UNIM: A Unified Any-to-Any Interleaved Multimodal Benchmark

Supplementary Material

A. Potential Limitations and Future Work

A.1. Potential Limitations

In this paper, we introduce the first large-scale unified any-to-any interleaved multimodal benchmark UNIM and propose a unified any-to-any interleaved multimodal agentic model UNIMA as baseline. Although this work represents a pioneering effort that significantly advances the development of interleaved multimodal learning, we acknowledge that it still has certain limitations.

Limited Modality Combinations and Types of UNIM. Although UNIM is the first unified any-to-any interleaved multimodal benchmark that encompasses 7 modalities: text, image, audio, video, document, code, and 3D, which represents an important expansion and pioneering step beyond previous interleaved benchmarks that primarily focused on image-text scenarios, we acknowledge certain constraints. First, human perception of the world extends beyond these 7 modalities. Second, the current benchmark includes only a restricted set of modality combinations, while real-world interleaved multimodal scenarios may involve far more complex and interleaved configurations.

Dependence of UNIMA's Performance on External Tools. The performance of UNIMA largely depends on the multimodal understanding and generation capabilities of the external tools it invokes. Although we employ state-of-the-art tools, their inherent limitations may still influence the overall performance of UNIMA.

System Complexity and Computational Overhead of UNIMA. To enable fine-grained comprehension, reasoning and generation, UNIMA invokes up to six external tools, introducing substantial complexity and computational overhead into the system. Compared with end-to-end models, this multi-step and sequential reasoning process leads to higher system complexity.

A.2. What To Do Next with UNIM and UNIMA

Building on this pioneering work, we believe that future research on ***interleaved multimodal learning*** can be further advanced along several key directions.

From Modular to End-to-End Interleaved Multimodal Foundation Models. An important future direction is the development of fully end-to-end foundation models. Current end-to-end MLLMs do not fully support flexible any-to-any multimodal interleaving, as they are unable to process arbitrary modality combinations or multiple items of the same modality. Therefore, an important future direction is to develop a unified encoder–decoder architecture supporting arbitrary combinations of seven modalities.

Improving Multi-Capability Coordination in Any-to-Any MLLMs. Although current any-to-any MLLMs have demonstrated certain effectiveness in several basic capabilities, their performance remains suboptimal in multi-capability coordination, particularly in complex tasks. Therefore, enhancing the model's performance to integrate and coordinate multiple capabilities emerges as a research direction worthy of in-depth exploration.

Cross-Modal Synergy and Complementarity. Future work may further investigate the synergistic and complementary interactions among modalities in interleaved multimodal scenarios. Instead of treating each modality as an independent source of information, models may benefit from explicitly modeling their dynamic interplay across tasks, contexts, and heterogeneous signal-quality conditions. Achieving stable and synergy-based multimodal integration is expected to clearly improve semantic consistency, generation quality, and generalization in complex interleaved multimodal settings.

Dynamic Reasoning Mechanisms and Contextual Adaptation. Future work may further investigate dynamic reasoning and contextual adaptability in interleaved multimodal settings. Rather than assuming fixed modality contributions, models should adjust the influence of modality features based on task context, focusing on the most informative modalities at different stages. This adaptive strategy can enhance robustness and enable more context-aware cross-modal reasoning.

Interleaved Rewarding. Future work may further explore an interleaved reward mechanism that explicitly incorporates both semantic accuracy and the handling of interleaved multimodal structures into the optimization objective, thereby improving consistency and robustness in complex modality composition scenarios.

Self-Verification and Iterative Refinement under Interleaved Settings. Future work may incorporate self-verification and iterative refinement mechanisms, enabling models to assess and locally revise generation outputs to better handle errors arising from one-shot processing in complex interleaved multimodal tasks.

Cognitive-Style Interleaved Modeling. Future research may further investigate cognitive-style interleaved modeling, where models emulate human multimodal reasoning strategies by identifying primary perceptual modalities, establishing reasoning order, and leveraging complementary modalities for validation or refinement. This cognitive-inspired approach is expected to enhance interpretability and robustness under complex interleaved multimodal scenarios.

B. Task Definition

The proposed task requires the model to perform comprehension and generation under the *any-to-any modalities interleaved* format. Specifically, the input is an interleaved sequence S_{in}, and models aim to generate a valid interleaved output sequence S_{out}. Both S_{in} and S_{out}

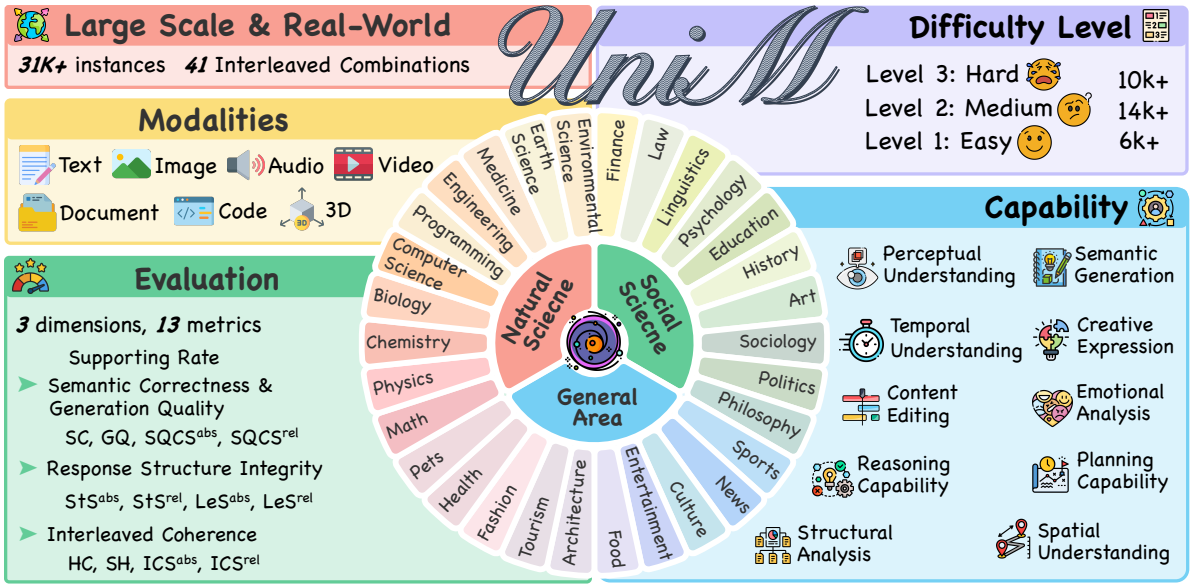


Figure 1. Overview of UNIM benchmark.

contain modality placeholder tags embedded in natural language, which denote non-text modalities.

Let $\mathcal{M}$ denote the set of modalities. The input sequence is defined as

$$S_{\text{in}} = [x_1, x_2, \ldots, x_{L_{\text{in}}}]. \tag{1}$$

Each sequence element x_j satisfies

$$x_j \in \mathcal{V} \cup \{\langle m_k^{\text{in}} \rangle \mid m_k \in \mathcal{M} \setminus \{\text{text}\}\}, \tag{2}$$

where $\mathcal{V}$ denotes natural language tokens, m_k represents a specific modality type, and $\langle m_k^{\text{in}} \rangle$ indicates a modality placeholder tag appearing in S_{in}.

For the model $\mathcal{F}$, the objective is to sequentially produce the output sequence S_{out} based on the given input sequence S_{in}. Each output token y_j is generated as

$$y_j \sim \mathcal{F}_\theta(\cdot \mid S_{\text{in}}, y_1, \ldots, y_{j-1}), \quad j = 1, 2, \ldots, L_{\text{out}}. \tag{3}$$

Similarly, the format of y_j should satisfy

$$y_j \in \mathcal{V} \cup \{\langle m_k^{\text{out}} \rangle \mid m_k \in \mathcal{M} \setminus \{\text{text}\}\}, \tag{4}$$

where $\mathcal{V}$ denotes natural language tokens, and $\langle m_k^{\text{out}} \rangle$ represents a placeholder tag of the modality that appears in the output sequence. Therefore, the output sequence can be expressed as

$$S_{\text{out}} = [y_1, y_2, \ldots, y_{L_{\text{out}}}], \tag{5}$$

To ensure format consistency and the feasibility of evaluation, we impose the following constraints on the placeholders in both the input and output sequences.

Constraint 1 (Directionality). Placeholders appearing in the input sequence should only occur in the input text, and those in the output sequence should only occur in the output text:

$$\begin{aligned} \forall\, m_k \in \mathcal{M} \setminus \{\text{text}\} : \quad & \langle m_k^{\text{in}} \rangle \in S_{\text{in}} \Rightarrow \langle m_k^{\text{in}} \rangle \notin S_{\text{out}}, \\ & \langle m_k^{\text{out}} \rangle \in S_{\text{out}} \Rightarrow \langle m_k^{\text{out}} \rangle \notin S_{\text{in}}. \end{aligned} \tag{6}$$

Constraint 2 (Uniqueness). Each placeholder appears exactly once in its corresponding sequence:

$$\begin{aligned} \forall\, \langle m_k^{\text{in}} \rangle \in S_{\text{in}}, \quad & |\langle m_k^{\text{in}} \rangle| = 1, \\ \forall\, \langle m_k^{\text{out}} \rangle \in S_{\text{out}}, \quad & |\langle m_k^{\text{out}} \rangle| = 1. \end{aligned} \tag{7}$$

C. Details of UNIM

In this section, we provide an exposition of UNIM, with Fig. 1 serving as a comprehensive overview.

C.1. Construction Process

In this section, we provide a detailed description of its construction process, quality control protocol, and statistical characteristics. Fig. 2 provides a high-level overview of data construction and annotation workflow.

C.1.1. Data Sources Collection

For multimodal data collection, we rely mainly on three sources. First, we curate samples from existing high-quality public datasets. Second, we obtain publicly accessible real-world multimodal content from famous social media. Third, we supplement the multimedia materials with open resources.

Public Datasets. We first obtain high-quality multimodal resources from existing public multimodal

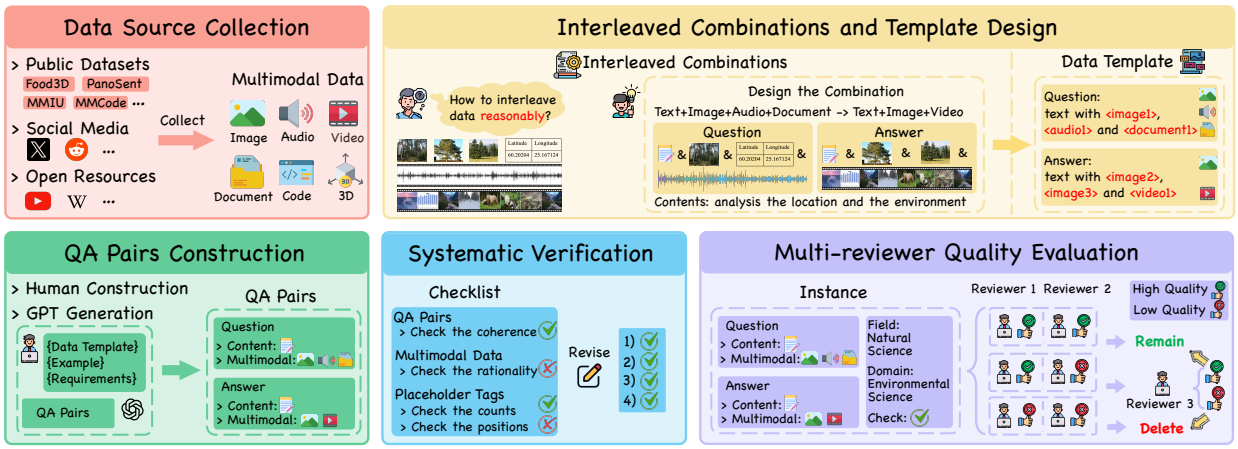


Figure 2. Overview of dataset construction.

Table 1. Public dataset sources.

Modality	Public Dataset
Image	FoodieQA [13], MuSLR [36], TaxaBench-8k [24], MSEarth [41], LAB-Bench [9], Chempile [18] OmniCorpus [12], OmniEarth-Bench [28], MMRB [3], StateControl-Bench [34] iMaterialist [7], DeepFashion [14]
Audio	Panosent [15], CMI-Bench [16], ESC [22]
Video	Video-MME [5], MuMa-ToM [25], MovieChat [26], Social-IQ [39], PhysLab [23], EgoTextVQA [42], HowTo100M [17]
Document	MedTrinity-25M [35], CaseSumm [8], FDABench [30], Synfintabs [1]
Code	MMCode [10], ScreenSpot-Pro [11], CodeEditorBench [6]
3D	MetaFood3D [2], General-Bench [4]

Table 2. Multimodal files formats.

Modality	File Format
Image	PNG, JPG, JPEG
Audio	WAV, MP3
Video	MP4, WEBM
Document	PDF, DOC, CSV, TXT, PNG, JPG
Code	Markdown Block
3D	JPG+MTL+OBJ, OFF, PLY

datasets and perform systematic filtering and structured organization. These datasets typically offer well-defined annotation schemes and stable data quality, providing reliable and diverse raw materials for constructing interleaved multimodal instances. By integrating various task types and modality forms, we ensure sufficient diversity and modality coverage in the foundational data. The specific data sources we use are listed in Table 1.

Social Media. Second, we harvest large quantities of real-world multimodal data from publicly accessible social media platforms (e.g., Twitter, Reddit), including posts and vlogs. Data on these platforms are highly dynamic, open-ended, and span various media types such as text, images, audio, and video, making them more representative of real-world scenarios. Under strict adherence to platform access and usage policies, we filter and preliminarily organize the collected multimodal data items, enabling them to serve as an important source for constructing interleaved multimodal instances.

Open Resources. Finally, we draw on a variety of highly accessible open resources to complement modality types that are relatively underrepresented in the previous sources, thereby strengthening the overall multimodal coverage. We acquire modality-specific materials from platforms that host particular types of data, for example, long-form videos from YouTube, document-hosting websites, code repositories, and libraries containing 3D models. After basic filtering and structured organization, these multimodal data items serve as an important supplement for constructing interleaved multimodal instances, substantially enhancing both modality richness and scenario diversity within the dataset.

Table 2 summarizes the data file formats associated with each modality in UNIM.

C.1.2. Interleaved Combinations and Template Design

After obtaining the multimodal files, we begin by manually analyzing the characteristics of each modality and designing appropriate **interleaved combinations** based on their semantic relatedness, information density, and interaction patterns. Specifically, while preserving the semantic information of each original modality, we reorganize images, audio, video, documents, code, and 3D into coherent interleaved structures that can jointly express information within a single question–answer pair. The positions, quantities, and interleaving orders of different modalities in both input and output are semanti-

cally driven, ensuring that the combinations are diverse yet aligned with realistic interleaved multimodal comprehension and generation requirements.

Following this principle, we construct **instance templates** in which multimodal elements are inserted into text using placeholder tags such as '<image1>', '<audio2>', '<code3>', and '<3d4>'. For the multimodal files collected in UNIM, we design a total of 41 interleaved combinations. The interleaved combinations are presented in Table 3.

C.1.3. QA Pairs Construction

In the QA-pair construction stage, we adopt a collaborative strategy of "human-led authoring with model-assisted expansion" to ensure both semantic quality and scalable data generation.

First, human annotators manually compose five high-quality representative QA pairs for each template designed in the previous stage, taking into account the characteristics of different task types. These manually created examples clarify task intent and provide guidance for subsequent expansion, ensuring coherent reasoning, self-consistent semantics, and natural cross-modal relationships.

Further, we leverage GPT-5 mini [20] to expand these human-designed examples. Annotators provide the model with the templates, exemplars, and explicit construction requirements, enabling it to generate additional candidate QA pairs under controlled structural constraints. The full prompt used is shown in Fig. 8.

C.2. Quality Control

To ensure the reliability and high quality of the dataset, we design and conduct a two-stage quality control procedure consisting of a systematic validation and multi-reviewer evaluation.

C.2.1. Systematic Verification

First, each constructed instance undergoes a systematic verification process conducted by human reviewers. Reviewers examine, item by item, the semantic coherence of the QA content, the correctness of the alignment between multimodal files and textual descriptions, and the accuracy of both the number and positions of placeholder tags. If inconsistencies or unreasonable elements are identified, reviewers are required to revise or rewrite the sample to ensure structural clarity, logical completeness, and correct modality representation. We develop an internal tool to support this reviewing and revising process, as illustrated in Fig. 15.

C.2.2. Multi-reviewer Quality Evaluation

After the initial revisions, all instances proceed to an independent multi-reviewer evaluation stage. Each instance is assigned to two reviewers who independently assess its quality, examining whether multimodal information is accurate, whether the task satisfies the template requirements, whether the reasoning is rigorous, and whether any ambiguities or missing content are present. Instances that both reviewers rate as high quality are retained directly, whereas those rated as low quality by both are removed. For instances with conflicting assessments, a third reviewer is introduced to provide an adjudication, ensuring the reliability and consistency of the final decision. We develop an internal tool to support this process, as illustrated in Fig. 16.

Level 3: Hard

Comprehension: Inputs contain highly complex modalities (e.g., 3D, code, video) or involve $\geq$5 intricate interleavings across modalities.
Generation: Outputs involve complex structured generation (e.g., 3D point cloud completion, video editing, code translation) or $\geq$4 interleavings across modalities.
Reasoning: Involves multi-step reasoning, deep logical analysis, or reliance on highly specialized domain knowledge.
Task: Task objectives are complex or open-ended, possibly comprising multiple interdependent subtasks.

Figure 3. Definition of hard level.

C.3. Progressive Difficulty Taxonomy

To systematically characterize the complexity of UNIM and quantitatively assess model performance on it, we establish a progressive difficulty taxonomy, which categorizes dataset instances into three hierarchical levels: Hard, Medium, and Easy.

Detailed Definitions of Different Difficulty Levels. We begin by manually defining the difficulty criteria across four core dimensions: comprehension, generation, reasoning, and task. These four dimensions jointly describe the overall difficulty of a multimodal instance. The detailed definitions and decision criteria for each difficulty level are provided in Fig. 3, Fig. 4 and Fig. 5.

Top-down Classification Process from Hard to Easy. We employ a top-down process for difficulty classification. Beginning from the hard level and proceeding downward, each instance is evaluated dimension by dimension across the four criteria: comprehension, generation, reasoning, and task. An instance is classified to a given difficulty level if it satisfies at least two of the four criteria for that level. This process ensures hierarchical consistency in the classification standards while enhancing the precision and reproducibility of the diffi-

Table 3. Interleaved combinations in UNIM. I: Image, V: Video, A: Audio, D: Document, C: Code.

Input	D	A+D	I	I+A	V+A	V+A	I+V+A	I+D	D	A+I	A	I+A	A	D
Output	I	A	I	I+A	I	A	V+A	A	D	A+V	I	D	A	A
Input	V	I	I+A	I+D	I	3D	A	V+I	A	I+A	V	A	V+A	V+A
Output	I	A	I	I	D	I+D	3D	V	D	A	A	V	V+A	D
Input	C	I	D	I+A+D	3D	I+V+A	I+A+D	D+A	I+A	D	I	A+V+D		
Output	C	C	C	C	3D	A	I	I	V	I+A	V+A	A		

Level 2: Medium

Comprehension: Inputs involve moderately complex modalities (e.g., documents, audio) or typically exhibit between three and four interleavings across modalities.
Generation: Outputs involve localized editing or limited structured generation (e.g., image editing, object detection) or typically contain between two and three interleavings across modalities.
Reasoning: Requires one to two steps of cross-modal or cross-instance reasoning, possibly involving commonsense judgment.
Task: Task objectives consist of multiple subtasks that are generally independent of one another.

Figure 4. Definition of medium level.

Level 1: Easy

Comprehension: Inputs include relatively simple modalities (e.g., text, image) or typically contain no more than two simple interleavings across modalities.
Generation: Outputs consist of straightforward, direct generation without any editing or structural modification.
Reasoning: Involves only single-step perception and alignment, without cross-modal reasoning or reliance on commonsense or external knowledge.
Task: Task objectives are singular, with short processes and no additional constraints.

Figure 5. Definition of easy level.

culty categorization.

C.4. Capabilities to Evaluate

To comprehensively evaluate model performance across different interleaved multimodal scenarios, we conduct a structured analysis of the capabilities required by the task types in UNIM, taking into account their semantic demands and cross-modal reasoning patterns. Based on this analysis, we derive ten core capability dimensions. The detailed definitions for each capability are provided in Table 4. For every instance, we further annotate a set of soft labels that indicate the competencies a model must possess in order to answer that instance effectively.

C.5. Task Types

To comprehensively characterize the range of tasks covered in UNIM, we organize the task types and categorize them by modality. Table 5 presents the major task types associated with each modality.

C.6. Detailed Statistics

We perform a detailed statistical analysis of UNIM. Table 6 reports the number of each modality files contained in the dataset. Then, we present the number of instances across the easy, medium and hard levels for each domain, as summarized in Table 7.

C.7. Data Example

We provide some representative data examples of UNIM as shown in Fig 17, Fig 18, Fig 19, Fig 20, Fig 21, Fig 22, Fig 23, Fig 24, Fig 25, Fig 26, Fig 27, Fig 28, Fig 29, Fig 30, Fig 31 and Fig 32.

D. Details of Evaluation Suite

D.1. Semantic Correctness & Generation Quality

To ensure cross-modal comparability and stable evaluation outcomes in assessing semantic correctness and generation quality, all experiments are conducted under a unified multimodal evaluation framework. This section provides the full technical details underlying the evaluation system described in the main text.

D.1.1. Semantic Correctness

Semantic Correctness measures the degree to which the model's generated content is semantically aligned with the ground truth. In the interleaved multimodal generation settings, a model may output text, images, audio, video, documents, code, or 3D data, but non-text modalities cannot be directly compared to the ground truth for semantic alignment. To address this limitation, we convert all modalities into textual representations prior to evaluation, enabling multimodal responses to be assessed within a unified semantic space using a consistent comparison protocol.

Table 4. Definitions of capabilities.

ID	Capability	Definition
C1	Perceptual Understanding	Identifying and interpreting content, entities, attributes, and semantic relations from multimodal inputs.
C2	Spatial Understanding	Comprehending spatial relationships, geometric structures, 3D scene characteristics, and physical layouts.
C3	Temporal Understanding	Understanding event or action sequences, temporal dynamics, and time-dependent dependencies.
C4	Semantic Generation	Generating multimodal content that is semantically coherent, contextually appropriate, and stylistically consistent.
C5	Content Editing	Modifying, refining, or transforming existing content, including stylistic adjustments and targeted edits.
C6	Creative Expression	Producing content with creativity, expressiveness, and aesthetic value.
C7	Reasoning Capability	Performing mathematical, spatial, logical, and causal reasoning under multimodal conditions.
C8	Emotional Analysis	Recognizing and interpreting emotions, intentions, mental states, and affective cues.
C9	Structural Analysis	Understanding structured or symbolically encoded information such as documents, tables, and code.
C10	Planning Capability	Decomposing tasks and making dynamic adjustments to ultimately achieve the target in complex scenarios.

Table 5. Definitions of task types.

Modality	Task Type
Text	Text summarization; Text generation; Machine translation; Sentiment analysis; Stance detection; Named entity recognition; Text style transfer; ...
Image	Image captioning; Image generation; Semantic segmentation; Image editing; Visual localization; Visual question answering; Image style transfer; ...
Audio	Audio captioning; Audio generation; Audio timbre analysis; Audio sentiment analysis; Audio content analysis; ...
Video	Video captioning; Video localization; Visual question answering; Video editing; Text-to-video generation; Audio-to-video generation; Viewpoint-transformation video synthesis; ...
Document	Document OCR; Document generation; Document description; Document summarization; Document question answering; ...
Code	Code generation; Code testing; Code debugging; Code translation; Code understanding; Code question answering; Algorithm implementation; ...
3D	3D content understanding; 3D question answering; 3D point cloud completion; 3D object recognition; 3D generation; ...

Table 6. The number of multimodal files in UNIM.

Image	Audio	Video	Document	Code	3D
68,651	54,329	2,559	4,728	1,207	820

We convert non-text modalities into textual captions. This procedure converts the entire multimodal output into a unified textual representation. The ground truth undergoes the same process during data preparation, ensuring that model response and ground truth are compared within an aligned semantic space. The captioning tools used for each modality are summarized in Table 8.

After completing the textual alignment, we assess semantic correctness using an LLM-as-a-Judge scoring paradigm. The evaluation model takes as input the responses with captions and the ground truth, and outputs a score drawn from a predefined five-level semantic consistency criteria, with possible values $\{1, 2, 3, 4, 5\}$. The scoring process focuses exclusively on semantic equivalence, disregarding style, tone, coherence, or surface form; variations in phrasing, ordering, or mild numerical rounding are permitted as long as the underlying meaning is preserved. Core factual errors, critical omissions, or contradictions result in lower scores. For each instance, the resulting discrete scores are further mapped onto the continuous interval 0-1 for normalization. The scoring criteria are provided in Table 9. The full prompt

Table 7. Detailed statistics of UNIM.

