ModelArts

Data Preparation

Issue	01
Date	2026-07-03

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: https://www.huaweicloud.com/intl/en-us/

Contents

1 Data Preparation Functions

Function

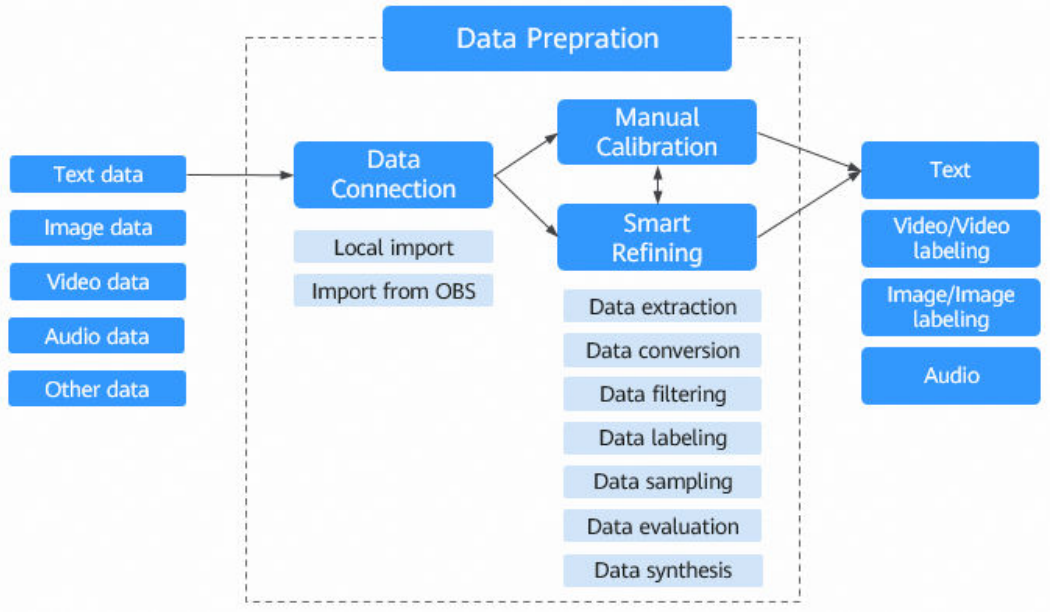
During the development of large models, data quality and processing efficiency directly impact model performance. However, developers often face challenges such as difficult data acquisition, inconsistent data quality, and low processing efficiency. These challenges not only increase the cost of model training but also limit the generalization capability of models. How to efficiently prepare high-quality training data has become an urgent issue. ModelArts data preparation provides one-stop, full-process data processing and management services. With built-in industry-level data processing operators and automated pipelines, it systematically handles data acquisition, processing, and publishing. This helps you efficiently convert massive amounts of raw, multimodal data into highly available and pure training datasets, improving data quality and processing efficiency, significantly reducing model training costs, and enhancing model generalization capabilities.

Data Preparation Process

ModelArts provides end-to-end data development features. You can use **Data Connection**, **Manual Calibration**, and **Smart Refining** to develop model datasets. **Smart Refining** covers the entire data processing process, including data processing and synthesis, helping developers quickly generate datasets required for model development.

Figure 1-1 shows the overall data preparation process.

Figure 1-1 Data preparation process



- **Data Connection**: Data acquisition is the first step of data engineering. Data from different sources and in different formats can be imported to the platform. An original dataset can be generated. With this feature, you can easily import a large amount of data to the platform to prepare for subsequent smart refining and model development. For details, see **Data Connection**.
- **Manual Calibration**: You can manually calibrate datasets on the visualized labeling page, generate standard datasets in one-click mode, and synchronize the datasets to **My Data** for tasks such as smart refining. For details, see **Manual Calibration**.
- **Smart Refining**: provides one-stop operations such as data processing and data synthesis to ensure that the original data meets various service requirements and model training standards and to obtain a dataset that meets model development requirements. For details, see **Smart Refining**.

Data Asset Management

The data asset management module provides a one-stop multimodal data management center for developers. It breaks down data silos and implements full-link closed-loop management from data ingestion, version control, quality preview, to final calling. ModelArts manages multimodal data, including text, image, audio, and video data. It categorizes data into preset and user-defined assets based on the source, supporting both general capability development and specialized, domain-specific customizations. For details, see **Data Asset Management**.

Data Types Supported by the ModelArts Platform

ModelArts provides the most comprehensive data processing functions in the industry. It can process text, image, audio, video, and platform-compatible datasetsModelArts also allows you to customize datasets and supports widely

used dataset formats such as **Alpaca** and **ShareGPT**, enabling flexible processing of diverse data.

The platform's smart refining and management capabilities provide you with comprehensive datasets for developing models.

Table 1-1 lists the data types supported by the platform. For details about the data format requirements of each type, see **Dataset Format Requirements**.

Table 1-1 Data Types Supported by the Platform

Data Type	Content	Supported File Format	Requirements on Datasets
Text	Document	docx and pdf.	**Format Requirements for Text Datasets**
	Pre-trained text	jsonl	
	Single-turn Q&A	jsonl and csv	
	Single-turn Q&A (with a system persona)	jsonl and csv	
	Multi-turn Q&A	jsonl	
	Multi-turn Q&A (with a system persona)	jsonl	
	Q&A sorting	jsonl and csv	
	Direct Preference Optimization (DPO)	jsonl	
	DPO (with a system persona)	jsonl	

Data Type	Content	Supported File Format	Requirements on Datasets
Image	Image	• Image + JSONL (optional) – Images can be in JPG, JPEG, PNG, or BMP format. – JSONL is an optional file type. If the JSONL file exists, ensure that the following conditions are met: The image file indexed in the JSONL file must exist. The JSONL file must be stored in the root directory of the dataset and named **annotation.jsonl**. The JSONL file supports only UTF-8 encoding.	**Format Requirements for Image Datasets**
Video	Video	mp4 and avi	**Format Requirements for Video Datasets**
	Video + Annotation	• Video + JSONL – Supported video formats: MP4 and AVI – Annotation files must be in JSONL format. The encoding format can only be UTF-8.	
Audio	Audio	• Audio + JSONL – Audio file: The .mp3, .flac, .wav, .opus, .aac and .m4a files are supported. Audio files can be stored in the **root** directory or a lower-level directory. – Annotation file format: Optional. UTF-8-encoded JSONL files are supported. Each line describes the relative path of an audio file in the dataset and other information.	**Format Requirements for Audio Datasets**
Other	Custom	You can customize dataset types based on specific scenarios. Mainstream Alpaca and ShareGPT datasets are supported.	**Format Requirements for Other Datasets**

2 Dataset Format Requirements

2.1 Format Requirements for Text Datasets

ModelArts supports the creation of text datasets. During the creation, you can import data in various formats. **Table 2-1** lists the format requirements.

Constraints

- Import from OBS: The size of a single file or compressed package cannot exceed 20 GB. If multiple files are imported, the total file size cannot exceed 20 GB.
- Local import: The size of a single file cannot exceed 1 GB, and the number of files cannot exceed 20.
- JSONL files support only UTF-8 encoding.

Table 2-1 Format requirements for text datasets

File Content	File Format	Format Requirement
Document	docx and pdf	Original document content.

Pre-trained text	jsonl	**text** indicates the text data used for pre-training. The following is an example:
		{
		{"text": "The Road is a 2006 post-apocalyptic novel by American writer Cormac McCarthy. The book details the grueling journey of a father and his young son over a period of several months across a landscape blasted by an unspecified cataclysm that has destroyed industrial civilization and almost all life."
		}

<table>
<tr><th>File Content</th><th>File Format</th><th>Format Requirement</th></tr>
<tr><td>Single-turn Q&A</td><td>jsonl</td><td>If the data is in JSONL format, the Alpaca, ShareGPT, and Standard formats are supported. The following are examples of datasets in different formats.
• Alpaca format
Non-CoT data: The data consists of Q&A pairs. instruction indicates the question, and output indicates the answer. The following is an example:
<pre>{
  "instruction": "Please recommend a book.",
  "input": "",
  "output": "Certainly! I recommend you read The Future of Autonomous Driving."
}</pre>CoT data: The data consists of Q&A pairs. instruction indicates the question, and output indicates the answer. The output must contain the think tag pair to indicate the thinking process. The following is an example:
<pre>{
  "instruction": "Can you recommend some books to me?"
  "input": "",
  "output": "<think>The user requests book recommendations but does not provide specific preferences. It is suitable to recommend a technology book that covers a wide range of topics and has a forward-looking perspective.</think>I recommend The Future of Autonomous Driving."
}</pre>• ShareGPT format
Non-CoT data: The data consists of Q&A pairs. In a conversation, the values from the human and gpt roles represent the question and answer, respectively. The following is an example:
<pre>{
  "conversations": [
    {
      "from": "human",
      "value": "Can you recommend some books to me?"
    },
    {
      "from": "gpt",
      "value": "As an expert in book recommendations, I recommend you read The Future of Autonomous Driving."
    }
  ]
}</pre>CoT data: The data consists of Q&A pairs. In a conversation, the values from the human and gpt roles indicate the question and answer, respectively. The value of the gpt role must contain the think tag pair to indicate the thinking process. The following is an example:
<pre>{
  "conversations": [
    {
      "from": "human",
      "value": "Can you recommend some books to me?"
    },</pre></td></tr>
</table>

File Content	File Format	Format Requirement

		<pre>{
		"from": "gpt",
		"value": "<think>As an expert in book recommendations, I should recommend books based on current technological trends and popular reading preferences.</think>I recommend you read The Future of Autonomous Driving."
		}
		]
		}</pre>
		• **Standard format**
		Non-CoT data: The data consists of Q&A pairs. **context** indicates the question, and **target** indicates the answer. The following is an example:
		<pre>{
		"context": "Hello, please introduce yourself.",
		"target": "I am a Pangu model."
		}</pre>
		CoT data: The data consists of Q&A pairs. **context** indicates the question, and **target** indicates the answer. **target** must contain the think tag pair to indicate the thinking process. The following is an example:
		<pre>{
		"context": "Hello, please introduce yourself.",
		"target": "<think>OK. The user asks me to introduce myself. First, I need to clarify the user's identity and usage scenario...</think>I am a Pangu model."
		}</pre>

	csv	• Non-CoT data: The first column in the CSV file corresponds to **context**, and the second column corresponds to **target**. The following is an example: <pre>"Hello, please introduce yourself.","I am a Pangu model."</pre> • CoT data: In the CSV file, the first column corresponds to **context**, the second column corresponds to **target**, and **target** must contain the think tag pair. The following is an example: <pre>"Hello, please introduce yourself","<think> The user asks me to introduce myself. First, I need to clarify the user's identity and usage scenario...</think>I am a Pangu model."</pre>

File Content	File Format	Format Requirement
Single-turn Q&A (with a system persona)	jsonl	If the data is in JSONL format, the Alpaca, ShareGPT, and Standard formats are supported. The following are examples of datasets in different formats. • **Alpaca format** Non-CoT data: **system** indicates the role, and **instruction** and **output** indicate the question and answer, respectively. The following is an example: `{ "system": "You are a book recommendation expert.", "instruction": "Please recommend books based on the user's requirements.", "input": "", "output": "As a book recommendation expert, I recommend you read The Future of Autonomous Driving." }` CoT data: **system** indicates the role, **instruction** indicates the question, and **output** indicates the answer. The output must contain the think tag pair to indicate the thinking process. The following is an example: `{ "system": "You are a book recommendation expert.", "instruction": "Please recommend books based on the user's requirements.", "input": "", "output": "<think>As a book recommendation expert, I should consider both technological trends and popular reading preferences.</think>I recommend you read The Future of Autonomous Driving." }` • **ShareGPT format** Non-CoT data: The data consists of Q&A pairs. **system_prompt** indicates the role. In a conversation, the values of from the **human** and **gpt** roles indicate the question and answer, respectively. The following is an example: `{ "system_prompt": "Role name: book recommendation expert.", "conversations": [{ "from": "human", "value": "Can you recommend some books to me?" }, { "from": "gpt", "value": "As an expert in book recommendations, I recommend you read The Future of Autonomous Driving." }] }` CoT data: The data consists of Q&A pairs. **system_prompt** indicates the role. In a conversation, the values corresponding to the **human** and **gpt** roles indicate the question and answer, respectively. The value of the **gpt** role must contain the think tag pair to

File Content	File Format	Format Requirement

		indicate the thinking process. The following is an example:
		{
		"system_prompt": "Role name: book recommendation expert.",
		"conversations": [
		{
		"from": "human",
		"value": "Can you recommend some books to me?"
		},
		{
		"from": "gpt",
		"value": "<think>As an expert in book recommendations, I should recommend books based on current technological trends and popular reading preferences.</think>I recommend you read The Future of Autonomous Driving."
		}
		]
		}
		• **Standard format**
		Non-CoT data: The data consists of Q&A pairs. **system** indicates the role. **context** indicates the question, and **target** indicates the answer. The following is an example:
		{
		"system": "Witty and humorous",
		"context": "Hello, please introduce yourself.",
		"target": "Haha, hello. I'm your smart assistant."
		}
		CoT data: The data consists of Q&A pairs. **system** indicates the role. **context** indicates the question, and **target** indicates the answer. **target** must contain the think tag pair to indicate the thinking process. The following is an example:
		{
		"system": "Witty and humorous",
		"context": "Hello, please introduce yourself.",
		"target": "<think>OK. The user asks me to introduce myself. First, I need to clarify the user's identity and usage scenario...</think>I am a Pangu model."
		}

	csv	• CSV non-CoT data: The first column in the CSV file corresponds to **system**, and the second and third columns correspond to **context** and **target**, respectively. The following is an example: "You're a smart and humorous Q&A assistant.","Hello, please introduce yourself.","Hello. I'm your smart assistant." • CSV CoT data: In the CSV file, the first column corresponds to **system**, and the second and third columns correspond to **context** and **target**, respectively. **target** must contain the think tag pair to indicate the thinking process. The following is an example: "You are a smart and humorous Q&A assistant.","context":"Hello, please introduce yourself.","<think>The user asks me to introduce yourself. First, I need to clarify the user's identity and usage scenario.</think>Hello. I'm your smart assistant."

File Content	File Format	Format Requirement
Multi-turn Q&A	jsonl	If the data is in JSONL format, the Alpaca, ShareGPT, and Standard formats are supported. The following are examples of datasets in different formats. • **Alpaca format** Non-CoT data: Array format. It consists of one or more turns of Q&A pairs. **instruction** indicates the question, **output** indicates the answer, and **history** indicates the historical conversation. The following is an example: `{ "instruction": "I'm looking for books related to autonomous driving.", "input": "", "output": "I recommend you read The Future of Autonomous Driving.", "history": [["I'm looking for a book that can help me understand future technology trends.", "Future technologies cover a wide range. Which areas are you more interested in?"]] }` CoT data: Array format. It consists of one or more turns of Q&A pairs. **instruction** and **output** indicate the question and answer, respectively. **history** indicates the historical conversation. The **think** tag pair indicates the thinking process. The following is an example: `{ "instruction": "I'm looking for books related to autonomous driving.", "input": "", "output": "<think>Since the user has specified the field, I should recommend a representative book with a complete structure and a balanced perspective on both technology and industry.</think>I recommend you read The Future of Autonomous Driving.", "history": [["I'm looking for a book that can help me understand future technology trends.", "<think>As a book recommendation expert, the first step should be to confirm the specific technology field the user is interested in, rather than directly providing a book title.</think>Future technologies cover a wide range of fields. Could you tell me which genres or topics interest you most?"]] }` • **ShareGPT format** Non-CoT data: Array format. It consists of one or more turns of Q&A pairs. In a conversation, the values corresponding to the **human** and **gpt** roles indicate the question and answer, respectively. The following is an example: `{ "conversations": [{ "from": "human", "value": "Hello"`

File Content	File Format	Format Requirement

		<pre> },
		    {
		      "from": "gpt",
		      "value": "Hi! How can I help you?"
		    },
		    {
		      "from": "human",
		      "value": "Can you recommend some books to me?"
		    },
		    {
		      "from": "gpt",
		      "value": "Certainly! Based on your interests, I recommend you read The Future of Autonomous Driving."
		    }
		  ]
		}</pre> CoT data: Array format. It consists of one or more turns of Q&A pairs. The values corresponding to **human** and **gpt** roles in a conversation indicate the question and answer, respectively. The value corresponding to the **gpt** role contains the think tag pair to indicate the thinking process. The following is an example:
		<pre>{
		  "conversations": [
		    {
		      "from": "human",
		      "value": "I want to read a book about future technology trends."
		    },
		    {
		      "from": "gpt",
		      "value": "<think>I should confirm the specific field first.</think>Future technologies cover a wide range. Which areas are you more interested in?"
		    },
		    {
		      "from": "human",
		      "value": "Autonomous driving."
		    },
		    {
		      "from": "gpt",
		      "value": "<think>I should recommend a representative book in the specified field. </think>I recommend The Future of Autonomous Driving."
		    }
		  ]
		}</pre> • **Standard format**
		Non-CoT data: Array format. It consists of one or more turns of Q&A pairs. **context** indicates the question, and **target** indicates the answer. The following is an example:
		<pre>[
		  {
		    "context": "Hello",
		    "target": "Hello. How can I help you?"
		  },
		  {
		    "context": "Please introduce Huawei Cloud products.",
		    "target": "Huawei Cloud provides products and services including</pre>

File Content	File Format	Format Requirement

		but not limited to compute, storage, and network."
		}
		]
		CoT data: Array format. It consists of one or more turns of Q&A pairs. **context** and **target** indicate the question and answer, respectively. **target** of at least one turn of Q&A contains the think tag pair, indicating the thinking process. The following is an example:
		[
		{
		"context": "Hello",
		"target": "Hello. How can I help you?"
		},
		{
		"context": "Please introduce Huawei Cloud products.",
		"target": "<think>Okay, the user asked me to introduce Huawei Cloud products. First, I need to recall...</think>Huawei Cloud provides products and services including but not limited to compute, storage, and networking."
		}
		]

File Content	File Format	Format Requirement

Multi-turn Q&A (with a system persona)	jsonl	If the data is in JSONL format, the Alpaca, ShareGPT, and Standard formats are supported. The following are examples of datasets in different formats.
		• **Alpaca format**
		Non-CoT data: Array format. It consists of one or more turns of Q&A pairs. **system** indicates the role, **instruction** indicates the question, **output** indicates the answer, and **history** indicates the historical conversation. The following is an example:
		`{ "system": "You are a book recommendation expert.", "instruction": "I'm looking for books related to autonomous driving.", "input": "", "output": "I recommend you read The Future of Autonomous Driving.", "history": [["I'm looking for a book that can help me understand future technology trends.", "Future technologies cover a wide range. Which areas are you more interested in?"]] }`
		CoT data: Array format. It consists of one or more turns of Q&A pairs. **system** indicates the role. **instruction** and **output** indicate the question and answer, respectively. **history** indicates the historical conversation. **output** must contain the think tag pair to indicate the thinking process. The format is as follows:
		`{ "system": "You are a book recommendation expert.", "instruction": "I'm looking for books related to autonomous driving.", "input": "", "output": "<think>Since the user has specified the field, I should recommend a representative book with a complete structure and a balanced perspective on both technology and industry.</think>I recommend you read The Future of Autonomous Driving.", "history": [["I'm looking for a book that can help me understand future technology trends.", "<think>As a book recommendation expert, the first step should be to confirm the specific technology field the user is interested in, rather than directly providing a book title.</think>Future technologies cover a wide range of fields. Could you tell me which genres or topics interest you most?"]] }`
		• **ShareGPT format**
		Non-CoT data: Array format. It consists of one or more turns of Q&A pairs. **system_prompt** indicates the role. In the conversation, the values corresponding to the **human** and **gpt** roles indicate the question and answer, respectively. The following is an example:

File Content	File Format	Format Requirement

		<pre>{
		  "system_prompt": "Book recommendation expert",
		  "conversations": [
		  {
		  "from": "human",
		  "value": "Hello"
		  },
		  {
		  "from": "gpt",
		  "value": "Hello, what can I help you with?"
		  },
		  {
		  "from": "human",
		  "value": "What book do you recommend?"
		  },
		  {
		  "from": "gpt",
		  "value": "I recommend The Future of Autonomous Driving."
		  }
		  ]
		}</pre> CoT data: Array format. It consists of one or more turns of Q&A pairs. **system_prompt** indicates the role. The values corresponding to **human** and **gpt** roles in a conversation indicate the question and answer, respectively. The value corresponding to the **gpt** role contains the think tag pair to indicate the thinking process. The following is an example: <pre>{
		  "system_prompt": "Book recommendation expert",
		  "conversations": [
		  {
		  "from": "human",
		  "value": "I want to read a book about future technology trends."
		  },
		  {
		  "from": "gpt",
		  "value": "<think>I should confirm the specific field first.</think>Future technologies cover a wide range. Which areas are you more interested in?"
		  },
		  {
		  "from": "human",
		  "value": "Autonomous driving."
		  },
		  {
		  "from": "gpt",
		  "value": "<think>I should recommend a representative book in the specified field. </think>I recommend The Future of Autonomous Driving."
		  }
		  ]
		}</pre> • **Standard format** Non-CoT data: Array format. It consists of one or more turns of Q&A pairs. **system** indicates the persona, **context** indicates the question, and **target** indicates the answer. The following is an example:

File Content	File Format	Format Requirement

		<pre>[
		  {
		  "system": "Witty and humorous"
		  },
		  {
		  "context": "Hello, please introduce yourself.",
		  "target": "Haha, hello. I'm your smart assistant."
		  }
		]</pre> CoT data: Array format. It consists of one or more turns of Q&A pairs. **system** indicates the role. **context** and **target** indicate the question and answer, respectively. **target** of at least one turn of Q&A contains the think tag pair, indicating the thinking process. The following is an example: <pre>[
		  {
		  "system": "Witty and humorous"
		  },
		  {
		  "context": "Hello, please introduce yourself.",
		  "target": "<think>Okay, the user asks me to introduce myself. First, I need to clarify the user's identity and usage scenario...</think> Haha, hello! I'm your smart assistant."
		  }
		]</pre>

Q&A sorting	jsonl	**context** indicates the question. The order of **targets** 1, 2, and 3 represent the order of human-preferred answers. The most preferred answer is placed at the forefront. <pre>{
		  "context": "context content,"
		  "targets": [
		  "a",
		  "b",
		  "c"
		  ]
		}</pre>

	csv	• CSV format: The first column in the CSV file corresponds to **context**, and the other columns are answers. <pre>"How do you understand the line 'A thousand sails pass by the sunken ship'?","This line is more than a beautiful image that describes the scene where new ships still pass by the sunken ship like a thousand sails. It contains profound philosophy and expresses the natural law that new things will inevitably replace old things.","This line can be used to describe natural scenery, expressing the grandeur of nature and the vitality of life."</pre>

<table>
<tr><th>File Content</th><th>File Format</th><th>Format Requirement</th></tr>
<tr><td>Direct Preference Optimization (DPO)</td><td>jsonl</td><td>

Non-CoT data: context indicates the question, target indicates the expected correct answer, and bad_target indicates the incorrect answer that does not meet the expectation. The following is an example:
Single-turn DPO
<pre>{
  "context": [
    "Hello, please introduce yourself."
  ],
  "target": "I am a Pangu model,"
  "bad_target": "I cannot answer this question."
}</pre>Multi-turn DPO
<pre>{
  "context": [
    "Hello, please introduce yourself.",
    "I am a Pangu model,"
    "Please introduce Huawei Cloud products."
  ],
   "target": "Huawei Cloud provides products and services including but not limited to compute, storage, and network.",
  "bad_target": "I cannot answer this question."
}</pre>
CoT data: context indicates the question, target indicates the expected correct answer, and bad_target indicates the incorrect answer that does not meet the expectation. At least one answer contains the think tag pair, indicating the thinking process. The following is an example:
Single-turn DPO
<pre>{
  "context": [
    "Hello, please introduce yourself."
  ],
  "target": "<think>OK. The user asks me to introduce myself. First, I need to clarify the user's identity and usage scenario...</think>I am a Pangu model.",
  "bad_target": "<think>OK. The user asks me to introduce myself...</think>I cannot answer this question."
}</pre>Multi-turn DPO
<pre>{
  "context": [
    "Hello, please introduce yourself.",
    "I am a Pangu model,"
    "Please introduce Huawei Cloud products."
  ],
   "target": "<think>Okay, the user asked me to introduce Huawei Cloud products. First, I need to recall...</think>Huawei Cloud provides products and services including but not limited to compute, storage, and networking.",
  "bad_target": "<think> Okay, the user asked me to introduce Huawei Cloud products...</think>I cannot answer this question."
}</pre>

</td></tr>
</table>

File Content	File Format	Format Requirement
DPO (with a system persona)	jsonl	• Non-CoT data: **system** indicates the role. **context** indicates the question, **target** indicates the expected correct answer, and **bad_target** indicates the incorrect answer that does not meet the expectation. The following is an example: Single-turn DPO with persona `{ "system": "You are a witty and humorous Q&A assistant.", "context": [ "Hello, please introduce yourself." ], "target": "Haha, hello. I'm your smart assistant. How can I help you?", "bad_target": "I cannot answer this question." }` Multi-turn DPO with persona `{ "system": "You are a witty and humorous Q&A assistant.", "context": ["Hello, please introduce yourself.", "Haha, hello. I'm your smart assistant. How can I help you?", "Please introduce your products."], "target": "We offer a wide range of products, including compute, storage, and network products.", "bad_target": "I cannot answer this question." }` • CoT data: **system** indicates the role. **context** indicates the question, **target** indicates the expected correct answer, and **bad_target** indicates the incorrect answer that does not meet the expectation. At least one correct answer contains the think tag pair, indicating the thinking process. The following is an example: Single-turn DPO with persona `{ "system": "You are a witty and humorous Q&A assistant.", "context": ["Hello, please introduce yourself."], "target": "<think> The user asked me to introduce myself. First, I need to clarify the user's identity and usage scenario.</think>Haha, hello! I'm your smart assistant. How can I help you?", "bad_target": "I cannot answer this question." }` Multi-turn DPO with persona `{ "system": "You are a witty and humorous Q&A assistant.", "context": ["Hello, please introduce yourself.", "Haha, hello. I'm your smart assistant. How can I help you?", "Please introduce your products."], "target": "<think>The customer wants to know about our products.</think>We offer a wide range of products, including compute, storage, and network products.", "bad_target": "I cannot answer this question." }`

2.2 Format Requirements for Image Datasets

ModelArts supports the creation of image datasets. lists the format requirements.

