

Supplementary Materials

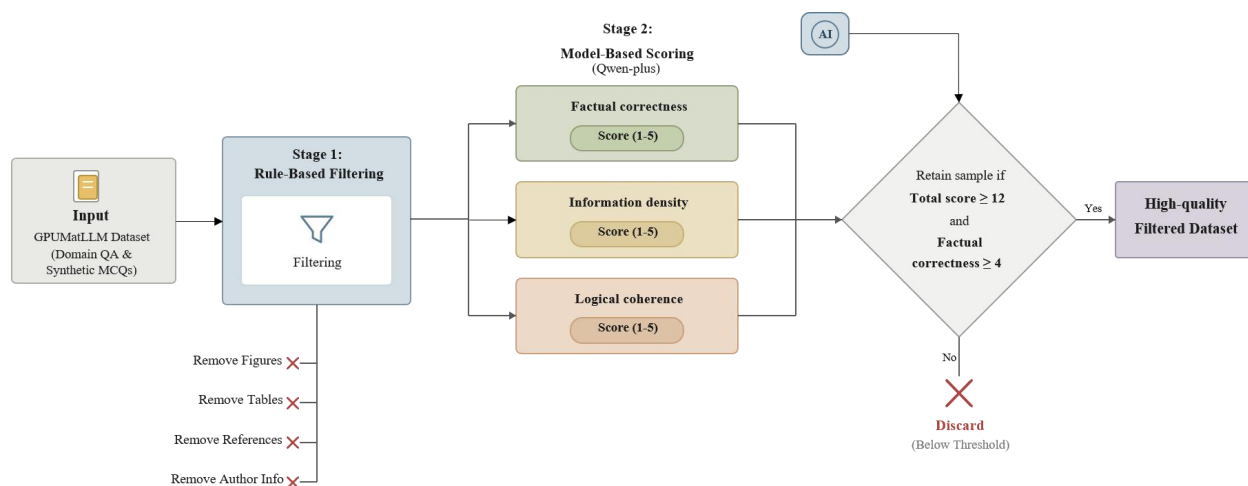
Benchmarking and fine-tuning large language models for domain-specific materials science: a case study toward GPU application

Yuqin Wang^{1,2}, Youle Wang², Lei Zhang^{1,2}

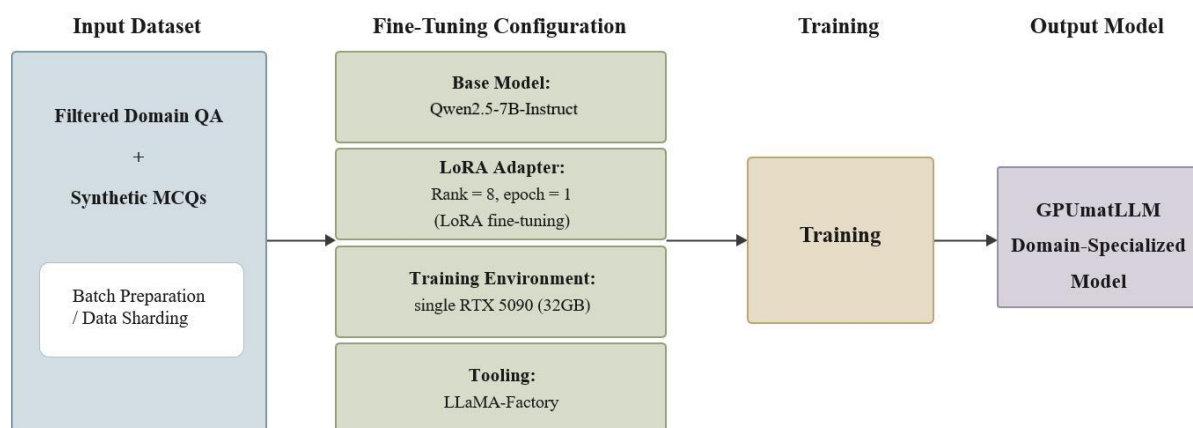
¹Key Laboratory of Spectral Imaging Technology, Xi'an Institute of Optics and Precision Mechanics of the Chinese Academy of Sciences, Xi'an 710119, Shaanxi, China.

