AI Engineering — Final Exam Revision Guide

A condensed, high-density revision sheet across all 7 chapters. Designed for last-week study. Use this after you have read each chapter file at least once.

Last-minute revision checklist

- ☐ I can write the bigram formula and compute it on a small corpus.
- ☐ I can sketch a Transformer block (attention → residual → norm → FFN → residual → norm).
- ☐ I can write scaled dot-product attention and explain the $\sqrt{d_k}$.
- ☐ I can compare MHA / GQA / MQA.
- ☐ I can compute softmax with temperature for given logits.
- ☐ I can describe BPE in 4 lines with a worked example.
- ☐ I can compare SFT / RLHF / DPO.
- ☐ I can state the Chinchilla rule (~20 tokens/param).
- ☐ I can write the 6-step RAG pipeline end to end.
- ☐ I can compare BM25 / dense / hybrid retrieval and define RRF.
- ☐ I can sketch HNSW and compare it to flat search.
- ☐ I can define Recall@k, Precision@k, MRR, faithfulness.
- ☐ I can distinguish agent vs workflow with one sentence.
- ☐ I can describe ReAct, Planner-Executor, Hierarchical, A2A.
- ☐ I can write LoRA decomposition $\Delta W = (\alpha/r)BA$ and trainable count.
- ☐ I can list EU AI Act four risk tiers.
- ☐ I can apply the 80% rule for disparate impact.
- ☐ I have practiced at least one coding task per chapter.

One-page summary per chapter

Ch1 — Introduction

What's it about? Motivation for the course; bigram LM; Bayes formula in ASR; foundation-model paradigm. Must-know formulas: $P(w_{1..T}) = \prod P(w_t \mid w_{<t})$; $\hat{P}(w \mid w') = c(w', w)/c(w')$; $\arg\max_w P(x \mid w)P(w)$. Must-know diagrams: chain rule; ASR Bayes split. Common exam Qs: ChatGPT acronym; bigram numerical; foundation-model definition. Cheat-sheet line: "LLM = giant next-word predictor; foundation model = pretrain + adapt."

Ch2 — Foundation Models / LLMs

What's it about? Internals of LLMs end-to-end. Must-know formulas: - Softmax-T: $e^{z_i/T}/\sum e^{z_j/T}$ - Self-Attention: $\text{softmax}(QK^T/\sqrt{d_k})V$ - Cross-entropy LM loss - BT preference: $\sigma(r_a - r_b)$ - DPO loss (Section 4.9 in Ch2 file) - FLOPs $\approx 6ND$ - Chinchilla: $D^* \approx 20N$ Must-know algorithms: BPE; self-attention; multi-head; greedy/top-k/top-p; SFT; RLHF (PPO); DPO. Common exam Qs: BPE example; attention by hand; SFT vs RLHF vs DPO; scaling-law multiple choice. Cheat-sheet line: "Tokenize → embed → N transformer blocks → unembed → softmax → sample."

Ch3 — Prompt Engineering

What's it about? Steering an aligned LLM via prompts. Must-know patterns: persona, format, constraint, recipe, reflection. Must-know algorithms: zero/few-shot; CoT; self-consistency; tool-call loop. Common exam Qs: design a JSON-output prompt; explain primacy/recency; sketch tool-call procedure. Cheat-sheet line: "System sets the rules; user asks; model answers; tool calls bridge to the world."

Ch4 — RAG

What's it about? Augmenting LLMs with retrieved documents. Must-know formulas: cosine; BM25; RRF $\sum 1/(k+r)$; Recall@k. Must-know steps: Document preparation → query embed → retrieve → rerank → assemble → generate. Common exam Qs: BM25 vs dense; HNSW vs IVF; cross-encoder rerank; Recall@k vs faithfulness. Cheat-sheet line: "Retrieval = recall, reranker = precision, generator = the writer."

Ch5 — Agents

What's it about? LLMs in a loop with tools. Must-know patterns: ReAct; PE; Hierarchical; A2A. Must-know components: tools, state, budget, stopping. Common exam Qs: define agent vs workflow; identify architecture from trace; failure modes. Cheat-sheet line: "Agent = Model + Control Plane (tools + memory + budget)."

Ch6 — Fine-tuning

What's it about? Updating model parameters cheaply. Must-know formulas: $\Delta W = (\alpha/r)BA$; trainable params $r(d+k)$. Must-know algorithms: full FT; LoRA; QLoRA; adapters. Common exam Qs: derive params; pros/cons LoRA vs Adapter; when not to fine-tune. Cheat-sheet line: "LoRA = small low-rank patch on top of frozen base; QLoRA = same with 4-bit base."