Constraints

- Import from OBS: The size of a single file or compressed package cannot exceed 20 GB. If multiple files are imported, the total file size cannot exceed 20 GB.
- Local import: The size of a single file cannot exceed 1 GB, and the number of files cannot exceed 20.
- JSONL files support only UTF-8 encoding.

2.3 Format Requirements for Video Datasets

ModelArts supports the creation of image datasets. During the creation, you can import data in various formats. **Table 2-2** lists the format requirements.

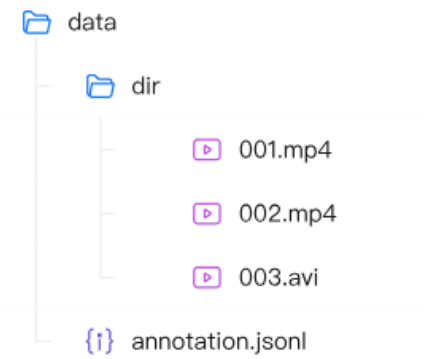
Constraints

- Import from OBS: The size of a single file or compressed package cannot exceed 20 GB. If multiple files are imported, the total file size cannot exceed 20 GB.
- Local import: The size of a single file cannot exceed 1 GB, and the number of files cannot exceed 20.
- JSONL files support only UTF-8 encoding.

Table 2-2 Format requirements for video datasets

File Content	File Format	Requirement
Video	mp4 and avi	Videos in MP4 and AVI formats can be uploaded. Videos can be stored in multiple folders, and each folder can contain videos in MP4 or AVI format.

File Content	File Format	Requirement

Video + Annotation	Video + JSONL	• The video format can be MP4 and AVI.
		The following is an example.
		 
		For details about the annotation file in JSONL format, refer to the following:
		{
		"video_fn": "13/ad098173-af09-48fe-95c3-e72fd629688e.mp4",
		"prompt": "A person pours a clear liquid from a bottle into a shot glass, then lifts the glass to their mouth and drinks the shot. The background includes a red coat and other indistinct background elements.",
		"long_prompt": "A person is seen pouring a clear liquid from a green glass bottle into a small glass. The individual is wearing a white shirt with a lace collar and a beige cardigan. The background appears to be a cozy indoor setting, possibly a cafe or a restaurant, with red and white elements visible, such as a red coat hanging on the wall and a white table. The person carefully pours the liquid, ensuring it is filled to the brim of the glass. The liquid is clear and has some green leaves floating in it. The person then holds the glass up, possibly to show the contents or to prepare for a drink."
		}

2.4 Format Requirements for Audio Datasets

ModelArts supports the creation of audio datasets. **Table 2-3** lists the format requirements.

Constraints

- Import from OBS: The size of a single file or compressed package cannot exceed 20 GB. If multiple files are imported, the total file size cannot exceed 20 GB.
- Local import: The size of a single file cannot exceed 1 GB, and the number of files cannot exceed 20.
- JSONL files support only UTF-8 encoding.

Table 2-3 Format Requirements for Audio Datasets

File Content	File Format	Requirement
Audio	Audio + JSONL (optional)	• Audio file: The .mp3, .flac, .wav, .opus, .aac and .m4a files are supported. Audio files can be stored in the **root** directory or a lower-level directory. Annotation file format: Optional. UTF-8-encoded JSONL files are supported. Each line describes the relative path of an audio file in the dataset and other information. The following is an example.  For details about the annotation file in JSONL format, refer to the following: {"audio_name":"dir/001.mp3","caption":"1"} {"audio_name":"dir/002.flac","caption":"2"} {"audio_name":"dir/003.wav","caption":"3"}

2.5 Format Requirements for Other Datasets

In addition to text, image, video, and audio datasets, you can also import other types of datasets, that is, custom datasets used for model training, such as the common open-source datasets in Alpaca and ShareGPT formats.

Import from OBS: The size of a single file or compressed package cannot exceed 20 GB. If multiple files are imported, the total file size cannot exceed 20 GB.

Local upload: The size of a single file cannot exceed 1 GB, and a maximum of 20 files can be uploaded at a time.

This section describes the requirements for common open-source dataset formats.

Alpaca Dataset Format Requirements

Alpaca is a common dataset format used by open-source models (such as the DeepSeek and Qwen series) and is the main dataset format used for fine-tuning open-source models. It is especially used for instruction-tuning. The data format provides a clear task description (instruction), input, and output.

Typical Alpaca dataset format:

```
[
  {
    "instruction": "Human instruction (required)",
    "input": "Human input (optional)",
    "output": "Model answer (required)",
    "system": "System prompt (optional)",
    "history": [
      [
        "First-turn instruction (optional)",
        "First-turn answer (optional)"
      ],
      [
        "Second-turn instruction (optional)",
        "Second-turn answer (optional)"
      ]
    ]
  }
]
```

Field description:

- instruction: Task instruction, which tells the model what operation needs to be performed.
- input: Input required for the task. If the task is open-ended or does not require explicit input, this field can be an empty string.
- output: Expected output of the task, which is the content that the model needs to generate given the instruction and input. To train a model that incorporates a CoT or thinking process, you can wrap the reasoning process within <think> and </think> tags or by prepend a prompt like "Let's think step by step."
- system: System prompt, which specifies the style or role. This field is optional.
- history: A list of tuples, each representing the instruction and response of each turn of conversation in the historical messages. During instruction supervision fine-tuning, the responses in the historical messages are also used for model learning. This field is optional.

Features:

- The Alpaca data format is simple and easy to understand.
- The task instruction and input content are separated, making it suitable for various natural language processing tasks, such as text generation, translation, and summarization.

ShareGPT Dataset Format Requirements

The ShareGPT format comes from the dataset that records the conversations between ChatGPT and users. It is mainly used for the training of dialog systems. It gathers and organizes multiple exchanges that mimic real user-AI interactions. ShareGPT datasets support diverse role types, such as **human**, **gpt**, **observation**, and **function**. They are presented in the **conversations** column based on different role objects.

Typical ShareGPT dataset format:

```
[
  {
    "conversations": [
```



```
        {
            "from": "human",
            "value": "human instruction"
        },
        {
            "from": "function_call",
            "value": "tool parameter"
        },
        {
            "from": "observation",
            "value": "tool result"
        },
        {
            "from": "gpt",
            "value": "model answer"
        }
    ],
    "system": "system prompt (optional)",
    "tools": "tool description (optional)"
  }
]
```

- conversations: A list of conversations, including the role and content of each turn of conversation. This field is mandatory. The role fields are defined as follows:
  - human: The instruction given by humans in a conversation.
  - function_call: Tool calling. The tool is an AP that provides a certain function.
  - observation: The result of function_call.
  - gpt: The answer provided by the model based on the instruction given by humans.

  Note: **human** and **observation** in roles must be in odd positions, and **gpt** and **function** must be in even positions.
- system: System prompt. It is optional.
- tools: A description of function_call. It is optional.

Features:

The ShareGPT format is closer to the way humans interact with AI and is suitable for building and fine-tuning conversational models.

Suggestions

- The Alpaca format is suitable for single-turn instruction-tuning, such as task-oriented dialogs, Q&A systems, or tool calls. Its structured design simplifies the understanding and response of models to explicit instructions. It is often used for lightweight fine-tuning (such as LoRA fine-tuning) or basic capability training (such as text generation and translation).
- The ShareGPT format focuses on multi-turn dialog scenarios. It records the interaction history between users and assistants through the **conversations** field. It is suitable for training conversational models (such as chatbots and customer service assistants) and performs better in tasks that require dialog coherence, such as context understanding, emotional dialogs, or complex reasoning.
- The two formats can be used together, with the former enhancing basic capabilities and the latter improving interaction experience.

3 Data Connection

3.1 Creating a Data Connection

Scenarios

In data processing and model training scenarios, efficiently and accurately importing various datasets into ModelArts is essential to support subsequent smart refining and model training. Traditional data import methods, however, are often limited—supporting only basic task naming and offering minimal data format conversion capabilities. These constraints lead to difficulties in data import and slow processing efficiency, making the need for a more flexible and efficient data import solution increasingly urgent. ModelArts addresses these challenges with an enhanced data import feature that supports multiple common data types. You can now customize task names and descriptions during task creation. The platform also supports data format conversion and direct dataset publishing, significantly improving both the flexibility and efficiency of data processing. This enhancement meets the diverse needs of users during the data preparation phase.

Prerequisites

- You have registered a Huawei ID and enabled Huawei Cloud services, performed real-name authentication, and ensure your account is not frozen or in arrears before using ModelArts. For details, see **Signing Up for a HUAWEI ID and Enabling Huawei Cloud Services** and **Real-Name Authentication Introduction**.
- Configure an agency.

  Certain ModelArts functions require access to services like OBS. Before using ModelArts, ensure your account has been authorized to access these services. For details, see **Configuring Agency Authorization for ModelArts with One Click**.
- Before creating an import job, prepare data based on TR202603005599_148 .

Constraints

- The size of a data file or compressed package imported through a data connection cannot exceed 20 GB, and the total file size cannot exceed 20 GB.

- When importing data from OBS, you can only specify a folder as the OBS path, not a file.

Creating a Data Connection Task

1. Log in to the **ModelArts console**.
2. In the left navigation pane of the management console, choose **Data Preparation** > **Data Connections**.
3. In the upper right corner, click **Create Data Connection**. On the displayed page, configure the parameters.

Table 3-1 Parameters required for creating a data connection

Parameter		Description	Example Value
Basic Information	Task Name	Custom task name. The default value is **data-connect-*YYYYMMDDHHMMSS***. The name must start with a letter and end with a letter or digit. It can contain 2 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.	data-connect-20260425083507
	Description	Description of a custom data connection. It can contain a maximum of 200 characters. Only letters, digits, spaces, hyphens (-), underscores (_), commas (,), periods (.), brackets, colons (:), and commas are allowed.	-
Data Import	Dataset Type	Text, image, video, audio, and other types of datasets are supported. For details about the dataset format requirements, see **Dataset Format Requirements**.	Text Document docx
	Connection Method	OBS and local upload are supported.	Object Storage Service

Parameter		Description	Example Value
	Storage Location	• If **Connection Method** is set to **Object Storage Service**, you can select **Object Storage Service – Bucket** or **Object Storage Service – Parallel File System**. Click to select an OBS storage address or manually enter an address. The storage address must start with **obs://** or **/** and end with a slash (**/**). It cannot contain double slashes (**//**) except in the prefix. For example, **obs://bucketname/path/** or **/bucketname/path/**. Ensure individual files or compressed packages do not exceed 1 GB in size. • If **Connection Method** is set to **Local upload**, you can select select **Object Storage Service – Bucket** or **Object Storage Service – Parallel File System**. Click to select an OBS storage address or manually enter an address. The storage address must start with **obs://** or **/** and end with a slash (**/**). It cannot contain double slashes (**//**) except in the prefix. For example, **obs://bucketname/path/** or **/bucketname/path/**. A maximum of 20 files can be uploaded. The size of a single file cannot exceed 1 GB. It is recommended that no more than 10 files be uploaded at a time.	obs://bucketname/path/
Generate Dataset	Dataset Name	Name of a custom dataset. The name must start with a letter and end with a letter or digit. It can contain 2 to 63 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.	dataset
	Dataset Property	Click to configure dataset properties as required, such as the industry and language.	-

Parameter		Description	Example Value
	Description	Description of a custom dataset. Only letters, digits, spaces, hyphens (-), underscores (_), commas (,), periods (.), brackets, colons (:), and commas are allowed. The value can contain a maximum of 200 characters.	-
	Dataset Status	Only published datasets can be used by downstream tasks such as model training. • If you select **Publish Dataset**, the generated dataset is in the **Online** state on the **Asset Management** > **Data** > **My Data** page and can be directly used by downstream model training jobs. • If you do not select **Publish Dataset**, the generated dataset will be in the **Offline** state on the **Asset Management** > **Data** > **My Data** page and cannot be directly used by downstream model training jobs. You need to manually publish the dataset before using it.	Select **Publish Dataset**.
	Extended Info	You can configure dataset copyright information as required. The dataset copyright function records and manages copyright details to ensure legal data use. It specifies the source, owner, and license of the dataset. By entering this information, you can track the data's origin and set usage rules, protecting copyrights and preventing disputes.	-

4. After setting the parameters, click **Create Now** in the lower right corner of the page. On the **Data Connections** page, you can view the task status of the dataset. If the status is **Success**, the data connection task is successful.
5. After the dataset is generated, view the generated dataset on the **Asset Management** > **Data** > **My Data** page. For more information, see **My Data**.

Follow-Up Operations

After a data connection task is successfully executed, it can be used for **Smart Refining**.

3.2 Managing Data Connections

On the **Data Connections** page, you can manage all tasks. You can view the name/ID, status, generated dataset name, operation time, operator, and supported operations of each data connection task. This section describes how to manage data connection tasks.

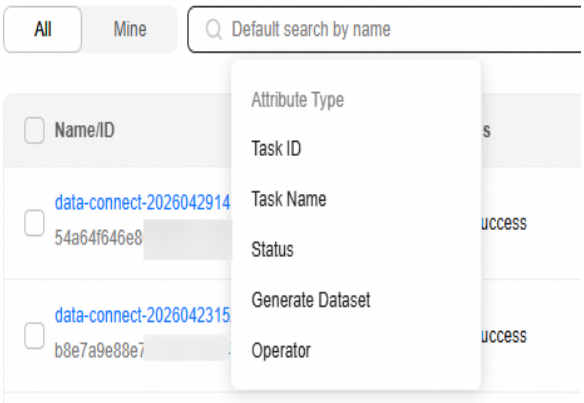
Data Connection Task Management

Data connection tasks can be filtered, searched, deleted, and retried.

- **Filtering and searching for data connection tasks**

  In the search box on the **Data Connections** page, you can filter tasks by task ID, task name, status, or generated dataset to quickly find the target task. You can also click **Mine** to display only the data connection tasks created by the current user.

  Figure 3-1 Filtering datasets

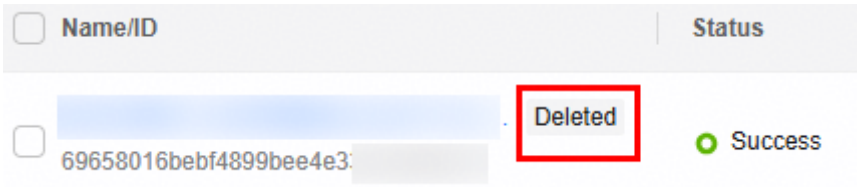


- **Deleting a data connection task**

  You can delete a created data connection task. On the **Data Connections** page, locate the task you want to delete and click **Delete** in the **Operation** column. In the displayed dialog box, confirm the deletion.

  Deleted tasks are not permanently removed. If you accidentally delete a task, you can restore it for future use. In the upper right corner of the **Data Connections** page, click **Show Deleted Tasks** to view the deleted tasks (labeled with **Deleted**).

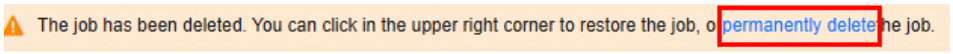
Figure 3-2 "Deleted" label of a deleted task



You can perform the following operations on deleted tasks:

- Restoring a task: Click **Restore** in the **Operation** column and click **OK**.
- Permanently deleting a task: A permanently deleted task cannot be restored. Click the task name. In the upper part of the page, click **permanently delete**. In the displayed dialog box, enter **DELETE** and click **OK**.

Figure 3-3 Permanent deletion



- **Batch deleting data connection tasks**

  On the **Data Connections** page, select multiple data connection tasks and click **Delete** in the upper right corner of the page to batch delete the tasks.

- **Retrying a data connection task**

  For a task that fails to be executed, you can click **Retry** in the **Operation** column on the **Data Connections** page. In the displayed dialog box, select **Start Now** or **Re-edit** and click **OK**. The differences between the two operations are as follows:

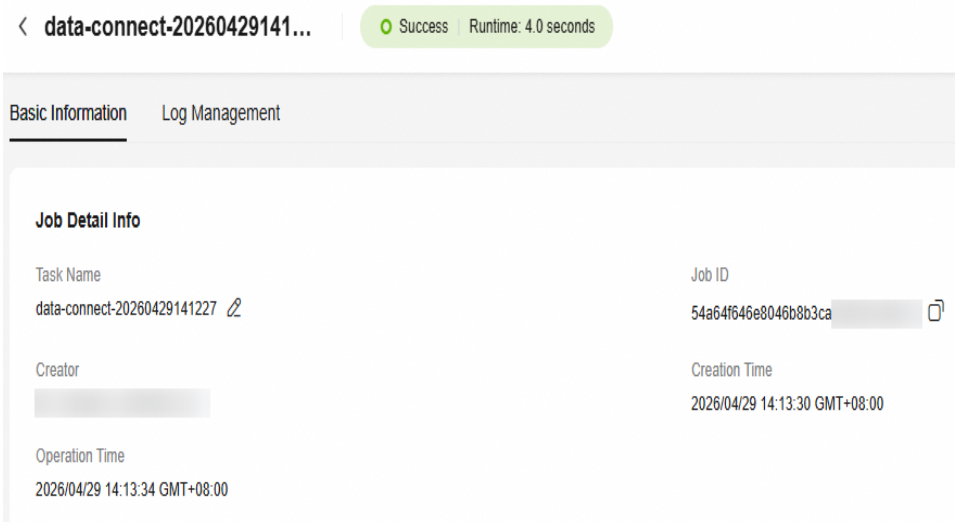
  - **Start Now**: Restart the data connection task without modifying any configuration of the task.
  - **Re-edit**: Go to the data connection configuration page again, modify the configuration, and restart the task.

Data Connection Task Details Management

The data connection task details page displays the details about the current task. On the **Data Connections** page, click a task name to go to the task details page. This page consists of the basic information and log management tab pages. The following describes the functions and operations on the two tab pages.

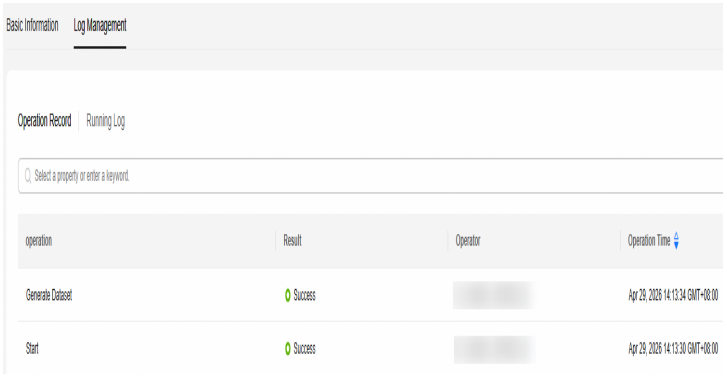
- **Basic Information**: This tab lists the dataset information, data configuration details, and information about the generated datasets of the task. You can delete a task in the upper right corner of the page. If a task fails to be executed, you can click **Retry** to restart the task.

Figure 3-4 Basic Information



- **Log Management**: On this page, you can view operation records and run logs. Operation records include all operations performed on the current task. Run logs track details of the execution process. Both of them help identify and troubleshoot issues that arise during task creation.

Figure 3-5 Log management



4 Data Refining

4.1 Smart Refining

4.1.1 Overview

Core Features

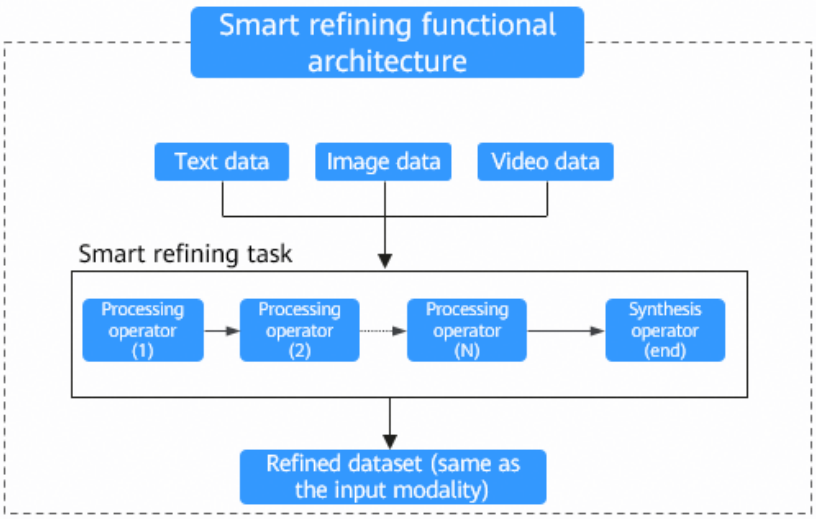
Smart refining is a key part of ModelArts data engineering, designed to address the dual challenges of data quality and quantity in foundation model training. It transcends the boundaries of traditional data processing by seamlessly integrating rule-based data processing (cleaning, filtering, deduplication, and more) with LLM-based data synthesis (rewriting, expansion, polishing, and more).

Through visualized operator orchestration, you can drag and drop multiple processing and synthesis operators to build an automated pipeline, much like assembling building blocks. The system follows your predefined logic to filter and optimize massive raw datasets layer by layer, ultimately outputting high-quality datasets that meet rigorous training requirements.

Functional Architecture

Smart refining takes text, image, and video datasets as input. It constructs smart refining tasks by orchestrating various data processing and synthesis operators to produce refined datasets. For details, see **Figure 4-1**.

Figure 4-1 Smart refining functional architecture



Benefits

- **Unified workflow**: Orchestrates data processing and synthesis in a single pipeline, eliminating the need to switch between modules and reducing intermediate data transfers.
- **Enhanced data quality**: Ensures high-quality input for the synthesis stage through rigorous, multi-level filtering using processing operators.
- **Flexible orchestration**: Supports the free combination of dozens of operators to satisfy diverse scenarios, from simple cleaning to complex data augmentation.
- **Efficient scale expansion**: Enables high-efficiency training data expansion by performing synthetic rewriting on top of cleaned, high-quality data.
- **Streamlined operation**: Offers a visualized, "what you see is what you get" orchestration experience, removing the need for manual scripting.
- **Workflow reproducibility**: Supports saving and reusing refining templates to ensure consistency across different data processing tasks.

4.1.2 Application Scenarios

Smart Refining is an integrated data preparation solution designed to streamline both cleaning and synthesis for large model training. Whether you are processing raw pre-training corpora by removing HTML tags and garbled characters, enhancing sparse seed data for SFT instruction fine-tuning, or ensuring security compliance through privacy anonymization, Smart Refining simplifies the workflow. It orchestrates a wide range of data processing operators into a cohesive pipeline, transforming messy or incomplete inputs into high-quality, diverse, and secure training datasets ready for model development.

Smart refining provides strong data processing and flexible operations, but getting started can be tough. This guide covers common smart refining scenarios to help you finish tasks easily and get high-quality data fast, speeding up model development.

Typical Scenarios

Smart refining is used in these typical scenarios, each with recommended operator combinations. Choose a scenario based on your needs.

- Choose **Scenario 1: Converting the Dataset Format** if you only need to convert the format of text datasets.
- Choose **Scenario 2: Cleaning and Improving Quality of Raw Corpus** if the data quality is poor and needs to be cleaned.
- Choose **Scenario 3: Expanding and Augmenting Training Data** if the data volume is insufficient and needs to be expanded.
- Choose **Scenario 4: Preparing Data for SFT** to prepare SFT data.
- Choose **Scenario 5: Processing Multimodal Data** to process image or video data.
- Choose **Scenario 6: Ensuring Data Compliance and Security** if you need to meet data compliance requirements.

Scenario 1: Converting the Dataset Format

Description

ModelArts supports multiple dataset formats. You need to convert data from one format to another without additional data processing.

Recommended operator orchestration sequence

```
Start node → end node
```

Expected results

The input data is converted into output data in different formats (Alpaca, ShareGPT, or platform-compatible).

Scenario 2: Cleaning and Improving Quality of Raw Corpus

Description

Raw data from the internet, internal systems, or third parties often has a lot of noise. So, it needs to be systematically cleaned for model training. For details about common corpus issues, see **Table 4-1**.

Table 4-1 Typical data issues

Data Issue	Form	Impact on the Model
Duplicate information	A large amount of identical or similar content exists in the data.	The trained model is overfitting.
Garbled data	Encoding errors and abnormal characters exist in the data.	The semantic understanding of the model is polluted.

Data Issue	Form	Impact on the Model
Sensitive and non-compliant information	Political, pornographic, or violent content exists in the data.	The model output has compliance risks.
Poor data quality	Sentences are not coherent, the logic is disordered, and sentences are incomplete.	The model generation quality is reduced.
Invalid data length	The data is too short to be meaningful or too long to be redundant.	The training efficiency is low.
Mixed data, not classified	Data from various domains is mixed and not classified by domain.	Unclassified domain data affects training efficiency.

Recommended operator orchestration sequence

Raw corpus → [Symbol standardization] → [Deduplication operator] → [Sensitive word filtering] → [Text length filtering] → [Incomplete sentence removal at paragraph ends] → [Pornographic text detection] → [Political text detection] → [Insult text detection operator] → [Pre-trained text classification] → Cleaned data

Expected results

- The data repetition rate is reduced by more than 90%.
- Low-quality samples are effectively removed.
- 100% of sensitive and non-compliant content is filtered out.
- The output data can be directly used for training or proceed to the next synthesis step.
- The output data can be classified by domain.

Scenario 3: Expanding and Augmenting Training Data

The cost of obtaining high-quality labeled data is high, and the existing data volume is insufficient to train a model with good performance.

