

POST-TRAINING OF LLMS

Week 2. Which model should we use and how?

CHAN YOUNG PARK

Assistant Professor, iSchool, School of Computing, The University of Texas at Austin

Recap: what did Week 1 establish?

- Base vs instruction-tuned checkpoint: what changed?
 - Continuing text generation vs responding to an instruction
- Post-training introduces a target behavior
 - The LM objective leaves important choices unspecified (intent, format, preference, constraints)
 - These are not simply “knowledge gaps”
- Lab: how are the two models different and why?
 - Base models don’t know when to stop (they repeat a lot)
 - Base models often miss the user’s intent
 - Post-trained models are more constrained (safety guardrails)
 - Post-training is what makes a model follow instructions and be to-the-point, helpful, and safe.

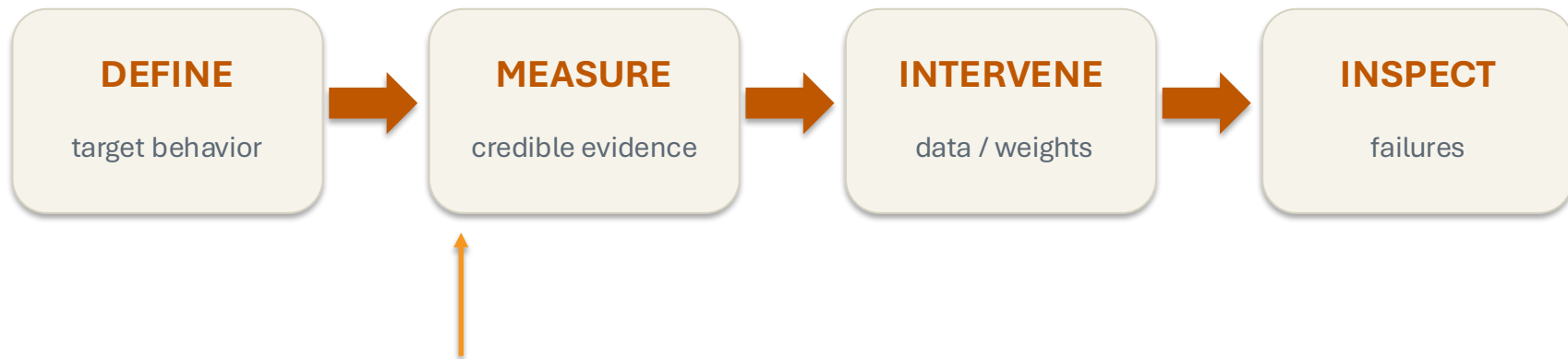
The central question of Week 2

- Before changing model weights, which model + prompt is a credible baseline?
- Is the remaining quality–cost gap worth post-training?

Model selection process

- Define target behavior
 - Task formulation
 - Construct or find dev/test data
 - Set an acceptable minimum performance threshold
- Choose a baseline model based on the task and practical constraints
- Explore different prompting techniques
- Compare models across a range of sizes and costs
- Decide: which model to use and whether post-training is worth it

The empirical loop



Week 2 lives before the first weight update

Step 1. Define target behavior

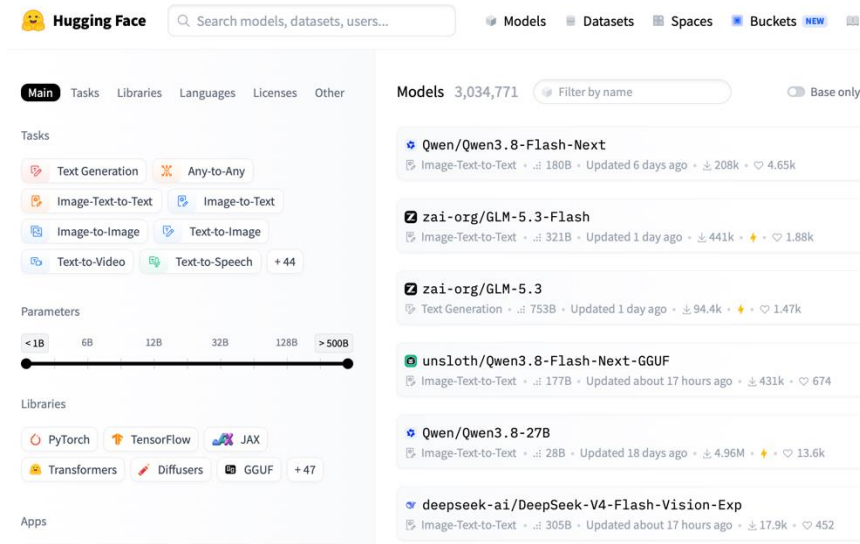
- Task formulation: what exactly is a task?
- How are you going to communicate it to a model?
 - Task description (definition, action guide, checklist, etc.)
 - Label definitions
 - Demonstrations (examples)
 - Output constraints
- Construct or find dev/test data
- Set an acceptable minimum performance threshold