Natural Science					Social Science					General Area				
ID	**Name**	**Easy**	**Medium**	**Hard**	**ID**	**Name**	**Easy**	**Medium**	**Hard**	**ID**	**Name**	**Easy**	**Medium**	**Hard**
#1	Math	806	777	48	**#11**	Finance	739	1694	135	**#21**	Sports	57	313	486
#2	Physics	40	591	374	**#12**	Law	256	634	121	**#22**	News	200	456	350
#3	Chemistry	312	617	113	**#13**	Linguistics	570	207	53	**#23**	Culture	128	170	100
#4	Biology	545	333	263	**#14**	Psychology	398	300	397	**#24**	Entertainment	301	188	299
#5	Compute Science	72	868	218	**#15**	Education	165	463	72	**#25**	Food	256	624	296
#6	Programming	15	64	743	**#16**	History	402	196	225	**#26**	Architecture	557	601	59
#7	Engineering	299	310	549	**#17**	Art	747	1492	279	**#27**	Tourism	363	617	105
#8	Medicine	666	462	77	**#18**	Sociology	65	879	182	**#28**	Fashion	700	160	85
#9	Earth Science	212	155	273	**#19**	Politics	204	153	56	**#29**	Health	662	348	215
#10	Env. Science	562	186	74	**#20**	Philosophy	299	191	0	**#30**	Pets	80	39	13

Table 8. Caption methods for each modality.

Modality	Caption Tool
Text	Original format retained
Image	GPT-5 mini
Video	Qwen3-Omni
Audio	Qwen3-Omni
Document	GPT-5 mini
Code	Original format retained
3D	PointLLM

is shown in Fig. 9.

D.1.2. Generation Quality

Generation quality measures the clarity, stability, usability, and overall perceptual performance of the model's outputs across different modalities. In the interleaved multimodal generation tasks, semantic correctness captures only whether the content produced by the model aligns with the intended meaning, but it does not reflect the visual, auditory, structural, or readability quality of the outputs. Practical applications typically rely on both content reliability and perceptual quality; thus, generation quality must be evaluated as a dimension independent from semantic correctness.

Because different modalities exhibit substantial differences in signal form, structural properties, and degradation patterns, it is difficult to apply a single unified metric. To address this, we design modality-specific quality assessment methods for seven distinct modalities and map all resulting scores onto 0-1 range, enabling cross-modal comparison.

Text. For the text modality, generation quality primarily reflects the completeness of the content, clarity of structure, fluency of language, and consistency of the output language. We adopt an LLM-as-a-Judge approach, where a GPT-5 mini evaluator constrained by a fixed prompt assesses the overall quality of the generated text. The evaluation follows a five-level quality criteria and is supported by a small set of examples to ensure consistency and comparability. The full criteria definition and prompt design are provided in Table 10 and Fig. 10.

Image. For the image modality, we employ the Natural Image Quality Evaluator (NIQE) as a no-reference quality metric to measure the deviation of generated images from natural scene statistics(NSS):

$$\mathrm{NIQE}(I) = \sqrt{(\mu_t - \mu_n)^\top \left(\frac{\Sigma_t + \Sigma_n}{2}\right)^{-1} (\mu_t - \mu_n)}. \tag{8}$$

Here, μ_t and Σ_t denote the mean vector and covariance matrix of the evaluated image in the NSS feature space, while μ_n and Σ_n represent the reference statistics estimated from a collection of high-quality natural images. The metric computes the Mahalanobis distance between these two feature distributions, reflecting the naturalness and degree of distortion of the image. Lower scores indicate that the generated image is closer to the distribution of natural images and thus exhibits higher perceptual quality. In implementation, each image is partitioned into local patches from which we compute luminance normalization coefficients, local variances, and other NSS features. The resulting statistics are then compared with the reference distribution to estimate their deviations.

Audio. The audio modality is evaluated using a fully statistics-driven, non-learning pipeline. Given an input waveform $x(t)$ resampled to 48 kHz, we first apply mono conversion, mean removal, and energy-based trimming. The magnitude spectrogram is computed as

$$S(f,t) = |\mathrm{STFT}(x)|. \tag{9}$$

The corresponding power spectrum is

$$P(f,t) = S(f,t)^2. \tag{10}$$

A set of robust signal-to-noise ratio estimates is computed, including the original energy ratio, subband stability, and a harmonic-percussive separation based metric. The effective SNR is defined as

$$\mathrm{SNR}_{\mathrm{eff}} = \max\big(\mathrm{SNR}_{\mathrm{orig}}, \mathrm{SNR}_{\mathrm{sf}}, \mathrm{SNR}_{\mathrm{hpss}}\big). \tag{11}$$

Table 9. Five-level rating criteria for *Semantic Correctness*.

Grade	Semantic Correctness Description
5	a) Completely equivalent; b) All key facts correct/covered; c) No contradictions; d) Units, ranges, and relational constraints remain consistent (paraphrasing, reordering, and minor rounding are allowable).
4	a) Almost equivalent; b) Most of key facts correct/covered; c) No major contradiction; d) Only minor omissions/ambiguity that do not affect the main conclusion.
3	a) Partially correct; b) Roughly half key facts correct; c) No major contradiction but noticeable; d) Notable omissions or minor misinterpretations, but the main conclusion is not fully overturned.
2	a) Low correctness; b) Less than half covered; c) Important errors/contradictions/confusions (numbers/entities); d) The core conclusion drifts, but still loosely on topic.
1	a) Almost incorrect; b) Nearly irrelevant; c) Mostly contradictory, or hallucinated; d) Mostly wrong/missing, or non-answers.

Table 10. Five-level rating criteria for text in *Generation Quality*.

Grade	Text Quality Description
5	a) Content is rich, self-consistent, and detailed; no major omissions or vague generalities; stands alone as a coherent text; b) Structure is clear and well-organized; transitions are smooth (e.g., "firstly…then…therefore…"); reasoning shows causal or hierarchical logic; c) Language is natural and grammatically flawless; diverse sentence structures; no syntactic errors; d) Entirely in one language; any foreign words appear only as necessary terminology.
4	a) Content is generally complete with sufficient detail but slightly shallow or missing minor points; b) Structure is good, with only mild jumps or awkward transitions that don't affect comprehension; c) Language is fluent with few minor grammatical or collocation issues; d) Mostly consistent language, with rare short foreign terms that do not disrupt flow.
3	a) Content covers the main idea but lacks depth or specific details; b) Organization somewhat weak; order or topic shifts slightly; meaning still clear overall; c) Several grammatical or spelling mistakes; simple or repetitive sentence patterns; d) Minor language switching between sentences or paragraphs, noticeable but not confusing.
2	a) Content is shallow, missing key details or explanations; very low information density; b) Poor structure; sentences disjointed; reader must infer connections; c) Frequent grammar errors; awkward or broken phrasing; readability low; d) Frequent in-sentence language mixing that affects readability.
1	a) Content is empty or meaningless; repetitive or irrelevant phrases; conveys no clear information; b) No logical order; severe contradictions; text barely comprehensible; c) Major grammatical breakdowns; unnatural or non-human syntax; d) Chaotic multilingual mixing (e.g., Chinese + English + Spanish, random spelling noise)

It is mapped through a logistic function,

$$q_{\mathrm{snr}} = \left[1 + \exp\{-k(\mathrm{SNR}_{\mathrm{eff}} - x_0)\}\right]^{-1}. \tag{12}$$

Global spectral flatness is quantified as

$$\mathrm{SF}_g = \mathrm{median}_t \left(\frac{\exp\left(\frac{1}{F}\sum_f \ln(S(f,t)+\epsilon)\right)}{\frac{1}{F}\sum_f (S(f,t)+\epsilon)} \right), \tag{13}$$

and the corresponding structural score is

$$q_{\text{struct}} = 1 - \text{SF}_g. \tag{14}$$

Dynamic range is extracted via percentile RMS and mapped to q_{dr}, while loudness is measured using integrated LUFS and normalized to q_{lufs}. Level stability, transient smoothness, and the crest factor are also included. The crest factor is computed as

$$\text{crest} = 20 \log_{10} \frac{\max_t |x(t)|}{\sqrt{\mathbb{E}[x(t)^2]}}. \tag{15}$$

Its normalized score is

$$q_{\text{crest}} = \text{band\_score}(\text{crest}). \tag{16}$$

Bandwidth quality is derived from the 95% energy frequency f_{95} and mapped to q_{bw}. Chroma stability and spectral contrast yield q_{chroma} and q_{contrast}.

High-frequency hiss is penalized through the regression slope of the log-power spectrum in the 4–12 kHz band, yielding p_{hiss}. Excessive high-frequency energy produces a penalty p_{hf}. Mid-section dropout or extended silence is handled by an adaptive coverage penalty p_{gap}, and low-contrast noise textures result in a penalty p_{lowC}.

Calibrated compensation factors are introduced to account for audio with pronounced structural or periodic components. Structural cues contribute

$$b_{\text{str}} = g_{\text{struct}}\big(q_{\text{contrast}},\, q_{\text{chroma}}\big), \tag{17}$$

while periodicity contributes

$$b_{\text{per}} = g_{\text{periodic}}\big(q_{\text{periodic}}\big). \tag{18}$$

The set of core quality terms $\{q_i\}_{i=1}^{N}$ is consolidated through a geometric-mean aggregation,

$$q_{\text{base}} = \exp\left(\frac{1}{N}\sum_{i=1}^{N} \ln q_i\right). \tag{19}$$

The final audio quality index is defined as

$$q_{\text{audio}} = q_{\text{base}}\, p_{\text{clip}}\, p_{\text{gap}}\, p_{\text{noise}}\, p_{\text{hiss}}\, p_{\text{hf}}\, p_{\text{lowC}}\, b_{\text{str}}\, b_{\text{per}}, \tag{20}$$

where $q_{\text{audio}} \in [0, 1]$.

Video. For the video modality, we adopt DOVER [32] to assess the generation quality of videos. Higher DOVER scores indicate better perceptual video quality.

Code. In the code modality, generation quality primarily reflects engineering quality rather than perceptual aesthetics. Accordingly, we adopt an LLM-as-a-Judge approach to perform structured quality assessment for each generated code segment. Specifically, the evaluator is configured as a language-agnostic and rigorous code reviewer, providing an overall quality judgment based on six dimensions: correctness, readability, design soundness, runtime efficiency, security, and testability. To enhance scoring consistency and interpretability, the prompt includes a five-level global quality criteria ranging from severe defects to engineering-grade excellence, along with representative few-shot examples covering different quality levels. These elements guide the LLM to make unified quality assessments across programming languages and coding styles. The complete prompt and criteria are provided in Table 11 and Fig. 11.

Document. In the document modality, the focus is not on image clarity or semantic correctness, but rather on the overall quality of tabular representation. Specifically, when a table presented as an image is recognized and transcribed into structured text, we assess whether the title and column headers are clear, whether the header and data rows are properly aligned, whether units and numerical values are consistent, and whether the table is sufficiently coherent to be interpreted and used independently.Therefore, we first apply OCR to transcribe the document image into raw text, then organize it into a Markdown-style table using simple heuristic rules. Additional formatting cues, such as consistency of decimal places and presence of measurement units, are extracted and provided to the evaluator as auxiliary hints. Building on this representation, we adopt an LLM-as-a-Judge paradigm that categorizes document quality into five levels, each specified by a corresponding set of criteria and supported by few-shot examples. This enables the model to make stable and consistent judgments regarding label clarity, structural organization, internal consistency, and overall self-containedness. The evaluation prompt and criteria are provided in Table 12 and Fig. 12.

3D. For the 3D modality, we employ the no-reference quality metric NR3D-Q to assess the generated 3D objects. This metric provides a unified score by jointly evaluating topology and completeness (Topology, T), geometric fidelity (Geometry, G), and sampling uniformity (Sampling Uniformity, S) for point clouds or mesh structures, with all results average and normalized to a common scale. The topology and completeness score (T) evaluates the global structural plausibility and integrity of a 3D object. For mesh representations, T reflects properties such as closure, watertightness, the proportion of non-manifold edges, self-intersections, and the number of connected components. For point clouds, T approximates the object's "surface-likeness" and structural continuity through intrinsic dimensionality estimation, local connectivity statistics, and normal-vector consistency. The geometry quality score (G) focuses on whether the local geometric structures are smooth, stable, and free from noticeable noise. It incorporates curvature statistics derived from local

Table 11. Five-level rating criteria for code in *Generation Quality*.

Grade	Code Quality Description
5	a) High-quality, professional-grade code with virtually no noticeable defects; b) Logically rigorous, structurally well-organized, and highly robust; c) Fully adheres to modern software engineering best practices, easy to maintain, and directly reusable.
4	a) High overall quality with a reasonable structure and clear logic; b) Contains only minor issues that do not affect usability; c) Aligns with most software engineering best practices and is easy to maintain.
3	a) Basically usable, with acceptable overall correctness but clear deficiencies; b) Contains multiple areas for improvement that may affect maintainability or reliability; c) The overall quality is functional but remains below desirable standards.
2	a) Contains multiple evident issues, with overall quality falling below acceptable standards; b) Barely runnable or only partially functional, requiring substantial repairs; c) Exhibits major deficiencies in logic, structure, performance, or security.
1	a) Contains numerous critical defects, resulting in extremely low overall quality; b) Exhibits chaotic logic, unclear structure, and significant risks; c) Largely non-reusable and unmaintainable, requiring a complete rewrite.

PCA, multi-scale normal stability, and neighborhood-level normal consistency. The sampling uniformity score (S) measures whether points or surface elements are evenly distributed in space, penalizing large sparse regions or excessively dense clusters. It is computed using statistics such as the distribution of kNN distances, the proportion of outliers, and the variation in mesh face areas or vertex valence.

Based on the above calculations, GQ is defined as the average of the scores assigned to all content items in the output of the instance, including both the text content and all multimodal content files.

D.1.3. Semantic-Quality Coupled Score

Evaluating SC and GQ in isolation often fails to fully capture a model's practical utility. To address this limitation, we introduce the Semantic-Quality Coupled Score (SQCS), which is computed for each instance using the normalized SC and GQ values:

$$\text{SQCS} = \text{SC} \cdot \left(\eta^{\text{SQCS}} + (1 - \eta^{\text{SQCS}}) \cdot \text{GQ}\right) . \quad (21)$$

The metric treats semantic correctness as the primary factor and incorporates generation quality as a modulation term that adjusts the score only when semantic conditions are satisfied. Specifically, when SC is low, the overall SQCS remains strongly suppressed even if GQ is high. Conversely, when SC is high, improvements in GQ drive the SQCS closer to the upper bound of SC, enabling finer discrimination of generation quality among semantically correct samples.

D.2. Response Structure Integrity

In this section, we provide a detailed elaboration of the two metrics under the *Response Structure Integrity* evaluation dimension, Strict Structure Score (StS) and Lenient Structure Score (LeS), including their formal definitions and computational procedures.

D.2.1. Strict Structure Score

The StS is designed to evaluate the strict structural consistency of a model's output. This metric requires that the types and quantities of modalities generated in the model's response precisely correspond to those in the ground truth. Any missing or redundant modalities, or discrepancies in the number of modality placeholder tags, are explicitly penalized.

Specifically, we define the modality set involved in score computation as

$$\mathcal{M}^{'} = \{m \mid g_m > 0 \vee r_m > 0\}, \quad (22)$$

where g_m denotes the number of placeholder tags of modality m in the ground truth, and r_m denotes the number of placeholder tokens of modality m in the model response. The set $\mathcal{M}^{'}$ thus contains all modalities that appear at least once in either the ground truth or the model response. If a modality does not appear in either $g_m = r_m = 0$, it is considered irrelevant and excluded from evaluation process. For each modality, we define the number of matches $n_m = min(g_m, r_m)$. The precision P_m and recall R_m for modality placeholder counts as

$$P_m = \begin{cases} \frac{n_m}{r_m}, & r_m > 0, \\ 0, & r_m = 0, \end{cases} \qquad R_m = \begin{cases} \frac{n_m}{g_m}, & g_m > 0, \\ 0, & g_m = 0. \end{cases} \quad (23)$$

The F1 score for each modality is calculated as

$$F1_m = \frac{2P_m R_m}{P_m + R_m}. \quad (24)$$

Table 12. Five-level rating criteria for document in *Generation Quality*.

Grade	Document Quality Description
5	a) All titles, labels, and column names are clear and unambiguous; b) The table structure is complete, with well-organized logical partitions and clear hierarchy; c) Units, naming conventions, capitalization, and punctuation are fully consistent; d) Numerical presentation is standardized, including uniform decimal places, units, and precision, with proper alignment; e) The surrounding context provides sufficient information for the table to be understood independently without additional explanation.
4	a) Most titles and column names are accurate, with only minor ambiguities; b) The table structure is generally reasonable, though some misalignment or slightly unclear grouping is present; c) Overall consistency is good, with only small variations in decimal places or units; d) Data presentation is mostly standardized, with mild irregularities that do not hinder usability; e) The table can largely be understood independently, though some contextual information is slightly lacking.
3	a) Contains noticeably ambiguous titles or labels whose meanings require contextual inference; b) Table structure is relatively disorganized, with unclear grouping, poorly defined relationships between columns; c) Multiple inconsistencies in formatting, naming, or units, reducing overall usability; d) Numerical presentation is irregular, with large variations in decimal places and missing or mixed units; e) Lacks sufficient contextual explanation, making the table difficult to interpret independently.
2	a) Titles or column names are missing or severely ambiguous, with clear potential for misinterpretation; b) The table structure is incomplete, exhibiting column misalignment, unclear row logic, or missing grouping; c) Formatting is highly inconsistent, with disordered or entirely missing units; d) Numerical presentation shows prominent issues, including irregular precision, inconsistent units, and potentially misleading values; e) The table is largely uninterpretable on its own and requires substantial supplementary explanation.
1	a) Titles or column names are extensively incorrect, missing, or unrecognizable; b) The table structure is entirely broken, with row–column relationships indistinguishable; c) No formatting consistency is preserved, with content unordered and misaligned; d) Numerical values are meaningless due to severe unit confusion, row mismatches, or erroneous combinations; e) The table is completely unusable on its own and requires re-OCR or full reconstruction from the original source.

Finally, the StS is defined as the average of $F1_m$, as shown in the following equation.

$$\text{StS} = \frac{1}{|\mathcal{M}'|} \sum_{m \in \mathcal{M}'} F1_m. \tag{25}$$

D.2.2. Lenient Structure Score

The LeS focuses on evaluating the degree of coverage at the modality level. This metric assesses whether the types of modalities generated in the response are consistent with those in the ground truth.

Let g_t denote the set of modality types appearing in the ground truth, and r_t denote the set of modality types appearing in the model response. We then define the modality overlap set as

$$\text{Overlap} = g_t \cap r_t. \tag{26}$$

The LeS is defined as the ratio of the number of overlapping modality types to the total number of modality types in the ground truth, as shown in the following equation.

$$\text{LeS} = \frac{|\text{Overlap}|}{|g_t|}. \tag{27}$$

D.3. Interleaved Coherence

Within the *Interleaved Coherence* dimension, we further refine it into two complementary perspectives: Holistic Coherence and Stylistic Harmony. The former reflects

the model's ability to maintain semantic consistency and logical completeness across modalities, while the latter captures its capacity to preserve a unified narrative tone and style during cross-modal generation, serving as a key indicator of the naturalness of the generated content.

D.3.1. Holistic Coherence

Holistic Coherence aims to assess whether a model can maintain global consistency in semantic logic, narrative structure, and modality transitions during multimodal interleaved generation. This perspective focuses on evaluating whether the model truly understands and integrates information across different modalities, rather than merely performing superficial modality stacking.

If the model's response demonstrates specific and accurate cross-modal references, with complementary information across modalities, natural logical flow, and well-organized interleaved labels that together produce a clear and coherent narrative, it should be rated high. Conversely, if there are conflicts or disconnections between modalities, such as evident logical leaps, semantic contradictions, or disordered modality sequencing that hinder overall comprehension—the response should be rated low. We define five-level rating criteria, with detailed definitions for each level shown in Table 13.

D.3.2. Stylistic Harmony

Stylistic Harmony is used to assess whether a model can maintain uniformity in register, tone, terminology, and visual style when generating interleaved multimodal content. In multimodal narratives, consistent style enhances the naturalness and credibility of the generated output: for instance, if the textual style is formal but the visual content appears cartoonish, or if terminology and naming are inconsistent across modalities, the overall reading experience will degrade significantly.

If the model's response demonstrates high coordination between linguistic and visual styles, with consistent tone, sentence structure, and rhetoric; and if key concepts, naming conventions, and modality tags are used uniformly, resulting in a smooth and natural overall reading experience, it should be rated high. Conversely, if the response exhibits chaotic register, inconsistent terminology, or obvious stylistic conflicts that make the content difficult to understand, it should be rated low. We define five-level rating criteria, with detailed definitions for each level shown in Table 14.

D.3.3. Interleaved Coherence Score

To provide a unified and reliable metric for evaluating model performance in interleaved multimodal generation, we introduce the Interleaved Coherence Score (ICS). Existing metrics often focus on semantic correctness or isolated stylistic quality but fail to capture how well a model maintains coherence when multiple modalities appear in an interleaved sequence. ICS addresses this gap by jointly assessing the semantic and structural alignment across modalities and the stylistic consistency of the generated output. This unified metric enables fair comparison across diverse multimodal configurations and highlights failure cases that conventional single-modality metrics cannot reveal.

ICS is defined as a weighted combination of SH and HC, given by

$$\mathrm{ICS} = \eta^{\mathrm{ICS}} \cdot \mathrm{HC} + (1 - \eta^{\mathrm{ICS}}) \cdot \mathrm{SH}\,. \tag{28}$$

To ensure the reproducibility of our evaluation process, we present the complete evaluation prompt used in the *Interleaved Coherence* dimension. This prompt guides the evaluation model to assign scores to generated content based on predefined criteria across two dimensions: Holistic Coherence and Stylistic Harmony. To enhance the stability of model scoring, we incorporate a small number of few-shot guiding examples, which demonstrate sample outputs at different score levels along with their corresponding explanations.

Before conducting the actual evaluation, we apply a unified text-based conversion to all non-text modalities. Specifically, we use caption tools to automatically caption the original image, audio, video, and other non-text inputs, and replace placeholder tags in the prompt with their transcribed textual representations. This ensures that the LLM performs scoring solely based on a standardized, text-only format. This preprocessing step guarantees comparability across different modalities and prevents biases arising from inconsistent perceptual capabilities of the evaluation model.

Finally, to standardize the evaluation scale across experiments, we linearly normalize all ICS from the original 1–5 scale to a 0–1 range. The caption tools used for each modality are listed in Table 15. The full prompt is shown in Fig. 13.

D.4. Supporting Rate

Since no current existing MLLM is perfect enough to support interleaved multimodal learning across the whole 7 modalities (text, image, audio, video, document, code, and 3D), there should be a common case in UNIM that baseline MLLMs are likely to be unable to accept certain modality types as input. To ensure fairness in evaluation, we emphasize full-modality visibility when conducting the assessment procedure. Specifically, a model must be able to receive and handle all modality information contained in an instance's input, ensuring that its outputs are based on a complete understanding of the query rather than being affected by missing inputs or modality incompatibilities.

Therefore, we treat a model's ability to support all input modalities of an instance as a prerequisite for eval-

Table 13. Five-level rating criteria for *Holistic Coherence*.

Grade	Holistic Coherence Description
5	a) Output highly matches the input, multimodal references accurate and specific; b) Different modality information complements each other without noticeable contradictions; c) Logic is rigorous, structure is reasonable, overall semantic/logical order is clear; d) Interleaved multimodal tags naturally ordered, reading/understanding experience smooth.
4	a) Output mostly matches the input, references generally reasonable; b) Different modality information mostly complementary, minor omissions or redundancy; c) Overall logic coherent, local repetitions or jumps minor, not affecting understanding; d) Interleaved tags mostly ordered, minor adjustments do not affect understanding.
3	a) Output has general relation to input, references are vague or partially unclear; b) Some modality blocks repeated or missing, minor contradictions exist; c) Local logic jumps or slight contradictions, requires extra reasoning; d) Interleaved tags partially disordered, local understanding may be difficult.
2	a) Output lowly matches the input, most references vague or incorrect; b) Cross-modal information repetitive or conflicting; c) Logic obviously chaotic, many contradictions; d) Interleaved tags order clearly disordered, understanding difficult.
1	a) Output is almost irrelevant to input, references missing or wrong; b) Different modalities barely complement, major contradictions; c) Logic completely collapsed, frequent contradictions; d) Interleaved tags order completely disordered, hard to understand.

Table 14. Five-level rating criteria for *Stylistic Harmony*.