Ch7 — Legal & Ethical

What's it about? Responsibility, fairness, compliance. Must-know terms: GDPR; EU AI Act risk pyramid; bias-as-design; automation bias; dual use; 80% rule. Must-know formulas: DI ratio; KL/PSI drift. Common exam Qs: define high-risk; describe bias as design problem; explain automation bias. Cheat-sheet line: "AI is not just software; legal + ethical + monitoring is part of engineering."

All must-know formulas (single-page reference)

1. Bigram MLE — $P(w \mid w') = c(w', w) / c(w')$
2. Chain rule — $P(w_{1..T}) = \Pi_t P(w_t \mid w_{<t})$
3. ASR posterior — $\text{argmax}_w P(x|w) P(w)$
4. Cosine — $\cos(u,v) = u \cdot v / (||u|| \, ||v||)$
5. Softmax-T — $P_i = \exp(z_i/T) / \Sigma_j \exp(z_j/T)$
6. Self-Attention — $\text{softmax}(QK^\top / \sqrt{d_k}) V$
7. Cross-entropy LM — $L = -1/T \, \Sigma \log P_\theta(w_t \mid w_{<t})$
8. Bradley-Terry — $P(a > b) = \sigma(r_a - r_b)$
9. DPO loss — $-\log \sigma(\beta \cdot [\Delta \log \pi_w - \Delta \log \pi_l])$
10. RLHF KL — $\max E[r] - \beta \, KL(\pi || \pi_{ref})$
11. Chinchilla — $D^* \approx 20 \cdot N$ tokens per param
12. FLOP cost — FLOPs $\approx 6 \cdot N \cdot D$

13. BM25 IDF(t) · TF·(k1+1) / (TF + k1·(1-b+b·|d|/avgdl))
14. RRF Σ_lists 1/(k + rank)
15. Recall@k |relevant ∩ topk| / |relevant|
16. Precision@k |relevant ∩ topk| / k
17. MRR (1/|Q|) Σ 1/rank_first_relevant
18. LoRA decomp $\Delta W = (\alpha/r)\ B\ A$
19. LoRA params r(d + k) per matrix
20. PSI drift Σ (p−q) log(p/q)
21. Disparate Impact DI = rate_min / rate_max

All must-know algorithms (single-page reference)

A. Bigram LM (Ch1)
B. BPE training (Ch2.2)
C. Scaled dot-product attention (Ch2.5)
D. Multi-head attention (Ch2.5)
E. Causal-masked attention (Ch2.5)
F. Greedy / Sampling / Top-k / Top-p (Ch2.7)
G. SFT training step (Ch2.10)
H. RLHF (high-level, PPO) (Ch2.10)
I. DPO loss step (Ch2.10)
J. Few-shot prompt construction (Ch3.3)
K. Self-consistency CoT (Ch3.4)
L. Tool-call loop / ReAct (Ch3.7 + Ch5)
M. RAG end-to-end (Ch4)
N. Reciprocal Rank Fusion (Ch4.4)
O. HNSW (concept) (Ch4.3)
P. Cross-encoder reranker (Ch4.5)
Q. Agent loop with budget (Ch5.1)
R. Planner-Executor (Ch5.2)
S. LoRA forward (Ch6.3)
T. Adapter forward (Ch6.3)
U. Fairness audit (Ch7.3)
V. Drift detection (PSI) (Ch7)

Difficult English terms with Bangla notes

English	Bangla
autoregressive	এক টোকেন করে আগেরগুলোর ওপর শর্ত
pretraining	প্রাথমিক বড়-আকারের প্রশিক্ষণ
fine-tuning	ছোট-পরিসরে চূড়ান্ত শোধন
alignment	প্রান্তিককরণ / পছন্দ-অনুসারী সমন্বয়
supervised	তত্ত্বাবধানে
preference	পছন্দ
reward model	পুরস্কার-মডেল
logit	সফট-ম্যাক্সের আগের স্কোর

