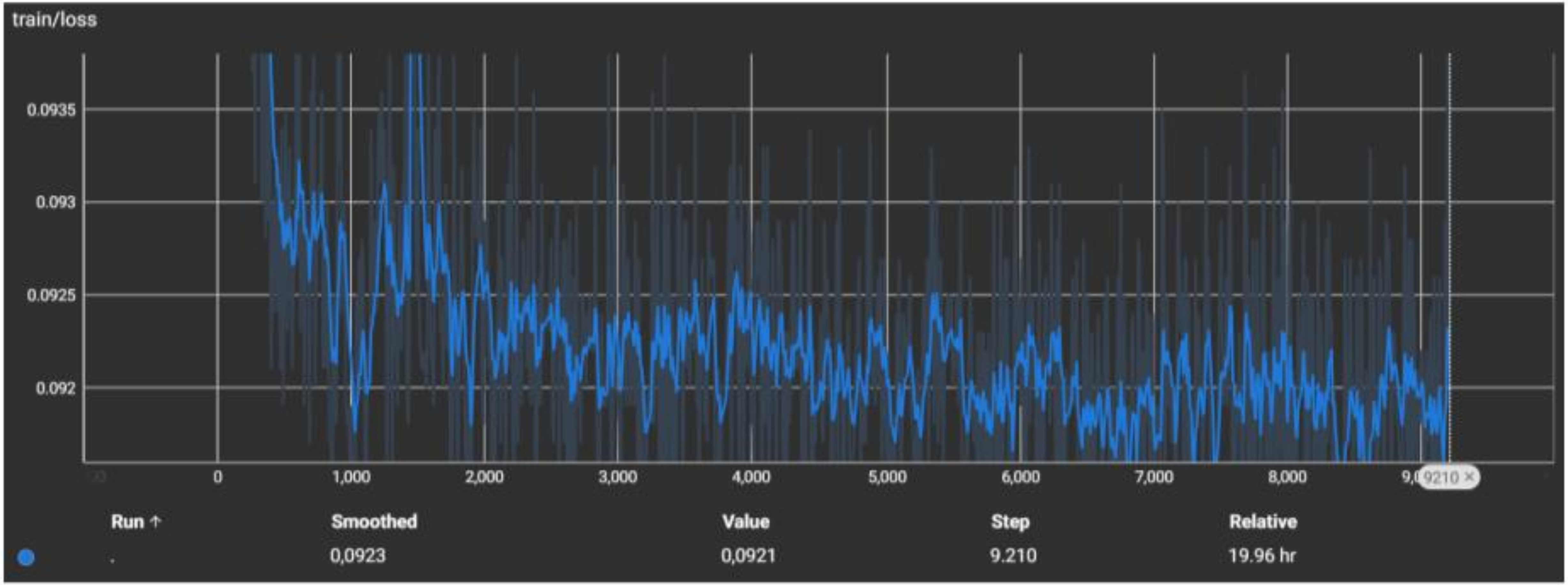
Results reported in the original work

Metric	Sampled model	Uniform model
$S_{\text{Lagrangian}} = 1$	93.4%	89.9%
$S_{\text{Contraction}} = 1$	95.8%	90.2%
$S_{\text{Object}} = 1$	95.9%	90.5%
$P_{\text{Length}} = 0$	97.4%	99.6%

Reference only — evaluation setup differs from this work

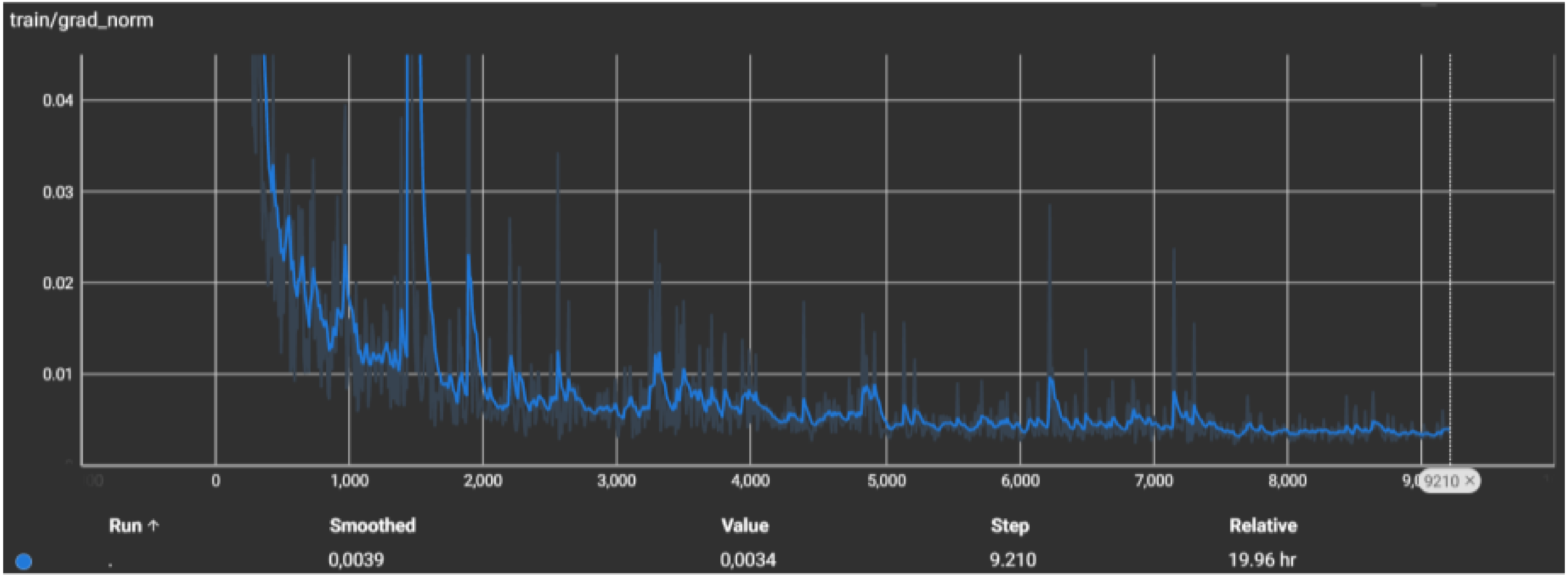
[1] Y. S. Koay, R. Enberg, S. Moretti, and E. Camargo-Molina, “Generating particle physics Lagrangians with transformers,” arXiv:2501.09729 (2025).