Grade	Stylistic Harmony Description
5	a) Style highly consistent, cross-modal narration uniform, tone and sentence structures without deviation; b) Terminology fully consistent, key concepts, tags, and naming completely aligned; c) Expression and visual style fully aligned, wording, sentence structures, and rhetorical/visual style coordinated, overall experience smooth.
4	a) Style mostly consistent, minor deviations; b) Terminology generally consistent, key concepts, tags, and naming mostly aligned; c) Expression and visual style mostly aligned, minor deviations, overall experience unaffected.
3	a) Style partially consistent, noticeable differences in narration; b) Terminology partially mixed, key concepts, tags, naming sometimes inconsistent; c) Expression and visual style partially aligned, wording, sentence structures, or rhetorical/visual style obviously deviate, reading/watching experience affected.
2	a) Style inconsistent, cross-modal narration uncoordinated; b) Terminology not uniform, key concepts, tags, naming frequently mixed; c) Expression and visual style not aligned, wording, sentence structures, or rhetorical/visual style conflicting, understanding affected.
1	a) Style completely inconsistent, cross-modal narration chaotic; b) Terminology completely wrong, key concepts, tags, naming incorrect or inconsistent; c) Expression and visual style completely chaotic, wording, sentence structures, or rhetorical/visual style extremely uncoordinated, almost impossible to understand.

uation: an instance is considered supported only if the model can effectively receive and process every modality present in its input. We then define the Supporting Rate (τ) of a model as the proportion of supported instances relative to the entire UNIM dataset. This metric reflects the model's breadth of adaptability and coverage across the multimodal input space.

Building on the supporting rate as a conditional modifier, we further distinguish two types of metrics in our evaluation suite: $\mathcal{X}^{abs}$ and $\mathcal{X}^{rel}$. $\mathcal{X}^{abs}$ denotes the average performance computed only over the subset of instances that a model can support, that is, those for which

Table 15. Caption tools for each modality.

Modality	Caption Tool
Image	GPT-5 mini
Video	Qwen3-Omni
Audio	Qwen3-Omni
Document	GPT-5 mini
Code	Original format retained
3D	Three-view rendering → GPT-5 mini

Table 16. Metrics list of UNIM evaluation suite.

Metric	Range	Dimension
$\tau \uparrow$	[0, 1]	/
SC $\uparrow$	[0, 1]	Semantic Correctness & Generation Quality
GQ $\uparrow$	[0, 1]	Semantic Correctness & Generation Quality
SQCS^{abs} $\uparrow$	[0, 1]	Semantic Correctness & Generation Quality
SQCS^{rel} $\uparrow$	[0, 1]	Semantic Correctness & Generation Quality
StS^{abs} $\uparrow$	[0, 1]	Response Structure Integrity
LeS^{abs} $\uparrow$	[0, 1]	Response Structure Integrity
StS^{rel} $\uparrow$	[0, 1]	Response Structure Integrity
LeS^{rel} $\uparrow$	[0, 1]	Response Structure Integrity
HC $\uparrow$	[0, 1]	Interleaved Coherence
SH $\uparrow$	[0, 1]	Interleaved Coherence
ICS^{abs} $\uparrow$	[0, 1]	Interleaved Coherence
ICS^{rel} $\uparrow$	[0, 1]	Interleaved Coherence

the model can fully process all input modalities. This reflects the model's true capability within its valid operating range. $\mathcal{X}^{rel}$, in contrast, characterizes the model's overall effectiveness on UNIM by incorporating its coverage across the full dataset. Together, $\mathcal{X}^{abs}$ and $\mathcal{X}^{rel}$ capture model performance from modality supportability and holistic benchmark completeness.

D.5. Metric List

Overall, the UNIM evaluation suite consists of 3 dimensions and 13 metrics, as illustrated in Table 16. It provides a comprehensive assessment of MLLMs performance in interleaved multimodal learning.

E. Details of UNIMA

E.1. Receiving Module

The Receiving Module serves as the entry point for multimodal interleaved inputs spanning seven modalities. Its primary function is to apply modality-specific expert tools to extract and convert raw inputs into dense captions for each item.

Receiving Module integrates four specialized tools:
GPT-5 mini [20] handles text, image, document, and code comprehension, performing semantic parsing and symbolic reasoning to extract structured textual representations.
Qwen3-Omni Thinker [37] is dedicated to audio and video understanding, capturing temporal dependencies and cross-modal correlations in continuous spatio-temporal signals.
Qwen3-VL [27] serves as the visual grounding module, extracting object-level details and bounding-box coordinates as structured visual evidence.
PointLLM [38] processes 3D point-cloud data, enabling geometric reasoning and spatial perception within three-dimensional environments.

E.2. Traceable Evidence Reasoning Module

Traceable Evidence Reasoning (TER) module includes two parts: Structured Evidence Reasoning Chain and Verification Submodule.

E.2.1. Structured Evidence Reasoning Chain

The main purpose of **Step 1** is to generate *task-conditioned dense caption (TCDC)* and the *paraphrased question* in parallel by using the output of Receiving Module and original question. TCDC differs from conventional dense captioning that merely provides detailed semantic descriptions of visual or video content. It explicitly incorporates the task context into the caption generation process, allowing the semantic content, granularity, and style of the caption to be adaptively constrained by the current task. Formally, it can be expressed as

$$Y = f_{\text{caption}}(X \mid T), \tag{29}$$

where X denotes the non-textual input modality and T represents the task context. The resulting captions automatically emphasize semantics relevant to T while filtering out irrelevant details. Paraphrased question is a rephrased version of the original question. It emphasizes the key point of the original question, making the description clearer. Then, all multimodal evidence is first normalized and reorganized into unified, task-conditioned representations that support the subsequent stages of structured reasoning. All prompt templates used across these tools are summarized in Fig. 33.

In **Step 2**, UNIMA determines whether the user query requires quantitative computation beyond pure textual reasoning. The module reads both the paraphrased question and the cross-modal TCDC evidence, and checks for the presence of numerical operations, data processing, statistical inference, table manipulation, or any task that cannot be reliably solved through language reasoning alone. If such requirements are detected, Step 2 automatically invokes the Code Interpreter, executes the necessary computation, and returns a concise *data report* summarizing the inputs, intermediate steps, and final results. If not required, the *data report* is omitted. Fig. 34 shows the prompt template.

In **Step 3**, UNIMA converts the intermediate outputs (TCDC, paraphrased question, and data report) from the Receiving and TER modules into three structured

components that guide the Generating Module. *modalities content* component specifies all non-text modalities that must appear in the final interleaved output (e.g., <image2>, <audio1>, <video1>). For each instance, the agent produces a concise, task-aligned dense caption derived from the TCDC and the paraphrased question. The result is a JSON grouping of modality instances by type. *text content* defines the connective narrative that will be interleaved with non-text modalities. Instead of generating the final prose, the agent outputs a structured planning string of the form'<image1>xx<audio1>xx<video1>xx', where the segments between placeholders provide high-level textual transitions. The third part is *tool list*, for each non-text placeholder tag in modalities content, the agent selects the appropriate generation tool from the system's tool set (Qwen3-Omni for audio, GPT-Image-1 for images, Sora-2 for video, PCDreamer for 3D). The mapping is returned as a JSON dictionary linking each modality instance to its corresponding tool. Together, these three components provide a complete, machine-readable plan that specifies what to generate, how it fits into the interleaved sequence, and which tool will handle each modality. The prompt template of this step is shown in Fig. 35.

In **Step 4**, UNIMA converts the structured plan generated in the previous stage into a unified, executable *final report*. Given the modalities content, text content, and tools list, the assembly agent first performs a strict consistency check to ensure that all modality tags follow the correct naming pattern and that every placeholder appearing in text content corresponds to an instance defined in both modalities content and tool list, with matching modality types and instance IDs. The module does not alter captions, tool assignments, or narrative flow; instead, it verifies structural coherence and formatting correctness. Once the cross-structure alignment is confirmed, the agent bundles the three validated components into the final report, which serves as the deterministic intermediate representation consumed by the Generating Module. Fig. 36 shows the detail of prompt template of this step.

E.2.2. Verification Submodule

The Verification Submodule ensures that the final report produced by the TER module is both correct and aligned with the user's original question before being passed to the Generating Module. It consists of two tightly coordinated components: the **Checker** and the **Judger**. Both operate on clearly defined inputs and produce outputs that guarantee the reliability of the entire pipeline.

Inputs and Outputs. The Verification Submodule receives two inputs: (1) the *user input question*, which specifies the task and constraints, and (2) the final report produced at the end of TER reasoning. Its output is either a validated final report that can safely enter the Generating Module, or an updated and corrected final report produced after the Judger performs targeted repairs.

Checker. The Checker compares only two items, the original user question and the final report. It does not inspect intermediate reasoning and therefore acts as an external critic. The Checker evaluates whether the final report directly addresses the user's task, maintains internal logical consistency, and satisfies all explicit constraints such as modality requirements, formatting rules, and domain assumptions. If the final report satisfies these conditions, the Checker outputs it unchanged to the Generating Module. If any issue is detected, the Checker rejects the report and activates the Judger.

Judger. Once triggered, the Judger inspects the entire chain of intermediate outputs generated during TER reasoning, starting from Step 1 up to the final report. This includes TCDC outputs for each modality, the Data Report, structured understanding and planning artifacts, and all subsequent reasoning steps. The Judger conducts a backward analysis: it begins from the final report and examines whether the identified issue could have originated from an earlier step. If not, it moves one step backward and checks the correctness and sufficiency of that step's output. This process continues until the Judger identifies the earliest step that contains the underlying error or misalignment.

Upon locating the faulty step, the Judger rolls back the reasoning to that point and re-executes the downstream reasoning from that step to the end, generating a revised sequence of intermediate results and a new final report. This corrected output is then sent back to the Checker for validation.

Iterative Loop. Together, the Checker and Judger form an iterative verification loop. The Checker evaluates the final report; if it is unsatisfactory, the Judger diagnoses and repairs the earliest faulty step and regenerates a new final report. This loop continues until the Checker confirms that the report fully answers the user's question and complies with all constraints. To prevent infinite regress, a small upper limit is imposed on the number of verification cycles, and the Judger is designed to make minimal, localized corrections whenever possible.

Once validated, the final report is forwarded to the Generating Module as a reliable, fully verified representation of the reasoning outcome.

E.3. Generating Module

The Generating Module also incorporates four specialized tools. In this stage, GPT-5 mini receives and reads the final report produced by the TER module and invokes the appropriate tools to generate the final output content strictly according to the specifications outlined

in the report.

As shown in Fig. 37, which is the prompt template of the Generating Module. The Generating Module executes the final report produced in Step 4. It generates every non-text modality strictly according to its caption and assigned tool, without modifying any content or structure. After all modalities are produced, it follows the sequence specified in text content and inserts each output into its corresponding placeholder to form the final interleaved result.

The four tools are introduced as follows:

Qwen3-Omni Talker [37] is responsible for audio generation, converting textual or semantic descriptions into natural and temporally coherent speech signals.

Sora-2 [21] handles video generation, synthesizing high-fidelity dynamic scenes conditioned on multimodal textual descriptions and temporal semantics.

GPT-Image-1 [19] is dedicated to *image generation*, producing high-fidelity and semantically aligned visual content conditioned on the textual descriptions generated by GPT-5.

PCDreamer [31] performs 3D point-cloud completion, reconstructing or refining spatial geometries based on the inferred semantic and structural context.

The inputs to all generative tools are textual descriptions of modality-specific outputs produced by GPT-5 mini, which interprets the *final report* generated by the TER Module and converts it into task-conditioned generation instructions for each corresponding modality.

E.4. Ablation Study

For the ablation study of UniMA, we additionally sample 600 regular, intention-clear QA pairs from various domains to isolate the contribution of individual components. Using such well-defined queries allows us to evaluate the TER Module under controlled conditions, where the system must identify relevant evidence and organize modality-specific information with precision. This setup enables a focused examination of its impact on StS & LeS, SQCS, and ICS.

To obtain a clean ablation environment, the Receiving Module and Generating Module are kept unchanged, and only internal components of TER are perturbed. Specifically, we consider three interventions: **(a)** replacing the task-conditioned TCDC representation with unconditioned dense caption; **(b)** removing the entire SERC, so that the TCDC and paraphrased questions are fed directly to the Generating Module without any intermediate reasoning or evidence planning; and **(c)** disabling the Verification Submodule so that no internal checking, localization, or backtracking is performed.

The results reveal distinct and complementary roles among the components. Removing the reasoning chain causes a dramatic collapse in structural metrics, with StS dropping from roughly 52% to 16% and LeS from about 82% to 21%, highlighting the necessity of explicit evidence structuring for maintaining correct modality types, modality counts, and interleaving order. Substituting TCDC with vanilla captions leads to substantial degradation in SQCS and ICS due to the loss of task-aware grounding and reduced semantic density, whereas structural scores remain relatively stable, indicating that content quality is highly dependent on TCDC, but structural conformance is dominated by the reasoning chain. Disabling the Verification Submodule degrades both structural and semantic dimensions, particularly when placeholder mismatches and alignment errors cannot be corrected through iterative checking and backtracking.

Overall, TCDC provides semantic grounding, the reasoning chain governs structural organization, and the Verification Submodule ensures robustness. Their synergy enables UniMA to maintain high correctness, coherent multimodal alignment, and strong structural integrity in complex interleaved multimodal generation.

F. Extended Experiments

In this section, we provide more experimental setting details and extended experimental results on UniM.

F.1. Experimental Settings

In this section, we provide the detailed experimental settings. For AnyGPT [40], NExT-GPT [33], and MIO [29], we adopt the corresponding default configuration settings. Moreover, since AnyGPT and NExT-GPT do not support multiple inputs of the same modality, we perform a concatenation-based adaptation of files within each modality to ensure completeness of the input content. Further, we provide detailed hyperparameter settings for UniMA as presented in Table 18.

F.2. Comparison of Multimodal Flexibility with Baseline Models

We compare UniMA with representative baseline models in terms of multimodal flexibility, covering multi-item input / output, any-modality combinations, any-number inputs, and full modality coverage, as summarized in Table 17.

F.3. Rationality Verification of Evaluation Suite

F.3.1. StS and LeS

To validate the rationality of our proposed StS and LeS metrics, we design experiments to examine whether these scores exhibit predictable, directionally consistent, and proportionally reasonable responses when the modality types and placeholder tags counts in the model's output are subject to controlled perturbations.

Table 17. Comparison between baseline models and UNIMA on multimodal flexibility. In / Out -Modality: supported input / output modality. Multi-I Input / Output: multiple items of the same modalities in the input / output. Any-M Input / Output: any modality combinations in the input / output. Any-I Input / Output: any number of items of the same modality in the input / output.

Model	Multi-I Input	Multi-I Output	Any-M Input	Any-M Output	Any-I Input	Any-I Output	In-Modality	Out-Modality
AnyGPT	✗	✗	✗	✗	✗	✗		
NExT-GPT	✗	✗	✗	✗	✗	✗		
MIO	✓	✓	✗	✗	✗	✗		
UNIMA	✓	✓	✓	✓	✓	✓		

Table 18. Detailed hyperparameter settings for UNIMA.

Model	Mode	Temperature	Top_p	Top_k	Max_Tokens
GPT-5 mini	/	1	1	/	16,384
Qwen3-Omni	Non-Thinking	0.7	0.8	20	8,192
Qwen3-VL	Instruct	0.7	0.8	20	4,096
PointLLM	Eval	0.2	0.9	50	1,024

We implement two sets of experiments: (1) perturbation of modality types, (2) perturbation of modality placeholder tag count.

For the perturbation of modality types, we construct five response variants based on the ground truth, denoted as RT_k, where $k \in \{-2, -1, 0, +1, +2\}$. When $k < 0$, RT_k is obtained by removing $|k|$ modality types from the ground truth, with all associated placeholders removed accordingly. When $k > 0$, RT_k is constructed by adding k additional modality types not present in the ground truth, introducing one placeholder per added modality. The case $k = 0$ corresponds to no modification, where the ground truth is used as the response.

For the perturbation of modality placeholder tag count, we similarly construct five response variants based on the ground truth, denoted as RN_k, where $k \in \{-2, -1, 0, +1, +2\}$. When $k < 0$, RN_k is obtained by removing $|k|$ placeholder tags across existing modalities in the ground truth, ensuring that no modality is left with zero placeholders. When $k > 0$, RN_k is constructed by adding k additional placeholder tags across existing modalities in the ground truth. The case $k = 0$ again corresponds to no perturbation, where the ground truth is used without modification.

To ensure experimental validity, we select 200 instances from the dataset whose ground truth contains at least two non-text modalities and in which at least one modality has more than two placeholder tags. The modality type and placeholder tag perturbation settings are then applied independently to these selected instances. The experimental results align with our expectations, substantiating the validity of StS and LeS.

F.3.2. SQCS and ICS

Parameters Selection. Since both SQCS and ICS rely on tunable weighting factors to balance their constituent sub-dimensions, it is necessary to empirically examine how different weight configurations align with human annotations. This validation on real data enables the determination of optimal parameter settings before conducting large-scale evaluations.

During the construction of the automated scoring dataset, for each model, we randomly sample three instances per difficulty level across all domains of UNIM within its support set, and obtain the baseline model generated responses for each instance. This procedure results in a collection of 1,080 model response samples in total. To obtain reliable human annotations, We recruit three evaluators with prior experience in multimodal assessment. A unified instruction session is conducted to standardize the scoring criteria for semantic correctness & generation quality, and interleaved coherence to ensure the reliability of human evaluation. Subsequently, each evaluator independently assesses all 1,080 samples. The final human score for each instance is computed as the average of the three evaluators' scores. To streamline the human evaluation process, We develop an internal tool for human evaluation, as illustrated in Fig. 14.

We use human annotations as the ground truth and align each automated score with its corresponding human rating at the sample level. For every candidate weight configuration, we compute the resulting SQCS or ICS values and assess their alignment with human judgments through Pearson correlation analysis. In addition, we plot KDE curves to examine the similarity of score distributions. The weight setting that yields higher correlation and more consistent distributional patterns is regarded as the more appropriate configuration.

For the SQCS formulation, Fig. 6 presents the score distributions obtained under three weighting factors, $\eta^{\text{SQCS}} \in \{0.6, 0.7, 0.8\}$. All three SQCS curves display distributional shapes that are highly consistent with those of the human annotations, although slight differences remain in their correlation strength. Among the tested configurations, $\eta^{\text{SQCS}} = 0.7$ yields the highest Pearson correlation coefficient, with a value of Pearson correlation coefficient $r = 0.974$, and its KDE curve shows the closest alignment with the human rating distribution. As shown in Fig. 7, we conducted the same form of analysis for the ICS formulation under weighting factors $\eta^{\text{ICS}} \in \{0.7, 0.8, 0.9\}$. The results indi-

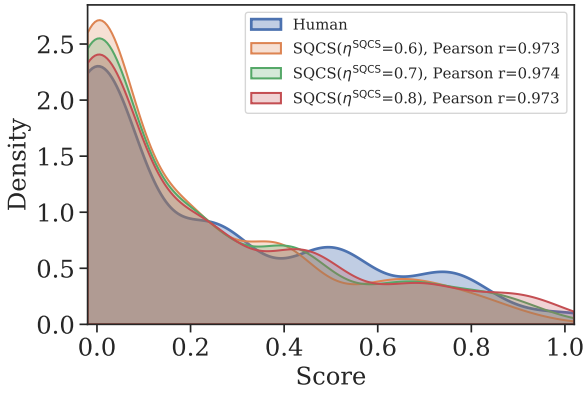


Figure 6. Parameters selection experiments results of SQCS.

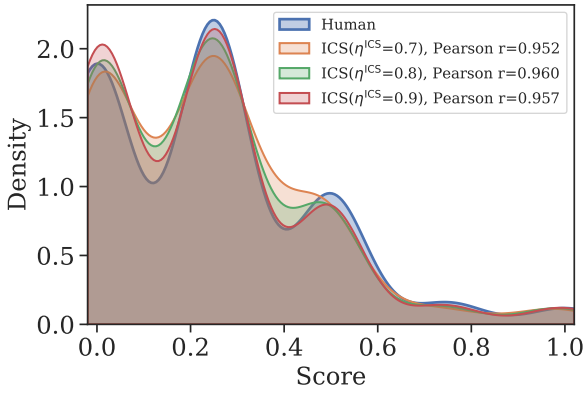


Figure 7. Parameters selection experiments results of ICS.

cate that different values of η^{ICS} substantially influence the balance between coherence and style consistency, which in turn affects the alignment with human *logical_coherence* annotations. Among the tested configurations, $\eta^{\text{ICS}} = 0.8$ achieves the highest Pearson correlation coefficient, reaching $r = 0.960$.

Therefore, we adopt $\eta^{\text{SQCS}} = 0.7$ as the weighting factor for SQCS and $\eta^{\text{ICS}} = 0.8$ as the weighting factor for ICS, and use these optimal configurations in all subsequent experiments.

Rationality Verification. After determining the optimal weighting factors, we further evaluate the consistency between the final SQCS and ICS metrics and the human semantic-quality annotations.

Based on the automated scoring dataset and the corresponding human annotations, we compute the automated score for each sample using the finalized weighting configurations and align it with the paired human rating at the instance level. We then perform sample-level Pearson correlation analyses.

The experimental results demonstrate that the proposed SQCS and ICS metrics exhibit a high degree of alignment with human evaluation.

F.4. Experimental Results

F.4.1. Domain-Level Performance

In this section, we provide detailed domain-level experiment results, as illustrated in Table 19, Table 20, Table 21, Table 22, Table 23, and Table 24.

F.4.2. Difficulty-Level Performance

In this section, we provide detailed difficulty-level experiments results, as illustrated in Fig. 38, Fig. 39, Fig. 40, and Fig. 41.

G. Case Study

This section presents 13 representative case studies that illustrate how the model behaves under different interleaved multimodal task settings. Each case is accompanied a corresponding figure, including Fig. 42, Fig. 43, Fig. 44, Fig. 45, Fig. 46, Fig. 47, Fig. 48, Fig. 49, Fig. 50, Fig. 51, Fig. 52, Fig. 53, and Fig. 54, covering diverse input–output modality combinations and varying levels of cross-modal interaction.

These cases provide qualitative insights into the model's performance in terms of semantic alignment, structural completeness, and stylistic consistency. They serve as a complementary perspective to the quantitative results presented earlier, offering a more comprehensive understanding of the model's strengths and limitations in realistic interleaved multimodal scenarios.

H. Ethic Statement

The UNIM benchmark uses multimodal assets that are sourced exclusively from publicly available and open-access datasets, as well as openly accessible content from the web. All the data used in this benchmark is derived from well-established open-source repositories and complies with their respective licenses, ensuring full transparency and legal compliance.

We confirm that no sensitive, private, or personally identifiable information is included in any of the multimodal assets. Additionally, we have carefully reviewed the contents of the datasets to ensure that there are no harmful, biased, or otherwise problematic elements that could negatively impact the integrity of the research or the communities involved. Ethical considerations regarding fairness, inclusivity, and non-discrimination have been taken into account throughout the data collection process.

Furthermore, the UNIM benchmark was developed with a focus on promoting responsible AI research, and we are committed to ensuring that our benchmark is used in a way that upholds ethical principles, including respect for privacy and the prevention of harm.

QA Pairs Construction Prompt

You are an excellent multimodal expert. Based on the following original data, please construct a data (Question-Answer pair) entry that strictly conforms to the JSON format below. Please design a multimodal interleaved Question-Answer pair. You can place different pieces of information from the original data into the input or output of the Question-Answer pair.

[Original Data] **{MULTIMODAL FILES}**

[Examples] **{Human Construction Examples}**

[Question-Answer Pair JSON Template]

```
{
  "field":  " ",
  "domain": " ",
  "id": "",
  "question": {
    "modality" : {
      "image1": "url",
      "image2": "url",
      "code1": "markdown block",
      "document1": "url"
    },
    "content": "Interleave <image1>, <image2>, <code1>, <document1> tags at the appropriate positions in the text and clearly indicate that the answer must include images to support or illustrate the explanation."
  },
  "answer": {
    "modality": {
      "image3": "url",
      "image4": "url",
    },
    "content": "This is the ground truth answer for the question. Interleave <image3> and <image4> placeholder tags at suitable positions within the text."
  }
}
```

[Construction Requirements]

- You need to design appropriate question-answer pair and clearly indicate in the question which specific modalities other than text are required to be included in the answer.
- The content of the question is the entire input fed into the model. The QA pair should be open-form.
- Give the JSON directly, no additional output information.
- The <> placeholder tags should be the components of the text sentence, not just a single word. For example, the <> placeholder tags can serve as the subject, object, or other components of the sentence.
- Please note that the <> placeholder tags of the question should not appear in the answer. The each <> placeholder tag ONLY appear once.

Figure 8. QA pairs construction prompt example.

Semantic Correctness Evaluation Prompt

You are a strict automatic semantic correctness grader. Score the semantic correctness of the Model Response relative to the Ground Truth. Ignore coherence, style, tone, length, politeness, formatting. Treat paraphrases, wording changes, ordering changes, and unit conversions as equivalent if they preserve meaning. Numbers may be mildly rounded, but the value/unit/range, comparative relations, causal/temporal conditions must remain semantically equivalent. If there are core factual errors, contradictions, hallucinations, or key omissions, lower the score. Return ONLY a JSON object with a single key 'Semantic Correctness', whose value MUST be one of: 1, 2, 3, 4, 5.

[Evaluation Dimensions and Scoring Guidelines]

(1) Semantic Correctness (1–5 points) **{Five-level rating *CRITERIA* for Semantic Correctness}**

[Response Format]

Strictly assign a score from 1, 2, 3, 4, 5 for each dimension.
Return JSON only, e.g.:

```
{
  "Semantic Correctness": 3
}
```

[Generated Answer to Evaluate] **{MODEL RESPONSE} {GROUND TRUTH}**

Figure 9. Prompt template for *Semantic Correctness*.