English	Bangla
hallucination	মিথ্যা-উদ্ভাবন
chain-of-thought	চিন্তার ধারাবাহিকতা
self-consistency	স্ব-সঙ্গতি
structured output	কাঠামোবদ্ধ আউটপুট
tool calling	টুল আহ্বান
recency / primacy bias	সাম্প্রতিক / প্রাথমিক পক্ষপাত
chunk	ছোট খণ্ড
embedding	ভেক্টর-উপস্থাপন
retrieval	পুনরুদ্ধার
recall	পুনরুদ্ধার-হার
precision	নির্ভুলতা
faithfulness	প্রসঙ্গের প্রতি বিশ্বস্ততা
hybrid	সম্মিলিত
agent loop	এজেন্ট লুপ
control plane	নিয়ন্ত্রণ স্তর
autonomy	স্বয়ংক্রিয়তা
planner / executor	পরিকল্পনাকারী / নির্বাহক
prompt injection	প্রম্পট ইনজেকশন আক্রমণ
low-rank	লো-র‍্যাঙ্ক
catastrophic forgetting	বিপর্যয়মূলক বিস্মৃতি
automation bias	স্বয়ংক্রিয়তার পক্ষপাত
dual use	দ্বৈত ব্যবহার
liability	দায়বদ্ধতা
compliance	আনুগত্য / নিয়ম-পালন
oversight	তদারকি
redaction	তথ্য মুছে ফেলা
drift	বণ্টন-পরিবর্তন

Common exam traps (course-wide)

1. Wrong denominator in bigram MLE.
2. Forgetting $\sqrt{d_k}$ scaling in attention.
3. Confusing causal mask logic — only past, not future.
4. Mixing SFT vs RLHF vs DPO roles (which uses a reward model? PPO? closed form?).
5. Confusing top-k vs top-p.
6. Treating LLaMA as Chinchilla-optimal (it is purposely over-trained).
7. Treating fine-tuning as a knowledge update — use RAG for fresh facts.
8. Confusing recall (retrieval) with faithfulness (generation).
9. Dense beats BM25 always — false; BM25 still wins on rare exact terms.
10. Workflow vs agent confusion — agents have a loop.
11. Planner-Executor doubles cost — yes, because it adds a planner LLM.
12. GDPR $\neq$ AI Act — different scopes.
13. 80% rule is only a screen — fairness is multidimensional.
14. Adapter vs LoRA latency: LoRA can be merged $\rightarrow$ zero latency; adapter cannot.

Short answer templates

"Erläutern Sie X." > X is . It works by . It is used in . Its main advantage over Y is . A common pitfall is .

"Vergleichen Sie A und B." > | Aspect | A | B | > |—|—|—| > | Definition | … | … | > | Cost / complexity | … | … | > | Strength | … | … | > | Weakness | … | … | > | When to choose | … | … |

"Berechnen Sie …" > 1. State the formula. 2. Substitute given values. 3. Compute step-by-step. 4. State units. 5. Interpret.

"Welche Vor- und Nachteile hat X?" > Two-column bullet list (advantages / disadvantages), 2–3 each, with one sentence justification each.

Strategy: theory questions

- Read the question twice; underline keywords (vergleichen, erläutern, berechnen).
- Restate definitions in your own words first.
- Use one concrete example.
- Add a limitation — examiners reward critical thinking.

Strategy: math questions

- Always state the formula first, even if you can do it in your head.
- Show substitutions explicitly.
- Re-check signs and denominators (most-common bigram trap).
- Sanity-check the final number (probabilities $\in [0,1]$, counts non-negative).

Strategy: coding questions

- Pseudocode first, code second.
- Use clear names: bigram_prob, not bp.
- Annotate complexity in a comment.
- If asked for input/output, write them on the side.
- For algorithms, mention any edge cases: empty input, division by zero, OOV tokens.

High-probability exam topics (final shortlist)

1. Bigram LM computation and interpretation (Ch1.2).
2. Self-attention computation by hand (Ch2.5).
3. BPE worked example (Ch2.2).
4. Sampling strategies (Ch2.7).
5. SFT vs RLHF vs DPO comparison (Ch2.10).
6. Chinchilla scaling law (Ch2.11).
7. Tool calling procedure (Ch3.7).
8. End-to-end RAG pipeline (Ch4.1) and BM25 vs dense vs hybrid (Ch4.4).
9. Cross-encoder rerankers (Ch4.5).
10. Agent definition and ReAct trace (Ch5.0–5.2).
11. LoRA math: $\Delta W = (\alpha/r)BA$ and trainable counts (Ch6.3).

12. EU AI Act risk pyramid (Ch7.2).
13. Disparate impact / 80% rule (Ch7.3).

Final exam-day reminders

- Bring a calculator.
- Watch the task verb: explain $\neq$ define $\neq$ compute — answer exactly what the verb asks.
- For multi-part questions, answer the easy parts first.
- For long answers, leave a blank line between paragraphs to ease grading.
- Time-budget: ~1 minute per mark.

Good luck. Viel Erfolg bei der Klausur!

End of Final Revision Guide.