Applicable scenarios

- Data is scarce in vertical domains.
- The labeling cost is too high.
- The data scale needs to be quickly expanded.
- Data diversity is insufficient.

Recommended operator orchestration sequence

Raw data → Data cleaning → Data generation → Expanded data

Table 4-2 Operators to use

Operator	Function	Configuration Suggestion
Data cleaning	Ensures seed data quality.	Ensure strict screening criteria.
Data generation	Generates diverse expressions.	Select an appropriate rewriting strategy to generate diverse data.

Expected results

- The data scale is expanded by 3 to 10 times.
- The semantic consistency is maintained.
- The expression diversity is improved.

Notes

- The synthesis operator must be placed at the end of the workflow.
- Only data synthesis of the same modality is supported.

Scenario 4: Preparing Data for SFT

Prepare high-quality datasets for SFT of foundation models.

Applicable scenarios

- Fine-tuning of general assistant models
- Customization of industry-specific models
- Dialogue capability optimization
- Task-oriented model training

Recommended operator orchestration process

```
Raw instruction data → Data cleaning → Text generation (optional) → Dataset generation
```

Table 4-3 Data format processing

Input Format	Processing Method	Output Format
Unstructured text	Format conversion operator	Alpaca/ShareGPT
Existing Alpaca	Quality filtering + rewriting	Optimized Alpaca
Existing ShareGPT	Quality filtering + rewriting	Optimized ShareGPT

Key quality control points

- Instruction clarity check
- Answer accuracy verification
- Format consistency assurance

Scenario 5: Processing Multimodal Data

Process datasets that contain multiple modalities, such as images and videos.

Applicable scenarios

- Video understanding data sorting

Recommended operator orchestration process (using images as an example)

```
Image dataset → Image deduplication → Image extraction → Image metadata filtering → Image detection → Processed data
```

Table 4-4 Key points for processing each modality

Modality	Key Point	Notes
Image	Size, format, and quality	Unified resolution
Video	Frame rate, resolution, and segment	Unified video encoding

Important constraints

A data processing task needs to be created for each modality separately.

Scenario 6: Ensuring Data Compliance and Security

Ensure that the training data complies with regulatory requirements and enterprise security policies.

Applicable scenarios

- Personal information protection (*GDPR*|*Personal Information Protection Law*)
- Sensitive data filtering

Recommended operator orchestration process

```
Raw data → Sensitive word filtering → Compliance data
```

Operators to use

Operator	Function	Compliance Requirements
Sensitive word filtering	Filters sensitive content in personal information.	Personal privacy requirements must be met.

4.1.3 Creating a Smart Refining Task

Scenarios

Smart Refining is an integrated data preparation solution designed to streamline both cleaning and synthesis for large model training. Whether you are processing raw pre-training corpora by removing HTML tags and garbled characters, enhancing sparse seed data for SFT instruction fine-tuning, or ensuring security compliance through privacy anonymization, Smart Refining simplifies the workflow. It orchestrates a wide range of data processing operators into a cohesive pipeline, transforming messy or incomplete inputs into high-quality, diverse, and secure training datasets ready for model development.

Prerequisites

- You have registered a Huawei ID and enabled Huawei Cloud services, performed real-name authentication, and ensure your account is not frozen or in arrears before using ModelArts. For details, see **Signing Up for a HUAWEI ID and Enabling Huawei Cloud Services** and **Real-Name Authentication Introduction**.
- You have configured an agency.

  Certain ModelArts functions require access to services like OBS. Before using ModelArts, ensure your account has been authorized to access these services.
- You have applied for the compute resources required for smart refining.
- If you need to use a custom dataset, import the dataset to ModelArts by referring to **Creating a Data Connection**.

Billing

Billing is based on the actual execution duration or usage of CPU operators.

Constraints

- **Synthesis operator**: The synthesis operator **must be placed at the last node** of the workflow. It cannot be inserted between processing operators or followed by other filtering operators.
- **Modality**:
  - Only **intra-modal** synthesis (e.g., text input to text output) is supported.
  - **Cross-modal** generation (such as generating Q&A pairs from input text or generating images from output text) is not yet supported.
- **Functions**:
  - Custom synthesis instructions (prompts) and the template functions are not supported.
  - Online debugging of synthesis tasks is not supported.
  - The output fields of synthesis operators are fixed, but the original fields in the input dataset are automatically retained.
- **Data volume and quality inspection**:
  - You cannot customize the number of output records of synthesized data. (Synthesized data is automatically generated based on the input.)

- You cannot perform automatic quality inspection or filtering on the results.
- The output is saved to a new dataset and is not automatically merged with the original dataset.

Creating a Smart Refining Task

1. Log in to the **ModelArts console**. In the navigation pane on the left, choose **Data Preparation** > **Data Refining**.
2. In the upper right corner, click **Create Smart Refining Task**, configure information, and click **Next Step**.

Table 4-5 Parameters for creating a smart refining task

Parameter		Description	Example Value
Basic Information	Name	Custom task name. The default value is **data-refine-*YYYYMMDDHHMMSS***. The name must start with a letter and end with a letter or digit. It can contain 2 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.	data-refine-20260423102952
	Description	Description of a custom smart refining task. Only letters, digits, spaces, hyphens (-), underscores (_), commas (,), and periods (.) are allowed. It can contain a maximum of 200 characters.	-
Selecting Dataset		Choose either ModelArts **Preset Data** or **My Data**. The supported dataset types are text, images, and videos. If you select **My Data**, upload a dataset. For details, see **Creating a Data Connection**.	Preset data **code-alpaca**
Refinement Template		ModelArts offers ready-to-use smart refinement templates. These templates include preset service processing tools and parameters for specific applications. You can use these templates directly. If your application does not need the services in the refinement template, skip this step and click **Next Step**.	Single-turn Q&A process

3. Select and orchestrate data operators based on the specific scenario. After you select operators, the operator orchestration area is displayed on the right. Configure the operator sequence and parameters, and click **Save and Next** in the lower right corner.

   For details about the applications, see **Application Scenarios**. For details about the operators, see **Preset Smart Refining Operators**.

   Note: Data orchestration is the key and most complex part of smart refining. Many scenarios and constraints must be considered. For text data like single-turn or multi-turn dialogues **(with or without persona settings)**, the start and end nodes (**Start Node** and **End Node**) handle both data input/output and format conversion. The following describes scenarios involving data format conversion:

   - When an arbitrary-format dataset enters the start node, it must be converted into the platform-compatible dataset format to enable processing by subsequent operators. Once all processing operators have completed their tasks, the end node will, by default, convert the output dataset back into the same format as the original input dataset.
   - The end node can be configured to output the dataset in any format, allowing you to choose the target format.
   - If no other operators are added between the start node and the end node, the system handles the task based on the following two conditions:
     - If the end node is set to the same format as the start node's input, no operations are performed on the dataset. In this case, the **Save and Next** button will be grayed out, preventing further configuration.
     - If the end node is set to a format different from the input, the task is treated as a pure format conversion. You can go to the next step to complete the subsequent configuration of the smart refining task.
   - After you click **Save and Next**, the current smart refining orchestration task, including the orchestration steps and previous configurations, will be saved. If the task is not complete, you can continue the subsequent configuration after the smart refining task is started next time.

4. Configure the generated dataset and resource information, and click **Run**.

Table 4-6 Parameter description

Parameter		Description	Example Value
Generate Dataset	Dataset Name	Name of a custom dataset. The name must start with a letter and end with a letter or digit. It can contain 2 to 63 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.	dataset

Parameter		Description	Example Value
	Storage Location	Choose **Object Storage Service – Bucket** or **Object Storage Service – Parallel File System** as the storage type. Click to select an OBS storage address or manually enter an OBS storage address. The storage address must start with **obs://** or **/** and end with a slash (**/**). It cannot contain double slashes (**//**) except in the prefix. For example, **obs://bucketname/path/** or **/bucketname/path/**.	obs://bucketname/path/
	Dataset Property	Configure dataset properties as required. You can configure tags by industry or language, or customize labels.	-
	Description	Only letters, digits, spaces, hyphens (-), underscores (_), commas (,), and periods (.) are allowed. It can contain a maximum of 200 characters.	-

Parameter		Description	Example Value
	Dataset Status	Only published datasets can be used by downstream tasks such as model development and training. • If you select **Publish Dataset**, the generated dataset is in the **Online** state on the **Asset Management** > **Data** > **My Data** page and can be directly used by downstream model training jobs. • If you do not select **Publish Dataset**, the generated dataset will be in the **Offline** state on the **Asset Management** > **Data** > **My Data** page and cannot be directly used by downstream model training jobs. You need to manually publish the dataset before using it.	Select **Publish Dataset**.
Resource Configuration	Resource Pool Type	Select a resource pool type as required.	Public resource pool
	Reference Specifications for Execution CPU Operator Instance	Data operators need compute resources for processing. Choose CPU or NPU resources based on the task and operator type.	**NPU (1 card) | (24 vCPUs) | MEMORY (192 GB)**

The smart refining task is complete when its **Latest Status** changes to **Dataset Generation Success**. The new data will be in **Asset Management** > **Data** > **My Data**.

Best Practice: Operator Orchestration Design Principles

Principle 1: Cleanse data before processing it.

Recommended sequence: Deduplicate → Format → Filter → Augment→ Synthesize

Principle 2: Reduce data before expanding it. Use the filter operator to reduce the data volume and then use the synthesis operator to expand it, improving the overall processing efficiency.

Principle 3: Place the synthesis operator at the end. The synthesis operator can only be used as the last processing step.

Principle 4: Maintain modal consistency. The entire workflow processes the same type of data and does not cross modalities.

Recommended Operator Orchestration Templates

Template 1: Basic data cleaning

```
Input → Format verification → Deduplication → Length filtering → Output
```

Template 2: Data cleaning + quality improvement

```
Input → Format verification → Deduplication → Sensitive word filtering → Quality score-based filtering → Output
```

Template 3: Data cleaning + synthesis and expansion

```
Input → Deduplication → Sensitive word filtering → Quality filtering → Q&A rewriting and synthesis → Output
```

Template 4: Full-process refinement

```
Input → Format conversion → Deduplication → Sensitive word filtering → Quality scoring → Length filtering → Rewriting and synthesis → Output
```

Follow-Up Operations

After a dataset is published, it can be directly used for model development, such as **model training**.

4.1.4 Managing Smart Refining Tasks

All refining tasks are displayed on the **Data Refining > Smart Refining** page. You can view the name/ID, associated dataset, latest running status, latest generated dataset, latest running time, creator, and supported operations of each task. This section describes how to perform operations on a refining task.

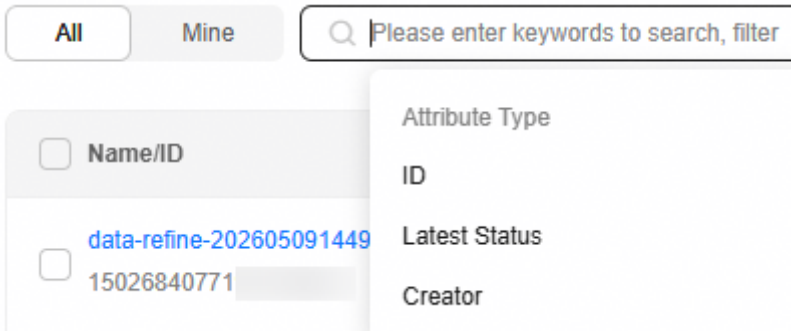
Managing Refining Tasks

Refining tasks can be filtered, searched, started, stopped, retried, edited, and deleted. The following describes these operations.

- **Filtering and searching for refining tasks**

  In the search box on the **Smart Refining** page, you can filter tasks by ID, Latest Status, or Creator, or enter a keyword to quickly find the target task. You can also click **My Created** to filter the refining tasks created by the current login user.

Figure 4-2 Filtering and searching for refining tasks



- **Starting a refining task**

  On the **Smart Refining** page, if you want to run a refining task whose **Latest Status** is Dataset generated, Not started, or Stopped again, click **Start** in the **Operation** column. In the **Start Smart Refining** panel, modify the refining configuration as required and click **OK**. For details about the parameters, see **Table 4-6**.

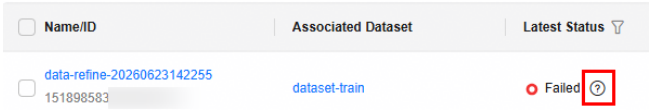
- **Stopping a refining task**

  On the **Smart Refining** page, click **Stop** in the **Operation** column of a running refining task. In the dialog box that is displayed, click **OK** to stop the task.

- **Retrying a refining task**

  On the **Smart Refining** page, for a task that fails to be executed, hover the pointer over the icon next to **Failed** to view the failure cause. After the error is handled, click **Retry** in the **Operation** column. In the **Start Smart Refining** dialog box, modify the refining configuration as required and click **OK** to retry the refining task. For details about the parameters, see **Table 4-6**.

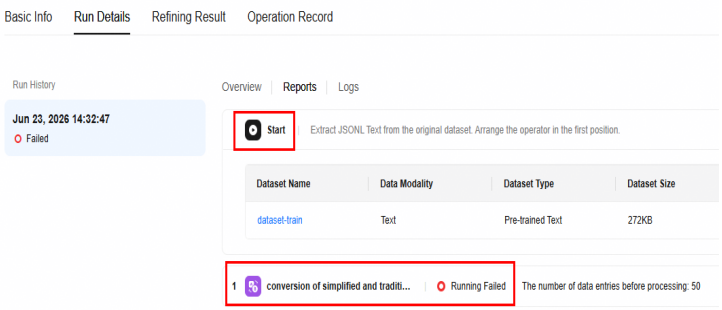
  Figure 4-3 Failed

  On the **Smart Refining** page, click the name of a refining task. On the **Run Details** tab page, click **Report** to view the operators that fail to run in the refining task.

  – For a new task: After the retry, the task is restarted from the operator that fails to run.
  – For a historical task: After the retry, the task is restarted from the Start operator.

Figure 4-4 Running report



- **Editing a refining task**

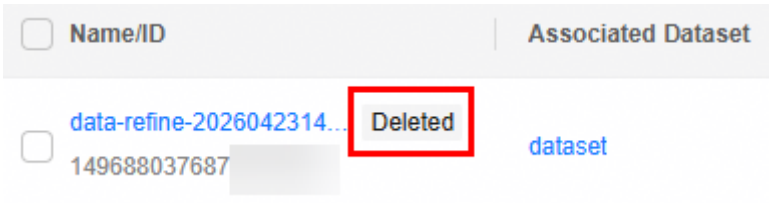
  On the **Smart Refining** page, for a refining task whose latest running status is not Running, click **Edit** in the **Operation** column. On the **Edit Smart Refining** page, adjust the operator sequence or parameters and click **Save** to complete the task editing.

- **Deleting a refining task**

  On the **Smart Refining** page, locate the row where the target refining task resides and click **Delete** in the **Operation** column. In the displayed **Delete** dialog box, click OK.

  Deleted tasks are not permanently removed. If you accidentally delete a task, you can restore it. On the **Smart Refining** page, click **Show deleted items** to view the deleted tasks (with the **Deleted** label on the right of the task name).

  Figure 4-5 Deleted tasks

  You can perform the following operations on deleted tasks:

  – Restoring a task: Click **Restore** in the **Operation** column and click **OK**.

  – Permanently deleting a task: A permanently deleted task cannot be restored. Click the task name. In the upper part of the page, click **permanently delete**. In the displayed dialog box, enter **DELETE** and click **OK**.

    Figure 4-6 Permanent deletion

- **Deleting refining tasks in batches**

  Select multiple refining tasks and click **Delete** in the upper right corner of the page to delete them in batches.

Managing Refining Task Details

On the **Smart Refining** page, click a task name to go to the task details page. The refining task details page displays the details of the current task. In the upper right corner of the page, you can click the start, retry, delete, or stop button based on the task status. For details, see **Managing Refining Tasks**.

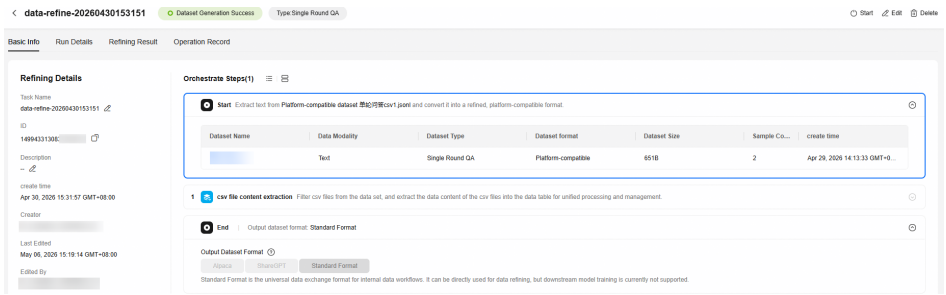
There are four tab pages on the refining task details page: **Basic Info**, **Run Details**, **Refining Result**, and **Operation Record**.

Basic Info

The left side of the **Basic Info** tab lists the refining details, including the task name, ID, task description, creation time, creator, last editing time, and editor. The task name and task description can be modified.

The right side of the **Basic Info** tab lists the orchestration details of the data operators used by the task.

Figure 4-7 Basic Info



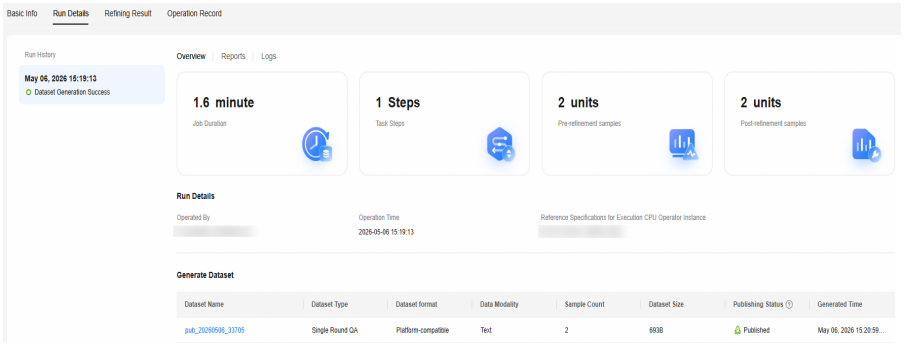
Run Details

The left side of the **Run Details** tab lists the running records of the task, including the execution time and result status.

The right side of the **Run Details** tab lists the task overview, report, and logs. The details are as follows:

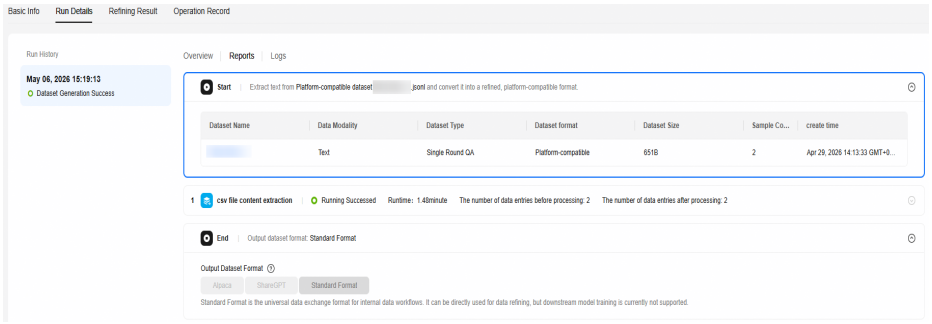
- **Overview**: displays the task duration, task steps, number of samples before refining, and number of samples after refining recorded in each execution. The details include the operator, operation time, and generated dataset.

Figure 4-8 Overview



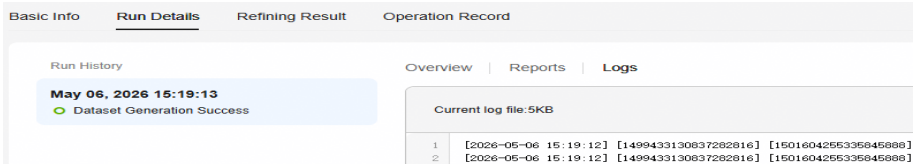
- **Reports**: displays the status of each data operator in the orchestration step, their run time, the number of data samples before and after processing, and the number of optimization hits. The report helps you intuitively determine whether the data processed by each operator meets the expectation.

Figure 4-9 Reports



- **Logs**: displays logs created while each refining task runs. These logs help you find issues quickly. You can search for specific logs using keywords or regular expressions on the log page. For details, see **Figure 4-10**.

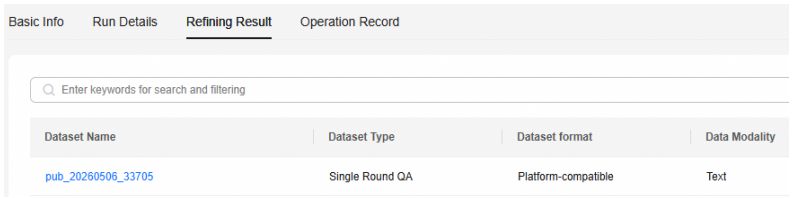
Figure 4-10 Logs



Refining Result

This tab page displays information about the datasets generated by the refining task, including the links to the datasets in the data assets.

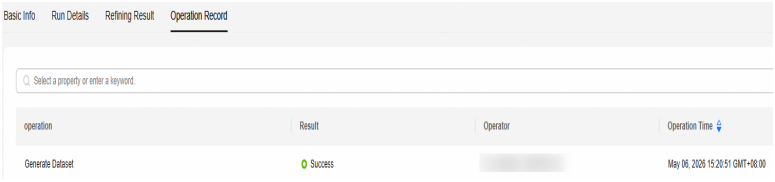
Figure 4-11 Refining Result



Operation Record

This tab page records all operations of the refining task, helping you view the operation status of the current task.

Figure 4-12 Operation Record



4.1.5 Smart Refining Templates

Description

Smart refining templates on ModelArts are best practices for data processing. They combine complex operations into simple workflows you can use with one click.

Core functions:

1. You do not need to start from scratch. You can reuse tested solutions from experienced engineers to ensure your data processing is professional and logical.
2. You only need to select a template that fits your service scenario when creating a task. The system will automatically load the full operator link. You can run the workflow directly or fine-tune it based on data characteristics, reducing data preparation time from hours to minutes.
3. The template-based system standardizes data processing, prevents quality issues from personal settings, and ensures high-quality training sets.

Built-in Templates

ModelArts provides the following built-in templates, covering the processing of text, image, and video data. For details about how to use these templates, see **Table 4-7**.

Table 4-7 Built-in templates

Template Name	Supported Dataset Modality	Supported Dataset Type	Use Case	Involved Operators
Word processing workflow	Text	Document	Use built-in operators to extract and process Word files and generate pre-trained text data.	Word Document Content Extraction
				Personal Data Anonymization
				Switching Between Simplified Chinese and Traditional Chinese

Template Name	Supported Dataset Modality	Supported Dataset Type	Use Case	Involved Operators
				Symbol Standardization
				Sensitive Word Filtering
				Filtering of the Incomplete Sentence at the End of a Paragraph
				N-gram Feature Filtering
				Text Length Filtering
				Pre-trained Text Classification
PDF processing workflow	Text	Document	Use built-in operators to extract and process PDF files and generate pre-trained text data.	PDF Content Extraction
				Personal Data Anonymization
				Switching Between Simplified Chinese and Traditional Chinese
				Symbol Standardization
				Sensitive Word Filtering
				Filtering of the Incomplete Sentence at the End of a Paragraph
				N-gram feature filtering
				Text Length Filtering
				Pre-trained Text Classification

Template Name	Supported Dataset Modality	Supported Dataset Type	Use Case	Involved Operators
Pre-trained text processing workflow	Text	Pre-trained text	Use built-in operators to process pre-trained text and generate cleaned pre-trained text data.	Switching between simplified Chinese and traditional Chinese
				Symbol Standardization
				N-gram feature filtering
				Text Length Filtering
				Pre-trained Text Classification
Image processing workflow	Image	Image	Use built-in operators to extract, deduplicate, label, and filter images, and generate summaries to produce processed image and text data.	Image and Text Extraction
				Image Metadata Filtering
				Image Deduplication
				Dangerous Situation Image Detection
				Pornographic Image Detection
				Violent and Terrorism Image Detection
Video processing workflow	Video	Video	Use built-in operators to split, deduplicate, label, and filter video clips, and generate processed video data.	Video Clipping
				Video Metadata Filtering
				Video Aspect Ratio Filtering
				Pornographic Video Detection
				Terrorism Video Detection
				Political Video Detection
				Motion Range Scoring
				Aesthetics Scoring

Template Name	Supported Dataset Modality	Supported Dataset Type	Use Case	Involved Operators
Single-turn Q&A process	Text	Single-turn Q&A	Use built-in operators to process single-turn Q&A data and generate cleaned single-turn Q&A data.	Personal Data Anonymization
				Switching between simplified Chinese and traditional Chinese
				Symbol Standardization
				Sensitive Word Filtering
				Filtering of the Incomplete Sentence at the End of a Paragraph
				N-gram feature filtering
				Text Length Filtering
Q&A ranking process	Text	Q&A sorting	Use built-in operators to process Q&A ranking data and generate cleaned Q&A ranking data.	Personal Data Anonymization
				Switching between simplified Chinese and traditional Chinese
				Symbol Standardization
				Sensitive Word Filtering
				Filtering of the Incomplete Sentence at the End of a Paragraph
				Text Length Filtering
Multi-turn Q&A process	Text	Multi-turn Q&A	Use built-in operators to process multi-turn Q&A data and generate cleaned multi-turn Q&A data.	Personal Data Anonymization
				Switching between simplified Chinese and traditional Chinese

Template Name	Supported Dataset Modality	Supported Dataset Type	Use Case	Involved Operators
				Symbol Standardization
				Sensitive Word Filtering
				Filtering of the Incomplete Sentence at the End of a Paragraph
				Text Length Filtering
Preference optimization process	Text	Preference optimization	Use built-in operators to process preference optimization data and generate cleaned preference optimization data.	Personal Data Anonymization
				Switching between simplified Chinese and traditional Chinese
				Symbol Standardization
				Sensitive Word Filtering
				Removing incomplete sentences at the end of a paragraph
				Text Length Filtering

4.1.6 Preset Smart Refining Operators

Smart refining operators are classified into **processing operators** and **synthesis operators**. A complete data processing process can be implemented by combining and orchestrating operators.