Text Quality Evaluation Prompt

You are a strict evaluator of text quality. Your task is to assign a single integer score from 1 to 5 for the answer, based solely on its textual quality — not its factual correctness or relevance.

[Evaluation Dimensions and Scoring Guidelines]

(1) Text Quality (1–5 points) **{Five-level rating *CRITERIA* for Text Quality}**

[Sample References for Scoring]

Text Quality Samples:

5: "The bloodhound stands as a legendary figure among scent-tracking dogs, combining anatomical precision with cinematic beauty. <image2> Its olfactory bulb is proportionally forty times larger than that of humans, enabling it to distinguish a single human scent among thousands. <image3> Historically, bloodhounds have been employed in everything from medieval manhunts to modern forensic investigations, revered for their near-mythic perseverance. <image4> Their legacy embodies both science and devotion: the perfect harmony between instinct and discipline."

4: "The bloodhound is renowned for having the best sense of smell among all dog breeds. <image2> These dogs are equipped with an exceptionally large olfactory bulb in their brains, allowing them to detect scents over great distances. <image3> Additionally, bloodhounds have been historically employed in various tracking roles due to their remarkable scent-detection abilities, such as in search and rescue missions. <image4>"

3: "The bloodhound is a kind of dog famous for its strong smelling power. <image2> It can find things or people by smell because its nose is very good. <image3> People use bloodhounds to help look for missing people or to follow trails in forests. <image4>"

2: "Bloodhound smell very strong, it can 找到 people in forest quickly sometimes. <image2:> dog nose grass picture, 光线 warm. Nose big so many police 使用 it for track， but sentences not good order. <image3>"

1: "狗 nose wow bloodhound smellings muy bueno?? <image2> imageno 混乱 blur 光线 hardfind 人 person trackerz olor sniff haha ###, 英语中文español一起转来转去—读不懂."

[Response Format]

Strictly assign a score from 1, 2, 3, 4, 5 for each dimension.
Return JSON only, e.g.:

```
{
  "Text": 4
}
```

[Text to Evaluate] {TEXT}

Figure 10. Prompt template for text in *Generation Quality*.

Code Quality Evaluation Prompt

Your task is to assign a single holistic score from 1 to 5 for a given code snippet. This score must reflect the overall quality across these six aspects: correctness, readability, design, performance, security, testability

[Evaluation Dimensions and Scoring Guidelines]

(1) Code Quality (1–5 points) **{Five-level rating *CRITERIA* for Code Quality}**

[Sample References for Scoring]

Code Quality Samples:

5:

```
def average_even(nums):
    """Avg of even ints; """
    """raise on non-iterable."""
    try:
        it = iter(nums)
    except TypeError:
        raise TypeError("nums must be iterable")
    t = 0
    c = 0
    for x in it:
        if isinstance(x, int) and x % 2 == 0:
            t += x
            c += 1
    return t / c if c else 0
```

4:

```
def average_even(nums):
    """Return average of even ints."""
    t = 0; c = 0
    for x in nums:
        if isinstance(x, int) and x % 2 == 0:
            t += x; c += 1
    return t / c if c else 0
```

3:

```
def average_even(nums):
    t = 0; c = 0
    for x in nums:
        if isinstance(x, int) and x % 2 ==
0:
            t += x; c += 1
    return t / c if c else 0
```

2:

```
def average_even(nums):
    total = 0
    c = 0
    for x in nums:
        if x % 2 == 0:
            total += x
            c += 1
    return total / c
```

1:

```
def average_even(nums):
    s = 0
    for x in nums:
        if x % 2 == 0:
            s += x
    return s / len(nums)
```

[Response Format]

Strictly assign a score from 1, 2, 3, 4, 5 for each dimension.
Return JSON only, e.g.:

```
{
  "Code": 4
}
```

[Code to Evaluate] {CODE}

Figure 11. Prompt template for code in *Generation Quality*.

Document Quality Evaluation Prompt

You are a strict professional document quality assessor. You evaluate the expression and presentation quality of tabular documents (in Markdown-like form). You MUST NOT evaluate factual correctness or image clarity. Focus ONLY on how well the document is expressed as a table.

[Evaluation Dimensions and Scoring Guidelines]

(1) Document Quality (1–5 points) **{Five-level rating *CRITERIA* for Document Quality}**

[Sample References for Scoring]

Document Quality Samples:

5:

Nutrition Facts (per 100g serving)

Nutrient	Value	Unit
Sugar	12.5	g
Protein	3.0	g
Fat	5.0	g
Calories	500	kcal

Note: All measurements follow FDA standard units.

4:

Item	Value	Unit
Sugar	12.5	g
Protein	3.0	g
Fat	5.0	g
Calories	500	kcal

3:

Item	Value	Unit
Sugar	12.5	g
Protein	3	g
Fat	5	g
Calories	500	kcal

2:

Item	Val	unit
sugar	12.5	grams?
prot	3	g
fat content	5g	
calories	500	

1:

	Sug	12	|g
Protn	3..	g
Fat|5g
Cal	5 00

[Response Format]

Strictly assign a score from 1, 2, 3, 4, 5 for each dimension. Return JSON only, e.g.:

```
{
  "Document": 4
}
```

[OCR Text and Document to Evaluate] {OCR TEXT} {DOCUMENT}

Figure 12. Prompt template for document in *Generation Quality*.

Interleaved Coherence Evaluation Prompt

You are a strict evaluation assistant. Evaluate the following text according to the criteria below. Please strictly assign a score (1, 2, 3, 4, 5) for both Holistic Coherence and Stylistic Harmony.

[Evaluation Dimensions and Scoring Guidelines]

(1) Holistic Coherence (1–5 points) **{Five-level rating *CRITERIA* for Holistic Coherence}**
(2) Style Harmony (1–5 points) **{Five-level rating *CRITERIA* for Style Harmony}**

[Sample References for Scoring]

Holistic Coherence Samples:
5: "A serene forest scene <image: Highly detailed forest with sunlight perfectly filtering through leaves, mossy ground, winding path, birds and river depicted naturally; every element harmonious; line, color, and visual style fully consistent; wording perfectly aligned with visuals; overall presentation flawless; semantic and logical order extremely clear>."
4: "Forest scene mostly coherent <image: Green trees with uneven sunlight, ground and path roughly sketched, birds slightly small; some minor inconsistencies in colors and line style; wording mostly aligns with visuals; local repetitions or small jumps exist but understanding not affected; overall coherent>."
3: "Forest scene partially consistent <image: Trees roughly drawn, uneven proportions; birds and river in slightly different style; colors partially inconsistent; wording partially deviates from image tone; local logic jumps present; requires some reasoning to fully understand>."
2: "Forest scene low coherence <image: Trees bright neon green, birds cartoonish, river pink; ground unrealistic; terminology mixed; visual elements conflicting; reading experience difficult; multiple contradictions>."
1: "Completely incoherent scene <image: A cityscape with unrelated desert, river, and random birds; style chaotic; terminology wrong; wording and visual style entirely uncoordinated; logic collapsed; almost impossible to understand>."

Style Harmony Samples:
5: "A serene forest scene <image: Highly detailed forest with tall green trees, sunlight perfectly filtering through leaves, mossy ground, winding path, birds and river depicted naturally; all elements harmonious; colors balanced and realistic; line work clean and consistent; wording precisely aligned with visuals; expression and visual style fully coordinated; overall immersive and smooth>."
4: "Forest scene mostly consistent in style <image: Green trees under partly cloudy sunlight, ground and path clearly visible; birds perched naturally; river reflecting surroundings; brushstroke thickness varies slightly; some colors slightly stylized; minor deviations in shading style; terminology mostly consistent; overall reading experience coherent>."
3: "Forest scene partially consistent <image: Trees unevenly shaped, some branches disproportionate; river and birds drawn simpler than trees; colors slightly clashing; wording occasionally deviates in tone from image; visual style not uniform; terminology partially mixed; overall understandable but smoothness reduced>."
2: "Forest scene inconsistent in style <image: Trees neon green, river tinted pink; birds cartoonish while trees semi-realistic; ground unrealistic with flat shading; terminology not uniform; visual style conflicting; wording and visuals partially misaligned; comprehension affected>."
1: "Completely inconsistent style <image: A cityscape mixed with desert, random river and cartoon birds; all elements in clashing styles; terminology entirely wrong; wording and image unaligned; expression and visual style chaotic; presentation almost impossible to interpret>."

[Response Format]

Strictly assign a score from 1, 2, 3, 4, 5 for each dimension.
Return JSON only, e.g.:

```
{
  "Holistic Coherence": 4,
  "Style Harmony": 5
}
```

[Text to Evaluate] {TEXT}

Figure 13. Interleaved coherence evaluation prompt.

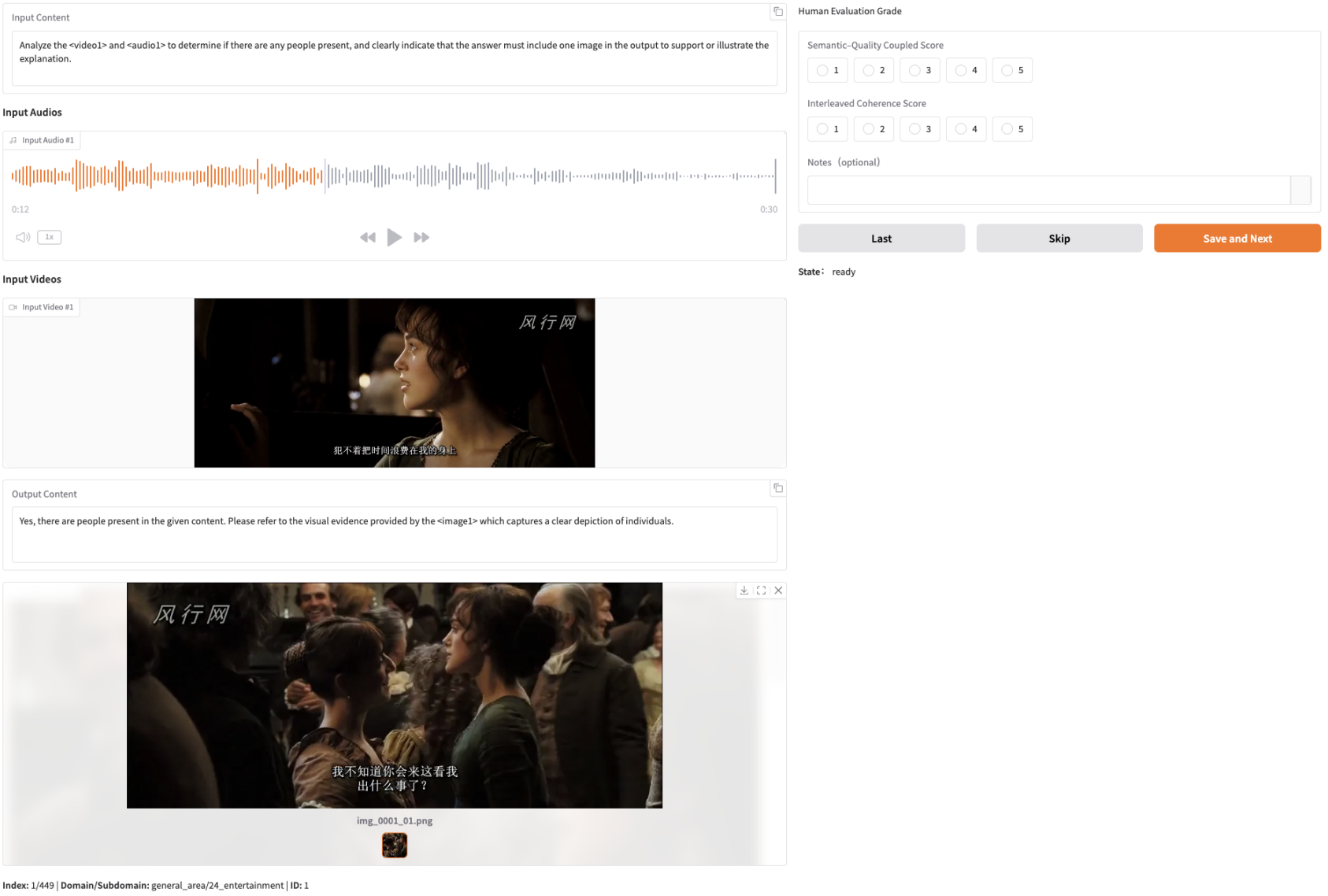


Figure 14. The screenshot of human evaluation tool we developed.

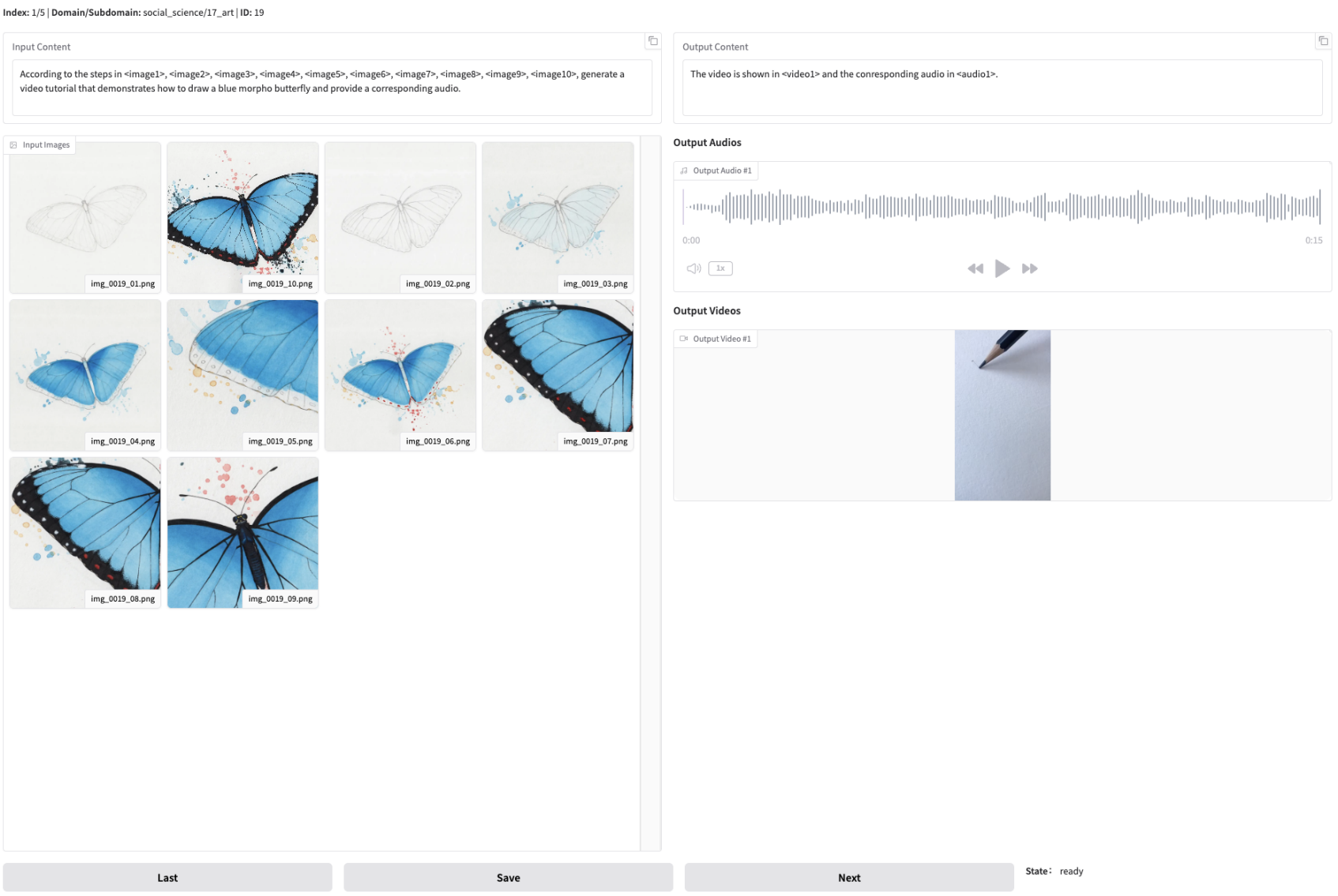


Figure 15. The screenshot of systematic verification tool we developed.

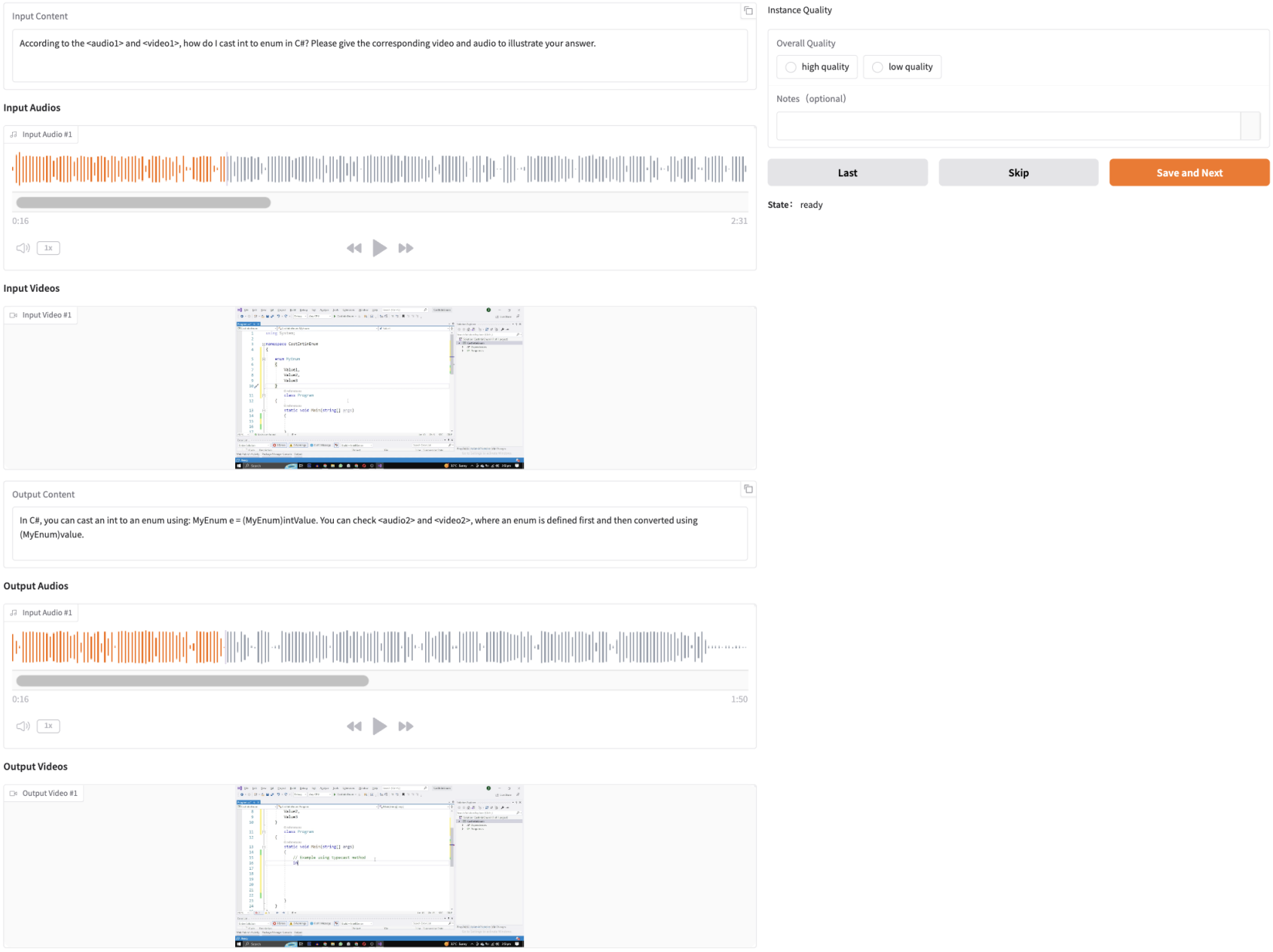


Figure 16. The screenshot of multi-reviewer quality evaluation tool we developed.

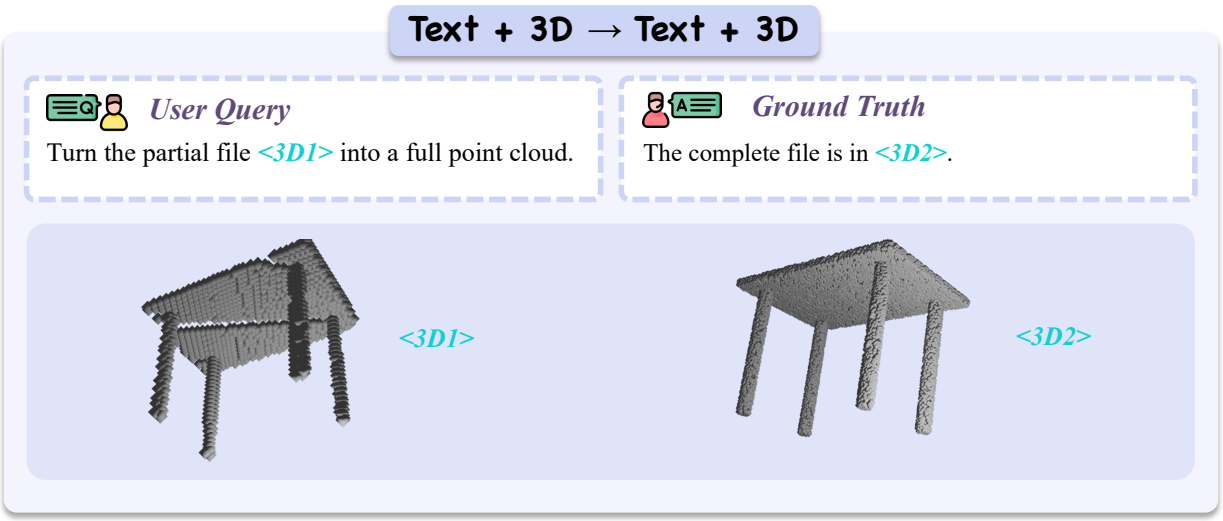


Figure 17. Data example of T + 3D to T + 3D interleaved combinations.

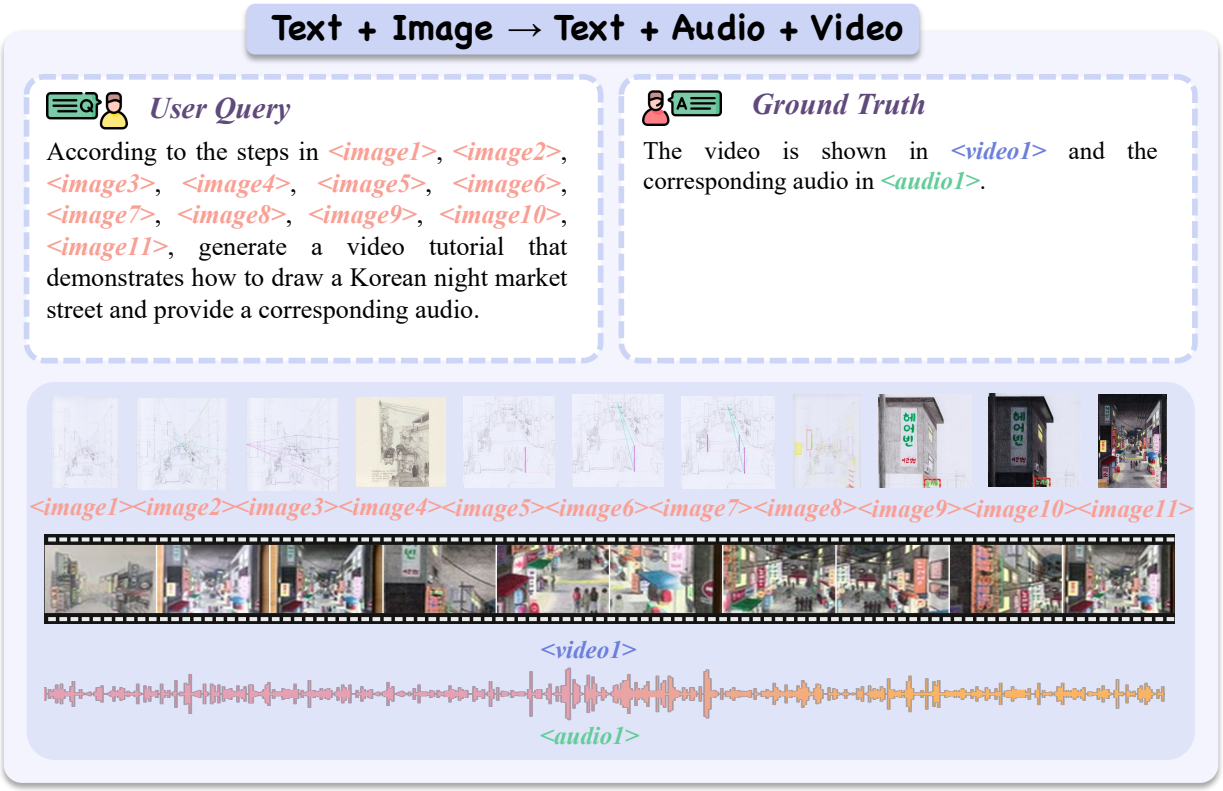


Figure 18. Data example of T + I to T + A + V interleaved combinations.

