

Project Agapé

Instruct Pipeline Updates

Free T4 cloud notebook • 12 September 2026

Ready-to-upload notebook: `Project_Agape_Instruct_Free_T4.ipynb`. Equivalent percent-cell source: `Project_Agape_Instruct_Free_T4.py`. Only code cells 2, 4, and 8 changed. The original raw-base notebook and completed artifacts are preserved.

Cell 2: checkpoint and new run

```
MODEL_ID = "meta-llama/Llama-3.2-1B-Instruct"
# Resolve this repository's SHA; do not reuse the raw-base SHA.
MODEL_REVISION = None
RUN_NAME = "agape-instruct-t4-" + time.strftime("%Y%m%d-%H%M%S")
```

Keep the existing locations of these assignments. Start a new run from Cell 2 after installing dependencies and restarting as instructed in Cell 1. Obtain access to the Instruct repository using the HF account behind the token. Train fresh SFT/DPO adapters and recompute the reference cache; raw-base adapters and reference scores are not interchangeable with Instruct artifacts.

The initial comparison retains `SFT_LR=1e-4`, `DPO_LR=1e-5`, and `DPO_BETA=0.1`. No learning-rate change is required for compatibility. A later, separately recorded sensitivity run could lower SFT to `2e-5` or DPO to `5e-6` to explore a smaller update; these are experimental choices, not established optima. Keep the seed, epochs, beta, and other factors fixed when comparing rates.

Cell 4: native template

Keep the existing tokenizer setup and vocabulary checks:

```
tokenizer = AutoTokenizer.from_pretrained(MODEL_ID, revision=REVISION)
tokenizer.padding_side = "right"
tokenizer.eos_token = "<|eot_id|>"
tokenizer.pad_token = "<|finetune_right_pad_id|>"
```

Delete the old manual `tokenizer.chat_template = (...)` assignment and insert:

```
assert tokenizer.chat_template, "Missing Instruct chat template"
CHAT_TEMPLATE_KWARGS = {"date_string": "12 Sep 2026"}
```

In `prompt_text()`, retain the role/content checks and replace the final return with:

```
return tokenizer.apply_chat_template(
    messages, tokenize=False, add_generation_prompt=True,
    **CHAT_TEMPLATE_KWARGS)
```

The fixed date is passed consistently so a template that uses date text cannot change the prompt across training and inference. Keep `add_special_tokens=False` when tokenizing rendered prompts. Keep DPO completions as plain response strings: TRL 0.23.1 appends EOS itself. Do not render a full assistant turn into the chosen/rejected completion and then append another EOT.

The supplied notebook additionally compares native full-conversation token IDs with prompt + completion + one EOT for every training alternative. It stops on a mismatch, including unexpected whitespace/template behavior, and saves the template hash and kwargs in the manifest. No vocabulary expansion or embedding resize is needed.

Cells 5-7: retain existing training

Keep NF4 double quantization, FP16, SDPA, gradient checkpointing, microbatch 1, accumulation 4, limits 768/512/256, and LoRA rank 8/alpha 16. Targets remain:

```
target_modules = ["q_proj", "k_proj", "v_proj", "o_proj",  
                  "gate_proj", "up_proj", "down_proj"]
```

Keep the new SFT adapter as the frozen reference for DPO on the same Instruct backbone. The existing named-adapter, precomputation, finite-score, hash, mask, checkpoint, and Drive logic remains intact. The prior measured raw-base memory peaks are not measurements of this new run; inspect this run's logs.

Cell 8: comparative probe

Disabling LoRA now gives the untouched Instruct checkpoint, which already has Meta SFT/RLHF. Relabel the first condition accordingly and retain the SFT-only and SFT+DPO conditions. All conditions use the same generic system message, prompt, template, quantization, and greedy decoding.

The supplied notebook reloads the saved tokenizer and uses this generation configuration:

```
probe_generation = copy.deepcopy(model.generation_config)  
stop_ids = probe_generation.eos_token_id  
stop_ids = [stop_ids] if isinstance(stop_ids, int) else list(stop_ids or [])  
stop_ids += [tokenizer.eos_token_id,  
             tokenizer.convert_tokens_to_ids("<|end_of_text|>")]  
probe_generation.eos_token_id = sorted(set(stop_ids))  
probe_generation.pad_token_id = tokenizer.pad_token_id  
probe_generation.do_sample = False  
probe_generation.num_beams = 1  
probe_generation.temperature = 1.0  
probe_generation.top_p = 1.0  
probe_generation.top_k = 50  
probe_generation.max_new_tokens = 192  
probe_generation.use_cache = True  
torch.cuda.reset_peak_memory_stats()
```

Inside the existing adapter-disabling/inference context:

```
generated = model.generate(**inputs, generation_config=probe_generation)
```

Probe results and interpretation

The result conditions become:

```
results = {  
    "Untouched Instruct (same NF4 quantization)": generate_condition(None),  
    "SFT adapter": generate_condition("sft"),  
    "Agapé SFT + DPO adapter": generate_condition("default"),  
}
```

The generation configuration and complete messages are saved with the probe results. Inference memory peaks are reset independently, correcting the cumulative-peak interpretation in the original probe cell.

Interpretation and validation

Compare contextual empathy, firm non-retaliation, useful alternatives, factual honesty, and repetition across all three outputs. The seen Sample A prompt remains a qualitative smoke probe; it is not independent evidence of improved jailbreak robustness. Instruct initialization tests incremental Agapé adaptation on top of existing SFT/RLHF, not developmental alignment versus RLHF from scratch. The prior repetitive raw-base result is consistent with inadequate chat adaptation, but one probe does not establish its sole cause.

All eight code cells compile, source and notebook match, and the unchanged seed export yields six SFT examples and twelve preference pairs. The Instruct checkpoint's tokenizer files require authenticated access; exact native serialization is enforced by cloud assertions. This new variant has not been GPU-executed, and no new behavioral or VRAM result is claimed.

Official references

[Meta Llama 3.2-1B-Instruct model card](#)

[Transformers 4.56.2 chat templates](#)

[TRL 0.23.1 DPO trainer](#)

[TRL 0.23.1 tokenization source](#)