Step 2. Choose a baseline model

- Adaptability: can we train or distill it later?
- Task: modality, language, behavior
- Interface: base/instruct, reasoning/non-reasoning, max context length
- Hardware: does the model fit in GPU memory?
- Deployment: supported inference engines

Task: modality, language, behavior

- <https://huggingface.co/models>



The screenshot shows the Hugging Face Models page. On the left, there are filters for Tasks (Text Generation, Image-to-Text, etc.), Parameters (ranging from <1B to >500B), and Libraries (PyTorch, TensorFlow, JAX, etc.). The main section displays a list of models, including Qwen/Qwen3.8-Flash-Next, zai-org/GLM-5.3-Flash, zai-org/GLM-5.3, unsloth/Qwen3.8-Flash-Next-GGUF, Qwen/Qwen3.8-27B, and deepseek-ai/DeepSeek-V4-Flash-Vision-Exp. Each model entry shows its modality, size, update time, and popularity metrics.

- <https://artificialanalysis.ai>

Artificial Analysis

- Intelligence
- Coding Agent Index
- Image & Video
- Speech
- Capability Indices
- Benchmarks
- Openness Index
- Output Tokens
- Cost
- Speed & Latency
- Providers

Models Coding Agents Speech, Image, Video Inference

Coding Agent Index

Performance, cost, and execution time for leading coding ag software engineering tasks



Quick LLM landscape

- **2023 Feb:** Llama 1 (7–65B) matches GPT-3 → Alpaca, Vicuna
- **2023 Jul:** Llama 2 (7B, 13B, 70B), the first with commercial use allowed
- **2023 Dec:** Mixtral 8×7B, Mixture-of-Experts; ~13B active parameters match 70B dense
- **2024 Feb/Jun:** Gemma 1 and 2; Gemma 2 9B and 27B were exceptionally strong for their size → on-device/edge LLMs
- **2024 Jun–Sep:** Qwen 2 and 2.5, strong multilingual support, 128K context
- **2024 Jul /Dec:** Llama 3.1 405B, then Llama 3.3 → 70B matches GPT-4o
- **2024 Dec, 2025 Jan:** DeepSeek V3 & R1: multi-head latent attention + GRPO, 671B MoE, reasoning becomes default
- **2025 Mar:** Gemma 3, 100+ languages, small multimodal language models
- **2025 Apr, Sep:** Qwen 3 and Qwen 3-Omni, 235B, dual mode (thinking / non-thinking); Omni handles speech and video, plus agentic tool use
- **2026 Aug:** DeepSeek V4, GLM 5.3, Kimi K3 came out; 1–2.8T models are becoming the norm with 1M-token context windows

Interface: reasoning and context length

- Reasoning vs non-reasoning models
 - Non-reasoning (e.g., DeepSeek V3): decode straight to an answer, one likely token at a time; fast and cheap
 - Reasoning (e.g., DeepSeek R1): spend extra tokens “thinking” before answering; they decompose hard logic, math, and code problems through an explicit chain of thought
- Max context length: 128K is the standard, and 1M-token windows are becoming more common (Kimi K3, GPT 5.6, Opus, ...)