Figure 19. Data example of T + V + A to T + A interleaved combinations.

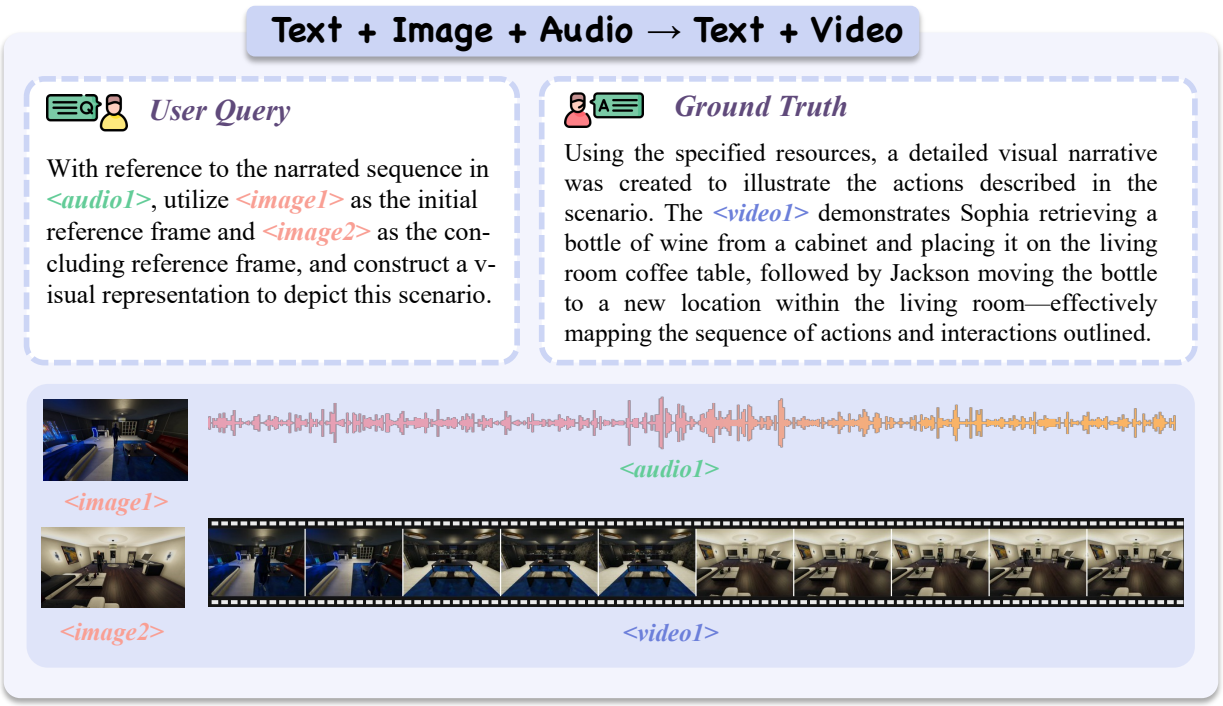


Figure 20. Data example of T + I + A to T + V interleaved combinations.

Text + Video → Text + Image

User Query

Find all missed moves in the game by analyzing the <video1>. Provide the screenshot of the game when this move was made.

Ground Truth

The missed moves of this game are: <image1>, <image2>, <image3>.

<video1>

<image1>

<image2>

<image3>

Figure 21. Data example of T + V to T + I interleaved combinations.

Text + 3D → Text + Image + Document

User Query

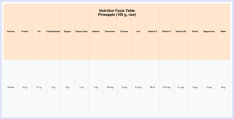
How can we identify and analyze the food item re-presented in this 3D mesh data, <3D1>? Please provide an explanation that incorporates the visual and textual elements appropriately.

Ground Truth

The 3D mesh data corresponds to a pineapple, which can be identified in <image1> by its spiky skin, leafy crown, and bright yellow flesh. Pineapples are juicy, sweet-tart tropical fruits, often eaten fresh or used in desserts and beverages. Their nutritional value is shown in <document1>, with 50 kcal per 100 g, 13 g of carbohydrates (10 g sugars), and high vitamin C content (47.8 mg), making pineapple a refreshing and immune-boosting fruit.

<3D1>

<image1>



<document1>

Figure 22. Data example of T + 3D to T + I + D interleaved combinations.

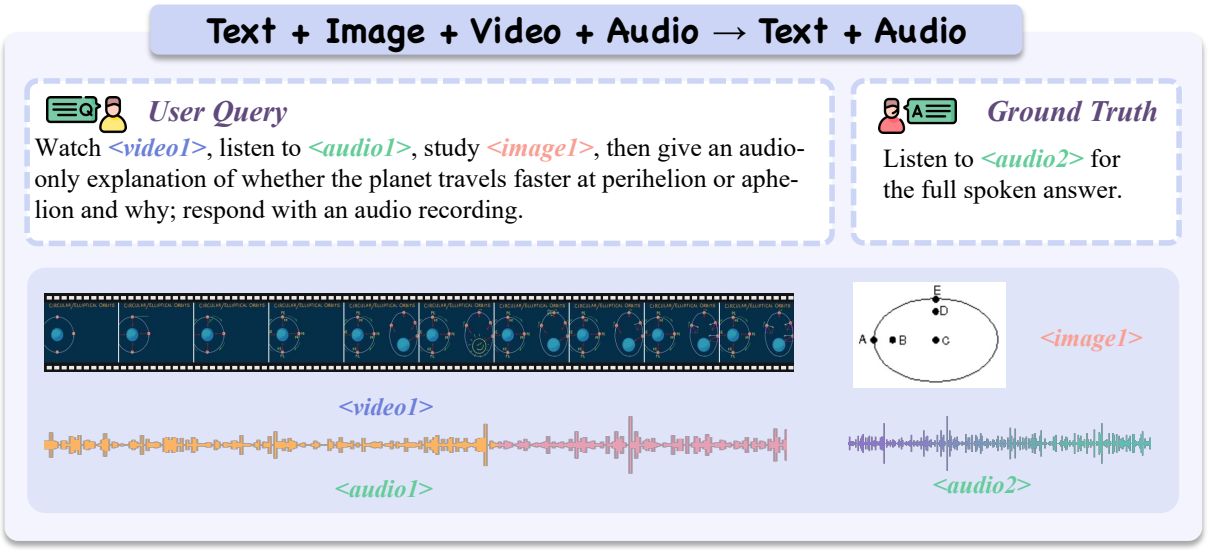


Figure 23. Data example of T + I + V + A to T + A interleaved combinations.

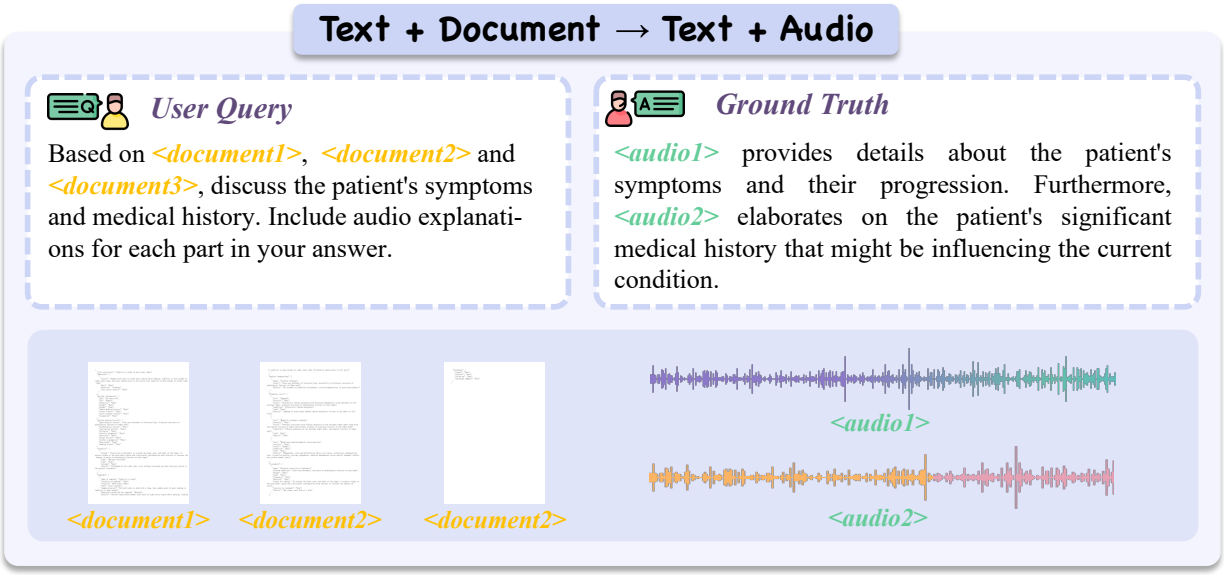


Figure 24. Data example of T + D to T + A interleaved combinations.

Text + Image + Audio → Text + Image

User Query

Based on the <image1>, analyze the scenario shown. The <audio1> provides an auditory description of the experiment setup. Describe the forces acting on the objects and predict their motion. Include one image in you're answer to visually support your explanation.

Ground Truth

In the given experiment, the primary forces at play include gravitational force, friction, and possibly a tension force if strings or ropes are involved. The predicted motion varies depending on these forces but could involve acceleration or deceleration along a plane. Refer to <image2> to see the highlighted key components and interaction areas.

<image1> <audio1> <image2>

Figure 25. Data example of T + I + A to T + I interleaved combinations.

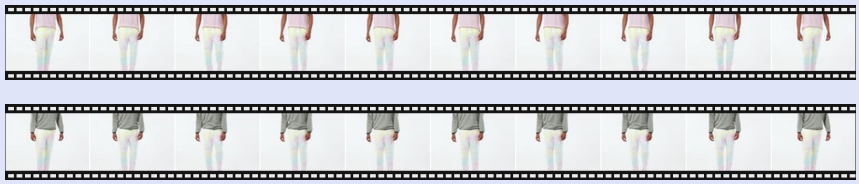
Text + Image + Video → Text + Video

User Query

Analyze the outfit combination where <video1> dis-plays the model before a costume change, and <image1> illustrates the new attire to be worn. P-rovide a detailed description along with a video of the model showcasing the style after the change.

Ground Truth

<video2> showcases a man wearing a new gray long-sleeved hoodie with a zippered collar, paired with pastel pink pants and white sneakers.



<video1>

<video2> <image1>

Figure 26. Data example of T + I + V to T + V interleaved combinations.

Text + Video → Text + Audio

User Query

Watch the clip within *<video1>* and consider the following query: Is the moving red object lighter than the moving rubber object? Please ensure your answer utilizes an audio that audibly demonstrates weight differences or provides relevant commentary.

Ground Truth

The moving red object is not lighter than the moving rubber object, as indicated by *<audio1>* where an expert explains weight distribution with sound evidence.

<video1>

<audio1>

Figure 27. Data example of T + V to T + A interleaved combinations.

Text + Code → Text + Code

User Query

How can the given C++ code for reversing a linked list be implemented in Python? Your answer must include the code representation. Refer to the *<code1>* for code details.

Ground Truth

To translate the given C++ code into Python, the function structure and logic can be directly adapted into Python syntax. Refer to *<code2>* for the implementation details, ensuring type consistency and idiomatic Python practices.

```cpp
class ListNode {
public:
    int val;
    ListNode *next;
};
ListNode* reverseList(ListNode* head) {
    ListNode* prev = nullptr;
    ListNode* current = head;
    ListNode* next = nullptr;

    while (current != nullptr) {
        next = current->next;
        current->next = prev;
        prev = current;
        current = next;
    }
    return prev;
}
```

<code1>

```python
class ListNode:
    def __init__(self, val=0, next=None):
        self.val = val
        self.next = next

def reverse_list(head):
    prev = None
    current = head
    while current is not None:
        nxt = current.next
        current.next = prev
        prev = current
        current = nxt
    return prev
```

<code2>

Figure 28. Data example of T + C to T + C interleaved combinations.

Text + Audio + Document → Text + Image

User Query

Discuss the impact of prohibitive measures described in <audio1>, referencing details from <document1> to support your explanation. Include <document2> to highlight relevant psychological theories and clearly ask the model to include one or more images to support or illustrate the explanation.

Ground Truth

Prohibitive measures, as explained through various theoretical lenses, have multifaceted impacts on individuals and societies. <image1> gives insight into the emotional and cognitive responses elicited by such measures. Furthermore, <image2> illustrates the direct correlation bet-ween these prohibitions and the resultant behavioral changes, integra-ting the key findings from empirical research. Delving into the broader societal implications, <image3> contextualizes these outcomes within wider social science perspectives, presenting a holistic view of the societal reactions to prohibitive policies.

<audio1>

<document1>

<document2>

<image1>

<image2>

<image3>

Text + Video + Audio → Text + Video + Audio

User Query

What are access modiiers in C# after observing the <audio1> and <video1>? Please provide the correspondding audio and video in your answer.

Ground Truth

In C#, access modifiers are used to control the accessibility of classes, methods, variables, and other members. The common ones are: public (visible to all code), private (visible only within the current class), protected (visible within the current class and its derived classes), internal (visible within the same assembly), protected internal (visible within the current assembly and derived classes), and private protected (visible only to derived classes within the current assembly). In <audio2> and <video2>, there is a demonstration showing how different modifiers define class members and how their visibility changes in different scopes.

<video1>

<video2>

<audio1>

<audio2>

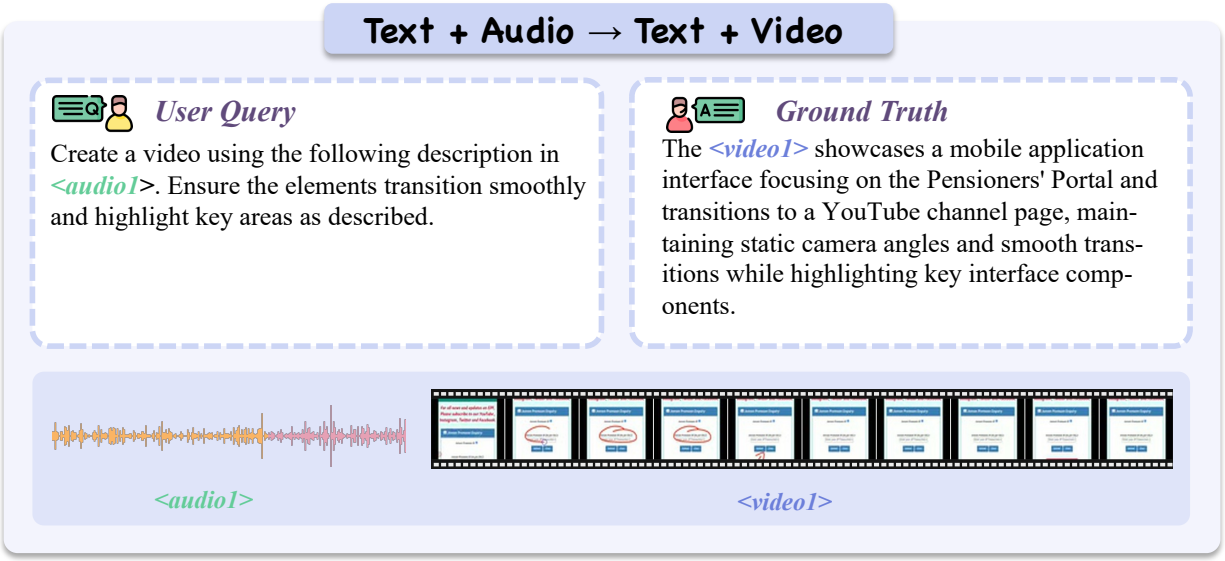


Figure 31. Data example of T + A + D to T + I interleaved combinations.

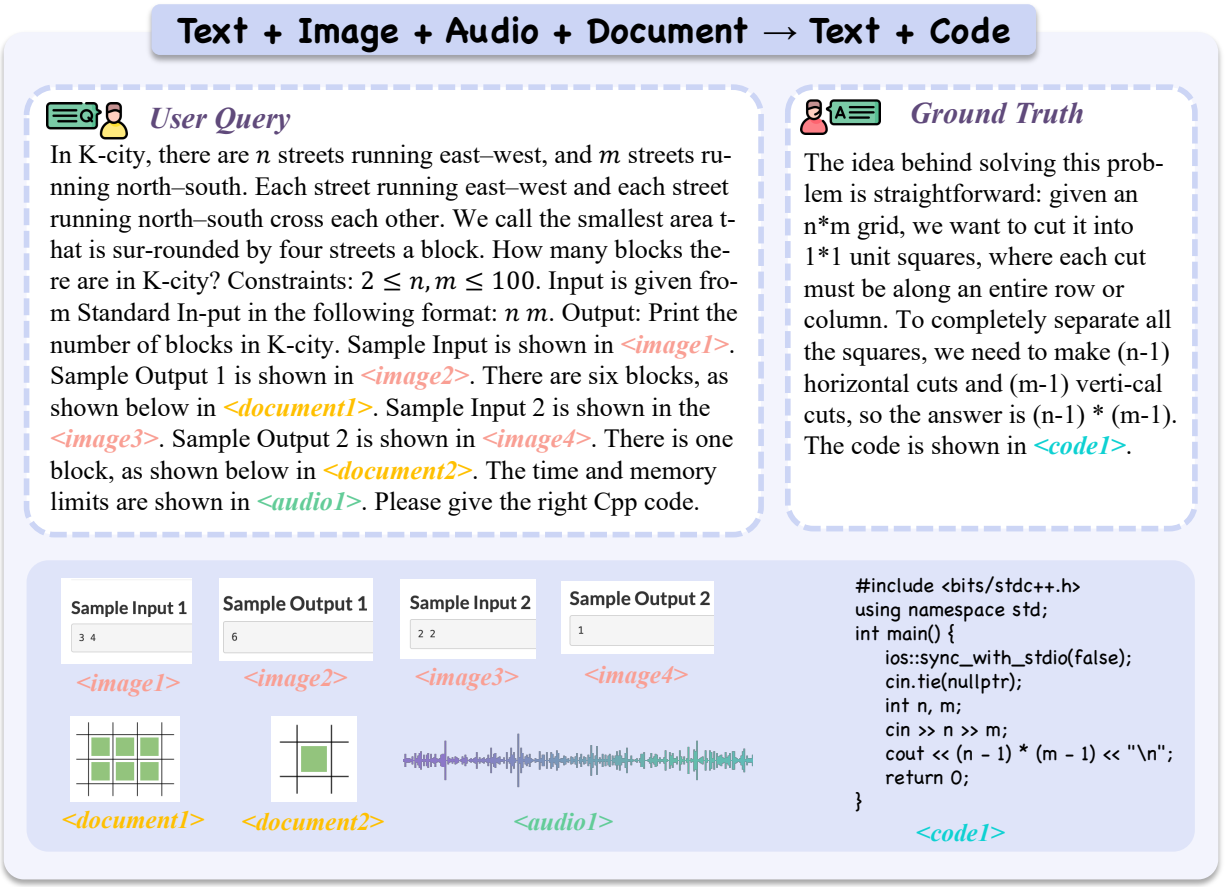


Figure 32. Data example of T + I + A + D to T + C interleaved combinations.

Prompt Template for Step 1 in TER Module

You are the Step 1 Planning Agent. Your job is to process your assigned input modality and generate TCDC captions strictly aligned with the Original Question. Output only the TCDC for your modality. For images, call grounding when needed.

[Input] **{DENSE CAPTION} & {ORIGINAL QUESTION}**

[Tasks]

For each instance:

- Generate a Task-Conditioned Dense Caption (TCDC) .
- Ensure strict alignment with the Original Question.
- Include ONLY task-relevant content; avoid redundancy or irrelevance.
- Keep instance correspondence, e.g., TCDC["image"]["image1"] → `<image1>`.

Output only the TCDC for your handled modality.
Cross-modal integration is performed by GPT-5.

For `<image*>` inputs:
If grounding is required, call Qwen3-VL and merge bounding-box JSON results into TCDC["image"]["`<imageX>`"]; otherwise, skip grounding.

GPT-5 collects all TCDC outputs across modalities and rewrites the Original Question into a concise, task-focused version (Paraphrased Question).

[Output Format]

```
{
 "TCDC": {
    "<modality1>": {
       "<instance_id1>": "<task-conditioned caption>",
       “<instance_id2>”: “<task-conditioned caption>”,
                    …
    },
    "<modality2>": {
       …
    }
 },
 "Paraphrased_Question":  "<rewritten question>"
}
```

Figure 33. Prompt template for step 1 in TER module.

Prompt Template for Step 2 in TER Module

You are Step 2 Planning Agent. Read the Paraphrased Question and TCDC, determine whether the task requires data analysis or computation. If yes, call Code Interpreter and produce a concise Data Report; otherwise output null. Only return is_data_related and Data_Report.

[Input] {PARAPHRASED QUESTION} & {TCDC}

[Tasks]

Decide whether the question involves:

- numerical computation,
- data processing,
- statistical inference,
- table operations,
- code execution,
- or quantitative reasoning that cannot be completed using textual reasoning alone.

If the question IS data-analysis-related: Invoke the Code Interpreter.
Generate a structured Data Report summarizing the executed computation.

The Data Report must contain:

- Inputs used
- Processing steps
- Key intermediate values
- Final quantitative results

If the question is NOT data-analysis-related: Do not generate a Data Report.

[Output Format]

```
{
 "is_data_related": true/false,
 "Data_Report": "<generated report or null>"
}
```

Figure 34. Prompt template for step 2 in TER module.

Prompt Template for Step 3 in TER Module

You are the Step 3 Planning Agent. Using TCDC, the Paraphrased Question, and the Data Report, construct three structures: Modalities_Content, Text_Content, Tool_List. Assign new modality tags sequentially. Output only these three fields.

[Input] {TCDC} & {PARAPHRASED QUESTION} & {DATA REPORT}

[Tasks]

Modalities_Content:
Using the TCDC, the Paraphrased Question, and the Data Report:
Decide which non-text modalities must appear in the final output (e.g., `<image2>`, `<image3>`, `<audio1>`, `<video1>`, `<threeD1>`, etc.).
For each required instance:

- Produce a dense, task-conditioned caption guiding that instance's generation.
- Group instances under: "image", "audio", "video", "threeD".
- Ensure every caption is: strictly aligned with the Original Question and Paraphrased Question, free of irrelevant or generic content.

Tag-ID Assignment Rule:

- For each modality, inspect the TCDC to identify the highest existing tag (e.g., `<image1>`, `<audio3>`, `<video2>`, `<threeD1>`).
- Any newly generated output instance for that modality must continue the numbering sequence. Example: if TCDC contains `<image1>`, the next generated image tag must be `<image2>`.
- This rule applies independently to each modality type.

(Include only the modalities and instances actually needed.)

Text_Content:
Plan the connective text that will appear BETWEEN modality placeholders in the final interleaved sequence. The sequence has the conceptual form: <yyy> xxxxxx <yyy> xxxxx <yyy> …
where:

- <yyy> are modality placeholders such as `<audio1>`, `<video1>`, `<threeD1>`, …
- xxxxxx is the Text Content between them.

Tool_List:
For each non-text instance in Modalities_Content, select exactly one generation tool from the following set:

- "Qwen3-Omni" : audio generation
- "GPT-Image-1" : image generation
- "Sora-2" : video generation
- "PCDreamer" : 3D point-cloud completion

[Output Format]

```
{
  "Modalities_Content": {
    "image":  { "<image id>": "<dense caption>", ...},  // captions for image outputs
    "audio":  { "<audio_id>": "<dense caption>", ...},  // captions for audio outputs
    "video":  { "<video_id>": "<dense caption>", ...},  // captions for video outputs
    "threeD":  { "<threeD_id>": "<dense caption>", ...},  //captions for 3D/point-cloud  outputs
      ...
  },
  "Text_Content": "<image1> ... <image2> ... <audio1> ... <video1> ...",
  "Tool_List": {
    "<instance_id>": "<tool_name>", ...
  }
}
```

Figure 35. Prompt template for step 3 in TER module.

Prompt Template for Step 4 in TER Module

You are the Step 4 Planning Agent. Validate tag consistency across Modalities_Content, Text_Content, and Tool_List, then package them into Final_Report without modifying content. Output only the Final_Report object.

[Input] **{MODALITIES CONTENT} & {TEXT CONTENT} & {TOOL LIST}**

[Tasks]

Perform a strict consistency check:

- Every placeholder (`<imageX>`, `<audioX>`, `<videoX>`, `<threeDX>`) appearing in
- Text_Content must appear as a key in both Modalities_Content and Tools_List.
- Modality prefixes must match their groups, e.g.:
  `<imageX>` → Modalities_Content["image"],
  `<audioX>` → Modalities_Content["audio"],
  `<videoX>` → Modalities_Content["video"], …
- Verify placeholder format strictly follows:
  <modalityName + positive integer>, e.g., `<image2>`, `<audio1>`, `<video3>`.

[Output Format]

```
{
 "Final_Report": {
   "Modalities_Content": { … },
   "Text_Content": "…",
   "Tool_List": { … }
 }
}
```

Figure 36. Prompt template for step 4 in TER module.

Prompt Template for Generating Module

You are the Generating Module. Execute the Final_Report strictly: generate each modality instance using the specified tool and its caption, then construct the final interleaved output by inserting these generated outputs into the exact positions defined by Text_Content. Produce only the final merged output.