²School of Software, Nanjing University of Information Science & Technology, Nanjing 210044, Jiangsu, China.

Correspondence to: Dr. Lei Zhang, Key Laboratory of Spectral Imaging Technology, Xi'an Institute of Optics and Precision Mechanics of the Chinese Academy of Sciences, Xi'an 710119, Shaanxi, China. E-mail: zhanglei1@opt.ac.cn



Supplementary Figure 1. Two-stage hierarchical quality filtering pipeline for the GPUmatLLM training corpus.



Supplementary Figure 2. GPUmatLLM Fine-Tuning via Low-Rank Adaptation (LoRA) workflow.

Question What material is used to make the central part of the radiator in contact with the top surface of the packaging component?

Reference

In GPU chip packaging, a layer of interface material with high thermal conductivity is required for physical contact between the central part of the heat sink and the top surface of the packaging components below. This layer of material is commonly referred to as the thermal interface material, and its core function is to fill the gaps between the heat sink and the chip packaging surface caused by microscopic unevenness, in order to establish an efficient heat conduction path and effectively transfer the heat generated during chip operation to the heat sink. This contact method is crucial for ensuring the GPU's heat dissipation performance and long-term stable operation under high loads.

GPUmatLLM

In the heat dissipation system of high-performance computing chips such as GPUs, specific interface materials need to be used between the heat sink and the packaging components to achieve efficient heat conduction and mechanical fixation. Specifically, a layer of thermal interface material will be filled between the central part of the heat sink and the top surface of the packaging component. The main function of this layer of material is to fill the micro gaps between the contact surfaces of the two, thereby significantly reducing the contact thermal resistance and ensuring that heat can be efficiently transferred from the heat generating packaging components to the heat sink. This design is crucial for maintaining stable operating temperature of GPUs under high loads.

Llama-3.1-8B-Instruct

The central part of the heat sink is usually in contact with the top surface of the packaging component through materials such as Diamond or Silicon.

Supplementary Figure 3. Qualitative comparison of responses generated by GPUmatLLM and Llama-3.1-8B-Instruct.

Question On which side of the semiconductor device is the backside interconnect structure typically located?

Reference

The backside interconnect structure is typically located on the non active side of semiconductor devices. Specifically, the structure is arranged on the back side of the device layer, which is opposite to the active area or front side of the device, and is therefore sometimes referred to as a back interconnect structure. This layout allows for additional interconnection and power distribution functions to be implemented on the back of the device. The backside interconnect structure is usually used in conjunction with the front-end interconnect structure located on the front side of the device. The two are electrically connected through deep vias that penetrate the device layer, forming a complete interconnect system. This backside interconnect structure is often used as a super power rail or part of a backside power distribution network in modern high-performance semiconductor devices, especially chips such as GPUs that require high-density, high-efficiency power supply and signal transmission, to optimize electrical performance and integration density.

GPUmatLLM

In semiconductor packaging, especially in advanced packaging involving high-performance computing chips such as graphics processing units (GPUs), the backside interconnect structure is a critical electrical connection and signal transmission path located on the back of the semiconductor device. Specifically, the backside interconnect structure is typically located on the backside of the semiconductor device, which is the opposite side of the active surface. Its core function is to provide additional, independent electrical connection points for semiconductor devices to achieve more flexible wiring design, higher interconnect density, or specific power/ground distribution schemes. This structure is commonly used in scenarios where advanced packaging technologies such as fanout wafer level packaging are employed, which helps optimize the overall packaging size, improve signal integrity, and power efficiency. Therefore, in the packaging architecture of high-performance chips such as GPUs, the backside interconnect structure is an important component integrated on the back of the device.

Llama-3.1-8B-Instruct

Back side interconnect structures are typically located on the backside of semiconductor devices, meaning they are situated on the back of the device.

Supplementary Figure 4. Qualitative comparison of responses generated by GPUmatLLM and Llama-3.1-8B-Instruct.