Tensorboard loss



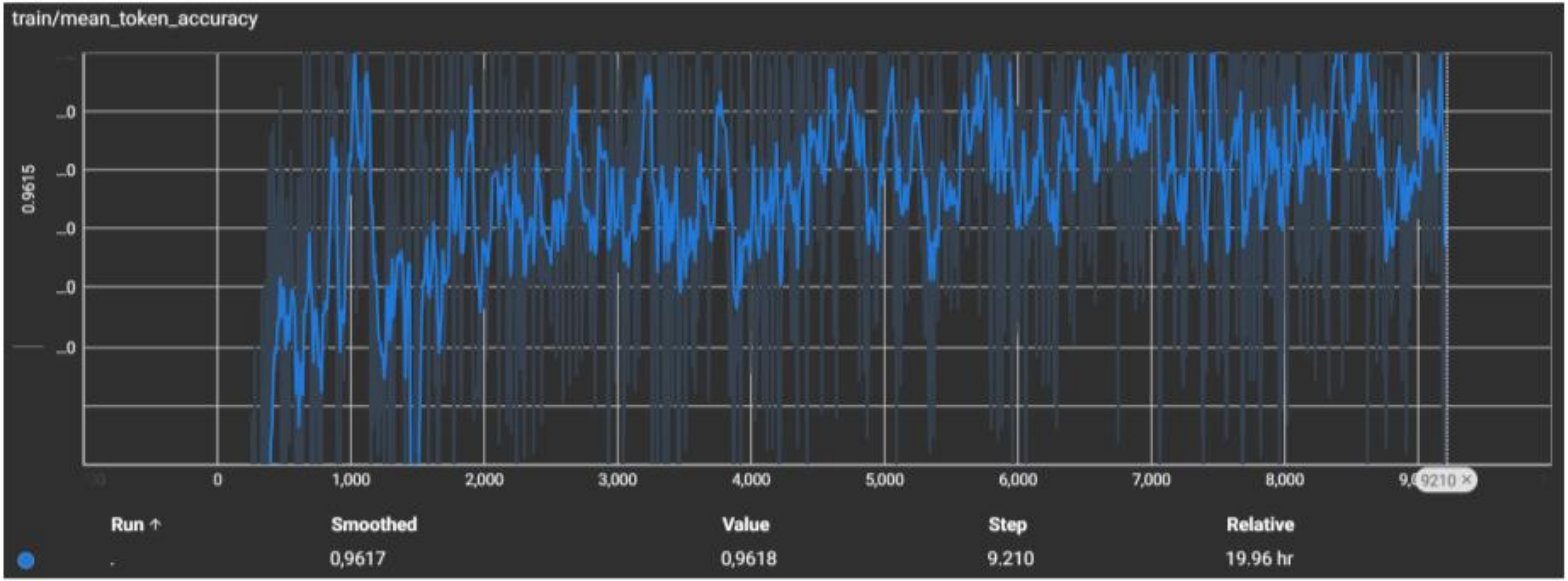
(a) Trend in training loss during SFT training.

Tensorboard gradient



(b) Changes in the gradient norm during SFT training.

Tensorboard token accuracy



(c) Trend in average token accuracy during SFT training.