[Input] {FINAL REPORT}

[Tasks]

For every modality instance in Final_Report.Modalities_Content:

- Identify its required generation tool from Final_Report.Tool_List.
- Use that tool exclusively.
  - `<imageX>` → GPT-Image-1
  - `<audioX>` → Qwen3-Omni
  - `<videoX>` → Sora-2
  - `<threeDX>` → PCDreamer

Generate the corresponding content strictly following the caption in Final_Report.Modalities_Content.

After generating all non-text modalities:

- Construct the final interleaved output using Final_Report.Text_Content as the sequence backbone.
- Insert each generated modality output at the exact placeholder
- positions (`<imageX>`, `<audioX>`, `<videoX>`, `<threeDX>`) indicated by Final_Report.Text_Content.
- Preserve the order and wording exactly.

[Output Format]

Return ONLY the final interleaved result with all placeholders replaced by the generated modality outputs.

Figure 37. Prompt template for generating module.

Table 19. Detailed experiment results on *Semantic Correctness & Generation Quality*.

Semantic Correctness & Generation Quality										
	#1	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**	**#8**	**#9**	**#10**
	#11	**#12**	**#13**	**#14**	**#15**	**#16**	**#17**	**#18**	**#19**	**#20**
	#21	**#22**	**#23**	**#24**	**#25**	**#26**	**#27**	**#28**	**#29**	**#30**
AnyGPT [40]										
	13.3	4.4	16.7	8.9	13.8	6.2	13.7	16.0	24.9	19.6
SC	16.9	18.9	33.4	6.5	23.3	51.9	5.4	17.4	4.8	2.0
	27.4	22.9	2.6	13.0	23.3	38.1	6.8	21.2	13.2	21.9
	44.8	24.7	49.8	44.0	32.3	39.1	32.9	22.0	54.0	35.0
GQ	45.7	26.1	45.8	6.7	28.1	48.5	9.5	19.6	5.8	1.9
	29.6	46.5	3.7	39.0	42.7	40.5	7.2	42.4	16.8	32.7
	11.1	3.4	14.1	7.4	11.0	5.0	10.9	12.3	21.4	15.7
$SQCS^{abs}$	14.2	14.7	27.9	4.6	18.3	43.8	4.0	13.2	3.4	1.4
	21.6	19.2	1.9	10.6	19.3	31.3	4.9	17.6	9.9	17.5
NExT-GPT [33]										
	3.3	3.0	1.4	59.7	8.1	0.9	1.6	0.6	2.7	2.6
SC	0.9	2.4	6.4	14.7	21.4	5.8	38.7	9.3	46.2	22.2
	1.7	2.5	9.7	2.5	5.7	13.8	4.9	2.4	4.2	6.4
	15.0	32.1	17.2	12.6	27.1	14.7	33.2	21.0	25.7	35.6
GQ	32.2	15.9	29.1	37.9	49.1	49.7	38.0	26.7	21.1	19.3
	29.0	37.1	23.7	34.6	47.2	39.3	9.8	37.1	20.8	21.7
	2.6	2.5	1.1	42.7	6.5	0.7	1.3	0.4	2.2	2.2
$SQCS^{abs}$	0.7	1.9	5.2	12.4	18.1	4.9	31.1	7.8	32.9	17.6
	1.5	2.1	7.9	2.0	4.8	11.8	3.8	2.0	3.4	5.2
MIO [29]										
	12.9	39.8	18.7	11.3	20.8	22.8	22.3	10.0	22.1	16.8
SC	9.6	20.7	22.5	14.5	43.0	44.3	1.9	19.9	23.0	52.6
	37.2	16.6	0.0	10.5	34.6	52.1	27.8	29.5	17.3	22.6
	14.5	19.6	47.1	25.0	33.0	34.6	12.1	43.1	41.7	20.1
GQ	21.4	34.9	28.6	18.1	43.5	31.8	12.3	41.0	47.0	49.0
	34.7	45.2	0.2	23.5	41.3	51.8	63.0	56.1	48.1	10.8
	9.6	30.2	15.7	8.7	16.6	18.3	16.4	8.3	18.3	12.7
$SQCS^{abs}$	7.3	16.6	17.7	10.9	35.7	35.2	1.4	16.4	19.4	44.5
	29.9	13.8	0.0	8.1	28.5	44.5	24.7	25.6	14.6	16.5
UNIMA										
	67.9	80.5	60.4	45.1	67.1	68.5	35.7	61.4	62.3	45.7
SC	78.9	75.0	73.2	81.9	79.1	83.7	52.1	78.9	89.0	92.2
	67.0	56.8	79.7	39.2	77.6	78.2	41.7	44.0	79.1	40.4
	84.9	89.3	85.6	69.1	83.6	97.4	45.3	83.1	93.4	93.0
GQ	87.5	82.0	78.3	96.4	84.6	82.3	85.2	80.7	92.6	67.9
	83.7	82.2	95.2	83.3	82.1	82.8	72.2	89.1	95.0	81.7
	65.3	78.6	57.6	40.5	64.4	68.2	30.7	58.6	61.6	44.9
$SQCS^{abs}$	76.3	70.9	68.6	81.5	76.0	79.7	50.0	75.0	87.4	83.4
	64.4	54.0	78.6	37.5	73.5	74.2	40.1	42.6	78.0	38.6

Table 20. Detailed experiment results on *Semantic Correctness & Generation Quality*.

	Semantic Correctness & Generation Quality									
	#1	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**	**#8**	**#9**	**#10**
	#11	**#12**	**#13**	**#14**	**#15**	**#16**	**#17**	**#18**	**#19**	**#20**
	#21	**#22**	**#23**	**#24**	**#25**	**#26**	**#27**	**#28**	**#29**	**#30**
AnyGPT [40]										
	99.9	79.6	98.8	98.0	95.0	99.4	47.4	92.5	95.3	98.3
τ	98.4	98.3	98.2	91.0	95.7	97.0	99.2	70.2	99.3	100
	96.5	98.7	100	40.0	95.7	99.7	99.7	96.3	98.9	77.3
	11.1	2.7	14.0	7.2	10.4	5.0	5.2	11.3	20.4	15.4
SQCS^{rel}	13.9	14.4	27.4	4.2	17.5	42.5	3.9	9.3	3.4	1.4
	20.9	18.9	1.9	4.3	18.4	31.2	4.9	16.9	9.8	13.5
NExT-GPT [33]										
	57.0	60.3	88.3	36.8	69.1	97.3	44.8	50.2	93.8	48.5
τ	98.2	98.3	97.6	36.4	94.3	95.8	99.4	96.9	98.1	20.0
	33.9	98.3	74.9	40.0	76.6	82.0	99.7	74.1	82.2	71.8
	1.5	1.5	0.9	15.7	4.5	0.7	0.6	0.2	2.1	1.1
SQCS^{rel}	0.7	1.9	5.0	4.5	17.0	4.7	30.9	7.5	32.2	3.5
	0.5	2.0	5.9	0.8	3.7	9.7	3.8	1.5	2.8	3.7
MIO [29]										
	57.0	60.3	87.3	36.8	51.8	97.3	44.8	50.2	93.8	12.2
τ	98.2	98.3	97.6	36.3	94.3	95.7	98.8	70.2	98.1	20.0
	33.9	98.3	74.9	40.0	76.2	82.0	99.7	52.9	82.2	77.3
	5.5	18.2	13.7	3.2	8.6	17.8	7.3	4.2	17.1	1.6
SQCS^{rel}	7.2	16.3	17.2	4.0	33.7	33.7	1.4	11.5	19.0	8.9
	10.1	13.6	0.0	3.2	21.7	36.5	24.6	13.5	12.0	12.8
UNIMA										
	100	100	100	100	100	100	100	100	100	100
τ	100	100	100	100	100	100	100	100	100	100
	100	100	100	100	100	100	100	100	100	100
	65.3	78.6	57.6	40.5	64.4	68.2	30.7	58.6	61.6	44.9
SQCS^{rel}	76.3	70.9	68.6	81.5	76.0	79.7	50.0	75.0	87.4	83.4
	64.4	54.0	78.6	37.5	73.5	74.2	40.1	42.6	78.0	38.6

Table 21. Detailed experiment results on *Response Structure Integrity*.

	Response Structure Integrity									
	#1	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**	**#8**	**#9**	**#10**
	#11	**#12**	**#13**	**#14**	**#15**	**#16**	**#17**	**#18**	**#19**	**#20**
	#21	**#22**	**#23**	**#24**	**#25**	**#26**	**#27**	**#28**	**#29**	**#30**
AnyGPT [40]										
	3.8	14.2	25.1	18.7	9.3	0	0	23.4	22.9	11.3
StS^{abs}	0.8	1.3	81.5	0.8	2.5	49.7	6.8	0.4	3.8	0.3
	29.8	11.2	0.6	37.0	10.9	1.4	7.2	5.4	11.2	9.9
	5.7	28.4	38.8	26.8	20.6	0	0	44.8	49.8	12.5
LeS^{abs}	1.2	2.2	86.4	1.9	4.2	54.3	8.7	0.6	6.1	0.5
	41.5	21.2	0.8	38.4	15.5	1.4	11.4	5.6	18.8	9.9
NExT-GPT [33]										
	2.2	0.7	0.9	5.9	8.0	0.0	1.1	0.5	0.9	0.3
StS^{abs}	0.6	0.3	6.3	0.7	0.6	1.4	0.1	1.1	1.6	0.7
	1.2	0.4	2.0	4.8	0.1	0.8	1.0	0.3	2.1	8.8
	2.7	0.7	1.0	6.0	8.4	0.0	1.4	0.5	1.0	0.3
LeS^{abs}	0.9	0.5	6.3	1.0	1.0	2.0	0.1	1.6	2.7	1.0
	1.7	0.5	2.0	4.8	0.3	0.8	1.5	0.3	3.9	8.8
MIO [29]										
	0.1	0.0	3.9	0.0	3.5	0.1	2.4	1.3	2.0	0.0
StS^{abs}	5.3	1.0	15.6	0.0	0.6	9.9	7.7	0.6	0.3	0.0
	0.4	1.5	0.0	4.1	0.8	0.1	0.1	1.8	16.2	7.8
	0.1	0.0	5.4	0.0	3.5	0.1	4.6	2.3	3.2	0.0
LeS^{abs}	7.7	1.7	16.2	0.0	1.1	11.8	11.9	1.6	0.3	0.0
	0.7	1.7	0.0	4.1	1.6	0.1	0.1	1.8	20.4	7.8
UNIMA										
	51.0	50.4	84.8	37.0	77.6	1.7	79.2	58.6	54.5	11.6
StS^{abs}	38.9	45.3	87.4	76.7	49.3	89.6	31.3	70.7	83.7	80.0
	78.9	75.5	56.7	84.6	73.8	93.9	51.3	56.3	57.4	36.5
	57.2	60.1	89.1	59.9	86.6	2.4	79.2	85.9	70.0	22.2
LeS^{abs}	53.5	57.1	90.6	88.8	82.8	93.4	52.0	95.1	97.0	80.0
	90.6	84.0	70.5	88.0	78.5	96.7	80.2	73.2	91.5	40.0

Table 22. Detailed experiment results on *Response Structure Integrity*.

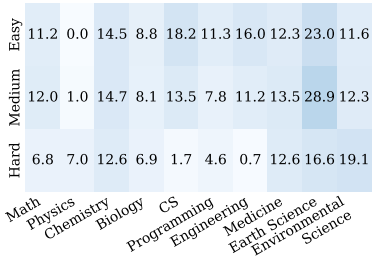
	Response Structure Integrity									
	#1	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**	**#8**	**#9**	**#10**
	#11	**#12**	**#13**	**#14**	**#15**	**#16**	**#17**	**#18**	**#19**	**#20**
	#21	**#22**	**#23**	**#24**	**#25**	**#26**	**#27**	**#28**	**#29**	**#30**
AnyGPT [40]										
	3.8	11.3	24.8	18.4	8.8	0	0	21.7	21.8	11.1
StS^{rel}	0.7	1.2	80.0	0.7	2.3	48.2	6.8	0.3	3.8	0.3
	28.8	11.1	0.6	14.8	10.4	1.4	7.2	5.2	11.1	7.7
	5.6	22.6	38.3	26.3	19.5	0	0	41.3	47.5	12.3
LeS^{rel}	1.2	2.2	84.9	1.7	4.0	52.7	8.6	0.4	6.1	0.5
	40.1	20.9	0.8	15.4	14.8	1.4	11.3	5.4	18.6	7.7
NExT-GPT [33]										
	1.2	0.4	0.8	2.2	5.5	0.0	0.5	0.3	0.8	0.1
StS^{rel}	0.6	0.3	6.1	0.2	0.6	1.3	0.1	1.1	1.6	0.1
	0.4	0.3	1.5	1.9	0.1	0.7	1.0	0.2	1.7	6.3
	1.5	0.4	0.9	2.2	5.8	0.0	0.6	0.3	0.9	0.1
LeS^{rel}	0.9	0.5	6.2	0.4	0.9	1.9	0.1	1.5	2.7	0.2
	0.6	0.5	1.5	1.9	0.3	0.7	1.5	0.2	3.2	6.3
MIO [29]										
	0.1	0.0	3.4	0.0	1.8	0.1	1.1	0.6	1.9	0.0
StS^{rel}	5.2	1.0	15.2	0.0	0.5	9.5	7.6	0.4	0.3	0.0
	0.2	1.5	0.0	1.7	0.6	0.1	0.1	1.0	13.3	6.1
	0.1	0.0	4.7	0.0	1.8	0.1	2.1	1.2	3.0	0.0
LeS^{rel}	7.6	1.7	15.8	0.0	1.0	11.3	11.8	1.2	0.3	0.0
	0.2	1.7	0.0	1.6	1.2	0.1	0.1	1.0	16.7	6.1
UNIMA										
	51.0	50.4	84.8	37.0	77.6	1.7	79.2	58.6	54.5	11.6
StS^{rel}	38.9	45.3	87.4	76.7	49.3	89.6	31.3	70.7	83.7	80.0
	78.9	75.5	56.7	84.6	73.8	93.9	51.3	56.3	57.4	36.5
	57.2	60.1	89.1	59.9	86.6	2.4	79.2	85.9	70.0	22.2
LeS^{rel}	53.5	57.1	90.6	88.8	82.8	93.4	52.0	95.1	97.0	80.0
	90.6	84.0	70.5	88.0	78.5	96.7	80.2	73.2	91.5	40.0

Table 23. Detailed experiment results on *Interleaved Coherence*.

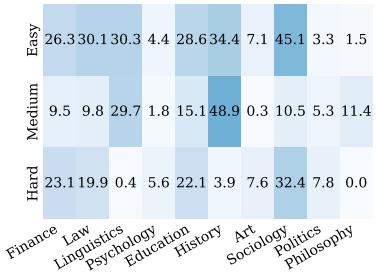
	Interleaved Coherence									
	#1	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**	**#8**	**#9**	**#10**
	#11	**#12**	**#13**	**#14**	**#15**	**#16**	**#17**	**#18**	**#19**	**#20**
	#21	**#22**	**#23**	**#24**	**#25**	**#26**	**#27**	**#28**	**#29**	**#30**
AnyGPT [40]										
	47.2	44.5	37.5	23.9	34.5	33.4	38.5	40.8	54.6	44.2
HC	65.3	36.1	58.4	8.7	35.8	60.3	12.0	28.4	5.9	2.0
	29.2	50.0	6.7	57.4	44.7	62.8	7.6	50.7	22.3	34.0
	58.3	53.9	46.6	25.1	40.4	38.9	43.9	39.5	65.5	51.3
SH	76.7	39.9	60.5	9.4	42.6	68.7	12.2	35.0	6.0	2.1
	33.0	60.2	8.2	75.9	53.5	67.9	8.1	57.0	24.1	30.8
NExT-GPT [33]										
	13.9	25.2	18.9	13.8	22.4	19.7	33.6	32.7	37.8	25.4
HC	28.0	16.7	19.4	28.8	26.8	11.0	29.2	25.8	23.2	0.0
	28.5	32.0	24.8	32.8	35.1	40.7	6.5	39.3	19.9	14.7
	15.8	26.4	20.4	14.1	23.2	20.5	39.1	34.9	49.8	27.0
SH	31.2	18.4	21.6	32.2	30.3	11.8	32.1	28.3	26.4	0.0
	29.9	35.0	28.3	36.1	40.0	47.0	6.8	44.4	21.7	16.7
MIO [29]										
	33.7	61.7	53.0	51.7	51.3	46.8	23.9	66.8	61.5	40.0
HC	46.8	48.6	52.9	43.6	70.0	61.7	32.6	33.9	53.4	57.6
	64.7	70.2	2.8	45.2	66.2	75.2	93.4	85.3	55.7	45.8
	47.2	79.3	71.2	68.9	67.3	53.1	33.7	79.6	79.8	60.8
SH	57.1	61.8	58.4	49.0	78.2	72.1	39.8	39.6	62.2	67.6
	81.6	81.9	3.9	54.4	83.8	90.9	93.7	89.4	61.6	64.7
UniMA										
	69.9	76.9	70.2	49.1	68.7	85.1	33.3	60.9	87.0	88.8
HC	88.0	67.3	48.0	74.9	68.7	65.0	77.1	62.1	88.7	56.3
	71.2	60.8	94.1	69.1	70.9	55.5	58.2	67.9	75.1	76.7
	69.6	78.9	77.1	53.7	71.4	89.0	45.3	63.8	89.1	91.6
SH	91.6	70.1	51.2	75.9	73.9	69.3	81.5	67.2	90.2	57.8
	72.4	63.6	94.7	74.9	79.3	68.6	64.2	76.1	77.6	79.0

Table 24. Detailed experiment results on *Interleaved Coherence*.

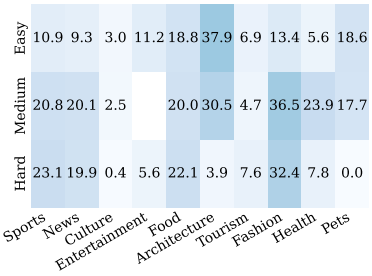
	Interleaved Coherence									
	#1	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**	**#8**	**#9**	**#10**
	#11	**#12**	**#13**	**#14**	**#15**	**#16**	**#17**	**#18**	**#19**	**#20**
	#21	**#22**	**#23**	**#24**	**#25**	**#26**	**#27**	**#28**	**#29**	**#30**
AnyGPT [40]										
	49.4	46.4	39.4	24.2	35.7	34.5	39.6	40.6	56.8	45.7
ICS^{abs}	67.6	36.8	58.8	8.9	37.2	62.0	12.1	29.7	5.9	2.0
	29.9	52.0	6.9	61.1	46.4	63.8	7.7	52.0	22.7	33.3
	49.3	36.9	38.9	23.7	33.9	34.3	18.7	37.5	54.1	44.9
ICS^{rel}	66.5	36.2	57.7	8.1	35.6	60.1	12.0	20.8	5.9	2.0
	28.9	51.3	6.9	24.4	44.4	63.6	7.7	50.0	22.4	25.8
NExT-GPT [33]										
	14.3	25.5	19.2	13.9	22.6	19.9	34.7	33.1	40.2	25.7
ICS^{abs}	28.7	17.1	19.8	29.5	27.4	11.2	29.8	26.2	23.8	0.0
	28.7	32.6	25.4	33.4	36.1	42.0	6.6	40.3	20.3	15.1
	8.1	15.4	16.9	5.1	15.6	19.3	15.6	16.6	37.7	12.5
ICS^{rel}	28.1	16.8	19.4	10.7	25.9	10.7	29.6	25.4	23.3	0.0
	9.7	32.1	19.0	13.4	27.6	34.4	6.6	29.9	16.7	10.8
MIO [29]										
	36.4	65.2	56.7	55.2	54.5	48.1	25.9	69.4	65.1	44.2
ICS^{abs}	48.9	51.3	54.0	44.6	71.6	63.8	34.1	35.0	55.2	59.6
	68.1	72.5	3.0	47.0	69.8	78.3	93.4	86.1	56.9	49.6
	20.7	39.3	49.5	20.3	28.3	46.8	11.6	34.8	61.1	5.4
ICS^{rel}	48.0	50.4	52.7	16.2	67.5	61.0	33.7	24.6	54.1	11.9
	23.1	71.3	2.3	18.8	53.2	64.2	93.1	45.6	46.8	38.4
UNIMA										
	69.8	77.3	71.6	50.0	69.2	85.9	35.7	61.5	87.4	89.3
ICS^{abs}	88.7	67.9	48.7	75.1	69.7	65.8	78.0	63.2	89.0	56.6
	71.4	61.3	94.2	70.3	72.6	58.1	59.4	69.6	75.6	77.2
	69.8	77.3	71.6	50.0	69.2	85.9	35.7	61.5	87.4	89.3
ICS^{rel}	88.7	67.9	48.7	75.1	69.7	65.8	78.0	63.2	89.0	56.6
	71.4	61.3	94.2	70.3	72.6	58.1	59.4	69.6	75.6	77.2



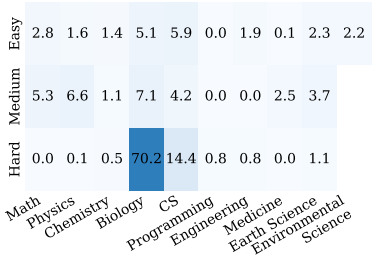
AnyGPT & natural science.



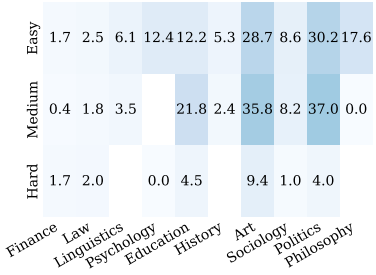
AnyGPT & social science.



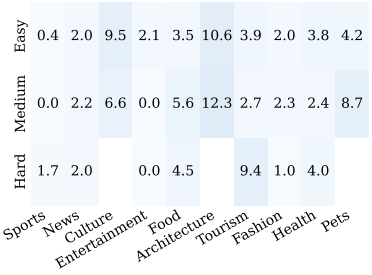
AnyGPT & general area.



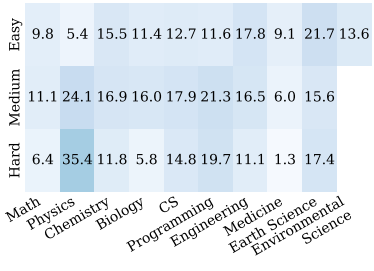
NExT-GPT & natural science.



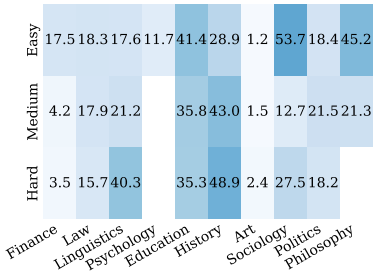
NExT-GPT & social science.



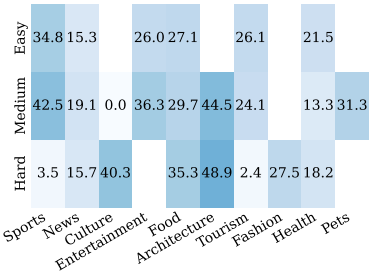
NExT-GPT & general area.



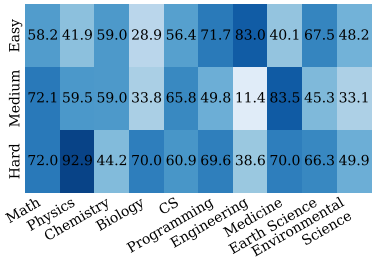
MIO & natural science.



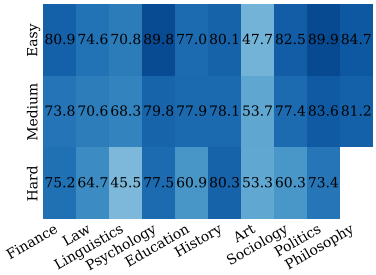
MIO & social science.



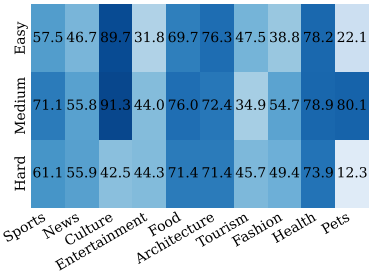
MIO & general area.



UniMA & natural science.

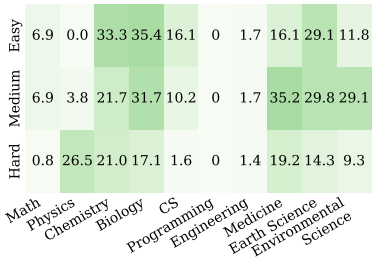


UniMA & social science.

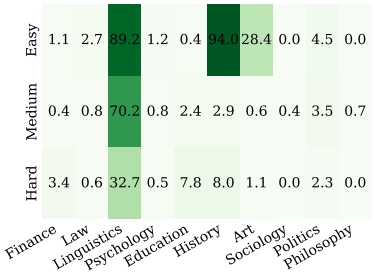


UniMA & general area.