GPU memory and model size

- For example, an A100 has 40 GB of VRAM
- Total VRAM = VRAM_Model + VRAM_KV Cache
- CUDA system overhead $\approx 20\%$ of VRAM_model
- VRAM_Model = #parameter (B) * coeff_quantization * 1.2
(FP16/BF16: 2 bytes; INT8: 1 byte)
- VRAM_KV Cache = $2 * \text{\#layers} * \text{KV Heads} * \text{Head dim} * \text{context length} * \text{coeff_quantization}$
- For example, Llama 3.1 8B (bf16) with a 128K context (bf16)
model: $8\text{B} * 2 * 1.2 = 19.2 \text{ GB}$
KV: $2 * 32 * 8 * 128 * 128\text{K} * 2 = 16 \text{ GB}$
total = 35.2 GB
- In most cases, A100/H100 can comfortably serve $\approx 13\text{B}$ and $\approx 30\text{B}$ models

GPU memory and model size

- Inference fit does not imply training fit
- Training needs far more VRAM: ≈ 15 bytes per parameter
- LoRA shrinks gradients and optimizer states
- Llama 3.1 8B: 16 + 120 GB vs. 16 + 7 GB (LoRA)

INFERENCE

weights
KV cache
activations
temporary buffers
fragmentation



FULL TRAINING ADDS

gradients
optimizer states
master weights (often)
more activations

Step 3. Explore prompting techniques

- Parts of a good prompt
 - Instruction: what you want the model to do
 - Context: background details that help the model understand the situation
 - Persona: the role you want the model to play (e.g., a math expert)
 - Format: how you want the answer structured (e.g., JSON or a bulleted list)
 - These map onto: task description (definition, action guide), label definitions, demonstrations (examples), and output constraints

Prompting Techniques

- **Few-shot prompting**: show a few input–output examples in the prompt
- **Chain-of-thought prompting**: “show your reasoning step by step...”
- **Role & persona prompting**: add “Act as a senior software engineer” to the system prompt
- **Prompt chaining**: break a complex task into a sequence of smaller steps (extract key facts → turn them into an outline → write the final report)
- **Meta-prompting**: “act as a prompt engineer...”, then review the output and iteratively refine the prompt
- **Self-consistency prompting**: raise the temperature, generate multiple distinct reasoning paths (CoT), then take a majority vote (best for math and logic)

Step 4. Compare models

- Compare models across a wide range of sizes (0.5B–70B) and across model families
- Cost efficiency: even if you plan to use closed-source frontier models, it is worth checking how open-source models do. It avoids unnecessary spending
- In some cases, latency (time to first token) and throughput (tokens/s) matter as much as quality

Model selection process

- Define target behavior
- Choose a baseline model based on the task and practical constraints
- Explore different prompting techniques
- Compare models across a range of sizes and costs
- Decide: which model to use, and whether post-training is worth it

LAB 2. MODEL SELECTION

Orbit parcel operations



- Every registered route, whether across one room or 900 light-years, has a normal delivery promise of exactly 2 minutes.
- Reverse gravity can make parcels float. Floating by itself is normal and is not evidence of damage.
- Blue plasma residue is harmless shipping dust. Green plasma residue is hazardous.
- Shipment status progresses in this order: BOOKED, IN_TRANSIT, GATE_LOCKED, DELIVERED.

Five actions and ten rules

ESCALATE**REFUND****REPLACE****REROUTE****REQUEST_INFO**

R01 green plasma / chrontal duplicate → ESCALATE

R02 post-lock reroute → ESCALATE

R03 missing decisive fact → REQUEST_INFO

R04 duplicate charge → REFUND

R05 >2 min late, no customer-caused hold → REFUND

R06 damaged + explicit refund → REFUND

R07 damaged + replacement requested → REPLACE

R08 pre-lock reroute → REROUTE

R09 ignore normal anomalies → REQUEST_INFO

R10 do not invent remedy → REQUEST_INFO

Explore different prompting techniques

P0

minimal

P1+ Task description
(definitions,
action guide)**P2**+ Demonstrations
(1-shot)**P3**+ Demonstrations
(balanced 5-shot)**P4**+ Boundary
+ checklist**P5**Your own
design

- What information/decision scaffold changed behavior the most?

Use dev to choose the prompt design, final to check it

DEVELOPMENT (20)