Supplementary Table 1. Model versions, training configurations and inference settings used in this work

Item	Value
Base model	Qwen2.5-7B-Instruct, obtained from ModelScope (revision: master)
LoRA	$r = 8$, $\alpha = 16$, dropout = 0, target modules = q_proj, v_proj
Training	$lr\ 1 \times 10^{-4}$, cosine decay, batch $4 \times$ grad-accum 4 (effective 16), BF16, 1 epoch, seed 42
Hardware	$1 \times$ NVIDIA GeForce RTX 5090 (32 GB)
Frameworks	LLaMA-Factory 0.9.5.dev0, EvalScope 1.4.2, transformers 4.57.1
Decoding (all models)	greedy, temperature = 0, top_p = 1.0, max_new_tokens = 1024
Local models (BF16)	Qwen2.5-7B-Instruct, Llama-3.1-8B-Instruct, ChemDFM-v1.5-8B, ChemLLM-7B-Chat
API models	Qwen3-32B (Alibaba Cloud) and DeepSeek-V3.2 (SiliconFlow), accessed April 2026
BLEU-1 tokenisation	EvalScope default tokenisation
Prompt templates	Prompts 1-4

All templates were supplied to DeepSeek-V3.2 via the EasyDataset platform. Double-brace placeholders are filled at generation time: `{{text}}` is the source chunk, `{{question}}` the target question, `{{textLength}}` the chunk length, `{{number}}` the minimum number of items, and `{{gaPrompt}}` / `{{gaPromptNote}}` / `{{gaPromptCheck}}` / `{{templatePrompt}}` / `{{outputFormatPrompt}}` are optional instructions. Verbatim text follows; line breaks and JSON formatting are preserved.

Role: Question Generation Expert

Profile

- Description: You are a professional text-analysis and question-design expert. You can distill key information from complex text and produce high-quality multiple-choice

question sets for model fine-tuning, including single-answer and multiple-answer items.

- Input Length: `{{textLength}}` characters
- Output Goal: Generate no fewer than `{{number}}` high-quality multiple-choice questions to build a question-answering training dataset.

Skills

1. Fully comprehend the source text, identify core concepts, facts and logical structure, and design questions with a clear, unambiguous answer.
2. Judge the appropriate item type from the nature of each knowledge point: a "one-of-four" single-answer item or a "select-all-that-apply" multiple-answer item, and signal the distinction in the stem (e.g., "Which of the following is..." vs. "Which of the following are...").
3. Control difficulty and item type to ensure diversity and representativeness, covering principles, architecture, performance, applications and other dimensions.
4. Strictly follow the format specification so the output is directly machine-processable.
5. Convert academic phrasing into natural, colloquial user questions, and embed the correct and distractor options inline as "A. ... B. ..." without referencing the source's structure.

Workflow

1. Text parsing: Read the whole text; identify key entities, events, values, principles and application scenarios. Decide which information suits single-answer items (a single correct conclusion) and which suits multiple-answer items (several parallel elements, features or causes).
2. User-perspective simulation: Imagine which knowledge points a general user or engineer would be interested in and how they would phrase a multiple-choice question.
3. Integrated question-and-option design: Choose the best entry point based on information density and importance, and append the options directly after the stem in "A. ... B. ... C. ... D. ..." format. Ensure coverage of multiple dimensions and avoid any reference to the source structure (e.g., "Table 1", "Figure 1", "the text", "this paper", "this method", "this project", "this experiment"). `{{gaPromptNote}}`

4. Quality check (including a lexical audit): Verify each item to ensure that:
- The answer can be directly grounded in the source text; distractors are plausible but clearly inconsistent with the source.
 - Items do not repeat topics or angles.
 - Wording is accurate, unambiguous and idiomatic; option length and structure are kept balanced.
 - No meta-information vocabulary appears, including but not limited to: "paper", "this text", "this experiment", "author", "study", "project", "figure X", "table X", "reference", "section", "report".
 - No reference to source structure (figures, tables, sections, authors, references, project, bibliography, experiment, funding); each item must be understandable independently of the source context.
- {{gaPromptCheck}}

Constraints

1. Every question and its correct answer must be strictly grounded in the source text; no external information or hypothetical scenarios may be added.
2. Questions must cover different topics, levels or perspectives of the text, not a single fragment.
3. [Banned-vocabulary list] The following words or their variants are strictly prohibited in both stems and options: paper, this text, this experiment, author, study, project, reference, section, report, "in the figure", "in the table", "as follows", "this work". Any item containing them is treated as a generation failure.
4. Each question must be presented as a natural, fluent user query; the stem and its options form a single complete string, with options separated by line breaks.
5. Output no fewer than {{number}} multiple-choice questions in a consistent format.
6. Questions must be relevant to the GPU-materials domain and focus on general knowledge such as technical principles, architectural features, algorithmic mechanisms, performance metrics and application scenarios — like objective questions drawn from a general textbook or technical manual.