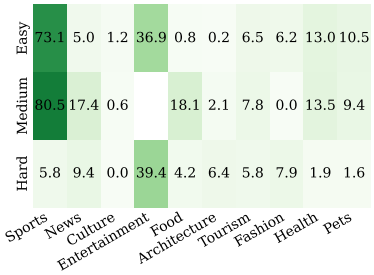
Figure 38. SQCS performance across baseline models, difficulty levels and domains.



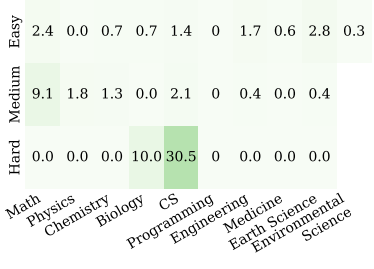
AnyGPT & natural science.



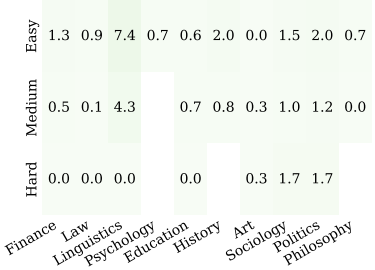
AnyGPT & social science.



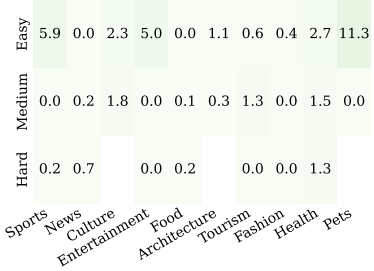
AnyGPT & general area.



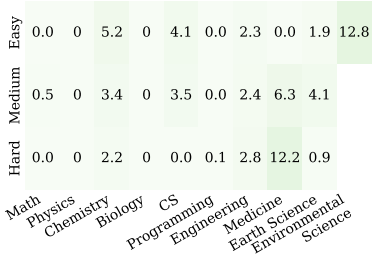
NExT-GPT & natural science.



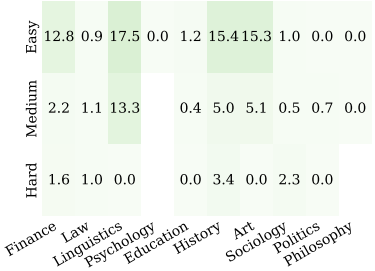
NExT-GPT & social science.



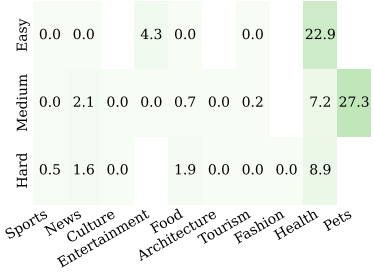
NExT-GPT & general area.



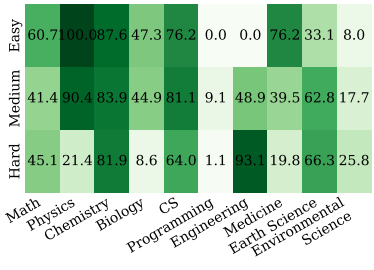
MIO & natural science.



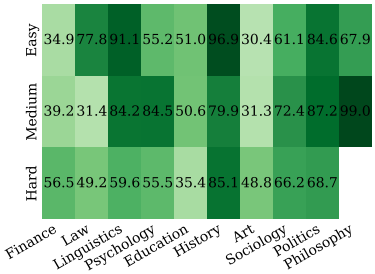
MIO & social science.



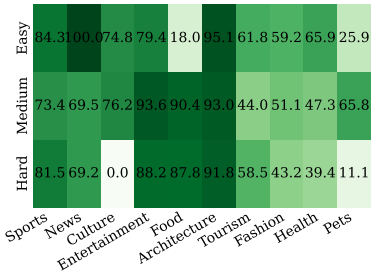
MIO & general area.



UniMA & natural science.

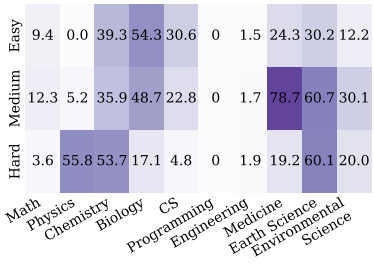


UniMA & social science.

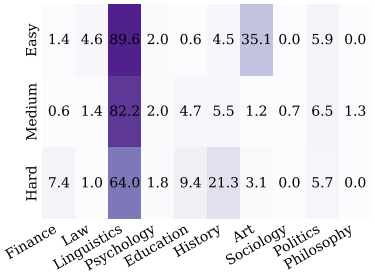


UniMA & general area.

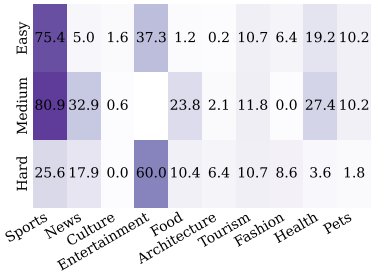
Figure 39. StS performance across baseline models, difficulty levels and domains.



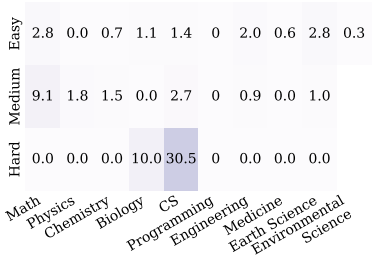
AnyGPT & natural science.



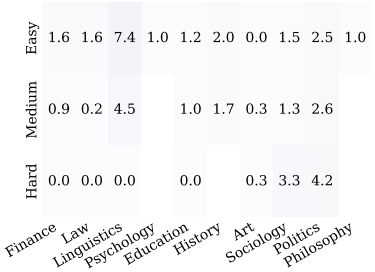
AnyGPT & social science.



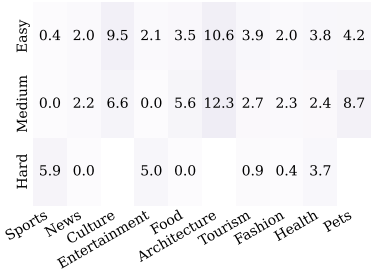
AnyGPT & general area.



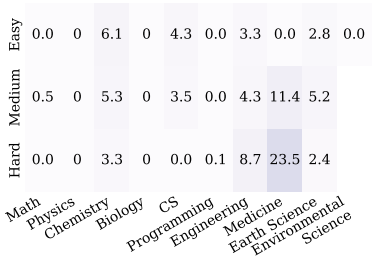
NExT-GPT & natural science.



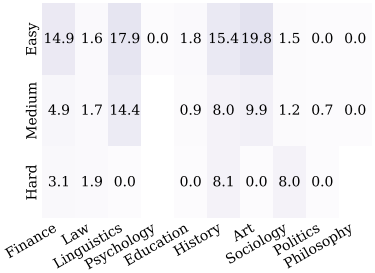
NExT-GPT & social science.



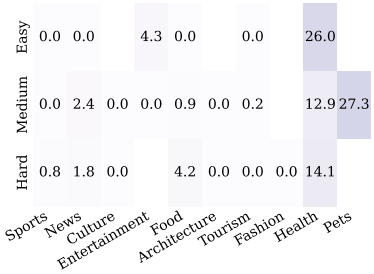
NExT-GPT & general area.



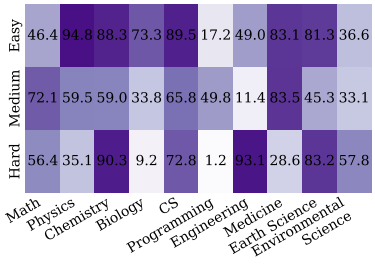
MIO & natural science.



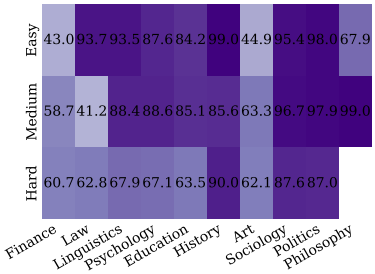
MIO & social science.



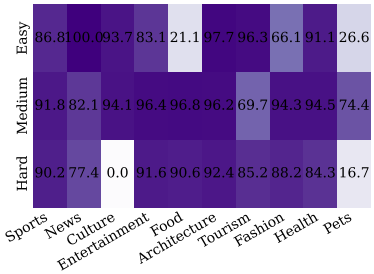
MIO & general area.



UniMA & natural science.

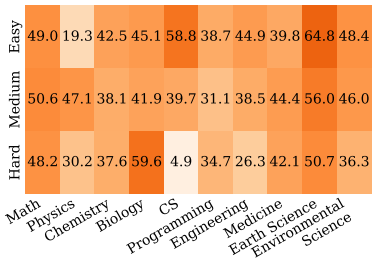


UniMA & social science.

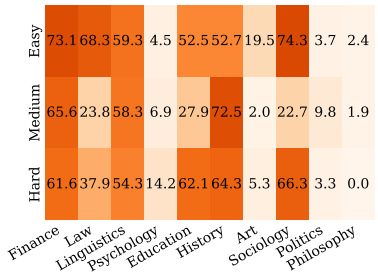


UniMA & general area.

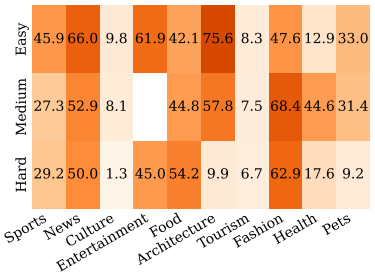
Figure 40. LeS performance across baseline models, difficulty levels and domains.



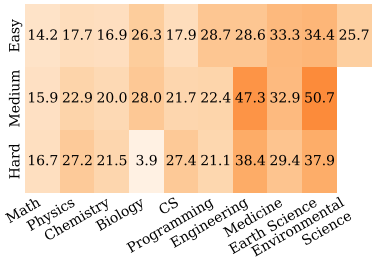
AnyGPT & natural science.



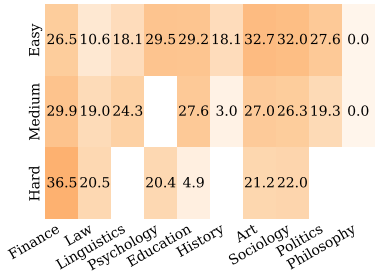
AnyGPT & social science.



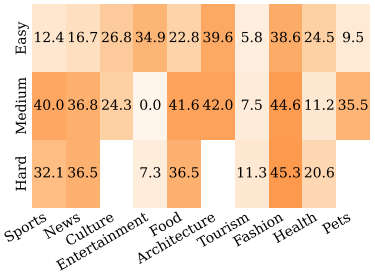
AnyGPT & general area.



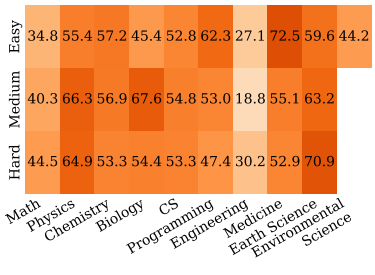
NExT-GPT & natural science.



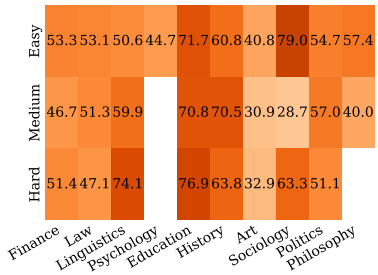
NExT-GPT & social science.



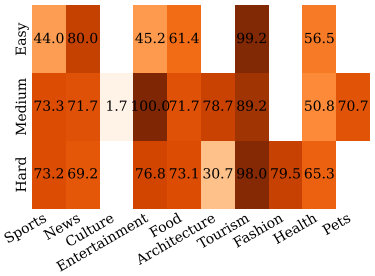
NExT-GPT & general area.



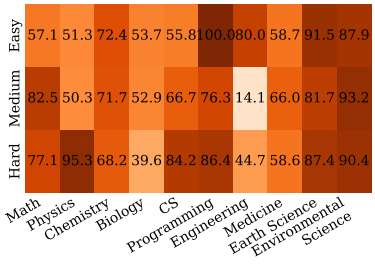
MIO & natural science.



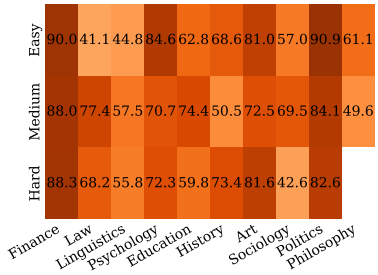
MIO & social science.



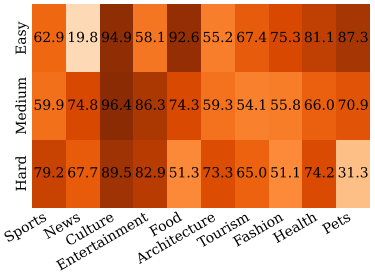
MIO & general area.



UniMA & natural science.



UniMA & social science.



UniMA & general area.

Figure 41. ICS performance across baseline models, difficulty levels and domains.

Text + Image + Audio → Text + Image + Audio

▶ **User Query:** Observe the *<image1>* where you can see construction workers and machinery. Listen to *<audio1>* for a description of the scene. Please provide an explanation of potential hazards at this construction site and ensure the answer includes one audio and one image.

<image1>

<audio2>

▶ **UniMA Response:** The photo shows an outdoor construction area with an excavator…three workers…; the supplied audio similarly notes…excavator on a mound…vegetation behind. Key hazards: Excavator swing radius…risk of being struck. Worker proximity…inside the exclusion zone… Uneven, loose terrain…risk of slips or instability. Scattered materials…trip/obstruction hazards. Lack of barriers…no visible barricades… *<image2>* Practical recommendations: establish a fenced exclusion zone…; assign a spotter…; stabilize loose soil…; clear debris…; install barricades/signage…; apply dust/hearing protection… A concise narrated summary…: *<audio2>*

<image2>

<audio2>

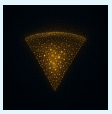
Figure 42. Case illustrating the responses produced by UNIMA to T+I+A as input.

Text + 3D → Text + Image + Document

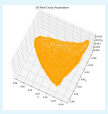
▶ **User Query:** Using the 3D mesh data provided by *<3D1>*, can you identify the type of food and analyze its characteristics?

<3D1>

▶ **UniMA Response:** Based on the provided 3D mesh data—including geometry, material, and texture mapping—the food item can be identified as a slice of pizza. The triangular fan-like outline, thin thick-ness profile, and golden–brown baked cheese texture all match a typical pizza slice. The following image shows a visualization from t-he point-cloud structure: *<image1>* The next figure is the top-down po-int-cloud projection … revealing a fan-shaped contour: *<document1>* Combining geometric shape, surface texture, and thickness distribution, the 3D object can be confidently identifyed as a baked thin-crust pizza slice with melted-and-browned cheese.



<image1>



<document1>

Figure 43. Case illustrating the responses produced by UNIMA to T+3D as input.

Text + Video + Audio → Text + Document

▶ **User Query:** Based on the explanations given in *<video1>* and *<audio1>*, create a summary document.

<video1>

<audio1>

▶ **UniMA Response:** I have already created a document to summarize the video and audio, the *<document1>* explains the core idea of cloning a GitHub repository. It highlights the difference between remote and local work, shows how a repository can be cloned using various tools, and c-larifies how developers manually synchronize changes between their local environment and GitHub. It serves as a quick reference for understand-ing the local-workflow fundamentals in Git.

<document1>

Figure 44. Case illustrating the responses produced by UNIMA to T+V+A as input.

Text + Image + Audio + Document → Text + Document

▶ **User Query:** Guided by <image1> and <document1>, act according to the directions in <audio1>, and generate text and a corresponding image to support your answer.

<image1>

<document1>

<audio1>

▶ **UniMA Response:** The photograph shows teachers and supporters holding red "ON STRIKE" and "Beverly Educators On Strike" placards … symbolizing organized labor action. The Boston.com article and audio instructions describe a class-action lawsuit filed by parents after the 12-day 2023 Beverly teachers' strike…alleging the union encouraged and sustained the strike through strike-fund resources, causing educational and emotional harm. Attorneys Ilya Hoekstetter and Daniel Suhar represent the parents, seeking about $17 million…focused on whether union-backed actions unlawfully prolonged the stoppage and caused measurable harm. Concise takeaway: the document and article show tension between educators' labor rights and community impacts…a prolonged strike becoming a major legal challenge over accountability and compensation. <document2>

<document2>

Figure 45. Case illustrating the responses produced by UniMA to T+I+A+D as input.

Text + Video + Audio + Document → Text + Audio

▶ **User Query:** How do the women on the left react to the woman on the right, considering both auditory cues from the conversation <audio1> and visual elements from their facial expressions <video1>? Examine their interactions and exp-lore supporting content from various topics <document1> they potentially discussed. You need to generate an audio that include your answer.

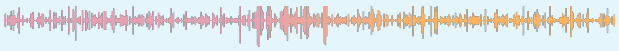
<video1>

<document1>

statement0	They are indifferent because it is not a major problem.
statement1	They show surprise and worry because she reports sensations they do not feel.
statement2	They are discussing their favorite skincare products.
statement3	They are talking about their favorite TV shows.

<audio1>

▶ **UniMA Response:** The women on the left react with surprise and genuine worry to the woman on the right. Evidence and reasoning - Auditory cues: the woman on the right reports sudden skin sensations…; the others ask "Are you okay ?" "Does it hurt?" showing concern. Visual cues: widened eyes, furrowed brows, leaning-in posture … signaling alarm and empathy. Document support: the table image's statement1…matches the interaction; other statements…do not align with the audio–visual evidence. Summary conclusion: both spoken content and reactions indicate they are surprised and worried about her unexpected physical reaction. <audio2>



<audio2>

Figure 46. Case illustrating the responses produced by UniMA to T+V+A+D as input.

Text + Image + Video + Audio → Text + Video + Audio

▶ **User Query:** In the video <video1> and its accompanying audio <audio1>, the University of Oxford's excellence in social sciences research is emphasized, visually supported by the image <image1>. How do Oxford's diverse programs prepare students for leadership roles and impactful careers in various sectors, as outlined in <audio2>? The answer must include a video to support or illustrate the explanation.

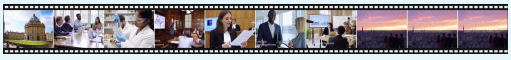
<image1>

<audio2>

<video1>

<audio1>

▶ **UniMA Response:** Through <video2> and <audio3>, the strengths of Oxford's social science programs are displayed—showing mentorship, interdisciplinary collaboration, and community engagement. It demonstrates how academic inquiry translates into leadership and problem-solving across public, private, and non-profit sectors.



<video2>

<audio3>

Figure 47. Case illustrating the responses produced by UniMA to T+I+V+A as input.

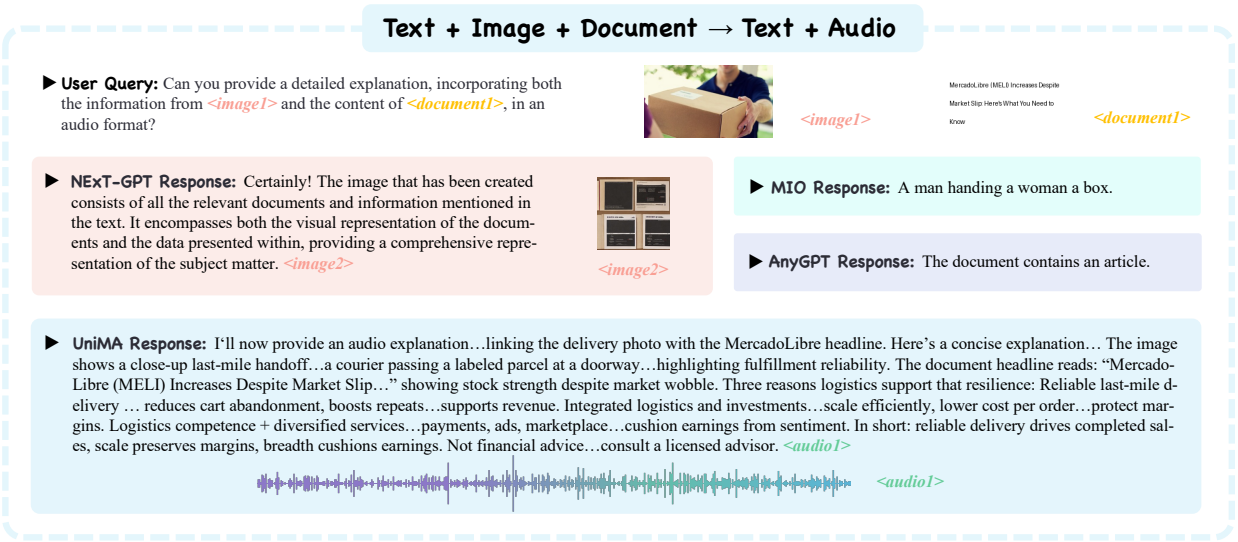


Figure 48. Case illustrating the responses produced by AnyGPT, NExT-GPT, MIO and UniMA to T+I+D as input.

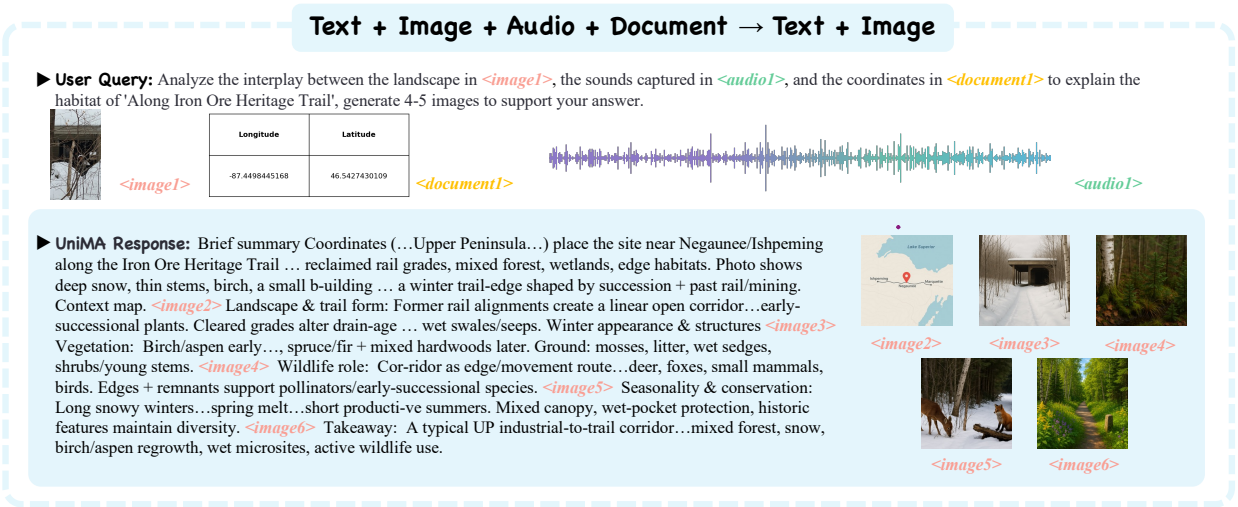
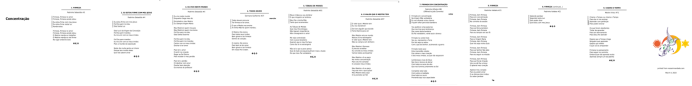


Figure 49. Case illustrating the responses produced by UniMA to T+I+A+D as input.

Text + Document → Text + Image + Audio

▶ **User Query:** Using the information found in *<document1>*, explain the impact of technology on modern educational practices. The answer should include an audio summary of key technological advancements and an image illustrating a virtual classroom setting, along with detailed explanation.



<document1>

▶ **UniMA Response:** The uploaded document outlines…virtual classrooms, technological tools, advantages/disadvantages, and evaluation … showing how technology reshapes modern education. Technology has shifted learning toward interactive, student-centered, data-driven models. Virtual classrooms integrate live video, whiteboards, chats, and LMS dashboards…enhancing engagement and flexibility. *<image1>* Caption (image1): a modern virtual classroom interface…illustrating connected learning. Key impacts include: pedagogy—multimedia or simulations support active learning; assessment—LMS analytics give instant feedback…ethics/validity concerns; access—online tools expand reach…digital divides; collaboration—chats/shared documents foster teamwork; challenges—screen overreliance and privacy issues. To summarize…this narration highlights major advances and their influence on teaching, assessment, and inclusion. *<audio1>*

<image1>

<audio1>

▶ **AnyGPT Response:** : The evolution of technology has transformed learning environments, emphasizing flexibility, accessibility, and personalized learning. *<image1>*

<image1>

▶ **NExT-GPT Response:** OK, I will provide an audio clip for you.

▶ **MIO Response:** This is an audio summary of key technological advancements and an image illustrating a virtual classroom setting.

Figure 50. Case illustrating the responses produced by AnyGPT, NExT-GPT, MIO and UniMA to T+D as input.