- run P0–P5
- inspect failures
- rank top three
- choose one



commit
to one prompt

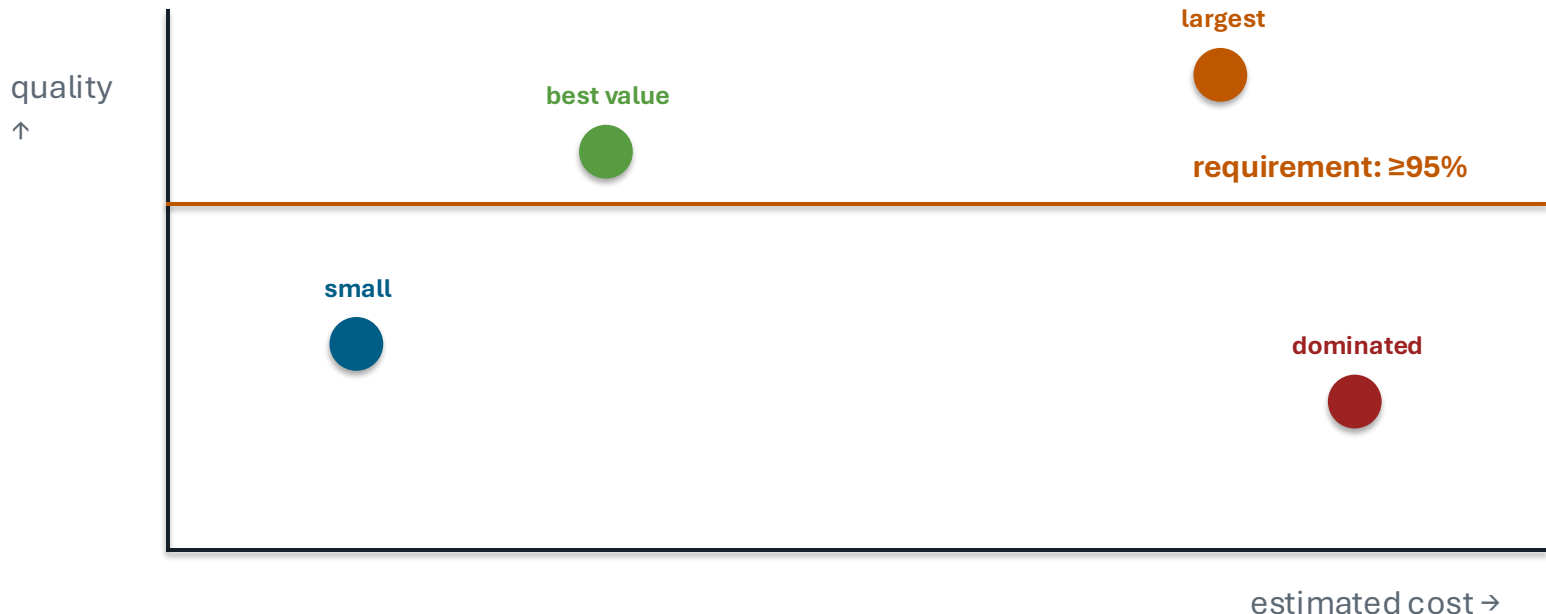
FINAL (30)

- do not revise prompt
- compare various models
- inspect cases

Model Comparison

- Hold fixed: prompt (strategy + demonstrations)
- Change: only the model
- We'll use the Tinker API to run the same prompt across several models

Make your decision



- Based on your investigation, if deploying today, which model would you use?
- Is the remaining quality - cost gap worth post-training?
- Choose among: Deploy the selected model as-is, Post-train/distill, More evidence needed

PROJECT GROUP FORMATION

- **The default group size is three students.** I may allow a small number of two-person groups in special cases, but because we need to keep the total number of project groups manageable, you should plan to form a group of three.
- **If you already have a group of three students,** fill out your information in the *Formed Teams* tab. If your group is fully formed and you have completed the sheet, you are welcome to leave.
- **If you are looking for teammates,** you can add your information in the *Looking for Teammates* tab.
- **If you (or your team) have a project idea to pitch** (max 3 mins) to the class to attract additional interest, please indicate Yes in the “Want to pitch to the class?” column in the Formed Teams tab. I will set aside some time during class for these short pitches.
- Then we’ll have time to meet classmates, discuss potential project topics, and form groups.

Project Pitch

- Introduce the team
- Project idea(s) in mind (the behavior or task of interest)
- Skills / background you bring
- What background or contribution the team needs

Interest-cluster conversation

- **People, society, and alignment:** care work, community governance, pluralistic/community alignment; fairness, cultural bias, health, and social impact
- **Reliability, knowledge, and language:** grounding, hallucination reduction, retrieval, factuality; multilingual behavior, calibration
- **Multimodal models and training methods:** visual understanding and MLLMs; self-distillation, diffusion language models, and method-oriented projects
- **Behavioral experiments and open exploration:** human-AI behavioral questions; unusual model effects, exploratory ideas, and students who are undecided