Table 4-8 Smart refining operators

Type	Category	Operator Name	Description
Text processing operators	Unclassified	**Start Node**	Serves as the first node in the smart refining orchestration process to receive the dataset to be refined. Provides data conversion for some text data **(single-turn dialogue, single-turn dialogue with persona setting, multi-turn dialogue, and multi-turn dialogue with persona setting)**. The conversion rules are as follows: • Platform-compatible datasets can skip conversion and go straight to the refining process. • Non-platform-compatible data (like Alpaca or ShareGPT formats) must first be converted to the platform-compatible format at the start node.

Type	Category	Operator Name	Description
		End Node	Serves as the end node in the smart refining orchestration process to output the refined dataset. Provides data conversion for some **text** data **(single-turn dialogue, single-turn dialogue with persona setting, multi-turn dialogue, and multi-turn dialogue with persona setting)**. The conversion rules are as follows: • If the dataset input at the start node is in the platform-compatible format, the end node outputs the dataset in the platform-compatible format by default. • If the dataset input at the start node is in a non-platform-compatible format (Alpaca or ShareGPT), the end node outputs the dataset in the same non-

Type	Category	Operator Name	Description
			platform-compatible format by default. • The end node can select a dataset format different from that of the start node as the final output dataset format.
	Data extraction	**Word Document Content Extraction**	Extracts text from a Word document and retain the contents, titles, and body of the original document, but does not retain images, tables, formulas, headers, and footers.
		CSV Content Extraction	Reads all text content from a CSV file and generates data in JSON format based on the key value of the file content type template.
		PDF Content Extraction	Extracts text from PDF files and converts the text into structured data. Texts, tables, and formulas can be extracted.

Type	Category	Operator Name	Description
	Data conversion	**Personal Data Anonymization**	Anonymizes or directly deletes sensitive personal information, such as mobile numbers, identity documents, email addresses, URLs, license plate numbers in China, IP addresses, MAC addresses, IMEIs, passports, and vehicle identification numbers.

Type	Category	Operator Name	Description
		Symbol Standardization	Searches for non-standardized symbols carried in the text for standardization and unified conversion. • Unified space: All Unicode spaces (such as U+00A0 and U+200A) are converted to standard spaces (U+0020). • DBC to SBC: Converts full-width characters in documents to half-width characters. • Punctuation normalization: The following symbols support a unified format: – {"?": "\?\? "} • Normalizes digits and symbols.

Type	Category	Operator Name	Description
		Custom Regular Expression Replacement	Uses the customized regular expression to replace the text content if the data items remain unchanged. Examples include: • Remove References and the content following References: **\nReferences[\s\S]*** • For the PDF content, remove the content before "0 Introduction". The content before the introduction is irrelevant to knowledge: **[\s\S]{0,10000}0 Introduction** • Delete the content irrelevant to knowledge before "1.1 Introduction to Java" from the PDF file: **[\s\S]{0,10000} 1\. 1 Introduction to Java**

Type	Category	Operator Name	Description	
		Date and Time Format Conversion	Automatically identifies the date, time, and week, and converts the date, time, and week based on the selected format.	
		Advertisement Data Filtering	Deletes a sentence that includes advertisement data from the text, based on a filtering granularity of a sentence.	
		Data filtering	Filtering Abnormal Characters	Searches for abnormal characters carried in each data record in the dataset and replaces the abnormal characters with null values. The data items remain unchanged. • Invisible characters, for example, U+0000-U+001F • Web page label symbols: <style></style> • Special space: [\u2000-\u2009]
	Custom Regular Expression Filtering		Deletes or retains the data that complies with the customized regular expression.	

Type	Category	Operator Name	Description	
	Custom Regular Expression Filtering		Deletes data that contains keywords.	
	Filtering the incomplete sentence at the end of a paragraph		Checks whether the content at the end of a paragraph is complete based on the sentence-level filtering granularity, and filters out the content if the content is incomplete.	
	Sensitive Word Filtering		Automatically detects and filters sensitive data such as pornography, violence, and politics in text.	
	Text Length Filtering		Retains the data within the specified length range based on the configured text length.	

Type	Category	Operator Name	Description	
	Sentence Feature Filtering		Uses punctuations in a document as sentence separators and collects statistics on the length of each sentence. If the average length of a document is greater than the configured length, the document is retained. Otherwise, the entire document is deleted. The filtering is based on the following: • Average length of sentences to be retained	
	Data labeling operators		Prohibited Text Detection	Analyzes the input Chinese text content and returns the JSON structured result indicating whether the text contains forbidden content.
		Personal Privacy Identification	Analyzes the input Chinese text content and returns the JSON structured result indicating whether the text contains privacy content.	
		Garbage Content Text Detection	Analyzes the input Chinese text content and returns the JSON structured result indicating whether the text contains junk content.	

Type	Category	Operator Name	Description	
		Ad Text Detection	Analyzes the input Chinese text content and returns the JSON structured result indicating whether the text contains junk advertisement content.	
		Pornographic Text Detection	Analyzes the input Chinese text content and returns the JSON structured result indicating whether the text contains pornographic content.	
		Abusive Text Detection	Analyzes the input Chinese text content and returns the JSON structured result indicating whether the text contains abusive content.	
		Politically Sensitive Text Detection	Analyzes the input Chinese text content and returns the JSON structured result indicating whether the text contains politically sensitive content.	

Type	Category	Operator Name	Description	
		Pre-trained Text Classification	Classifies the pre-trained text, such as news, education, and health. The supported languages include Chinese and English.	
	Text synthesis operators	Data synthesis	**Data Generation**	Generates similar Q&As from a single sample, injects specific character roles into Q&As, and allows one-click adjustment of Q&A difficulty to implement large-scale customized data synthesis.
	Video processing operators	Data extraction	**Video Duration Segmentation**	Segments the source video into short videos of fixed duration. The fixed duration can be configured, and the value ranges from 1 to 5 minutes.
Video Clipping			Splits a long video into short video clips based on the scene change. If the length of a clip exceeds the specified time threshold, the clip is further split by duration.	

Type	Category	Operator Name	Description	
Data conversion		Video Cropping	Video cropping is to crop unnecessary elements in a video, such as subtitles, logos, watermarks, borders, and dense text, and filter out video files whose area ratio after cropping exceeds the preset threshold. Before using this function, you need to execute the subtitle, logo, watermark, border, and dense text recognition operators.	
Data filtering		Video Metadata Filtering	Filters videos based on the video metadata (frame rate, resolution, and video duration) and retains only the videos that meet the specified conditions. Note: The standard frame rate of a movie is 24 FPS or 30 FPS.	

Type	Category	Operator Name	Description	
		Video Aspect Ratio Filtering	Filters videos based on the aspect ratio. The aspect ratio is a ratio of a width to a height of a video image.	
Data labeling		**Pornographic Video Detection**	Labels pornographic content.	
		Terrorism Video Detection	Labels violent and terrorism content.	
		Political Video Detection	Labels political content.	

Type	Category	Operator Name	Description	
		Motion Range Scoring	Calculates and scores the motion range of each pixel in each frame, and identifies videos with too fast motion (for example, > 100 optical flows) or too slow motion (for example, ≤ 2 optical flows). A larger value indicates faster motion.	
		Aesthetics Scoring	Scores the aesthetics of a video from the following dimensions: content (attractive and clear), composition (good object position), color (vital and pleasant), light (obvious contrast), and track (continuous and stable). Scores range from 0 to 1. Higher scores indicate superior aesthetics. A score > 0.95 signifies high-aesthetics video.	
		Watermark Detection	Identifies whether a video contains watermarks.	

Type	Category	Operator Name	Description	
		Subtitle Detection	Identifies whether a video contains subtitles.	
		Video Black Bar Detection	Identifies whether a video contains black bars.	
		Dense Text Detection	Identifies whether a video contains dense text. A video in which the proportion of dense text area exceeds the specified proportion is a video with dense text. By default, a video with a cropping area proportion greater than or equal to 7% is a video with dense text.	
		Video Classification	Returns the label classes of video content. There are 10 classes for L1, 39 classes for L2, 93 classes for L3, and 2219 classes for L4.	

Type	Category	Operator Name	Description	
		Video Synopsis Generation (Simplified)	Extracts frames from a video and generates a simplified video synopsis through model inference.	
		Video Synopsis Generation (Detailed)	Extracts frames from a video and generates a detailed Chinese video synopsis through model inference.	
		Chinese Video Synopsis Generation (Detailed)	Extracts frames from a video and generates a detailed Chinese video synopsis through model inference.	

Type	Category	Operator Name	Description	
		Posture Detection	Extracts eight frames from a video, marks key points on the images, outputs the task bounding box and key point coordinates, and determines whether there are persons in the video based on the coordinates.	
		Camera Motion Description	Calculates and infers optical flow by extracting frames from a video to output the lens type of the video.	
Image processing operators	Data extraction	**Image and Text Extraction**	Extracts JSON text and images from the compressed image-text package and performs structured parsing (Base64 encoding) on the images to facilitate the use of image-text processing operators.	
	Data filtering	**Image Metadata Filtering**	Cleans image/text data based on the image width and height, file size, and aspect ratio threshold.	

Type	Category	Operator Name	Description	
		Image Deduplication	Filters out duplicate image-text pairs after image structuring.	
	Data labeling	Pornographic Image Detection	Labels image operators.	
		Dangerous Situation Image Detection	Labels dangerous situation images.	
		Violent and Terrorism Image Detection	Filters out violent and terrorism images.	

Start Node

- Supported file formats: Applicable to all dataset types. However, it specifically provides data format conversion capabilities for "text > single-turn dialogue, single-turn dialogue with persona, multi-turn dialogue, and multi-turn dialogue with persona."

- Note: All dataset formats will be converted into the platform-compatible format after processing at the start node.

  Provides data format conversion for some text data **(single-turn dialogue, single-turn dialogue with persona setting, multi-turn dialogue, and multi-turn dialogue with persona setting)**. The conversion rules are as follows:

  – Platform-compatible datasets can skip conversion and go straight to the refining process.

  – Non-platform-compatible data (like Alpaca or ShareGPT formats) must first be converted to the platform-compatible format at the start node for subsequent processing by data operators.

- Parameter configuration example

  None

- Conversion example

  Dataset format before processing at the input node: platform-compatible, Alpaca, or ShareGPT format.

  Dataset format after processing at the input node: platform-compatible format.

End Node

- Supported file formats: Applicable to all dataset types. However, it specifically provides data format conversion capabilities for "**text > single-turn dialogue, single-turn dialogue with persona, multi-turn dialogue, and multi-turn dialogue with persona**." You can also choose to output data in different formats.
- Note:

  After smart refining is complete for a dataset, the end node can convert the data format for the specified data type. The conversion rules are as follows:

  – If the dataset input at the start node is in the platform-compatible format, the end node outputs the dataset in the platform-compatible format by default.

  – If the dataset input at the start node is in a non-platform-compatible format (Alpaca or ShareGPT), the end node outputs the dataset in the same non-platform-compatible format by default.

  – The end node can also select a dataset format different from that of the start node as the final output dataset format.

- Parameter configuration example

  None

- Conversion example

  Dataset format before processing at the input node: any format.

  Dataset format after processing at the output node: any format.

Word Document Content Extraction

- Applicable file format: document > docx
- Parameters:

Type of the content to be extracted: Extracts text from a Word document and retains the titles and body of the original document, but does not retain images, formulas, headers, and footers. Nested tables cannot be extracted.

- Parameter configuration example:

  No parameters need to be set. By default, the contents, title, and body of the original document are retained, and the images, tables, formulas, headers, and footers are not retained.

- Extraction example

  Local import: {"fileName":"JAVA from Beginner to Master.docx","original_path": "Local Import","text":"JAVA is a cross-platform..."}

  OBS import: {"fileName":"JAVA from Beginner to Master.docx","original_path": "nlp_data/word/JAVA from Beginner to Master.docx","text":"JAVA is a cross-platform..."}

  AI Gallery: {"fileName":"JAVA from Beginner to Master.docx","original_path": "Gallery Subscription","text":"JAVA is a cross-platform..."}

CSV Content Extraction

- Applicable dataset types: Text > single-turn Q&A, single-turn Q&A (with persona), and Q&A sorting.
- Parameter description

  Type of content to be extracted: Reads all text content from a CSV file and generates data in JSON format based on the key value of the file content type template.

- Parameter configuration example

  No parameters need to be set.

- Extraction example

  If the extracted CSV content is "Hello, please introduce yourself. I am Pangu model.", the extracted content is {"context":"Hello, please introduce yourself","target":"I am Pangu model."}

PDF Content Extraction

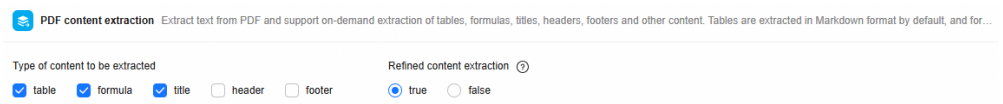
- Applicable dataset type: Document > PDF.
- Parameter description

  Type of content to be extracted: By default, the text, tables, formulas, and titles are retained. You can select the type to be saved. The types that are not selected will be removed.

  Refined content extraction: indicates whether to support image content extraction after layout analysis.

  Available formats for table extraction: The default format is Latex. The table can be converted to the Markdown format.

- Parameter configuration example



- Extraction example

  Local import: {"fileName":"JAVA from Beginner to Master.pdf","original_path":"Local Import","text":"JAVA is a cross-platform..."}.

  OBS import: {"fileName":"JAVA from Beginner to Master.pdf","original_path":"nlp_data/pdf/JAVA from Beginner to Master.pdf","text":"JAVA is a cross-platform..."}.

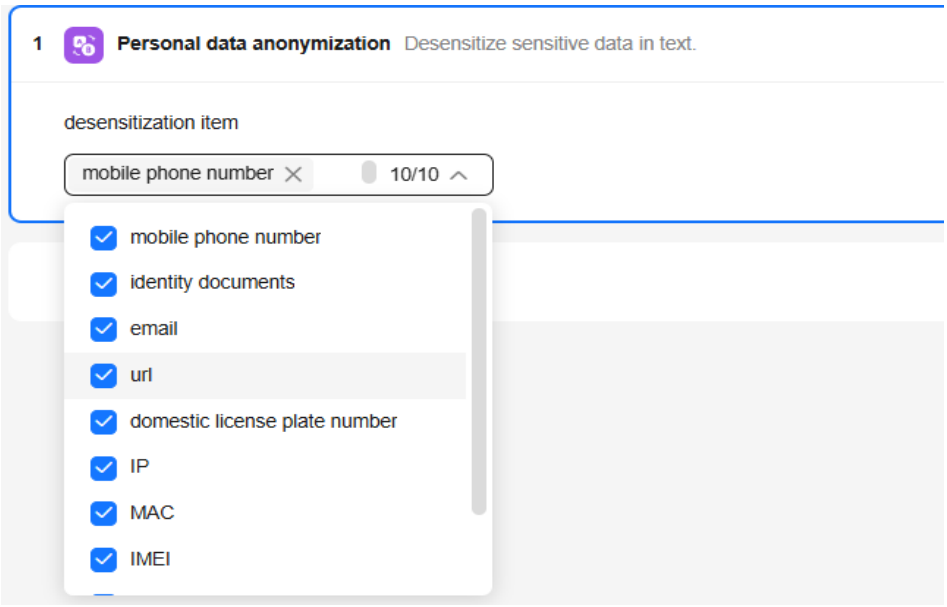
  AI Gallery: {"fileName":"JAVA from Beginner to Master.pdf","original_path":"Gallery Subscription","text":"JAVA is a cross-platform..."}.
- Operator restrictions

  The PDF content extraction process stops after 24 hours if it handles a lot of data. Split the data before running the process.

Personal Data Anonymization

- Applicable dataset type: Text.
- Parameter description

  Type of content to be converted: Anonymizes sensitive personal information in the text, such as mobile numbers, ID cards, email addresses, URLs, license plate numbers in China, IP addresses, MAC addresses, IMEIs, passports, and vehicle identification numbers. By default, all options are selected. You can also select some of them.
- Parameter configuration example



- Conversion example

  Before refining: "Data is from www.test.com."

  After refining: "Data is from *******."

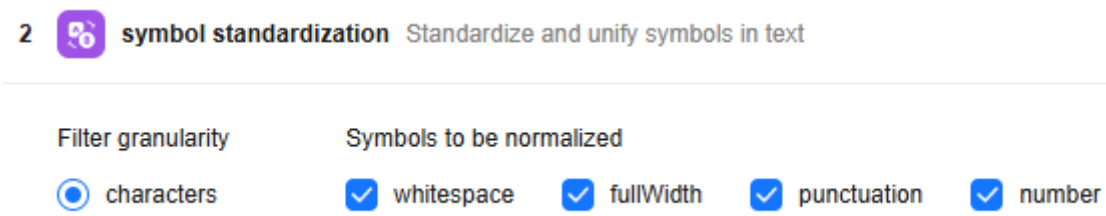
Symbol Standardization

- Applicable dataset type: Text.
- Parameter description

  Type of content to be converted: Non-standard symbols in the text can be converted to standard symbols. The non-standard symbols include spaces,

DBC symbols, punctuations, and number symbols. By default, all non-standard symbols are selected. The filtering granularity is character.

- Parameter configuration example



Custom Regular Expression Replacement

- Applicable dataset type: Text.
- Parameter description

  Type of content to be converted: Uses the customized regular expression to replace the text content if the data items remain unchanged.

- Parameter configuration example

- Conversion example

  Before refining: {"text":"This is the main content aeiou in the test aeiou. "}.

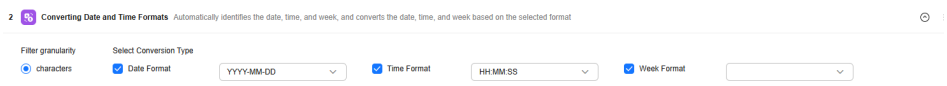
  After refining: {"text":"This is the main content 11111 in the test 11111. "}.

Date and Time Format Conversion

- Applicable dataset type: Text.
- Parameter description

  Type of content to be converted: Automatically identifies the date, time, and week, and converts the date, time, and week based on the selected format. The conversion types include date format, time format, and week format. By default, all of them are selected. You can also select some of them.

- Parameter configuration example



- Conversion example

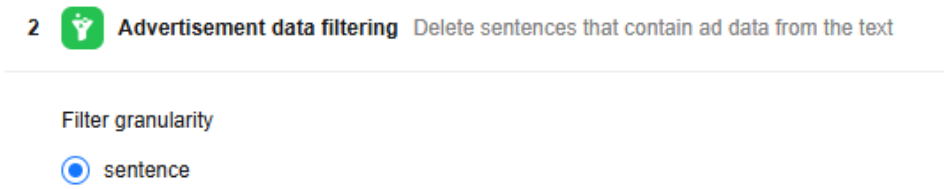
  Before refining: {"text":"Today is Monday, March 3, 2025. The rain is heavy in the morning. "}.

  After refining: {"text":"Today is Monday, 2025-03-03 00:00:00. The rain is heavy in the morning. "}.

Advertisement Data Filtering

- Applicable dataset type: Text.
- Parameter description

  Type of content to be filtered: Deletes a sentence that includes advertisement data from the text, based on a filtering granularity of a sentence.
- Parameter configuration example



- Filtering example

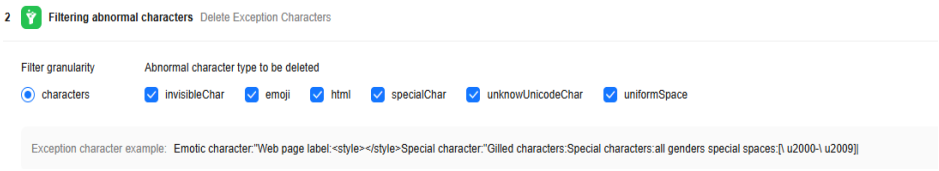
  Before refining: {"text":"Specific discount! Buy our products and enjoy a discount of up to 50%! Click the link below to avail the discount at https://example.com. Seize this opportunity now and take action! "}.

  After refining: {"text":""}.

Filtering Abnormal Characters

- Applicable dataset type: Text.
- Parameter description

  Type of content to be converted: Searches for abnormal characters carried in each data record in the dataset and replaces the abnormal characters with null values. The data items remain unchanged. Types of abnormal characters include invisible characters, emojis, web page labels, special characters, garbled characters, and special spaces. By default, all types are selected. You can also select some of them.
- Parameter configuration example



- Filtering example

  Before refining: {"text":"Test exception. <style></style>Haha. Limited-time offer! ☺"}.

  After refining: {"text":"Test exception. Haha. Limited-time offer!"}.

Custom Regular Expression Filtering

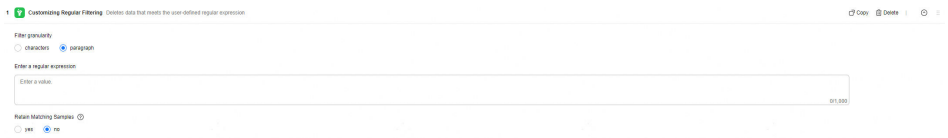
- Applicable dataset type: Text.
- Parameter description

  Type of content to be filtered: Filters content based on a custom regular expression. The filtering granularity can be character (default) or paragraph.

  Regular expression: Enter the regular expression required for custom regular expression filtering.

Retain Matching Samples: This parameter is displayed when the type of content to be filtered is paragraph. The default value is **false**.

- Parameter configuration example



- Filtering example

  Filtering out the content following "References"

  Before refining: {"text":"This is the body content. References [1] Author 1, Article 1, Journal 1, 2021.[2] Author 2, Article 2, Journal 2, 2022."}.

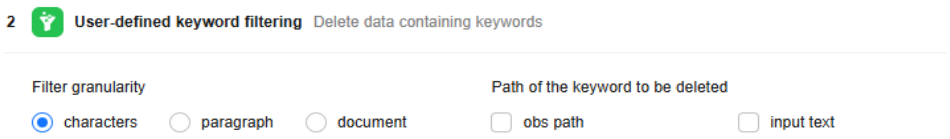
  After refining: {"text":"This is the body content. "}.

Custom Keyword Filtering

- Applicable dataset type: Text.
- Parameter description

  Type of content to be filtered: The filtering granularity can be character (default), paragraph, or document. The path of the keyword to be deleted supports keyword import from OBS and text input.

- Parameter configuration example



- Filtering example

  For example, filter by keyword **test**.

  Before refining: {"text":"Keyword test. This is a test data record. "}.

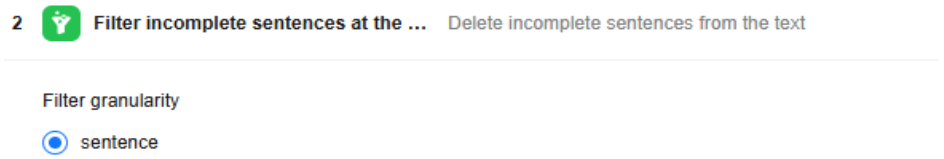
  After refining: {"text":"Keyword. This is a test data record. "}.

Filtering of the Incomplete Sentence at the End of a Paragraph

- Applicable dataset type: Text.
- Parameter description

  Type of content to be filtered: Checks whether the content at the end of a paragraph is complete based on the sentence-level filtering granularity, and deletes the content if the content is incomplete.

- Parameter configuration example



- Filtering example

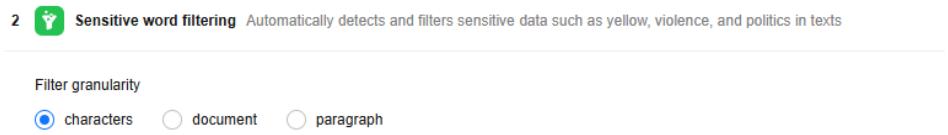
  Before refining: "Java is an object-oriented programming language. Use Java. "

After refining: "Java is an object-oriented programming language."

Sensitive Word Filtering

- Applicable dataset type: Text.
- Parameter description

  Type of content to be filtered: Automatically detects and filters sensitive data such as pornographic, violent, and political content in the text. Sensitive words need to be preset. The filtering granularity can be character (default), paragraph, or document.
- Parameter configuration example



- Filtering example

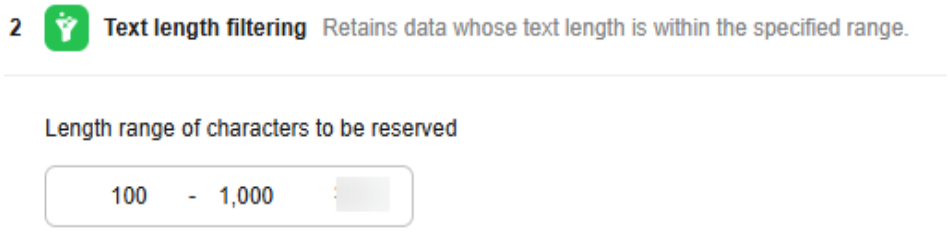
  Before refining: {"text":"prostitute test"}.

  After refining: {"text":"test"}.