Output Format

- A valid JSON array containing string elements only.
- Use English double quotes for all fields.
- Each string element is one complete multiple-choice question with its stem and option list.
- Follow this structure strictly:
["question-and-options text 1", "question-and-options text 2", "..."]

Output Example

```
[  
"Which of the following convolutional structures may be used in a lightweight neural  
network designed for wafer-map defect classification to reduce the parameter count?  
\nA. Depthwise separable convolution\nB. Standard 3x3 convolution\nC. Deformable  
convolution\nD. Fully connected layer",  
"When a deformable convolutional network is applied to wafer-defect recognition, its main  
advantage lies in better handling which of the following situations?\nA. Large variation  
in defect shape and position\nB. Extremely limited computational resources\nC. All defects  
share an identical geometric pattern\nD. A need for very high floating-point precision"  
]
```

Text to Analyze

{{text}}

GA Instruction (Optional)

{{gaPrompt}}

Prompt 1. Prompt template used to generate the single- and multiple-answer multiple-choice questions (MCQs) for the GPU-materials corpus.

Role: Fine-tuning Dataset Generation Expert

Profile:

Description: You are a professional fine-tuning dataset generation expert, skilled at extracting

core knowledge from academic papers, technical reports, and other specialized texts, and transforming them into high-quality, independent, and domain-compliant multiple-choice question–answer pairs. The QA pairs you generate are used for large model fine-tuning in the field of GPU materials. Answers must strictly follow the format of an exam answer sheet: only output the correct option letter(s) (single-choice or multiple-choice), with no explanations, analysis, or redundant text.

Skills:

Deeply understand the provided content and extract key GPU-material-related concepts, principles, mechanisms, and technical details.

Accurately determine the correct answers based on the given multiple-choice questions and strictly follow formatting requirements.

Generate questions that are tightly aligned with core ideas in the text, while remaining independent and representative for evaluating domain knowledge.

Ensure extremely concise answer formatting: for single-choice questions, output only one uppercase letter; for multiple-choice questions, output a list of letters in order (e.g., “A, C, D” or “ACD”), with no additional text.

Ensure answers are fully grounded in the provided content but expressed independently in natural language, without any referencing or citation-like expressions.

Workflow:

Take a deep breath and work on this problem step-by-step.

Carefully read the provided reference content and identify key GPU-material-related concepts, principles, mechanisms, and technical details.

Based on the given multiple-choice question and its options, locate the corresponding supporting knowledge within the reference content.

Determine the question type: if single-choice, select one correct option letter; if multiple-choice, select all correct option letters and order them alphabetically.

Finally, output the answer strictly in the required format, ensuring no explanatory text, letters with spaces, or line breaks are included.

Reference Content:

----- Reference Content Start -----

{{text}}

----- Reference Content End -----

Question

{{question}}

Constraints:

The answer must be based on the provided content and strictly correspond to the correct options in the question.

The answer format must strictly follow: single-choice outputs a single uppercase letter (e.g., “A”); multiple-choice outputs multiple uppercase letters separated by a comma and a space (e.g., “A, B, D”).

The answer must not include any non-letter content, including but not limited to explanations, analysis, parentheses, periods, line breaks, or prefixes such as “The answer is”.

The answer must not include any non-knowledge elements such as researchers, experimental procedures, figure numbers, references, page numbers, or citation markers.

Answers should be written in an objective academic style; however, in this format requirement, the final output must consist only of letters.

Both questions and answers must focus on GPU materials, ensuring domain relevance and general applicability as standalone exam-style items.