Output example

Example 24 / Qwen 3B raw zero-shot

Raw response

```
<think>
I need to construct the Lagrangian for the given fields, which include fermions (spin 1/2) transforming under SU(3),
↪  SU(2), and U(1) with specific charges, and scalars (spin 0) also transforming under SU(2) and U(1). The goal is to
↪  create kinetic terms, mass terms, and possible interaction terms respecting the gauge symmetries.

First, I will generate the kinetic terms for the fermions and scalars. Then, I will add mass terms if applicable.
↪  Finally, I will consider possible interaction terms between the fermions and scalars, ensuring they respect the
↪  gauge symmetries.

Kinetic terms for fermions: $\bar{\psi} \gamma^\mu D_\mu \psi$
Kinetic terms for scalars: $D_\mu A^\mu$
Mass terms for fermions: $\frac{m}{\sqrt{2}} \bar{\psi} \gamma^0 \psi$
Mass terms for scalars: $- \frac{1}{2} m^2 A^\mu A_\mu$

Interaction terms: $\bar{\psi} \gamma^\mu (i D_\nu) \psi A^\nu$ (Fermion-scalar interaction)

I will now generate these terms in the required tokenized format.
</think>
<answer>
HEL HEL HEL HEL HEL HEL HEL HEL HEL HEL
[... HEL repeated ...]
```

Input

```
FIELD SPIN 1/2 SU3 3 SU2 2 U1 2 HEL 1/2 FIELD SPIN 0 SU2 3 U1 - 7 / 2 FIELD SPIN 0 SU2 2 U1 - 1 FIELD SPIN 0 SU3 3 U1 -
↪  1 / 4 FIELD SPIN 0 SU3 3 SU2 2 U1 3 / 4
```

Output example

Example 76 / Qwen 7B zero-shot

Raw response

```
<think>
We need to generate all valid Lagrangian terms from the given fields. The fields are:
1. A SU(3) triplet (denoted as \( \phi \))
2. A U(1) septuplet (denoted as \( \psi \))

We will consider kinetic terms, quadratic terms, and interaction terms while ensuring gauge invariance.
</think>
<answer>
FIELD DERIVATIVE FIELD CONTRACTIONS ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID
↪  ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID
↪  ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID
↪  ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID
↪  ID ID ID ID ID [... ID repeated ...]
ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID
↪  ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID
```

Output example

Example 76 / Qwen 7B few-shot

Raw output

```
<think> The input fields are mapped to the corresponding tokenized Lagrangian structure shown in the dataset. We need to consider all possible terms involving the
↪  given fields, including kinetic terms, quadratic terms, and interaction terms, while respecting the gauge symmetries and group labels. </think> <answer> + FIELD
↪  SPIN 0 SU3 - 3 U1 1 / 5 ID2 FIELD SPIN 0 U1 7 / 4 ID3 + FIELD SPIN 0 SU3 - 3 U1 1 / 5 ID3 FIELD SPIN 0 U1 7 / 4 ID2 + DERIVATIVE ID2 FIELD SPIN 0 SU3 - 3 U1 1 / 5
↪  DAGGER ID4 DERIVATIVE ID3 FIELD SPIN 0 U1 7 / 4 CONTRACTIONS LORENTZ ID2 ID3 LORENTZ ID4 ID2 ID3 + DERIVATIVE ID3 FIELD SPIN 0 U1 7 / 4 DAGGER ID4 DERIVATIVE ID2
↪  FIELD SPIN 0 SU3 - 3 U1 1 / 5 CONTRACTIONS LORENTZ ID3 ID2 LORENTZ ID4 ID3 ID2 - COMMUTATOR_A DERIVATIVE SU3 ID2 COMMUTATOR_B DERIVATIVE SU3 ID4 COMMUTATOR_A
↪  DERIVATIVE SU3 ID3 COMMUTATOR_B DERIVATIVE SU3 ID5 CONTRACTIONS LORENTZ ID2 ID3 LORENTZ ID4 ID5 - COMMUTATOR_A DERIVATIVE U1 ID2 COMMUTATOR_B DERIVATIVE U1 ID4
↪  COMMUTATOR_A DERIVATIVE U1 ID3 COMMUTATOR_B DERIVATIVE U1 ID5 CONTRACTIONS LORENTZ ID2 ID3 LORENTZ ID4 ID5 </answer><|im_end|>
```

Task description zero_shot

```
You are given an input sequence that defines a set of particle fields by their spin and gauge group representations.

Your task is to generate the corresponding physically allowed Lagrangian terms in the required tokenized output format.

The output should contain all valid terms that can be constructed from the given fields, including kinetic terms, quadratic terms, interaction terms, gauge field strength terms, and the required index contractions, whenever they are allowed by the field content and gauge symmetries.

The output is not a particle name, not a natural language explanation, and not a copy of the input. It must be the full tokenized Lagrangian sequence.

Rules:
- Use only the fields given in the input.
- Generate only terms that are consistent with the gauge group representations.
- Preserve the exact token vocabulary and formatting shown in the examples.
- Use the tokenized format with FIELD, DERIVATIVE, COMMUTATOR_A, COMMUTATOR_B, CONTRACTIONS, ID tokens, DAGGER, and group labels where required.
- Do not repeat the prompt.
- Do not include role labels such as user or assistant.

Return your response in exactly this format:

<think>
Briefly explain what you are trying to construct.
</think>
<answer>
Only the final tokenized Lagrangian sequence.
</answer>

Examples:
```