Text Length Filtering

- Applicable dataset type: Text.
- Parameter description

  Type of content to be filtered: Retains data within the specified text length. By default, the length of the characters to be reserved ranges from 100 to 1000 characters, which can be modified. The minimum value is 1.
- Parameter configuration example



- Filtering example

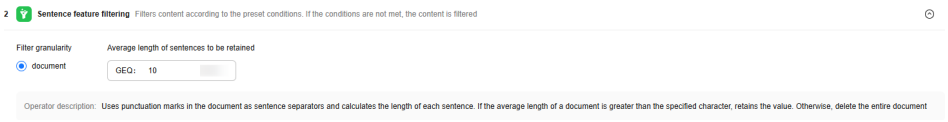
  Before refining: {"text": "Test length"}

  After refining: {"text":""}

Sentence Feature Filtering

- Applicable dataset type: Text.
- Parameter description

  Type of content to be filtered: Filters the content based on the document filtering granularity and the average sentence length to be retained. If the content does not meet the requirements, the content is filtered out. The default value is greater than or equal to 10 characters, which can be modified. The minimum value is **1**.
- Parameter configuration example



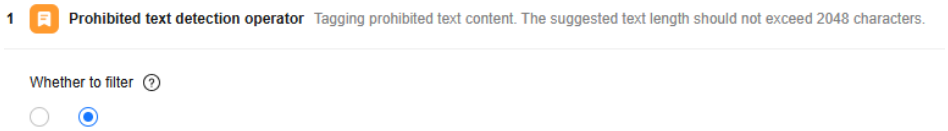
- Filtering example

  Before refining: {"text":"In a small village, there is a legend. In the legend, a mysterious fox appears in the village forest every full moon night. "}.

  After refining: {"text":""}.

Prohibited Text Detection

- Applicable dataset types: Q&A sorting, single-turn Q&A, and single-turn Q&A (with persona) > JSONL.
- Parameter description: If filtering is enabled, the filtering operator is used. Otherwise, the filtering operator is not used.
- Parameter configuration example

- Filtering example

  Before labeling:

  {"text": "Do you have QQ sales shareholder data?"}

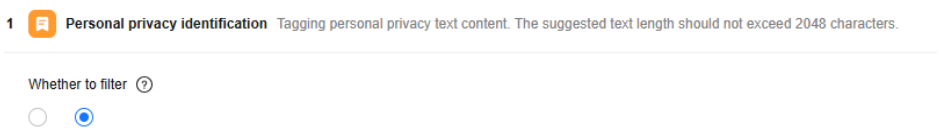
  After labeling:

  {"text":"Do you have QQ sales shareholder data?","text_ban_moderation":{"suggestion":"block","details":{"confidence":1.0,"label":"violation_info","risk_level":2,"segments":[{"segment":"QQ sales shareholder data"},{"segment":"Shareholder data"},{"segment":"Shareholder data & sales"},{"segment":"Sales shareholder data"}],"suggestion":"block"}}}

  suggestion indicates whether the file passes the check. **pass** indicates that the file passes the check and no problem occurs. **review** indicates that manual review is required. You can choose to bypass or block the file based on your review policy. **block** indicates that the file to be reviewed is problematic.

Personal Privacy Identification

- Applicable dataset types: Q&A sorting, single-turn Q&A, and single-turn Q&A (with persona) > JSONL.
- Parameter description: If filtering is enabled, the filtering operator is used. Otherwise, the filtering operator is not used.
- Parameter configuration example

- Filtering example

  Before labeling:

{"text": "You save my MAC address: 20-6E-D4-88-F3-98"}

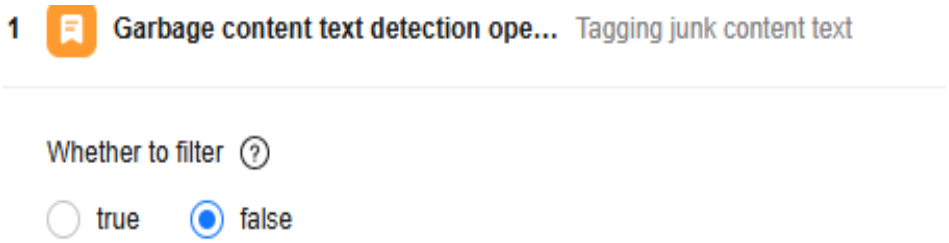
After labeling:

{"text": "You save my MAC address: 20-6E-D4-88-F3-98","text_pii_moderation":{"suggestion":"block","details":[{"start":33,"end":50,"length":17,"data":"20-6E-D4-88-F3-98","category":"MAC_ADDRESS"}]}}

suggestion indicates whether the file passes the check. **pass** indicates that the file passes the check and no problem occurs. **review** indicates that manual review is required. You can choose to bypass or block the file based on your review policy. **block** indicates that the file to be reviewed is problematic.

Garbage Content Text Detection

- Applicable dataset types: Q&A sorting, single-turn Q&A, and single-turn Q&A (with persona) > JSONL.
- Parameter description: If filtering is enabled, the filtering operator is used. Otherwise, the filtering operator is not used.
- Parameter configuration example



- Filtering example

  Before labeling:

  {"text": "[Kaiyuan fake certificate 848777596_qq Hefei fake certificate uhc0tm] What does it mean_Kaiyuan false certificate 848777596_qq Hefei false certificate uhc0tm Translation_Phonetic mark_Pronunciation_Usage_Example sentence_Online translation_Youdao dictionary"}

  After labeling:

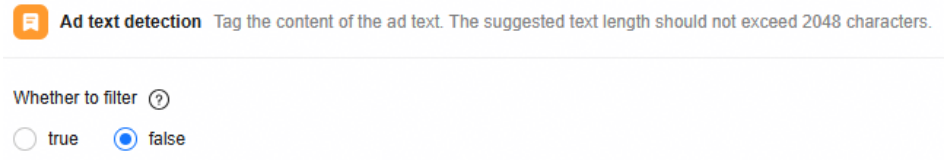
  {"text":"[Kaiyuan fake certificate 848777596_qq Hefei fake certificate uhc0tm] What does it mean_Kaiyuan false certificate 848777596_qq Hefei false certificate uhc0tm Translation_Phonetic mark_Pronunciation_Usage_Example sentence_Online translation_Youdao dictionary","text_spam_moderation":{"details":[{"confidence":1.0,"label":"abuse","risk_level":2,"segments":[{"segment":"tm's"}]},"suggestion":"block"]},"suggestion":"block"}}

  suggestion indicates whether the file passes the check. **pass** indicates that the file passes the check and no problem occurs. **review** indicates that manual review is required. You can choose to bypass or block the file based on your review policy. **block** indicates that the file to be reviewed is problematic.

Ad Text Detection

- Applicable dataset types: Q&A sorting, single-turn Q&A, and single-turn Q&A (with persona) > JSONL.

- Parameter description: If filtering is enabled, the filtering operator is used. Otherwise, the filtering operator is not used.
- Parameter configuration example



- Filtering example

  Before labeling:

  {"context":"On sale for inventory clearance. All items are only CNY2.","target":"The price is cheap."}

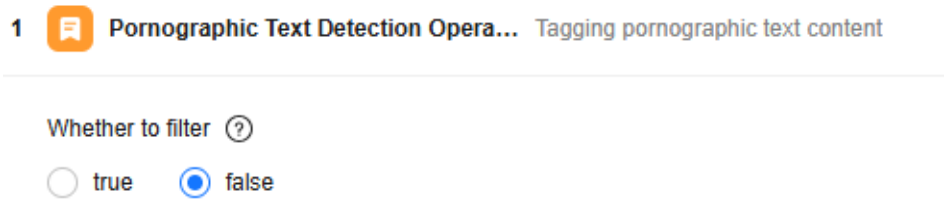
  After labeling:

  {"context":"On sale for inventory clearance. All items are only CNY2.","target":"The price is cheap.","text_ad_moderation":{"details":[],"suggestion":"pass"}}

  suggestion indicates whether the file passes the check. **pass** indicates that the file passes the check and no problem occurs. **review** indicates that manual review is required. You can choose to bypass or block the file based on your review policy. **block** indicates that the file to be reviewed is problematic.

Pornographic Text Detection

- Applicable dataset types: Q&A sorting, single-turn Q&A, and single-turn Q&A (with persona) > JSONL.
- Parameter description: If filtering is enabled, the filtering operator is used. Otherwise, the filtering operator is not used.
- Parameter configuration example



- Filtering example

  Before labeling:

  {"text": "XXX navigation, come and enjoy hardcore action now, fill your life with erotica and excitement, wait no more..."}

  After labeling:

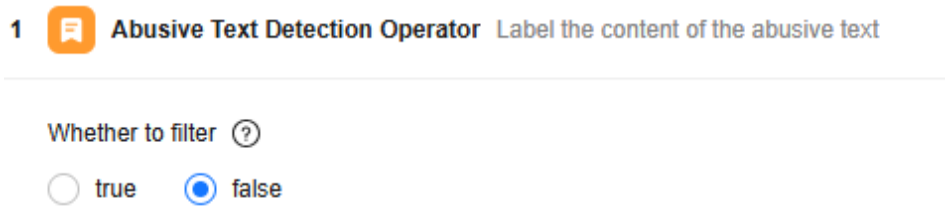
  {"text":"XXX navigation, come and enjoy hardcore action now, fill your life with erotica and excitement, wait no more...","text_porn_moderation":{"suggestion":"block","details":"[{'confidence': 1.0, 'label': 'porn_violence', 'risk_level': 2, 'segments': [{'segment': 'hardcore action'}, {'segment': 'XXX navigation'}], 'suggestion': 'block'}]"}}

  suggestion indicates whether the file passes the check. **pass** indicates that the file passes the check and no problem occurs. **review** indicates that

manual review is required. You can choose to bypass or block the file based on your review policy. **block** indicates that the file to be reviewed is problematic.

Abusive Text Detection

- Applicable dataset types: Q&A sorting, single-turn Q&A, and single-turn Q&A (with persona) > JSONL.
- Parameter description: If filtering is enabled, the filtering operator is used. Otherwise, the filtering operator is not used.
- Parameter configuration example



- Filtering example

  Before labeling:

  {"text":"Who wants to die with you? Die by yourself."}

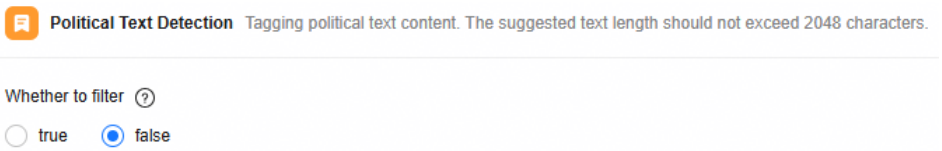
  After labeling:

  {"text":"Who wants to die with you? Die by yourself.","text_abuse_moderation":{"details":[{"confidence":0.9998,"label":"abuse","risk_level":2,"segments":[],"suggestion":"block"}],"suggestion":"block"}}

  suggestion indicates whether the file passes the check. **pass** indicates that the file passes the check and no problem occurs. **review** indicates that manual review is required. You can choose to bypass or block the file based on your review policy. **block** indicates that the file to be reviewed is problematic.

Politically Sensitive Text Detection

- Applicable dataset types: Q&A sorting, single-turn Q&A, and single-turn Q&A (with persona) > JSONL.
- Parameter description: If filtering is enabled, the filtering operator is used. Otherwise, the filtering operator is not used.
- Parameter configuration example



- Filtering example

  Before labeling:

  {"text": "But the authorities have never deigned to explain these online voices of doubt, opting instead for direct censorship."}

  After labeling:

{"text":"But the authorities have never deigned to explain these online voices of doubt, opting instead for direct censorship.","text_polInfo_moderation": {"suggestion":"block","details":"[{'confidence': 1.0, 'label': 'politics', 'risk_level': 3, 'segments': [{'segment': 'authorities'}], 'suggestion': 'block'}]"}}

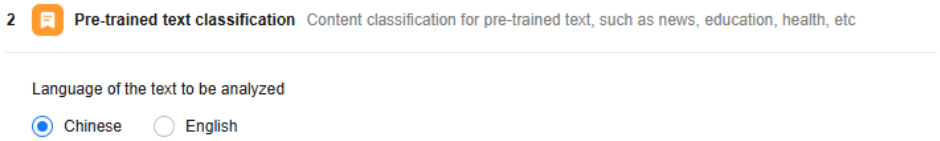
suggestion indicates whether the file passes the check. **pass** indicates that the file passes the check and no problem occurs. **review** indicates that manual review is required. You can choose to bypass or block the file based on your review policy. **block** indicates that the file to be reviewed is problematic.

Pre-trained Text Classification

- Applicable dataset type: Document > pre-trained text.
- Parameter description

  Type of content to be labeled: Classifies the content of the pre-trained text, for example, news, education, and health. The supported languages include Chinese and English. The default language is Chinese.

- Parameter configuration example



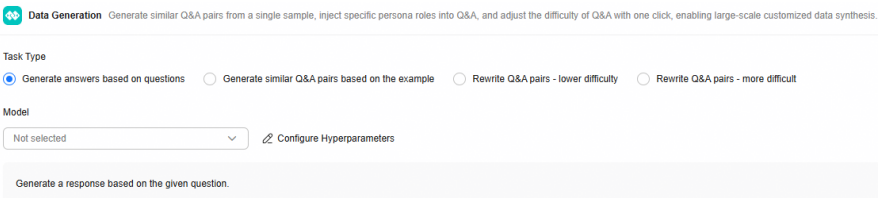
- Labeling example

  {"fileName":"News Labeling Test.docx","text":" (Beijing, March 3, reporter Xu Peiyu) The People's Bank of China released the financial market operation report in January this year. In January, the bond market in China issued bonds worth CNY5,102.75 billion. Of which, government bonds were issued for CNY1,018.5 billion, local government bonds for CNY557.57 billion, financial bonds for CNY704.21 billion, corporate credit bonds for CNY1,279.17 billion, credit asset-backed securities for CNY2.73 billion, and interbank certificates of deposit for CNY1,514.78 billion. \nAs of the end of January, the bond market custody balance in China was CNY178.2 trillion. Of which, the custody balance of the interbank market was CNY156.9 trillion, and that of the exchange market was CNY21.3 trillion. \nAs of the end of January, the custody balance of foreign institutions in the Chinese bond market was CNY4.2 trillion, accounting for 2.3% of the custody balance of the Chinese bond market. Of which, the bond custody balance of foreign institutions in the interbank bond market was CNY4.1 trillion. By bond type, foreign institutions held CNY2.0 trillion of government bonds (48.8%), CNY1.1 trillion of certificates of deposit (25.8%), and CNY0.9 trillion of policy bank bonds (20.8%). \n","pre_classification":"Economy"}

Data Generation

- Function: Generates similar Q&As from a single sample, injects specific character roles into Q&As, and allows one-click adjustment of Q&A difficulty to implement large-scale customized data synthesis.
- Applicable dataset type: Document > pre-trained text and single-turn Q&A.
- Parameter description
  - Generation scenario

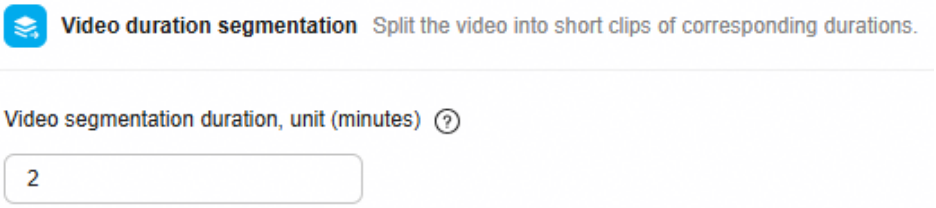
        - If the dataset type is document or pre-trained text, Q&A pairs can be generated based on the pre-trained text.
        - If the dataset type is single-turn Q&A, similar Q&A pairs can be generated based on samples, and Q&A pairs can be rewritten to be easier or more difficult.
    - Model: Select the model for data generation. Click **Configure Hyperparameter**. You can set default parameters or customized parameters as required.
- Parameter configuration example



Video Duration Segmentation

- Applicable file format: video > mp4/avi.
- Parameter description

  Video segmentation duration: You can set this parameter to determine the duration of the video after segmentation. The value ranges from 1 to 5 minutes. If the source video duration does not meet the segmentation requirements, the source video is retained.
- Function: The operator is used to segment the source video into short videos of fixed duration. The fixed duration can be configured, and the value ranges from 1 to 5 minutes. The operator efficiency is improved if you segment the video to reduce the video length and then use the shot segmentation function.
- Scenario:
  - Supported scenario
    - The video duration is longer than 1 minute.
  - Unresolved issue
    - The video duration is less than 1 minute.
- Parameter configuration example



Video Clipping

- Applicable file format: video > mp4/avi.

- Parameter description

  Video to be clipped: Videos that meet the resolution, duration, and frame rate criteria are clipped.

  Specifications after video clipped: The maximum duration of a single video slice can be customized. If the duration of the first clip slice exceeds the specified value, the video will be further clipped. The final clip result is less than or equal to the specified threshold.

- Scenario:
  - Supported scenario
    - There are significant scene changes, including direct switching or fade-in and fade-out.
  - Unresolved issue
    - The content captured in the same scenario changes, but the content similarity is high.

Video Cropping

- Applicable file format: video > mp4/avi.
- Parameter description

  Items to be cropped: Remove useless information such as subtitles, logos, watermarks, borders, and dense text from videos.

  crop_ratio_threshold: The ratio of the cropped video area to the original video area is the cropping area ratio. The default ratio threshold is 0.3.

  restore_over_cropped: Whether to retain the original video when the cropping ratio is greater than the threshold. If yes, the video is retained. Otherwise, the video is filtered out.

- Scenario:
  - Supported scenario
    - The subtitle, logo, watermark, border, and dense text recognition operators must be executed first.
  - Unresolved issue
    - The subtitle, logo, watermark, border, and dense text recognition operators have not been executed first.
    - After cropping, videos that are too small or have an improper aspect ratio cannot be retained.

Video Metadata Filtering

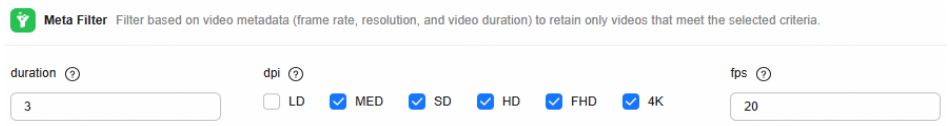
- Applicable file format: video > mp4/avi.
- Parameter description

  Resolution to be reserved: Select a resolution to be reserved. Videos that do not meet the selected resolution will be filtered out.

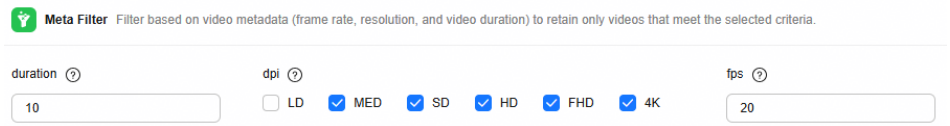
  Retention period: The default value is 3. Videos whose duration is shorter than the retention period will be filtered out.

Frame rate to be reserved: The standard frame rate of a movie is 24 FPS or 30 FPS. Videos whose frame rate is less than the frame rate to be reserved will be filtered out.

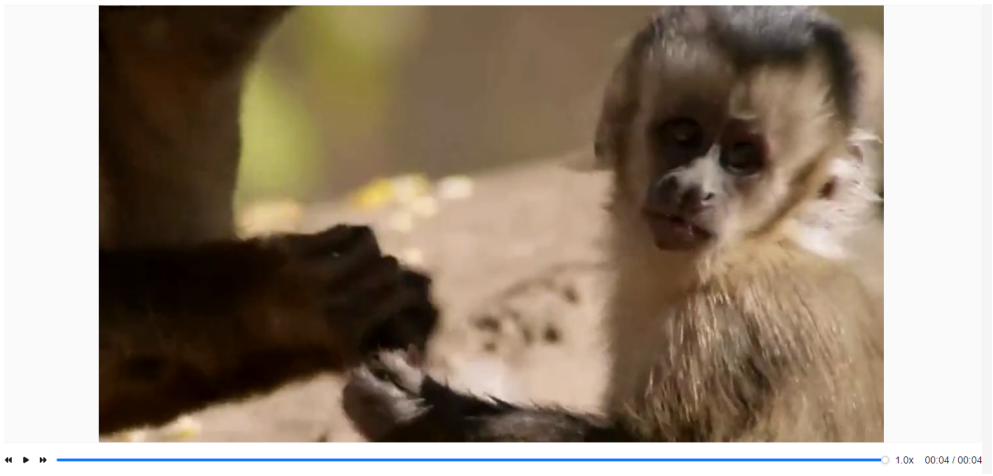
- Parameter configuration example



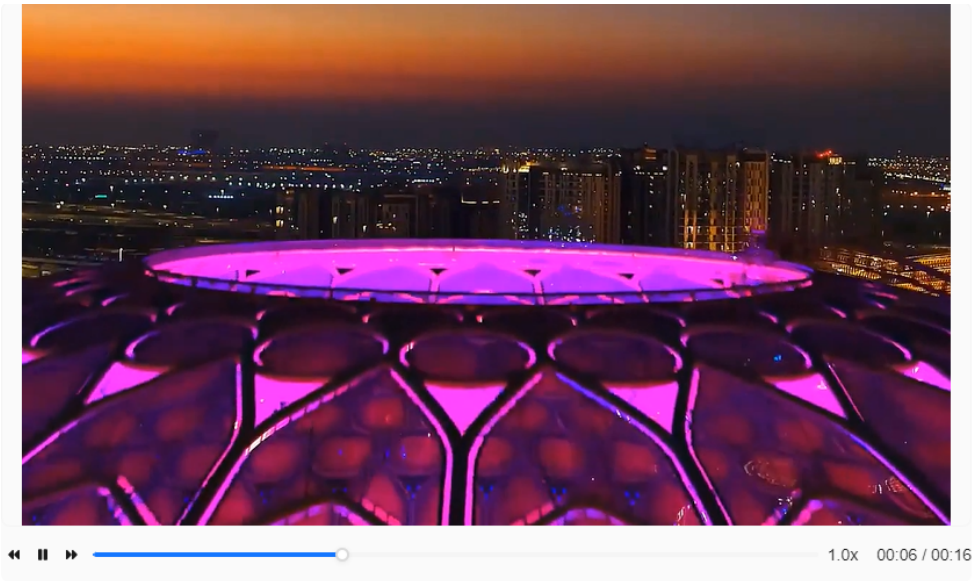
- Example: Set the retention period to be greater than or equal to 10s.



Before filtering: The duration of one video is 4s, and the duration of the other video is 16s.



After filtering: Only the video whose duration is 16s is retained.



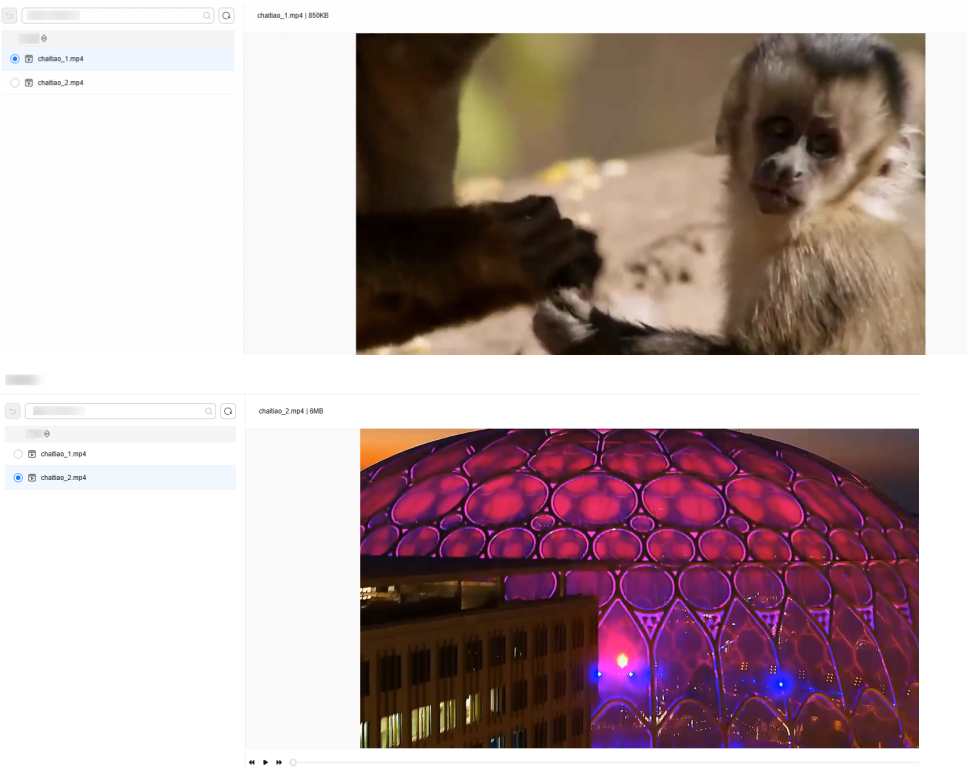
Video Aspect Ratio Filtering

- Applicable file format: video > mp4/avi.
- Parameter description

  Aspect ratio threshold: Videos whose aspect ratio exceeds the threshold will be filtered out. The threshold range is (1, 10). You can enter one decimal place.
- Filtering example

  Original video dataset:

  There are two videos, and their respective aspect ratios are 1.77 and 1.79.



Set the aspect ratio threshold to 1.78. After operator processing, only the video with the aspect ratio of 1.79 is retained.