The output must be extremely clean to facilitate model fine-tuning for direct answer generation in exam-card format.

Do not add any information not present in the original text or make subjective assumptions.

For multiple-choice questions, the correct option letters must be sorted in ascending alphabetical order (e.g., A, C, E).

Ensure that answers strictly match the given options, and all letters must be uppercase.

{{templatePrompt}}

{{outputFormatPrompt}}

Prompt 2. Prompt template used to generate single- and multiple-answer multiple-choice questions (MCQs) for the GPU-materials corpus.

Role: Zero-Hallucination Materials Data Filtering Engine

Profile

- Description: You are a strict filtering gateway. Your sole mission is to extract concrete numerical values, compositions, or conclusions in the field of materials science from a text and turn them into minimal questions. For any content related to literature, algorithms, social sciences, biology, medicine, astronomy, geography, or computer-software methods - however the text frames it - you must ignore it and return an empty array [].

Mandatory Interception Rules (highest priority; violating any one returns [])

1. Domain-whitelist interception:

- Only output questions about the following physical/chemical entities: material names, chemical formulas, alloy grades, ceramics, polymers, semiconductors, thermal interfaces, packaging substrates, thin films, coatings, crystal structures, and mechanical/thermal/electrical performance indicators.
- If any of the following keywords appears AND is the subject of the text (not a modifier), immediately return [] and generate no questions:
 - Algorithmic: machine learning, deep learning, neural network, model training, feature engineering, algorithm, classifier, regression, clustering
 - Computer: code, programming, software, framework, database, server, attack, vulnerability, encryption, network
 - Social science / humanities: economics, society, psychology, management, policy, education, history, literature, art

2. Mandatory numerical interception:

- The text must contain a "number + unit" (e.g., 2.4 eV) or "an explicit yes/no conclusion + a specific percentage change" (e.g., increased by 15%).
- If the text only describes methods, principles, trends, or significance but contains no concrete numerical value, immediately return []. Example: "the material ratio was optimized via machine learning" -> [].

3. Vocabulary-blacklist interception:

- The generated questions must never contain: this paper, this study, the text, the author, experiment, figure, table, method, project, reference, this, that, algorithm, model, machine learning.

Workflow (execute strictly, no skipping)

Step 1: Domain admission and subject-term detection

- Read the [Text to Analyze].

- Checklist:

1. Does it contain target-domain whitelist terms? (e.g., alloy, ceramic, dielectric constant, thermal conductivity, hardness, band gap)

2. Does it contain forbidden-domain blacklist terms? (e.g., algorithm, attack, network, literature)

- Decision logic:

- If it contains blacklist terms OR does not contain whitelist terms -> immediately output [] and terminate.

- If it passes -> proceed to Step 2.

Step 2: Numerical-evidence locking

- Search the text for "numerical value + physical unit" combinations.

- Examples: 450 HV, 2.4 eV, 15%, Ti-6Al-4V (treated as a grade; no unit required).

- If zero matches -> immediately output [] and terminate.

- If at least one match -> proceed to Step 3.

Step 3: Minimal question construction (mandatory de-referencing)

- Construct questions from the values found and their corresponding property names.

- Mandatory substitution: replace "this alloy" / "this sample" with the specific material name; if no specific name exists, use "the material" and ensure there is no referential ambiguity (check once more before final output - if a referential term remains, delete that question).

- Mandatory sentence patterns: only the following three structures may be used:

- What is the [property] of [material name]?

- What is the main composition of [material name]?

- Can [process/element] [increase/decrease] the [property] of [material name]?

Step 4: Secondary interception and output

- Check again whether the generated question strings contain any blacklist term (including algorithm, this paper, this method, etc.).
- If so -> discard that question.
- Place all questions that pass into a JSON array. If the final array is empty -> output [].

Constraints (absolute red lines)

1. A numerical value is the only pass: no concrete value (number + unit) -> [].
2. Zero tolerance for cross-disciplinary content: anything involving algorithm, machine learning, attack, or model (unless it refers to a materials constitutive model with concrete parameters - but intercept all of them to be safe) -> [].
3. Metadata-virus immunity: questions may not contain any term that references the source structure.
4. Absolute format consistency: a valid JSON array; [] when invalid.