Text + Code → Text + Code

▶ **User Query:** Please analyze the *<code1>* code snippet provided and identify the reasons for its failure. Also, the correct solution is required in the response.

```
class Solution:
    def fib(self, n: int) -> int:
        def dfs(k):
            if memo[k] != -1:
                return memo[k]
            memo[k] = dfs(k - 1) + dfs(k - 2)
            return memo[k]
        if n == 0:
            return 0
        if n == 1:
            return 1
        memo = [-1] * (n + 1)
        memo[0] = 0
        memo[1] = 1
        return dfs(n)
```

<code1>

▶ **UniMA Response:** The code fails due to a **SyntaxError** at the line `def fib(n:`—the missing parenthesis prevents Python from parsing the file at all. This is the sole error that makes the program fail before execution. Other issues such as naming the inner function `fib` (shadowing) and using uppercase variable names for mutable lists reduce readability but do not cause runtime errors.\n\nAfter fixing the syntax, the memoized recursion logic is fully valid. Two correct solutions are provided: (1) a repaired top-down memoization version faithful to the original intent, and (2) a recommended iterative DP version that runs in O(1) space. *<code2>* is the correct version of the code

```
class Solution:
    def fib(self, n: int) -> int:
        def fib_helper(k: int) -> int:
            if cache[k] != -1:
                return cache[k]
            cache[k] = fib_helper(k - 1) + fib_helper(k - 2)
            return cache[k]
        if n <= 1:
            return n
        cache = [-1] * (n + 1)
        cache[0] = 0
        cache[1] = 1
        return fib_helper(n)
```

<code2>

▶ **MIO Response:** Python

```
def fib(n: int) -> int:
    if n == 0:
        return 0
    elif n == 1:
        return 1
    else:
        return fib(n - 1) + fib(n - 2)
```

▶ **AnyGPT Response:** The code snippet is incomplete and lacks proper indentation. *<audio1>*

<audio1>

▶ **NExT-GPT Response:** The given recursive Fibonacci code is intended to compute Fib(n), but it fails because: The name Fib is referenced before being defined. There is no function or list named Fib in the current scope. Because of this undefined reference, the program will crash with a NameError. To fix this issue, the code must define a memoization table (commonly a list) or rename the variable properly.

Figure 51. Case illustrating the responses produced by AnyGPT, NExT-GPT, MIO and UniMA to T+C as input.

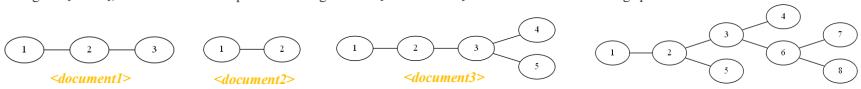
Text + Image + Audio + Document → Text + Code

▶ **User Query:** We have a tree with N vertices numbered 1 to N. The i-th edge in this tree connects Vertex a_i and Vertex b_i. Consider painting each of these edges white or black. There are 2^{N-1} such ways to paint the edges. Among them, how many satisfy all of the following M restrictions?
- The i-th $(1 \leq i \leq M)$ restriction is represented by two integers u_i and v_i, which mean that the path connecting Vertex u_i and Vertex v_i must contain at least one edge painted black.

Constraints
- $2 \leq N \leq 50$
- $1 \leq a_i, b_i \leq N$
- The graph given in input is a tree.
- $1 \leq M \leq \min\left(20, \frac{N(N-1)}{2}\right)$
- $1 \leq u_i < v_i \leq N$
- If $i \neg= j$, *either* $u\_i \neg= u\_j$ *or* $v\_i \neg= v\_j$
- All values in input are integers.

Input: Input is given from Standard Input in the following format:

$$N$$
$$a_1\ b_1$$
$$\vdots$$
$$a_{N-1}\ b_{N-1}$$
$$M$$
$$u_1\ v_1$$
$$\vdots$$
$$u_M\ v_M$$



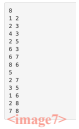
Sample Input 1


Sample Output 1


Sample Input 2


Sample Output 2


Sample Input 3


Sample Input 4


Sample Output 3

Sample Output 4


Output:Print the number of ways to paint the edges that satisfy all of the M conditions. Sample Input 1 is shown in *<image1>*. Sample Output 1 is shown in *<image2>*. The tree in this input is shown below in *<document1>*. All of the M restrictions will be satisfied if Edge 1 and 2 are respectively painted (white, black), (black, white), or (black, black), so the answer is 3. Sample Input 2 is shown in *<image3>*. Sample Output is shown in *<image4>*. The tree in this input is shown below in *<document2>*. All of the M restrictions will be satisfied only if Edge 1 is painted black, so the answer is 1. Sample Input is shown in *<image5>*. Sample Output 3 is shown in *<image6>*. The tree in this input is shown below in *<document3>*. Sample Input 4 is shown in *<image7>*. Sample Output 4 is shown in *<image8>*. The tree in this input is shown below in *<document4>*. The time and memory limits are shown in *<audio1>*. Please give the right Cpp code.

▶ **UniMA Response:** Solution idea (brief):\n- Use the Inclusion–Exclusion Principle over the M constraints.\n- For each constraint (u, v), find the set of edge indices on the unique path u–v and store it as a 64-bit bitmask (N-1 ≤ 49 so fits).\n- For each subset of constraints, take the union of their bitmasks, count fixed edges k, and add/subtract 2^{(N-1)-k} according to subset parity.\n- Complexity: O(M * 2^M + M * (N)) for path computations and subset enumeration; fits the limits. The correct C++ solution has been placed into *<code2>*. This program follows the inclusion–exclusion approach, uses BFS to reconstruct each u–v path, encodes edges in 64-bit masks, enumerates subsets of constraints (M ≤ 20), and prints the correct count.

```cpp
#include <bits/stdc++.h>
using namespace std;

int main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);

    int N;
    if (!(cin >> N)) return 0;
    int E = N - 1;
    vector<vector<pair<int,int>>> g(N+1);
    for (int i = 0; i < E; ++i) {
        int a, b; cin >> a >> b;
        g[a].push_back({b, i});
        g[b].push_back({a, i});
    }
    int M; cin >> M;
    vector<pair<int,int>> qs(M);
    for (int i = 0; i < M; ++i) cin >> qs[i].first >> qs[i].second;

    vector<unsigned long long> pathMask(M, 0ULL);
    for (int qi = 0; qi < M; ++qi) {
        int s = qs[qi].first;
        int t = qs[qi].second;
        vector<int> parent(N+1, -1), parentEdge(N+1, -1);
        queue<int> q;
        q.push(s);
        parent[s] = s;
        while (!q.empty()) {
            int u = q.front(); q.pop();
            if (u == t) break;
            for (auto [v, eid] : g[u]) {
                if (parent[v] == -1) {
                    parent[v] = u;
                    parentEdge[v] = eid;
                    q.push(v);
                }
            }
        }
        unsigned long long mask = 0ULL;
        int cur = t;
        while (cur != s) {
            int e = parentEdge[cur];
            mask |= (1ULL << e);
            cur = parent[cur];
        }
        pathMask[qi] = mask;
    }

    long long ans = 0;
    int totalSub = 1 << M;
    for (int s = 0; s < totalSub; ++s) {
        unsigned long long unionMask = 0ULL;
        int bits = 0;
        for (int i = 0; i < M; ++i)
            if (s & (1 << i)) {
                unionMask |= pathMask[i];
                ++bits;
            }
        int fixedEdges = __builtin_popcountll(unionMask);
        int freeEdges = E - fixedEdges;
        long long contribution = 0;
        if (freeEdges >= 0) contribution = (1LL << freeEdges);
        if (bits % 2 == 1) ans -= contribution;
        else ans += contribution;
    }

    cout << ans << '\n';
    return 0;
}
```

Figure 52. Case illustrating the responses produced by AnyGPT, NExT-GPT, MIO and UniMA to T+I+A+D as input.

Text + Image → Text + Code

▶ **User Query:** Please review the *<image1>* to assist in constructing HTML code that mirrors the webpage depicted. Ensure the code aligns with material design principles.

<image1>

▶ **UniMA Response:** The HTML layout has been updated and inserted into *<code1>*.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <title>About Us Layout</title>
  <style>
    * {
      box-sizing: border-box;
    }
    body {
      margin: 0;
      font-family: system-ui, -apple-system,
      'Segoe UI', Roboto, Arial, sans-serif;
      color: #222;
      background-color: #ffffff;
    }
    .page {
      display: flex;
      min-height: 100vh;
    }
    .sidebar {
      width: 32%;
      padding: 40px 32px;
      background-color: #f3f3f3;
    }.sidebar h1 {
      margin: 0 0 24px;
      font-size: 32px;
      font-weight: 700;
    }
    .sidebar ul {
      margin: 0;
      padding-left: 28px;
      font-size: 20px;
      line-height: 1.6;
    }
    .sidebar li {
      margin-bottom: 4px;
    }
    .content {
      flex: 1;
      padding: 40px 48px;
    }
    .content h1 {
      margin: 0 0 16px;
      font-size: 32px;
      font-weight: 700;
    }
    .content h2 {
      margin: 32px 0 8px;
      font-size: 28px;
      font-weight: 700;
    }
    .content p {
      margin: 0 0 8px;
      font-size: 20px;
      line-height: 1.5;
    }
    .logo-placeholder {
      width: 260px;
      height: 260px;
      background-color: #b3b3b3;
      margin-top: 4px;
      margin-bottom: 24px;
    }
    @media (min-width: 1200px) {
      .page {
        max-width: 1400px;
        margin: 0 auto;
      }
    }
  </style>
</head>
<body>
  <div class="page">
    <aside class="sidebar">
      <h1>Product Categories</h1>
      <ul>
        <li>Makeup</li>
        <li>Skincare</li>
        <li>Hair care</li>
        <li>Fragrance</li>
      </ul>
    </aside>
    <main class="content">
      <h1>About Us</h1>
      <div class="logo-placeholder" aria-label
      ="Company Logo"></div>
      <p>Our company is dedicated to providing the highest
      quality beauty products to all women around the world.</p>
      <h2>Core Values</h2>
      <p>At our core, we believe in sustainability and strive for ethical
      working conditions for our suppliers and employees.</p>
      <h2>Commitment to Sustainability</h2>
      <p>We are committed to using sustainable materials for all of
      our products and reducing our carbon footprint as much as possible.</p>
    </main>
  </div>
</body>
</html>
```

<code1>

▶ **MIO Response:**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=
  device-width, initial-scale=1.0">
  <title>Products</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container">
    <div class="row">
      <div class="col-md-6">
        <h2>Products</h2>
        <ul>
          <li><a href="#">Product 1</a></li>
          <li><a href="#">Product 2</a></li>
          <li><a href="#">Product 3</a></li>
          <li><a href="#">Product 4</a></li>
        </ul>
      </div>
    </div>
  </div>
</body>
</html>
```

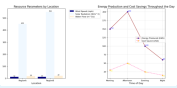
▶ **NExT-GPT Response:** The layout displayed in the image appears to be a simple and clean design with a focus on white space and simplicity.

▶ **AnyGPT Response:** *<image2>*

<image2>

Figure 53. Case illustrating the responses produced by AnyGPT, NExT-GPT, MIO and UNIMA to T+I as input.

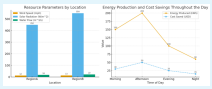
Text + Document → Text + Code



▶ **User Query:** As an expert Python developer, you are required to generate matplotlib code based on a given picture. Please analyze the *<document1>* and *<document1>* provide the relevant Python code to replicate the image using matplotlib. Ensure you include the use of figsize to set the image size.

▶ **UniMA Response:** he provided matplotlib script *<code2>* recreates the figure shown in *<image1>*. It defines a 1x2 layout using `plt.subplots(1, 2, figsize=(12, 5))`. The left subplot draws a grouped bar chart for Wind Speed, Solar Radiation, and Water Flow at two locations, including numeric labels above each bar and a legend. The right subplot plots two lines graphs for Energy Produced and Cost Saved throughout the day, with markers, dashed style for the second line, value labels at each point, a grid, axis labels, and a legend. The overall layout, titles, and data closely replicate the original image.

```
import numpy as np
import matplotlib.pyplot as plt
locations = ['RegionA', 'RegionB']
wind_speed = [12, 13]
solar_radiation = [450, 550]
water_flow = [18, 22]
x = np.arange(len(locations))
bar_width = 0.25
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 5))
b1 = ax1.bar(
    x - bar_width, wind_speed,
    width=bar_width, label='Wind Speed (mph)',
    color='navy'
)
```



<image1>

<code2>

```
b2 = ax1.bar(
    x, solar_radiation,
    width=bar_width, label='Solar Radiation
(W/m^2)',
    color='#e6f2ff'
)
b3 = ax1.bar(
    x + bar_width, water_flow,
    width=bar_width, label='Water Flow
(m^3/s)',
    color='#f4c27a'
)
ax1.set_title('Resource Parameters by
Location')
ax1.set_xlabel('Location')
ax1.set_xticks(x)
ax1.set_xticklabels(locations)
for bars in (b1, b2, b3):
    for rect in bars:
        height = rect.get_height()
        ax1.text(
            rect.get_x() + rect.get_width() / 2.0,
            height + 5,
            f'{int(height)}',
            ha='center', va='bottom', fontsize=8
        )
ax1.legend(loc='upper right')
times = ['Morning', 'Afternoon', 'Evening', 'Night']
energy_produced = [150, 200, 100, 60]
cost_saved = [30, 50, 25, 15]
ax2.plot(
    times, energy_produced,
    marker='o', color='brown',
    label='Energy Produced (kWh)'
)
ax2.plot(
    times, cost_saved,
    marker='s', linestyle='--', color='magenta',
    label='Cost Saved (USD)'
)
for t, e in zip(times, energy_produced):
    ax2.text(t, e + 3, f'{e}', ha='center', va='bottom', fontsize=8)
for t, c in zip(times, cost_saved):
    ax2.text(t, c + 3, f'{c}', ha='center', va='bottom', fontsize=8)
ax2.set_title('Energy Production and Cost Savings Throughout the Day')
ax2.set_xlabel('Time of Day')
ax2.set_ylabel('Value')
ax2.grid(True, linestyle='--', alpha=0.5)
ax2.legend(loc='upper right')
plt.tight_layout()
plt.show()
```

▶ **MIO Response:**

```
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
import matplotlib.colors as colors
from matplotlib.patches import Line2D
from matplotlib.lines import Line
from matplotlib.text import Text
from sklearn.cluster import KMeans
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import cross_val_score
from sklearn.model_selection
```

▶ **NExT-GPT Response:** I'm sorry.

▶ **AnyGPT Response:** : *<image>*

<image>

References

[1] Ethan Bradley, Muhammad Roman, Karen Rafferty, and Barry Devereux. Synfintabs: a dataset of synthetic financial tables for information and table extraction. *arXiv preprint arXiv:2412.04262*, 2024. 4

[2] Yuhao Chen, Jiangpeng He, Gautham Vinod, Siddeshwar Raghavan, Chris Czarnecki, Jinge Ma, Talha Ibn Mahmud, Bruce Coburn, Dayou Mao, Saeejith Nair, et al. Metafood3d: 3d food dataset with nutrition values. *arXiv preprint arXiv:2409.01966*, 2024. 4

[3] Ziming Cheng, Binrui Xu, Lisheng Gong, Zuhe Song, Tianshuo Zhou, Shiqi Zhong, Siyu Ren, Mingxiang Chen, Xiangchao Meng, Yuxin Zhang, et al. Evaluating mllms with multimodal multi-image reasoning benchmark. *arXiv preprint arXiv:2506.04280*, 2025. 4

[4] Hao Fei, Yuan Zhou, Juncheng Li, Xiangtai Li, Qingshan Xu, Bobo Li, Shengqiong Wu, Yaoting Wang, Junbao Zhou, Jiahao Meng, et al. On path to multimodal generalist: General-level and general-bench. In *Proceedings of the ICML*, 2025. 4

[5] Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multimodal llms in video analysis. In *Proceedings of the CVPR*, pages 24108–24118, 2025. 4

[6] Jiawei Guo, Ziming Li, Xueling Liu, Kaijing Ma, Tianyu Zheng, Zhouliang Yu, Ding Pan, Yizhi Li, Ruibo Liu, Yue Wang, et al. Codeeditorbench: Evaluating code editing capability of large language models. *arXiv preprint arXiv:2404.03543*, 2024. 4

[7] Sheng Guo, Weilin Huang, Xiao Zhang, Prasanna Srikhanta, Yin Cui, Yuan Li, Matthew R. Scott, Hartwig Adam, and Serge Belongie. The imaterialist fashion attribute dataset. In *Proceedings of the ICCVW*, pages 3113–3116, 2019. 4

[8] Mourad Heddaya, Kyle MacMillan, Hongyuan Mei, Chenhao Tan, and Anup Malani. Casesumm: a large-scale dataset for long-context summarization from us supreme court opinions. In *Proceedings of the NAACL*, pages 1917–1942, 2025. 4

[9] Jon M Laurent, Joseph D Janizek, Michael Ruzo, Michaela M Hinks, Michael J Hammerling, Siddharth Narayanan, Manvitha Ponnapati, Andrew D White, and Samuel G Rodriques. Lab-bench: Measuring capabilities of language models for biology research. *arXiv preprint arXiv:2407.10362*, 2024. 4

[10] Kaixin Li, Yuchen Tian, Qisheng Hu, Ziyang Luo, Zhiyong Huang, and Jing Ma. Mmcode: Benchmarking multimodal large language models for code generation with visually rich programming problems. *arXiv preprint arXiv:2404.09486*, 2024. 4

[11] Kaixin Li, Ziyang Meng, Hongzhan Lin, Ziyang Luo, Yuchen Tian, Jing Ma, Zhiyong Huang, and Tat-Seng Chua. Screenspot-pro: Gui grounding for professional high-resolution computer use. In *Proceedings of the ACM MM*, pages 8778–8786, 2025. 4

[12] Qingyun Li, Zhe Chen, Weiyun Wang, Wenhai Wang, Shenglong Ye, Zhenjiang Jin, Guanzhou Chen, Yinan He, Zhangwei Gao, Erfei Cui, et al. Omnicorpus: A unified multimodal corpus of 10 billion-level images interleaved with text. *arXiv preprint arXiv:2406.08418*, 2024. 4

[13] Wenyan Li, Crystina Zhang, Jiaang Li, Qiwei Peng, Raphael Tang, Li Zhou, Weijia Zhang, Guimin Hu, Yifei Yuan, Anders Søgaard, et al. Foodieqa: A multimodal dataset for fine-grained understanding of chinese food culture. In *Proceedings of the EMNLP*, pages 19077–19095, 2024. 4

[14] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of the CVPR*, pages 1096–1104, 2016. 4

[15] Meng Luo, Hao Fei, Bobo Li, Shengqiong Wu, Qian Liu, Soujanya Poria, Erik Cambria, Mong-Li Lee, and Wynne Hsu. Panosent: A panoptic sextuple extraction benchmark for multimodal conversational aspect-based sentiment analysis. In *Proceedings of the ACM MM*, pages 7667–7676, 2024. 4

[16] Yinghao Ma, Siyou Li, Juntao Yu, Emmanouil Benetos, and Akira Maezawa. Cmi-bench: A comprehensive benchmark for evaluating music instruction following. *arXiv preprint arXiv:2506.12285*, 2025. 4

[17] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the CVPR*, pages 2630–2640, 2019. 4

[18] Adrian Mirza, Nawaf Alampara, Martiño Ríos-García, Mohamed Abdelalim, Jack Butler, Bethany Connolly, Tunca Dogan, Marianna Nezhurina, Bünyamin Şen, Santosh Tirunagari, et al. Chempile: A 250gb diverse and curated dataset for chemical foundation models. *arXiv preprint arXiv:2505.12534*, 2025. 4

[19] OpenAI. Model: gpt-image-1, 2025. 17

[20] OpenAI. Introducing gpt-5, 2025. 5, 15

[21] OpenAI. Sora 2 is here, 2025. 17

[22] Karol J. Piczak. ESC: Dataset for Environmental Sound Classification. In *Proceedings of the ACM MM*, pages 1015–1018, 2015. 4

[23] Colin B Price and Ruth Price-Mohr. Physlab: a 3d virtual physics laboratory of simulated experiments for advanced physics learning. *Physics Education*, 2019. 4

[24] Srikumar Sastry, Subash Khanal, Aayush Dhakal, Adeel Ahmad, and Nathan Jacobs. Taxabind: A unified embedding space for ecological applications. In *Proceedings of the WACV*, pages 1765–1774, 2025. 4

[25] Haojun Shi, Suyu Ye, Xinyu Fang, Chuanyang Jin, Leyla Isik, Yen-Ling Kuo, and Tianmin Shu. Muma-tom: Multi-modal multi-agent theory of mind. In *Proceedings of the AAAI*, pages 1510–1519, 2025. 4

[26] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. Moviechat: From

dense token to sparse memory for long video understanding. In *Proceedings of the CVPR*, pages 18221–18232, 2024. 4

[27] Qwen Team. Qwen3-vl-30b-a3b-instruct model card, 2025. 15

[28] Fengxiang Wang, Mingshuo Chen, Xuming He, YiFan Zhang, Feng Liu, Zijie Guo, Zhenghao Hu, Jiong Wang, Jingyi Xu, Zhangrui Li, et al. Omniearth-bench: Towards holistic evaluation of earth's six spheres and cross-spheres interactions with multimodal observational earth data. *arXiv preprint arXiv:2505.23522*, 2025. 4

[29] Zekun Wang, King Zhu, Chunpu Xu, Wangchunshu Zhou, Jiaheng Liu, Yibo Zhang, Jiashuo Wang, Ning Shi, Siyu Li, Yizhi Li, et al. Mio: A foundation model on multimodal tokens. *arXiv preprint arXiv:2409.17692*, 2024. 17, 42, 43, 44, 45, 46, 47

[30] Ziting Wang, Shize Zhang, Haitao Yuan, Jinwei Zhu, Shifu Li, Wei Dong, and Gao Cong. Fdabench: A benchmark for data agents on analytical queries over heterogeneous data. *arXiv preprint arXiv:2509.02473*, 2025. 4

[31] Guangshun Wei, Yuan Feng, Long Ma, Chen Wang, Yuanfeng Zhou, and Changjian Li. Pcdreamer: Point cloud completion through multi-view diffusion priors. In *Proceedings of the CVPR*, pages 27243–27253, 2025. 17

[32] Haoning Wu, Erli Zhang, Liang Liao, Chaofeng Chen, Jingwen Hou, Annan Wang, Wenxiu Sun, Qiong Yan, and Weisi Lin. Exploring video quality assessment on user generated contents from aesthetic and technical perspectives. In *Proceedings of the ICCV*, pages 20144–20154, 2023. 10

[33] Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. NExT-GPT: Any-to-any multimodal LLM. In *Proceedings of the ICML*, pages 53366–53397, 2024. 17, 42, 43, 44, 45, 46, 47

[34] Zongru Wu, Rui Mao, Zhiyuan Tian, Pengzhou Cheng, Tianjie Ju, Zheng Wu, Lingzhong Dong, Haiyue Sheng, Zhuosheng Zhang, and Gongshen Liu. See, think, act: Teaching multimodal agents to effectively interact with gui by identifying toggles. *arXiv preprint arXiv:2509.13615*, 2025. 4

[35] Yunfei Xie, Ce Zhou, Lang Gao, Juncheng Wu, Xianhang Li, Hong-Yu Zhou, Sheng Liu, Lei Xing, James Zou, Cihang Xie, et al. Medtrinity-25m: A large-scale multimodal dataset with multigranular annotations for medicine. *arXiv preprint arXiv:2408.02900*, 2024. 4

[36] Jundong Xu, Hao Fei, Yuhui Zhang, Liangming Pan, Qijun Huang, Qian Liu, Preslav Nakov, Min-Yen Kan, William Yang Wang, Mong-Li Lee, et al. Muslr: Multimodal symbolic logical reasoning. *arXiv preprint arXiv:2509.25851*, 2025. 4

[37] Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, et al. Qwen3-omni technical report. *arXiv preprint arXiv:2509.17765*, 2025. 15, 17

[38] Runsen Xu, Xiaolong Wang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. Pointllm: Empowering large language models to understand point clouds. In *Proceedings of the ECCV*, pages 131–147, 2024. 15

[39] Amir Zadeh, Michael Chan, Paul Pu Liang, Edmund Tong, and Louis-Philippe Morency. Social-iq: A question answering benchmark for artificial social intelligence. In *Proceedings of the CVPR*, pages 8807–8817, 2019. 4

[40] Jun Zhan, Junqi Dai, Jiasheng Ye, Yunhua Zhou, Dong Zhang, Zhigeng Liu, Xin Zhang, Ruibin Yuan, Ge Zhang, Linyang Li, et al. Anygpt: Unified multimodal llm with discrete sequence modeling. In *Proceedings of the ACL*, pages 9637–9662, 2024. 17, 42, 43, 44, 45, 46, 47

[41] Xiangyu Zhao, Wanghan Xu, Bo Liu, Yuhao Zhou, Fenghua Ling, Ben Fei, Xiaoyu Yue, Lei Bai, Wenlong Zhang, and Xiao-Ming Wu. Msearth: A benchmark for multimodal scientific comprehension of earth science. *arXiv preprint arXiv:2505.20740*, 2025. 4

[42] Sheng Zhou, Junbin Xiao, Qingyun Li, Yicong Li, Xun Yang, Dan Guo, Meng Wang, Tat-Seng Chua, and Angela Yao. Egotextvqa: Towards egocentric scene-text aware video question answering. In *Proceedings of the CVPR*, pages 3363–3373, 2025. 4