Pornographic Video Detection

- Applicable file format: video > mp4/avi.
- Operator description: Labels pornographic content.
- Parameter configuration example

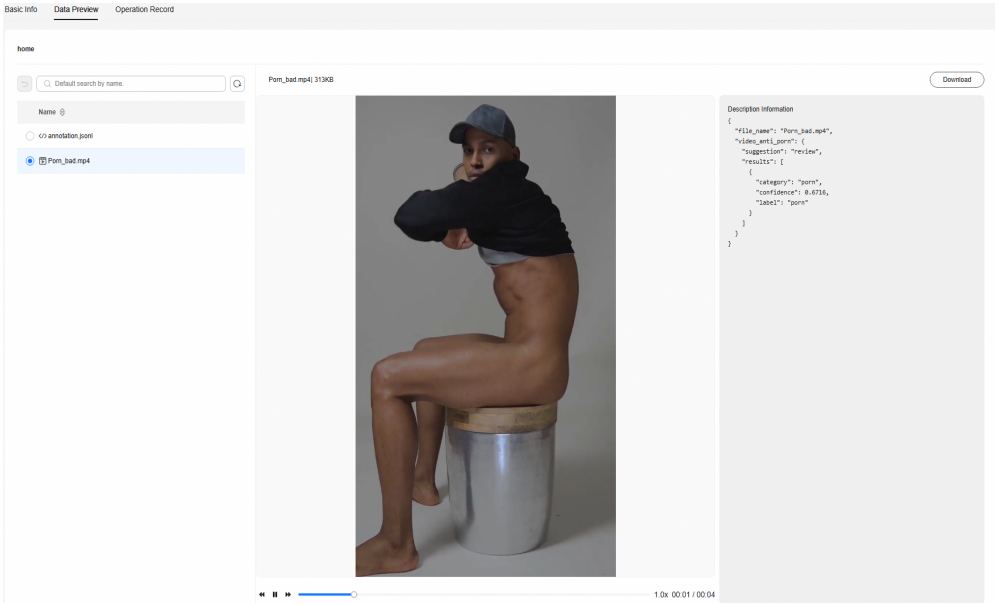
  No parameters need to be set.
- Detection example

  The results are stored in the annotation file as the video_anti_porn object.

  suggestion: indicates whether the file passes the check. **pass** indicates that the file passes the check and no problem occurs. **review** indicates that manual review is required. You can choose to bypass or block the file based on your review policy. **block** indicates that the file to be reviewed is problematic.

  confidence: detection confidence of the model. (Note that the confidence indicates the confidence of the model-provided suggestions.) If **suggestion** is **pass**, the value is 0. If **suggestion** is **review** or **block**, the value ranges from 0 to 1.

  label: label of the pornographic content detected by the model. If no pornographic content is detected, the value is empty.



Terrorism Video Detection

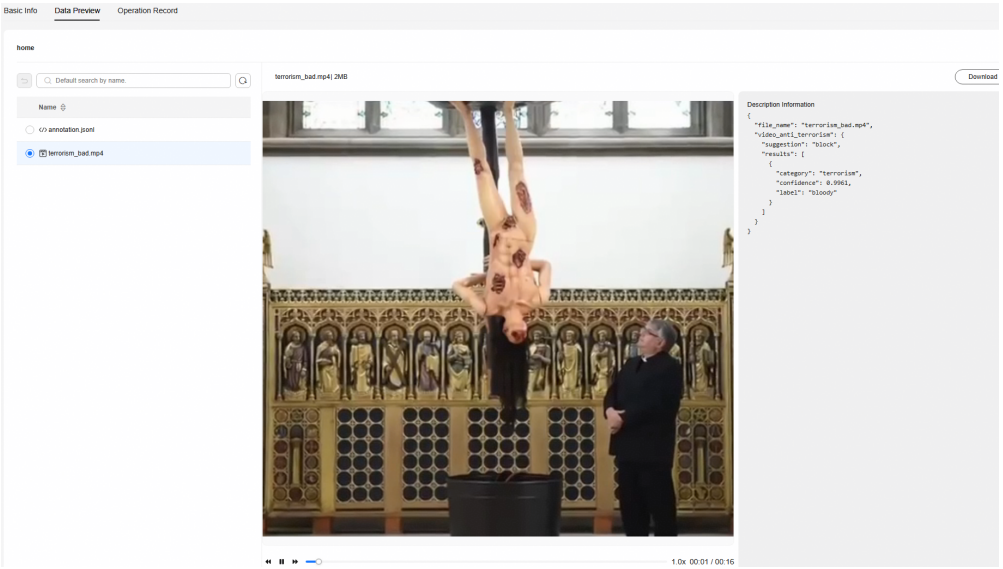
- Applicable file format: video > mp4/avi.
- Operator description: Labels violent and terrorism content.
- Parameter configuration example

  No parameters need to be set.
- Detection example: The results are stored in the annotation file as the video_anti_terrorism object.

  suggestion: indicates whether the file passes the check. **pass** indicates that the file passes the check and no problem occurs. **review** indicates that manual review is required. You can choose to bypass or block the file based on your review policy. **block** indicates that the file to be reviewed is problematic.

  confidence: detection confidence of the model. (Note that the confidence indicates the confidence of the model-provided suggestions.) If **suggestion** is **pass**, the value is 0. If **suggestion** is **review** or **block**, the value ranges from 0 to 1.

  label: label of the violent and terrorism content detected by the model. If no violent or terrorism content is detected, the value is empty.



Political Video Detection

- Applicable file format: video > mp4/avi.
- Operator description:

  Labels political content.
- Parameter configuration example

  No parameters need to be set.
- Scenario:

  This function mainly detects political figures in and outside China, negative political leaders in China, and terrorists and heretics outside China. Currently, the identification accuracy cannot be fully guaranteed.
- Detection example

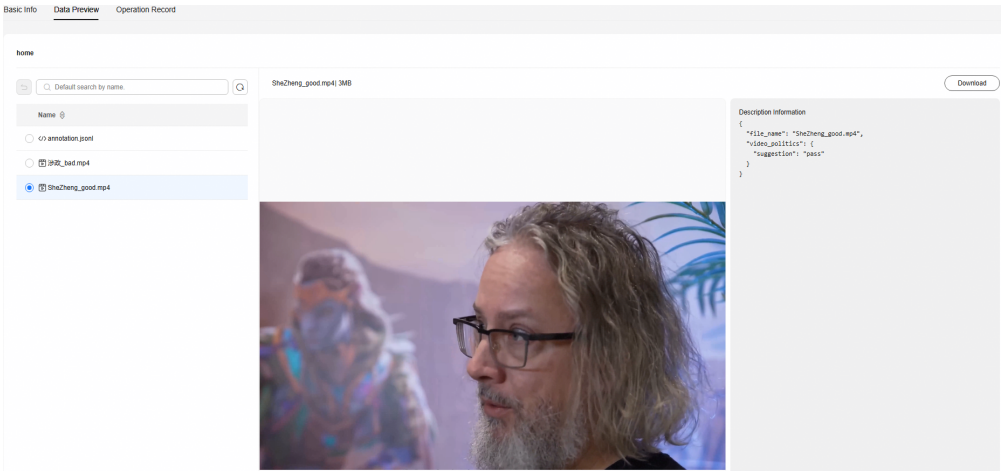
  The results are stored in the annotation file as the video_anti_politics object.

  suggestion: indicates whether the file passes the check. **pass** indicates that the file passes the check and no problem occurs. **review** indicates that manual review is required. You can choose to bypass or block the file based on your review policy. **block** indicates that the file to be reviewed is problematic.

  result: result returned by the model after file detection, including the suggestion, confidence, and label. One or more records can be returned.

  confidence: detection confidence of the model. (Note that the confidence indicates the confidence of the model-provided suggestions.) If **suggestion** is **pass**, the value is 0. If **suggestion** is **review** or **block**, the value ranges from 0 to 1.

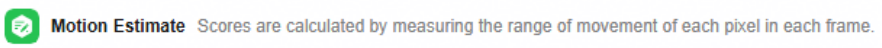
  label: label of the political content detected by the model. If no political content is detected, the value is empty.



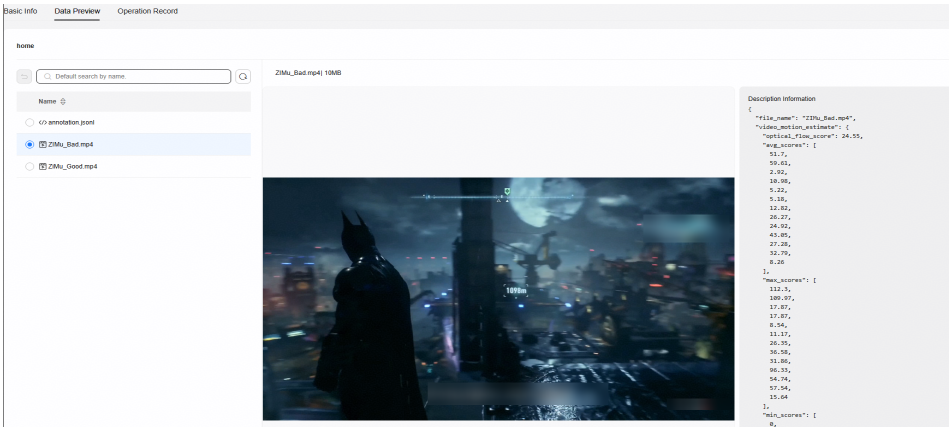
Motion Range Scoring

- Applicable file format: video > mp4/avi.
- Scoring description:

  Identifies videos with too fast or too slow motion. A larger value indicates faster motion. If the motion range is greater than 100 optical flows, the motion is too fast. If the motion range is less than or equal to 2 optical flows, the motion is too slow.
- Scenario:
  - Supported scenario
    - Identifies images with too large or too small motion range, and static images.
  - Unresolved issue
    - The parts with small fast/slow speed ratio cannot be identified.
- Parameter configuration example

- Scoring example: The motion range score is displayed in the JSONL file, as shown in the following figure.

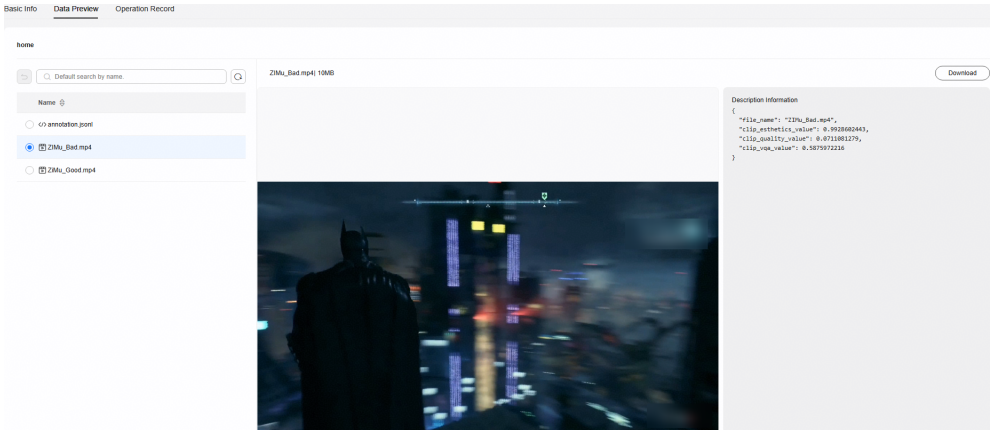
  

Aesthetics Scoring

- Applicable file format: video > mp4/avi.
- Scoring description

  Assess video aesthetics based on content (appealing, sharp), composition (well-positioned subjects), color (vibrant, pleasing), lighting (contrast), and trajectory (smooth, stable). The value range is (0, 1). A higher value indicates better aesthetics. A video whose score is greater than 0.95 is considered as a video with high aesthetics.
- Scenario:
  - Supported scenario
    - The recognition effect is better for videos with obvious aesthetic problems or quality.
  - Unresolved issue
    - Videos of the pixel game type cannot be processed.
    - The video is insensitive to watermarks.
- Parameter configuration example

- Scoring example: The aesthetics scores are stored in a JSONL file as the clip_esthetics_value object.

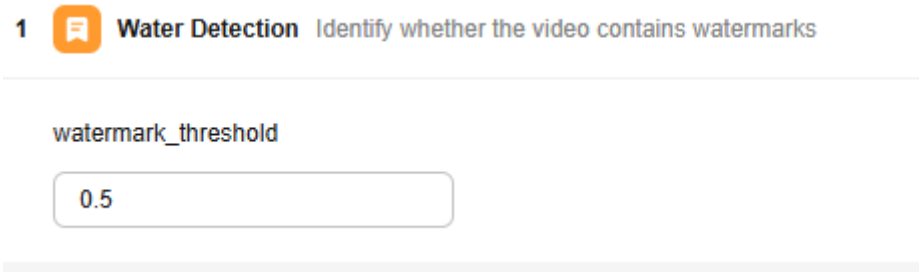


Watermark Detection

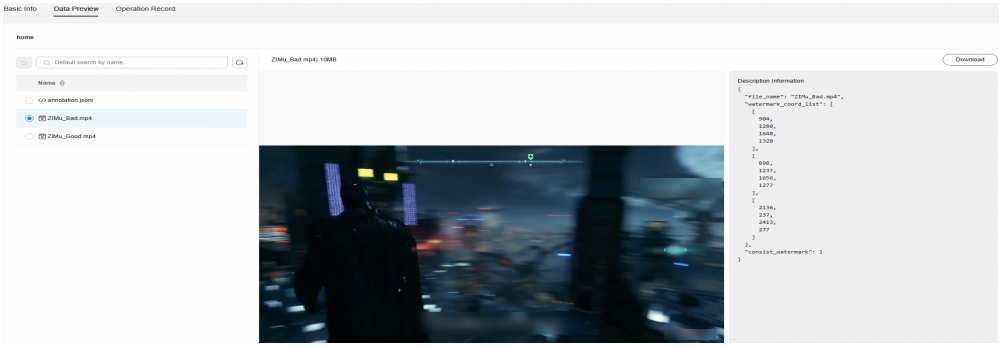
- Applicable file format: video > mp4/avi.
- Operator description:

  Identifies whether a video contains watermarks.
- Parameter configuration example

  watermark_threshold: If the watermark detection confidence is higher than this threshold, the watermark is detected. The default threshold is 0.5.
- Parameter configuration example



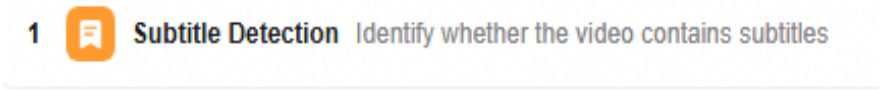
- Example: The JSONL file shows whether the watermark is identified. If the value of **consist_watermark** is 1, the watermark is identified. If the value is 0, no watermark is identified.



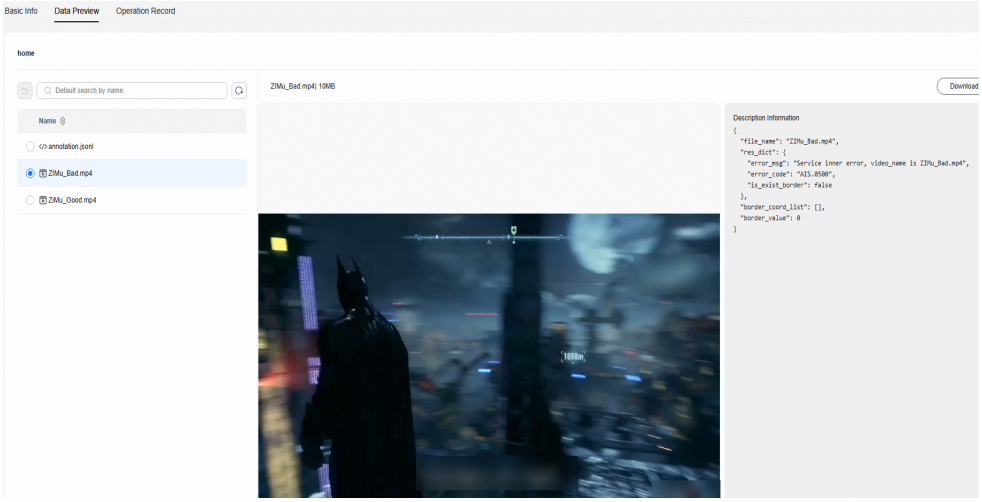
Subtitle Detection

- Applicable file format: video > mp4/avi.
- Operator description:

  Identifies whether a video contains subtitles.
- Parameter configuration example



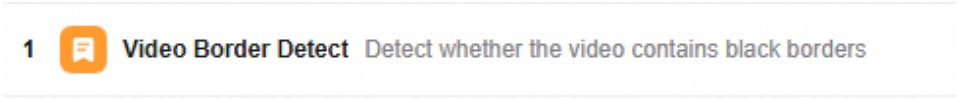
- Example: The JSONL file shows whether the subtitle is identified. If the value of **consist_subtitle** is 1, the subtitle is identified. If the value is 0, no subtitle is identified.



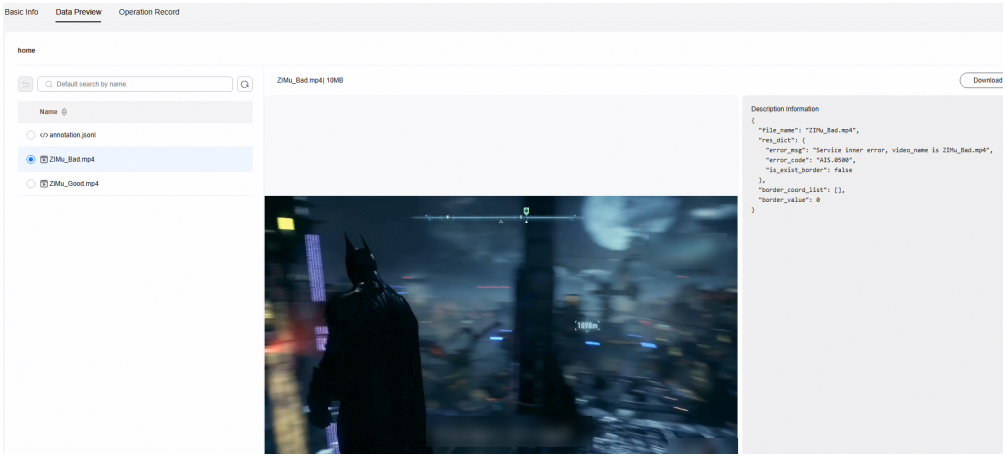
Video Black Bar Detection

- Applicable file format: video > mp4/avi.
- Operator description:

  Identifies whether a video contains black bars.
- Scenario:
  - Supported scenario
    - Only the black bars on the four edges of the video can be processed, and their color remains consistent with minimal variation.
  - Unresolved issue
    - Videos that do not contain black bars on the four edges and have color differences such as subtitles in the black bars cannot be processed.
- Parameter configuration example

- Example: If **border_value** is 1, black bars are identified. If **border_value** is 0, black bars are not identified.

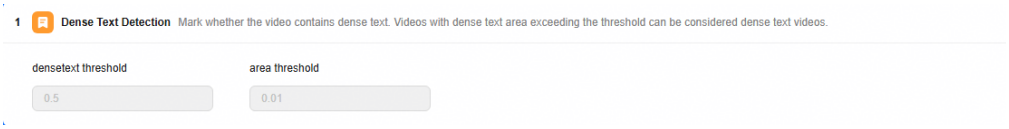
  

Dense Text Identification

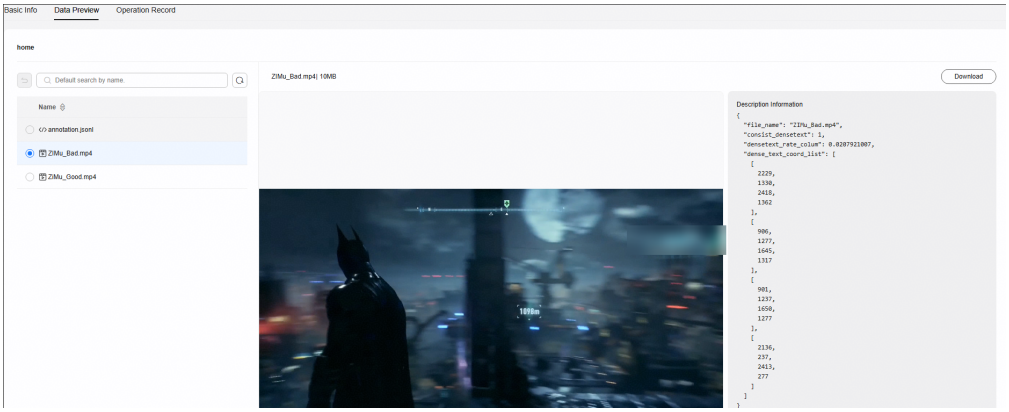
- Applicable file format: video > mp4/avi.
- Parameters:

  area threshold: A video whose dense text area ratio exceeds the threshold may be considered as a dense text video. Generally, the threshold of the dense text area ratio is 1%.

  densetext threshold: When the detection confidence exceeds the specified threshold, the video content may be considered to contain dense text. By default, **densetext threshold** is set to 0.5.
- Parameter configuration example

- Example: In the JSONL file, if the value of **consist_densetext** is 1, dense text is identified. If the value is 0, dense text is not identified.



Video Classification

- Applicable file format: video > mp4/avi.
- Operator description:

  Automatically classifies short video content and generates corresponding tags.
- Scenario:
  - Supported scenario
    - The preset categories can be classified.
  - Unresolved issue
    - The classification accuracy is not verified and is used only for uniform sampling.
    - Non-preset categories are not supported.
- Parameter configuration example

  No parameter configuration is required.
- Example of category labeling:

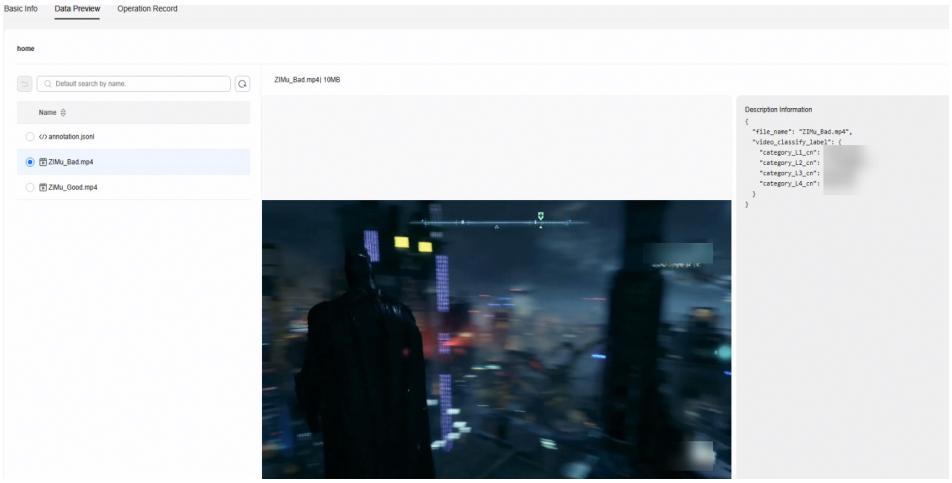
  The following information is displayed in the description:

  category_L1_cn: first-level category

  category_L2_cn: second-level category

  category_L3_cn: third-level category

  category_L4_cn: fourth-level category



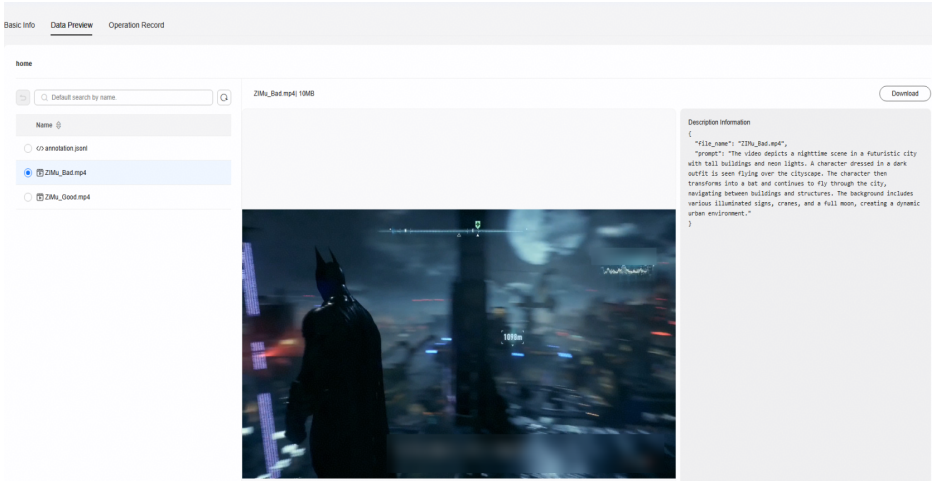
Video Synopsis Generation (Simplified)

- Applicable file format: video > mp4/avi.
- Operator description:

  Extracts frames from a video and generates a simplified video synopsis through model inference.
- Scenario:
  - Supported scenario
    - All videos can be briefly described.
  - Unresolved issue
    - The description method cannot be specified.
    - Only the viewing information (scenario, appearance, and behavior) of the video can be described. The deep content (such as news understanding, content interpretation, and well-known person recognition) of the video cannot be understood, and the audio cannot be processed.
- Parameter configuration example

  No parameter configuration is required.
- Example: The **prompt** field in the description indicates the simplified video synopsis.

Figure 4-13 Example

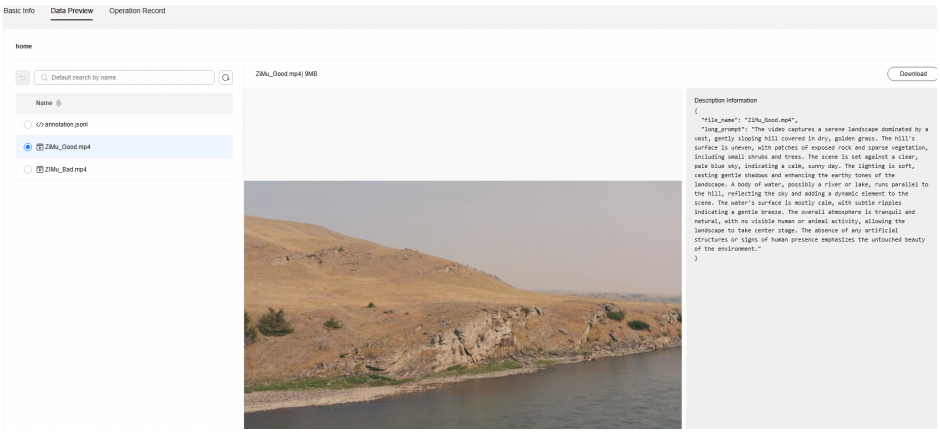


Video Synopsis Generation (Detailed)

- Applicable file format: video > mp4/avi.
- Operator description:

  Extracts frames from a video and generates a detailed video synopsis through model inference.
- Scenario:
  - Supported scenario
    - All videos can be described.
  - Unresolved issue
    - The description method cannot be specified.
    - Very detailed content, such as the quantity and action details, cannot be accurately described.
    - Only the viewing information (scenario, appearance, and behavior) of the video can be described. The deep content (such as news understanding, content interpretation, and well-known person recognition) of the video cannot be understood, and the audio cannot be processed.
- Parameter configuration example

  No parameter configuration is required.
- Example: The **long_prompt** field in the description indicates the detailed video synopsis.