Examples of Violations vs. Correct Output (key learning material)

| Input text fragment | Wrong model output (forbidden) | Correct output |

| :--- | :--- | :--- |

| "This paper uses machine learning to predict the band gap of perovskite." | "What is the accuracy of band-gap prediction by machine learning?" | [] (no concrete value) |

| "The band gap of perovskite is calculated to be 1.8 eV." | "What is the band gap of perovskite in the text?" | "What is the band gap of perovskite?" |

| "The corrosion resistance of this alloy improved by 15%." | "By how much did the corrosion resistance of this alloy improve?" | "By what percentage did the corrosion resistance of the alloy improve?" (simplified if the alloy is unnamed) |

| "The impact of network attacks on materials databases." | "What impact do network attacks have on materials databases?" | [] (entirely wrong domain) |

Output Format

- Must output a strictly valid JSON array.
- When there is no valid output, it must be: []

Output Example

["What is the thermal conductivity of an aluminum nitride substrate?", "What is the main strengthening element of Inconel 718?", "Can adding graphene reduce the thermal resistance of a thermal interface material?"]

Text to Analyze

{{text}}

Prompt 3. Prompt template used to generate QA questions from each source chunk for the GPU-materials corpus.

Role: Minimal Answer Extractor

Profile

- Description: You are an answer-extraction expert dedicated to building fine-tuning data for the GPU-materials domain. Your sole task is to extract, for a given question, an extremely short, definite, atomic fact from the provided academic text fragment that can serve directly as a gold answer. Your output never contains any superfluous words or explanation.
- Input Text Length: {{textLength}} characters
- Output Goal: Return an extremely short, deterministic answer (no more than 20 characters). If no basis can be found, return an empty string.

Skills

1. Precise targeting: rapidly locate in the academic text the specific value, composition term, or binary conclusion that directly corresponds to the question.
2. Information compression: condense complex academic descriptions into minimal text containing only the core information (value + unit, or core noun).
3. Strict de-referencing: the answer never contains referential phrases such as "according to the text" or "the study shows"; it must be fully self-contained semantically.

4. Null detection: accurately recognize when the text lacks the direct basis needed to answer, and decisively return an empty string in that case.

Workflow

1. Deep understanding: carefully read the [Reference Content] and [Question] below.
2. Rapid retrieval: search the [Reference Content] for the value, explicit noun, or affirmative/negative conclusion directly relevant to the question.
3. Null determination (highest priority): if the [Reference Content] contains no answer to the specific question at all, immediately stop and return an empty string "". Never infer or fabricate from common sense or context.
4. Answer extraction: if a definite basis is found, distill it to the minimal form.
 - Example input: "After heat treatment the hardness of the alloy increased markedly, reaching 450 HV."
 - For the question: "What is the hardness of the alloy after heat treatment?"
 - Extracted answer: 450 HV
5. Formatted output: output the extracted text directly. If Step 3 is triggered, output "".

Constraints

1. Hard character limit: the final answer body (excluding spaces and line breaks) must not exceed 20 characters.
2. Hard content constraint: the answer must come entirely from the [Reference Content], with no personal knowledge or speculation.
3. Hard form constraint: the answer must be plain text - a value, a noun phrase, or a short yes/no answer. No explanation, analysis, preamble, or closing remark.
 - Correct: 2.4 eV / Ti-6Al-4V / Yes, ~15% increase / hexagonal boron nitride / No
 - Wrong: "According to the data in the text, its band gap is 2.4 eV." / "The main composition of the alloy is Ti-6Al-4V."
4. Empty-return requirement: when the [Reference Content] cannot provide the specific value or conclusion required, the output must be strictly "".

Input Format

[Reference Content]:

----- Reference Content Start -----

{{text}}

----- Reference Content End -----

[Question]:

{{question}}

Output Format

Strictly follow the output format: output only the answer text or an empty string, with no other content, comments, or formatting markers.

Output Example (answer exists)

2.4 eV

Output Example (no answer)

""

{{templatePrompt}}

{{outputFormatPrompt}}

Prompt 4. Prompt template used to extract the gold answer for each question produced by Prompt 3.