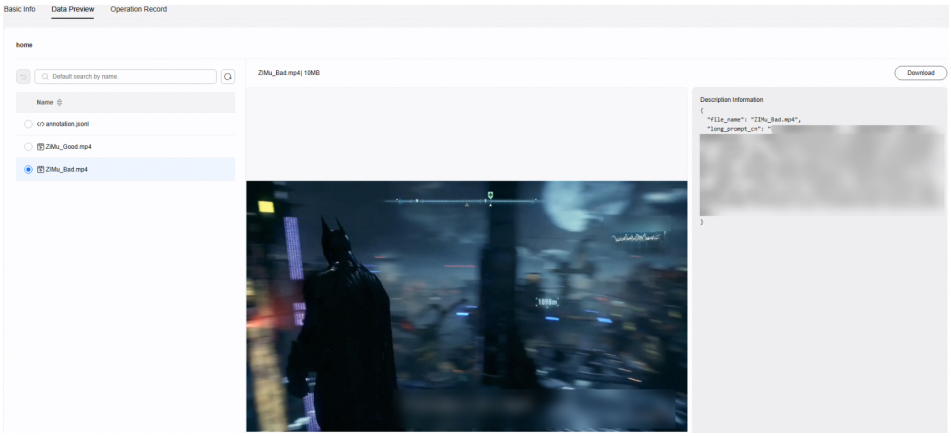


Chinese Video Synopsis Generation (Detailed)

- Applicable file format: video > mp4/avi.
- Operator description:

  Extracts frames from a video and generates a detailed Chinese video synopsis through model inference.
- Scenario:
  - Supported scenario
    - All videos can be described.
  - Unresolved issue
    - The description method cannot be specified.
    - Very detailed content, such as the quantity and action details, cannot be accurately described.
    - Only the viewing information (scenario, appearance, and behavior) of the video can be described. The deep content (such as news understanding, content interpretation, and well-known person recognition) of the video cannot be understood, and the audio cannot be processed.
- Parameter configuration example

  No parameter configuration is required.
- Example: The **long_prompt_cn** field in the description indicates the detailed video synopsis.



Posture Detection

- Applicable file format: video > mp4/avi.
- Operator description:

  The posture detection operator extracts eight frames of images from the video, marks key points on each frame of image, calculates the confidence, and calculates the number of images that meet the filtering conditions. If the number reaches a certain value, the video contains the corresponding number of persons.
- Scenario:
  - Supported scenario
    - Videos where the faces of persons are exposed can be processed.
  - Unresolved issue
    - If a person is partially blocked, the detection fails.
- Parameter configuration example

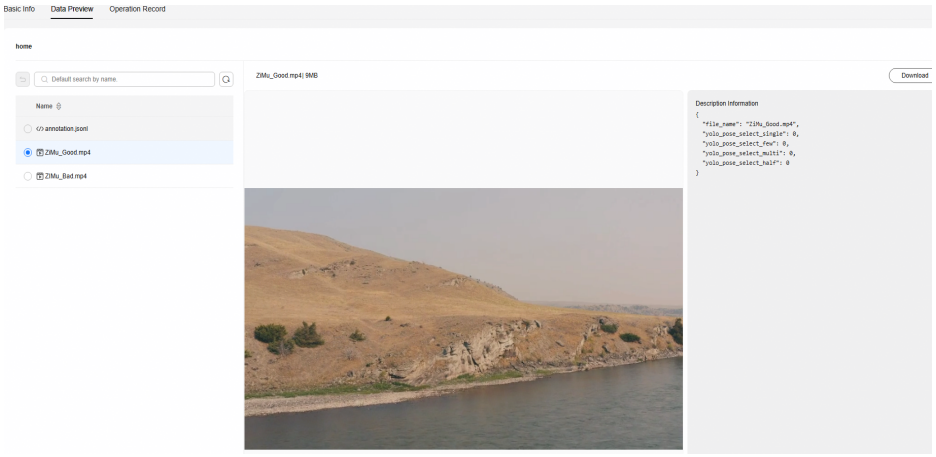
  No parameter configuration is required.
- Labeling example

  yolo_pose_select_single: indicates whether the posture of a single person is detected. If yes, the value is 1. If no, the value is 0.

  yolo_pose_select_few: indicates whether the posture of a small number of persons (usually 2 to 4) is detected. If yes, the value is 1. Otherwise, the value is 0.

  yolo_pose_select_multi: indicates whether the posture of multiple persons (usually four or more persons) is detected. If yes, the value is 1. Otherwise, the value is 0.

  yolo_pose_select_half: indicates whether the posture of half a person is detected. If yes, the value is 1. Otherwise, the value is 0.



Camera Motion Description

- Applicable file format: video > mp4/avi.
- Operator description:

  Calculates and infers optical flow by extracting frames from a video to output the lens type of the video.
- Scenario:
  - Supported scenario
    - The camera motion in the video is clear and not confusing.
  - Unresolved issue
    - If multiple camera motion combinations or unclear camera motion are used, the camera motion cannot be accurately identified. Only the preset categories can be identified.
- Parameter configuration example

  No parameter configuration is required.
- Labeling example

  motion: camera movement type.

  The tag range is: { 0: 'static', 1: 'others', 2: 'pull out', 3: 'push in', 4: 'static' , 5: 'tracking', 6: 'orbit', 7: 'spin', 8: 'tilt up', 9: 'tilt down', 10: 'pan right', 11: 'pan left' ,12: 'tracking' }

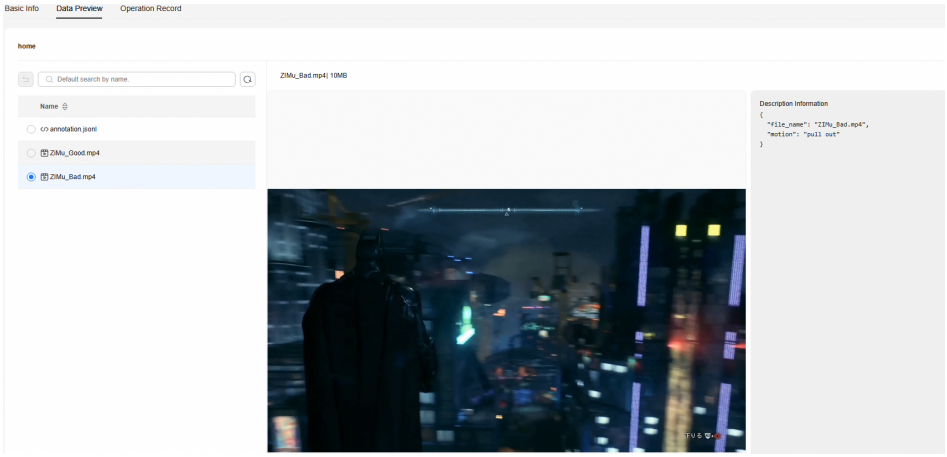


Image and Text Extraction

- Applicable file formats

  tar+jsonl: All images are saved as a TAR package. Images can be in JPG, JPEG, PNG, or BMP format. The image text is saved as a JSONL file. The image name in the JSONL file must be the same as that in the TAR package.

- Parameter description

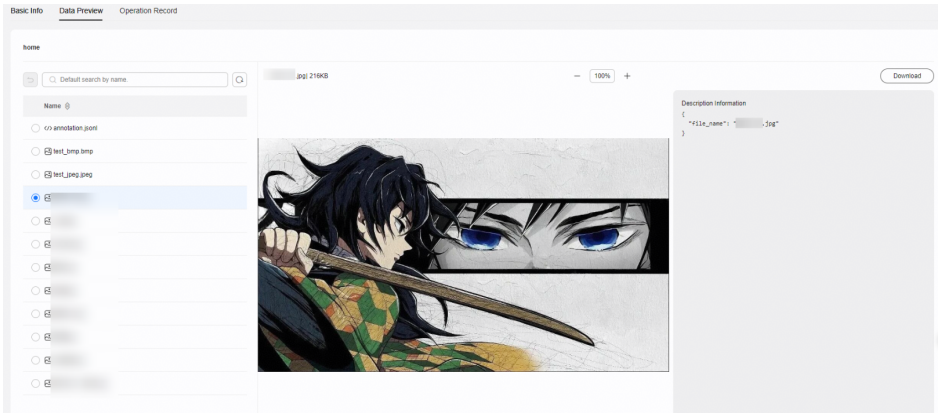
  Type of content to be extracted: Extract the JSON text and images from the image-text package and perform structured parsing on the images.

- Parameter configuration example

  No parameters need to be set.

- Extraction example

  Before refining:

  After refining:

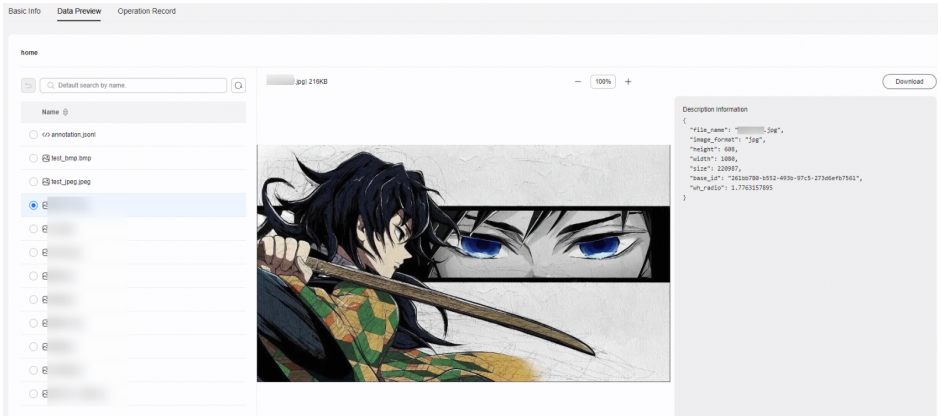
  

Image Metadata Filtering

- Applicable file formats:

  JPG, JPEG, PNG, and BMP

  tar: All images are saved as a TAR package. The images in the TAR package can be in JPG, JPEG, PNG, or BMP format.

- Parameter description

  Type of content to be filtered:

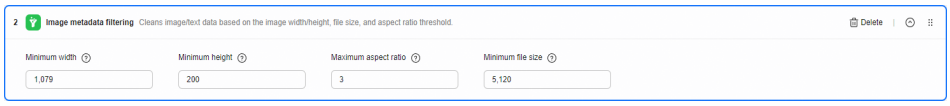
Minimum width: If the width of an image is less than the value of this parameter, the image will be filtered out.

Minimum height: If the height of an image is less than the value of this parameter, the image will be filtered out.

Minimum aspect ratio: If the aspect ratio of an image is greater than the value of this parameter, the image will be filtered out.

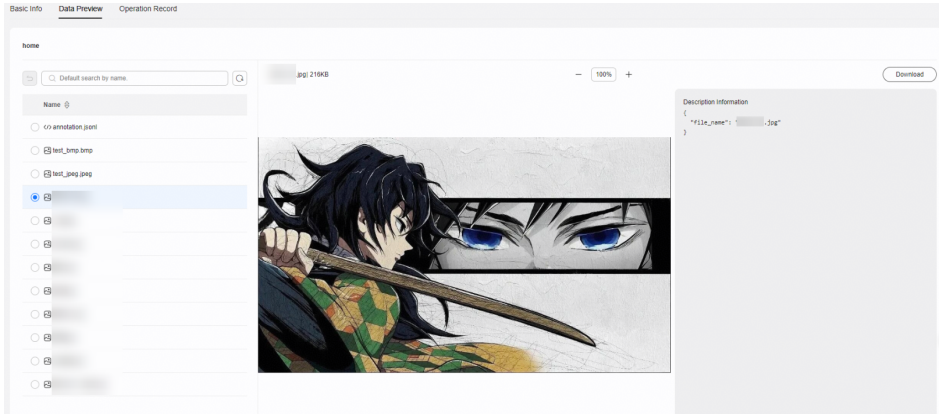
Minimum file size (B): If the file size is less than the minimum file size, the file will be filtered out.

- Parameter configuration example



- Filtering example

  Original dataset



  After filtering: Images whose width is less than 1079 are filtered out.

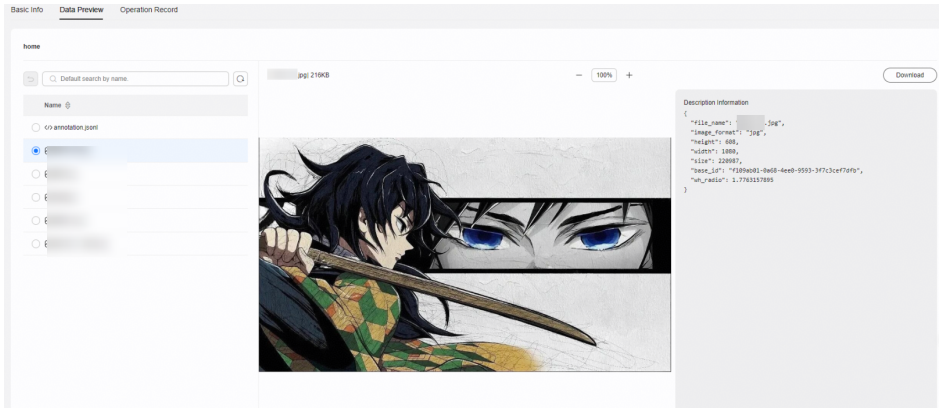


Image Deduplication

- Applicable file formats:

  JPG, JPEG, PNG, and BMP

  tar: All images are saved as a TAR package. The images in the TAR package can be in JPG, JPEG, PNG, or BMP format.

- Parameter description

Type of content to be filtered: After image structuring, duplicate image/text pairs are filtered out.

- Parameter configuration example

  No parameters need to be set.

Pornographic Image Detection

- Applicable file formats:

  JPG, JPEG, PNG, and BMP

  tar: All images are saved as a TAR package. The images in the TAR package can be in JPG, JPEG, PNG, or BMP format.

- Parameter description

  Type of content to be labeled: Score the pornographic content of the image. A higher score indicates a higher risk. The score range is (0, 100). Videos whose score is greater than or equal to 50 are considered pornographic videos.

- Parameter configuration example

  true: The filtering function is enabled.

  false: The filtering function is disabled.

- Detection example

  The results are stored in the annotation file as the image_porn object.

  suggestion: indicates whether the file passes the check. **pass** indicates that the file passes the check and no problem occurs. **review** indicates that manual review is required. You can choose to bypass or block the file based on your review policy. **block** indicates that the file to be reviewed is problematic.

  confidence: detection confidence of the model. (Note that the confidence indicates the confidence of the model-provided suggestions.) If **suggestion** is **pass**, the value is 0. If **suggestion** is **review** or **block**, the value ranges from 0 to 1.

  label: label of the pornographic content detected by the model. If no pornographic content is detected, the value is empty.

Description Information


```
{
  "base_id": "a93bc4e7-3166-4510-aa41-f603397c6690",
  "file_name": "[redacted].jpg",
  "height": 1279,
  "image_format": "jpg",
  "image_porn_label": {
    "suggestion": "review",
    "results": [
      {
        "category": "porn",
        "confidence": 0.5259,
        "label": "sexy"
      }
    ]
  },
  "size": 287324,
  "wh_radio": 0.712275215,
  "width": 911
}
```


Dangerous Situation Image Detection

- Applicable file formats:

  JPG, JPEG, PNG, and BMP

  tar: All images are saved as a TAR package. The images in the TAR package can be in JPG, JPEG, PNG, or BMP format.

- Parameter description

  Type of content to be labeled: Labels the content of dangerous situation images.

- Parameter configuration example

  No parameters need to be set.

- Detection example: The results are stored in the annotation file as the image_danger object.

  suggestion: indicates whether the file passes the check. **pass** indicates that the file passes the check and no problem occurs. **review** indicates that manual review is required. You can choose to bypass or block the file based on your review policy. **block** indicates that the file to be reviewed is problematic.

  confidence: detection confidence of the model. (Note that the confidence indicates the confidence of the model-provided suggestions.) If **suggestion** is **pass**, the value is 0. If **suggestion** is **review** or **block**, the value ranges from 0 to 1.

  label: label of the dangerous situation content detected by the model. If no dangerous situation content is detected, the value is empty.

  Description Information


```
{
  "base_id": "86d6891d-732b-4d72-a2d6-ac20ec314ca8",
  "file_name": "[redacted]v1.PNG",
  "height": 810,
  "image_danger_label": {
    "suggestion": "pass"
  },
  "image_format": "png",
  "image_terrorism_label": {
    "results": null,
    "suggestion": "pass"
  },
  "size": 923091,
  "wh_radio": 1.6259259259,
  "width": 1317
}
```


Violent and Terrorism Image Detection

- Applicable file formats:

  JPG, JPEG, PNG, and BMP

  tar: All images are saved as a TAR package. The images in the TAR package can be in JPG, JPEG, PNG, or BMP format.

- Parameter description

  Type of content to be labeled: Filters out violent and terrorism images.

- Parameter configuration example

  true: The filtering function is enabled.

  false: The filtering function is disabled.

- Scenario:

  This function applies only to violent and terrorism-related scenarios. Currently, the identification accuracy cannot be fully guaranteed.

- Detection example: The results are stored in the annotation file as the image_terrorism object.

  suggestion: indicates whether the file passes the check. **pass** indicates that the file passes the check and no problem occurs. **review** indicates that manual review is required. You can choose to bypass or block the file based on your review policy. **block** indicates that the file to be reviewed is problematic.

  confidence: detection confidence of the model. (Note that the confidence indicates the confidence of the model-provided suggestions.) If **suggestion** is **pass**, the value is 0. If **suggestion** is **review** or **block**, the value ranges from 0 to 1.

  label: label of the violent and terrorism content detected by the model. If no violent or terrorism content is detected, the value is empty.

  Description Information


```
{
  "base_id": "8176bb22-d490-41f7-9ee3-2ca840669459",
  "file_name": "[redacted].jpg",
  "height": 500,
  "image_format": "jpg",
  "image_terrorism_label": {
    "suggestion": "block",
    "results": [
      {
        "category": "terrorism",
        "confidence": 0.998,
        "label": "bloody"
      }
    ]
  },
  "size": 86858,
  "wh_radio": 1.334,
  "width": 667
}
```


4.2 Manual Calibration

You can manually calibrate datasets on the visualized labeling page, generate standard datasets in one-click mode, and synchronize the datasets to **My Data** for tasks such as smart refining.

Constraints

- A maximum of 10 manual calibration tasks can be created using an IAM account.
- The number of samples in a dataset cannot exceed 10,000.

Creating a Manual Calibration Task

1. Log in to the **ModelArts console**. In the navigation pane on the left, choose **Data Preparation** > **Data Refining**.
2. In the upper right corner of the **Manual Calibration** tab page, click **Create**. On the **Create Manual Calibration** page, configure related information and click **OK**.

Figure 4-14 Create Manual Calibration

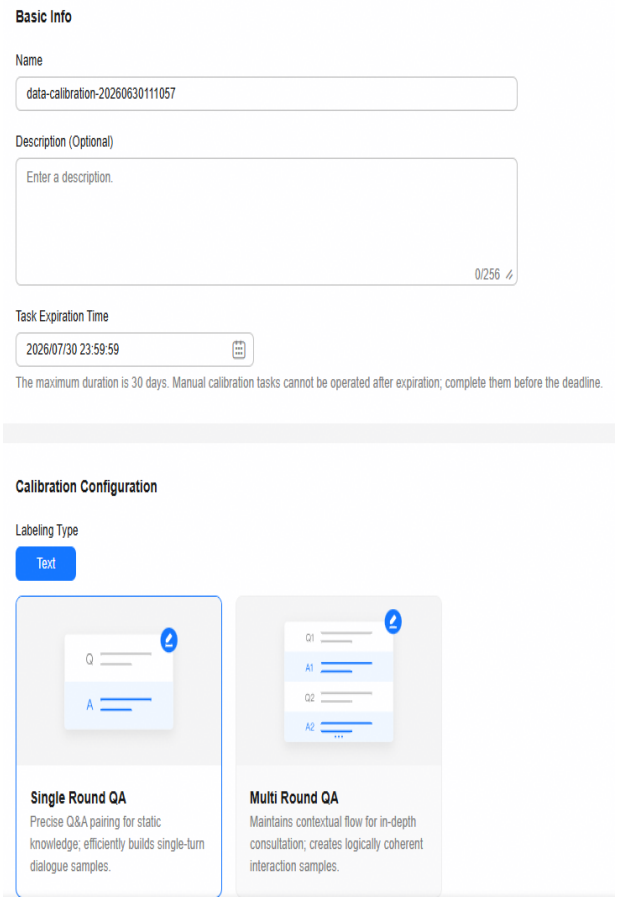


Table 4-9 Parameters for creating a manual calibration task

Parameter		Description	Example Value
Basic Information	Name	Custom task name. The default value is **data-calibration-***YYYYMMDDHHMMSS*. The name must start with a letter and end with a letter or digit. It can contain 2 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.	data-calibration-20260425164425

Parameter		Description	Example Value
	Description	Description of the manual calibration task. Only letters, digits, spaces, hyphens (-), underscores (_), commas (,), periods (.), brackets, colons (:), and commas are allowed. It can contain a maximum of 256 characters.	-
	Task due date.	The maximum duration is 30 days. Manual calibration tasks cannot be operated after expiration; complete them before the deadline.	2026/05/25 23:59:59
Calibration Configuration	Type	The text type is supported. You can select single-turn or multi-turn Q&A as needed. • Single-turn Q&A: Precise Q&A pairing for static knowledge; efficiently builds single-turn dialogue samples. • Multi-turn Q&A: Maintains contextual flow for in-depth consultation; creates logically coherent interaction samples.	Multi-turn Q&A
Associated Dataset		The system automatically filters the types of My Datasets available for manual calibration based on your selected calibration scenario. Click the card, select a dataset as required, and click **OK**.	Multi-turn Q&A chain-of-thought ShareGPT-0327-001

When the status of the manual calibration task changes to **Processing**, the task is created.

Executing a Calibration Task

Perform calibration on the visualized labeling page. Use the visual interface to fix errors, add missing labels, and use batch verification tools to guarantee high data accuracy. The following uses a multi-turn Q&A task as an example.

1. On the **Manual Calibration** tab page, click **Label** on the right of the manual calibration task whose status is **Processing**.
2. Proofread the input, thinking process, and output in the Q&A pair, and click **Submit**.

- Incorrect content: Modify the content as required.
- Invalid content: Click **Invalid Data** in the lower right corner of the page. Marking items as invalid will reset the annotations to their original state.

The task progress is displayed on the right of the manual calibration task name. After all data is labeled, the task status changes to **Completed**.

Generating Datasets

Summarize and export the calibration results, and integrate the calibration data. You can preview details, verify integrity, generate standard datasets in one-click mode, and synchronize the datasets to My Data for future use.

1. On the **Manual Calibration** tab page, click **Generate** on the right of the manual calibration task.
2. In the **Generate Dataset** pane, configure related information and click **OK**.

Table 4-10 Parameters for generating a dataset

Parameter	Description	Example Value
Dataset Name	Name of a custom dataset. The name must start with a letter and end with a letter or digit. It can contain 2 to 63 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.	dataset
Output Data Content	Select labeled data, unlabeled content, valid data, and invalid data as required.	Labeled data and valid data
Storage Location	Choose **Object Storage Service – Bucket** or **Object Storage Service – Parallel File System** as the storage type. Click to select an OBS storage address or manually enter an OBS storage address. The storage address must start with **obs://** or **/** and end with a slash (**/**). It cannot contain double slashes (**//**) except in the prefix. For example, **obs://bucketname/path/** or **/bucketname/path/**.	obs://bucketname/path/
Dataset Property	Click to configure dataset properties as required, such as the industry and language.	-

Parameter	Description	Example Value
Description	Description of a custom dataset. Only letters, digits, spaces, hyphens (-), underscores (_), commas (,), periods (.), brackets, colons (:), and commas are allowed. The value can contain a maximum of 100 characters.	-
Dataset Status	Only published datasets can be used by downstream tasks such as model training. • If you select **Publish Dataset**, the generated dataset is in the **Online** state on the **Asset Management** > **Data** > **My Data** page and can be directly used by downstream model training jobs. • If you do not select **Publish Dataset**, the generated dataset will be in the **Offline** state on the **Asset Management** > **Data** > **My Data** page and cannot be directly used by downstream model training jobs. You need to manually publish the dataset before using it.	Select **Publish Dataset**.

Viewing Calibration Task Details

On the **Manual Calibration** tab page, click the name of a manual calibration task to view the task details, including the number of samples that have been calibrated, total number of samples, total progress, basic information about the calibration task, associated datasets, and generated datasets.

Deleting a Manual Calibration Task

You can delete unnecessary manual calibration tasks. **Deleted manual calibration tasks cannot be restored. Proceed with caution.**

1. On the **Manual Calibration** tab page, click **Delete** on the right of the manual calibration task name.
2. In the **Delete Manual Calibration Task** dialog box, enter **DELETE** and click **OK**.

Follow-Up Operations

The generated dataset can be used for secondary manual calibration, **model training**, and other operations.

4.3 FAQs

1. **Can a synthesis operator be placed in the middle of a workflow?**

   No. In the current version, synthesis operators can only be placed at the end of a workflow. If you need to perform further processing on the synthesized results, you are advised to split the process into two separate tasks:

   a. Task 1: processing operator + synthesis operator

   b. Task 2: further processing of the synthesized results

2. **Is text-to-image cross-modal synthesis supported?**

   The current version does not support cross-modal synthesis. It only supports same-modality data synthesis, for example:

   - Text → text (Q&A rewriting)
   - Text → image (not supported)
   - Image → text (not supported)

5 Data Asset Management

5.1 Overview

Data assets encompass all datasets managed, stored, and accessible on the ModelArts platform, specifically including **Preset Data** and **My Data**.

- **Preset Data**: The platform offers high-quality preset datasets across four modalities—text, image, video, and audio—that are rigorously selected and preprocessed for immediate use. By eliminating the need for extensive data preparation, these ready-to-use resources significantly lower technical barriers and accelerate workflows for tasks such as smart refining, model training, fine-tuning, and evaluation. For details, see **Preset Data**.
- **My Data**: When you create a data connection or smart refining task on the console, the generated dataset is placed in the **My Data** list as a data asset. For details, see **My Data**.

5.2 Preset Data

ModelArts provides high-quality preset datasets for you. These datasets follow open-source rules and work with popular training frameworks. They help track dataset versions and reproduce experiments. Choose a suitable dataset for your needs and use it right away in the platform.

Scenarios

Typical scenarios of **Preset Data**:

- Use preset datasets along with your own data to complete smart refining to improve and create high-quality datasets for later tasks.
- Use a preset dataset for LLM pre-training and fine-tuning to enhance foundational capabilities, while leveraging human preference data to optimize response quality.
- Combine image, video, and audio data to build cross-modal capabilities and multimodal models.
- Use a dataset as a standard test set to evaluate model performance and establish baseline assessments of model capabilities.

Viewing Preset Data

1. Log in to the **ModelArts console**.
2. In the navigation pane, choose **Asset Management** > **Data**. Click the **Preset Data** tab. The preset datasets are displayed in cards. You can view information such as the dataset name, modality, type, description, update time, and number of samples on the preset data card.

   NOTE

   The preset data available in each region may vary.
3. Click a preset dataset card to view its details. The details include **Basic Info** and **Data Preview**.
   - **Basic Info** includes the name, modality, type, number of samples, dataset size, and description of the preset dataset, as well as extended information such as the dataset property, industry, language, and tags.
   - **Data Preview** allows you to display some typical samples of structured data (text and tables), view the samples on multiple pages, and view the original data structure. Unstructured data (images/audio) can be previewed in thumbnail mode.

Preset Datasets

ModelArts offers preset text and image datasets. For details, see **Table 5-1**. Choose a dataset that fits your scenario.

Table 5-1 List of preset datasets

Name	Preset Tag	Dataset Overview	Size	Samples	Language	Link
ai-expert-alpaca	Text, single-turn Q&A	This dataset has high-quality Q&A pairs for training large language models (LLMs). It focuses on three key areas: LLMs, retrieval-augmented generation (RAG), and agent systems. The dataset covers these advanced AI topics in both English and Chinese.	8.2 MB	11,235	Chinese, English	**https://huggingface.co/datasets/GXMZU/ai-expert-alpaca?utm_source=chatgpt.com**
GPT-4-LLM	Text, single-turn Q&A	Alpaca-CoT is a large, high-quality dataset for instruction fine-tuning that includes various task types.	33.47 MB	48,818	Chinese	**https://huggingface.co/datasets/QingyiSi/Alpaca-CoT/blob/main/alpacaGPT4/alpaca_gpt4_data.json**

Name	Preset Tag	Dataset Overview	Size	Samples	Language	Link
alpaca_data	Text, single-turn Q&A	Stanford Alpaca released this dataset, which has 52,000 English instruction samples created using self-supervised methods.	20.0 MB	52,002	English	https://huggingface.co/datasets/QingyiSi/Alpaca-CoT/tree/main/alpaca
alpaca_gpt4_data	Text, single-turn Q&A	This dataset was released by Instruction-Tuning-with-GPT-4. It contains 52,000 English instruction-following samples generated by GPT-4 using Alpaca prompts, and is used to fine-tune LLMs.	40.4 MB	52,002	English	https://huggingface.co/datasets/QingyiSi/Alpaca-CoT/tree/main/alpacaGPT4
code_alpaca	Text, single-turn Q&A	This dataset was released by CodeAlpaca and contains code generation tasks with 20,022 samples.	6.7 MB	20,022	English	https://huggingface.co/datasets/QingyiSi/Alpaca-CoT/tree/main/CodeAlpaca
lunara-aesthetic-image-variations	Image	This dataset contains original images and artworks created by Moonworks.	17.7 MB	36	Chinese	https://huggingface.co/datasets/moonworks/lunara-aesthetic-image-variations/tree/main

5.3 My Data

When you create a data connection or smart refining task on the console, the generated dataset is placed in the **My Data** list as a data asset. If you enable **Publish Dataset** on the data connection or smart refining task creation page, the generated dataset will be automatically brought online as an asset. If this option is not selected, the asset is in the offline state in the asset list.

Scenarios

Typical scenarios of **My Data**:

- Using a dataset to perform smart refining to generate high-quality datasets required for downstream tasks

- Using a dataset for LLM pre-training and fine-tuning to enhance foundational capabilities, while leveraging human preference data to optimize response quality
- Using a dataset as a test set to evaluate model performance and establish baseline assessments of model capabilities

Operation Guide

1. Log in to the **ModelArts console**.
2. In the navigation pane on the left, choose **Asset Management** > **Data** > **My Data**. You can view all datasets and the list of assets created by you. You can also filter dataset assets by dataset name, data modality, dataset type, status, and creator.
3. Click  on the right of the search bar. On the page that appears on the right, set the search bar. **Table 5-2** describes the configurable items.

   Table 5-2 My data list configuration

Category	Parameter	Description
Basic Settings	Text Wrapping	Enabled: Automatically wraps text within cells. Data asset entries will expand vertically to display all information. Disabled: Text will not wrap; data entries will remain on a single line and information may be truncated.
	Data Columns	• **Freeze none**: If the data list exceeds the screen width, all columns remain scrollable. • **Freeze first**: The first column is frozen while other columns remain scrollable. • **Freeze first two**: The first and second columns are frozen while other columns remain scrollable.
	Operation Column	When enabled, the **Operation** column is permanently pinned as the final column and its width cannot be adjusted.
Custom Columns	Display options	Select the column names you wish to display. **Dataset Name** and **Operation** are displayed by default; other columns can be toggled on or off. Column headers can be dragged to reorder them.

4. Select a dataset and perform the following operations in the **Operation** column:
   - **Bring Online**: Offline datasets can be brought online. Click **Bring Online**. In the dialog box that is displayed, confirm the operation. The dataset is brought online. A dataset that has been brought online can be used as data for subsequent development.

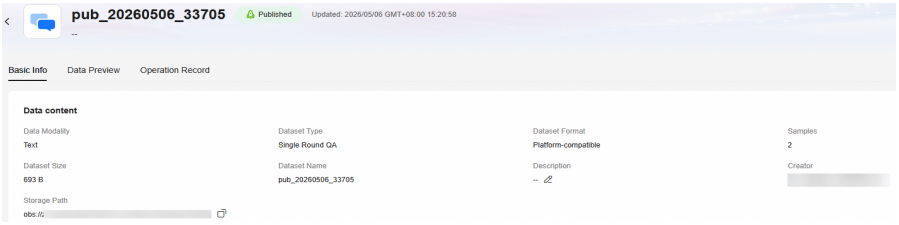
- **Take Offline**: Online datasets can be taken offline. Click **Take Offline**. In the dialog box that is displayed, confirm the operation. The dataset is taken offline. A dataset that has been taken offline cannot be used as data for subsequent development.
- **Delete**: Datasets can be deleted. Deleted datasets are not permanently removed. If you accidentally delete a dataset, you can restore it for future use. You can permanently delete a deleted dataset. Once permanently deleted, the dataset cannot be restored.
- **Restore**: You can restore a deleted dataset.

Data Asset Details Management

The data asset details page displays detailed information about the current dataset. In the dataset workspace, click a dataset name to go to the dataset details page. This page consists of the basic information, data preview, and operation records tab pages. The following describes the functions and operations on the three tab pages. You can delete a dataset in the upper right corner of the page. After you click **Delete**, the dataset will be permanently deleted. Exercise caution when performing this operation.

- **Basic Information** Displays the following asset information:
  - **Data content**: Information such as asset name, modality, type, sample count, size, description, creator, storage location, etc.
  - **Data Source**: Specifies the task that generated the data asset; you can navigate to the task via the **Source Task ID**.
  - **Extended Info**: Attributes and copyright information of the data asset; this information supports manual modification.

  Figure 5-1 Basic information about data assets



- **Data Preview**: Allows you to display three to five typical samples of structured data (text and tables), view the samples on multiple pages, and view the original data structure. Unstructured data (images/audio) can be previewed in thumbnail mode. Datasets can be downloaded.
- **Operation Record**: You can view operation records on this page. Operation records include all operations performed on the current dataset.

6 Using CTS to Audit ModelArts Data Services

ModelArts can connect to CTS. With CTS, you can obtain operations associated with ModelArts for later query, audit, and backtrack operations.

After CTS is enabled, CTS starts recording operations on ModelArts. The CTS console stores the operation records generated in the last seven days. This section describes how to view operation records of the last seven days on the CTS console.

Prerequisites

You have enabled CTS. For details, see **Cloud Trace Service User Guide**.

Key Data Preparation Operations Traced by CTS

Table 6-1 lists the key data preparation operations that can be traced by CTS.

Table 6-1 Key data management operations traced by CTS

Operation	Resource Type	Trace
Creating a dataset	dataset	createDataset
Deleting a dataset	dataset	deleteDataset
Updating a dataset	dataset	updateDataset
Publishing a dataset version	dataset	publishDatasetVersion
Deleting a dataset version	dataset	deleteDatasetVersion
Synchronizing the data source	dataset	syncDataSource
Exporting a dataset	dataset	exportDataFromDataset
Creating a dataset label	dataset	createLabel

Operation	Resource Type	Trace
Updating a dataset label	dataset	updateLabel
Deleting a dataset label	dataset	deleteLabel
Deleting a dataset label and its samples	dataset	deleteLabelWithSamples
Adding samples	dataset	uploadSamples
Deleting samples	dataset	deleteSamples

Viewing Traces

1. Log in to the **CTS console**.
2. Click  in the upper left corner and select the target region.
3. In the navigation pane on the left, choose **Trace List**.

   Each time you log in to the CTS console, the new edition is displayed by default. Click **Old Edition** in the upper right corner to switch to the trace list of the old edition.
4. Filter traces to view information about the target traces.

   For details, see **Querying Traces in CTS**.

7 Data Preparation Error Codes

The table below shows the error codes you might see when using the data preparation functions.

Data Connection

Table 7-1 Data connection error codes

Status Code	Error Code	Error Message	Description	Measure
400	ModelArts.72060001	Illegal argument [xxx].	The input parameter is invalid.	Check whether the input parameter is valid.
400	ModelArts.72060007	Dataset [xxx] is not existed.	The dataset does not exist.	Check whether the dataset exists.
400	ModelArts.72060009	Verification failed. Please check the content format is consistent with the template requirements.	Verification failed. The file format does not meet the template requirements.	View the sample data and check whether the source data is consistent with the template data.
400	ModelArts.72060012	Internal error.	Internal service error.	System error. Contact technical support.
400	ModelArts.72060020	Import job does not existed.	The connection task does not exist.	Check whether the connection task exists.

Status Code	Error Code	Error Message	Description	Measure
400	ModelArts.72060025	The file [{0}] size exceeds the maximum limit of [{1}].	File too large.	Check whether the data exceeds the limit.
400	ModelArts.72060041	Dataset is empty.	The dataset is empty.	Check whether the source data is empty.
500	ModelArts.72060046	Download log error.	Failed to download the log.	Try again later.
400	ModelArts.72060049	File format mismatch, require [xxx].	The file format does not meet the requirements.	Check whether the file name extension of the source data meets the requirements.
400	ModelArts.72060050	Data parsing error.	Data parsing error.	View the sample data and check whether the source data is consistent with the template data.
429	ModelArts.72060075	Too Many Requests. Please try again later.	Too many requests.	Too many requests. Try again later.
401	ModelArts.72060083	Failed to create the dataset. Please try again later.	Failed to generate the dataset. Try again later.	Failed to generate the dataset. Try again later.
500	ModelArts.72060084	Fail to get agency info. Please try again later.	Failed to obtain the agency information. Try again later.	Check whether the user has created an agency.
500	ModelArts.72060085	Failed to update import job. Please try again later.	Failed to update the connection task. Try again later.	Failed to update the connection task. Try again later.

Status Code	Error Code	Error Message	Description	Measure
400	ModelArts.72060086	Individual files not supported. Please select a folder.	A single file cannot be used to create a connection task. Select a folder.	Create a connection task using a folder. Ensure that the entered OBS path ends with a slash (/).
400	ModelArts.72060087	Required field [xxx] is missing in JSONL file [xxx] line [xxx]. Please ensure the file content follows the required format.	Failed to verify the data in JSONL format. The xxx field is missing in line xxx.	View the sample data and check whether the source data is consistent with the template data.
400	ModelArts.72060088	Image files referenced in JSONL file are missing. Please check the following non-existent filenames: [%s]	In the image dataset, the image indexed by the JSONL file does not exist.	Check whether the source data complies with the sample template and whether the image file of the JSONL index exists.
403	ModelArts.72060089	Current user is not authorized, please create agency first.	The current user has not created an agency. Create an agency first.	Create an agency.

Data Management

Table 7-2 Data management error codes

Status Code	Error Code	Error Message	Description	Measure
500	ModelArts.72010021	base server [{0}] is unavailable	The basic service [{0}] is unavailable.	Check whether OBS is available.

Status Code	Error Code	Error Message	Description	Measure
500	ModelArts.72010035	obs client failed caused by [{0}]	OBS client failure. Reason: [{0}]	Check the OBS configuration and network connection.
500	ModelArts.72010065	parse model info failed, caused by [{0}]	Failed to parse the model information. Reason: [{0}]	Check whether the model information format is correct.
400	ModelArts.72010001	dataset name [{0}] already exists (including soft-deleted)	Dataset name [{0}] already exists (including soft-deleted ones).	Use another dataset name.
500	ModelArts.72010068	generate table md5 failed. datasetName is [{0}].	Failed to generate the MD5 table.	Check whether the dataset file is complete.
500	ModelArts.72010067	Get metatable from obs error, datasetName is [{0}].	Failed to obtain the metadata table from OBS.	Check whether the OBS path and data exist.
500	ModelArts.72010068	generate dataset info failed. datasetName is [{0}].	Failed to generate the dataset information.	Check the dataset configuration.
404	ModelArts.72010003	dataset [{0}] not found in workspace [{1}]	Dataset [{0}] cannot be found in space [{1}].	Check whether the dataset name and space are correct.
500	ModelArts.72010022	Failed to bring the dataset [{0}] offline, caused by [{1}]	Failed to bring dataset [{0}] offline. Reason: [{1}].	Check the dataset status and ensure that no task is associated.
500	ModelArts.72010074	Permanently deleting the dataset failed.	Failed to permanently delete the dataset.	Check whether the OBS permission and data exist.

Smart Refining

Table 7-3 Smart Refining error codes

Status Code	Error Code	Error Message	Description	Measure
500	ModelArts.72020000	Internal Server Error.	Internal service error.	System error. Contact technical support.
400	ModelArts.72020001	The task operator [xxx] not exist.	Task operator not found.	Check whether the task operator exists.
400	ModelArts.72020002	The task operator required argument is missing.	Mandatory parameters are missing for the task operator.	Check whether the mandatory parameters of the operator are complete.
400	ModelArts.72020008	The job is running.	The refining task is running and cannot be deleted.	Wait until the refining task is complete or manually stop the task and then delete it.
400	ModelArts.72020020	Task not exist.	Task not found.	Check whether the refining task exists.
400	ModelArts.72020022	Dataset is not usable.	The dataset is unavailable.	Check whether the dataset is available.
400	ModelArts.72020028	This template does not exist.	Template not found.	Check whether the template exists.
400	ModelArts.72020041	Job status is not generating.	The dataset cannot be stopped from being generated because the refining job is not in the dataset generation state.	The dataset cannot be stopped from being generated because the refining job is not in the dataset generation state.

Status Code	Error Code	Error Message	Description	Measure
400	ModelArts.72020044	error:xxx	Error: xxx.	Common error. Contact technical support.
400	ModelArts.72020045	The job is not soft deleted.	The refining job has not been soft deleted and cannot be permanently deleted.	Delete the job and then delete it permanently.
400	ModelArts.72020047	The task is in the pending state, please try again later.	The job is waiting to be executed. Try again later.	The job is waiting to be executed. Try again later.
400	ModelArts.72020048	The job does not exist.	The refining job does not exist.	Check whether the refining job exists.
400	ModelArts.72020050	Failed to start job.	Failed to start the refining job.	Check whether the input parameters of the startup API are correct.
400	ModelArts.72020051	Operation failed. The current job status does not support this operation.	Operation failed. The current refining job status does not support this operation.	The current refining job status does not support this operation.
400	ModelArts.72020052	Failed to generate the dataset. The current job status does not support this operation.	Failed to generate the dataset. The current refining job status does not support this operation.	The current refining job status does not support this operation.
400	ModelArts.72020057	The input regular expression[xxx] has security risks, please re-enter.	The entered regular expression has security risks. Enter another one.	Check whether the regular expression entered for the operator parameter is valid.

Status Code	Error Code	Error Message	Description	Measure
403	ModelArts.72020100	operation is unsupported	Unsupported operation.	Sorry, but you do not have the permission to call this API.
503	ModelArts.72020067	read obs error, reason:[xxx]	An error occurred when reading OBS.	Check whether the OBS service is normal.
400	ModelArts.72020070	The log path is not exist.	Log not found.	Check whether the log file has been deleted from OBS.
503	ModelArts.72020076	The data management service is abnormal.	Data management service error.	Check whether the data management service is normal.
400	ModelArts.72020078	The source data set cannot be empty.	The source dataset cannot be empty.	Check whether the source dataset has been deleted.
503	ModelArts.72020079	The operator management service is abnormal.	The operator management service is abnormal.	Check whether the operator management service is normal.
400	ModelArts.72020080	The operators cpuArch does not match ModelArts cpuArch.	The operator cpuArch does not match the ModelArts resource pool cpuArch.	Change the flavor and try again.
400	ModelArts.72020081	Get ModelArts cpuArch failed. Please check the status of the ModelArts resource pool.	Failed to obtain the ModelArts resource pool cpuArch.	Check the ModelArts resource pool status.
500	ModelArts.72020087	Parameter settings are incorrect. Please adjust the parameter settings.	Parameters are incorrectly configured.	Check the operator parameter settings.

Status Code	Error Code	Error Message	Description	Measure
500	ModelArts.72020091	Internal service error. Contact technical support.	Internal service error.	Contact customer service.
500	ModelArts.72020092	The dependent service [xxx] is error.	The dependent service is incorrect.	Check whether the dependent service is normal.
400	ModelArts.72020107	List analysis files failed for task [xxx] node [xxx].	Obtain the analysis file list.	Check whether the OBS service is normal.
400	ModelArts.72020108	Get analysis file details failed for path [xxx].	Failed to obtain the analysis file details.	Check whether the OBS service is normal.
400	ModelArts.72020109	The total analysis file size exceed max download limit.	The total size of analysis files exceeds the upper limit.	The total size of analysis files exceeds the upper limit.
400	ModelArts.72020118	Record num should be more than 0 and less equal than 10w.	The number of records must be greater than 0 and less than or equal to 100,000.	Check the input parameters of the data synthesis operator.
400	ModelArts.72020119	Concurrency num should be more than 0 and less equal than 500.	The number of concurrent tasks must be greater than 0 and less than or equal to 500.	Check the input parameters of the data synthesis operator.
400	ModelArts.72020120	The target num should be more than 1 and less equal than 20.	The number of targets must be greater than 1 and less than or equal to 20.	Check the input parameters of the data synthesis operator.

Status Code	Error Code	Error Message	Description	Measure
400	ModelArts.72020121	The target num should equal 1.	The number of targets must be 1.	Check the input parameters of the data generation operator.
400	ModelArts.72020122	The output parameters of the synthesis task are incorrect.	The output parameters of the synthesis task are incorrect.	Check the input parameters of the data synthesis operator.
400	ModelArts.72020123	Invalid synthesis config.	Invalid synthesis configuration.	Check the input parameters of the data synthesis operator.
400	ModelArts.72020124	The model is not configured.	No model is configured.	Check the model configuration of the data synthesis operator.
400	ModelArts.72020125	The sample input length should be less than 20.	The sample input length must be less than 20.	Check the input parameters of the data synthesis operator.
400	ModelArts.72020128	Update cleanJob failed.	Failed to update the refining job.	Check whether the input parameters of the API are correct.
400	ModelArts.72020129	CleanJob name is invalid.	Invalid refining job name.	Check whether the refining job name contains invalid characters.
400	ModelArts.72020130	CleanJob description is invalid.	Invalid refining job description.	Check whether the refining job description contains invalid characters.
403	ModelArts.72020131	Only services deployed in the public resource pool are supported.	Only real-time services deployed in public resource pools are supported.	Only real-time services deployed in public resource pools are supported.
403	ModelArts.72020132	Current user is not authorized, please create agency first.	The current user is not authorized.	Create an agency.

Status Code	Error Code	Error Message	Description	Measure
500	ModelArts.72020133	Fail to get agency info. Please try again later.	Failed to obtain the agency information. Try again later.	Check whether the user has created an agency.
400	ModelArts.72020136	Input file does not exist.	The task input file does not exist.	Check whether the task input file has been deleted from OBS.
400	ModelArts.72020137	Resuming clean job failed.	Failed to retry the refining job.	Contact technical support.
400	ModelArts.72020138	Can not find [xxx] models in preset asset.	The model cannot be found in the preset assets.	Check whether the model exists in the preset assets.
400	ModelArts.72020139	Operator does not support dataset type: [xxx]. The supported dataset types: [xxx].	The operator does not support the current dataset type.	You are advised to change the operator in the task orchestration and try again.
400	ModelArts.72090002	operation is unsupported, reason: xxx	Unsupported operation.	The current refining job status does not support this operation.
400	ModelArts.72090005	parameter xxx is invalid, reason: xxx	Invalid parameter.	Check whether the parameters are valid.
400	ModelArts.72090006	parse operator manifest xxx failed, reason: xxx	Failed to parse the operator configuration file.	Check the operator configuration file.
400	ModelArts.72090007	operator parameter xxx is invalid, reason: xxx	The operator parameter settings are invalid.	Check whether the parameters are valid.
400	ModelArts.72090010	operator id xxx already exists	The operator ID already exists.	Change the operator ID.

Status Code	Error Code	Error Message	Description	Measure
400	ModelArts.72090011	operator xxx does not exist	The operator does not exist.	Check whether the operator exists.
400	ModelArts.72090012	operator [xxx] version [xxx] does not exist	The operator of this version does not exist.	Check whether the operator of the current version exists.
400	ModelArts.72090015	update operator [xxx] failed, reason: [xxx]	Failed to update the operator.	Check whether the operator update file is correct.
400	ModelArts.72090019	operator [{0}] version [{1}] format error	The format of the operator version number is incorrect.	Check the format of the operator version number.
400	ModelArts.72090020	operator [{0}] version [{1}] is too old	The operator version is too early.	Check the operator version.
400	ModelArts.72090023	operator package [{0}] format error	The format of the operator package is incorrect.	Check the format of the operator package.
400	ModelArts.72090024	invalid obs url: [{0}], reason: [{1}]	Invalid OBS path.	Check whether the OBS path is valid.
400	ModelArts.72090025	operator file [{0}] size exceeds limit [{1}]	The number of operator files exceeds the upper limit.	Check the size of the operator package.
403	ModelArts.72090026	read obs error, reason:[{0}]	The OBS file fails to be accessed.	Check the OBS configuration.
400	ModelArts.72090027	please purchase first before use	Purchase the service before use.	Subscribe to the service.

Status Code	Error Code	Error Message	Description	Measure
400	ModelArts.72090028	Parameter settings are incorrect. Please adjust the parameter settings	The parameters are set incorrectly.	Modify the parameters.
400	ModelArts.72090029	Data parsing error. For details, see the log	Data parsing failed.	Correct the data.
400	ModelArts.72090030	The permission [{0}] is incorrect. Please configure the permission	Permission errors.	Configure the correct permission.
400	ModelArts.72091001	read file [{0}] failed	Failed to read the file.	Check whether the file is correct.
403	ModelArts.72091010	The dependent resource [{0}] is incorrect. Ensure that the [{1}] is normal	Failed to obtain the dependent resource.	Check whether the dependent resource is available.
500	ModelArts.72091011	Internal service error. Contact technical support	An internal error occurred.	Contact technical support.
403	ModelArts.72091012	The dependent service [{0}] is incorrect. Check whether the service [{1}] is normal	The dependent service failed.	Check whether the service is available.

Manual Calibration

Table 7-4 Manual calibration error codes

Status Code	Error Code	Error Message	Description	Measure
403	ModelArts.72041001	You do not have permission to do this operation!	No operation permission.	Check whether the user has operation permissions.
400	ModelArts.72041020	Dataset bound to the task record num exceeds the limit!	The number of dataset records associated with the task exceeds the upper limit.	Check whether the number of samples in the associated dataset exceeds the upper limit.
409	ModelArts.72041015	Batch task name is already exist!	The batch task name already exists.	Check whether a batch task with the same name exists in the current project space.
400	ModelArts.72041037	Dataset size is empty	The size of the associated dataset is empty.	Check the size of the associated dataset and select a dataset whose size is not empty.
404	ModelArts.72041016	Batch task does not exist!	The batch task information does not exist.	Check whether the batch task exists in the current project space.
400	ModelArts.72042010	The current task status does not allow dataset generation.	A dataset cannot be generated in the current task status.	A dataset cannot be generated when the task is being created or fails to be created. Check the task status.
400	ModelArts.72042011	Dataset is being generated and cannot be operated.	This operation is not allowed because the dataset is being generated.	This operation is not allowed because the dataset is being generated. Try again later.

Status Code	Error Code	Error Message	Description	Measure
400	ModelArts.72042012	No data record meets the filter criteria. Therefore, the dataset cannot be generated.	The number of data records that meet the filter criteria is empty. The dataset cannot be generated.	The number of data records that meet the filter criteria is empty. The dataset cannot be generated. Check the filter criteria.
500	ModelArts.72042013	Failed to stop dataset generation, please try again later.	Failed to stop dataset generation.	Failed to stop dataset generation. Try again later.

