

Starting April 29, 2025, Gemini 1.5 Pro and Gemini 1.5 Flash models are not available in projects that have no prior usage of these models, including new projects. For details, see [Model versions and lifecycle](#) (/vertex-ai/generative-ai/docs/learn/model-versions#legacy-stable).

Design multimodal prompts

The Gemini API lets you send multimodal prompts to the Gemini model. The supported modalities include text, image, and video.

For general prompt design guidance, see [Prompt design strategies](#) (https://cloud.google.com/vertex-ai/generative-ai/docs/learn/prompts/prompt-design-strategies).

You can improve your multimodal prompts by following these best practices:

- [Prompt design fundamentals](#) (#fundamentals)
 - **Be specific in your instructions:** Craft clear and concise instructions that leave minimal room for misinterpretation.
 - **Add a few examples to your prompt:** Use realistic few-shot examples to illustrate what you want to achieve.
 - **Break it down step-by-step:** Divide complex tasks into manageable sub-goals, guiding the model through the process.
 - **Specify the output format:** In your prompt, ask for the output to be in the format you want, like markdown, JSON, HTML and more.
 - **Put your image first for single-image prompts:** While Gemini can handle image and text inputs in any order, for prompts containing a single image, it might perform better if that image (or video) is placed before the text prompt. However, for prompts that require images to be highly interleaved with texts to make sense, use whatever order is most natural.
- [Troubleshooting your multimodal prompt](#) (#troubleshooting)
 - **If the model is not drawing information from the relevant part of the image:** Drop hints with which aspects of the image you want the prompt to draw information from.

- **If the model output is too generic (not tailored enough to the image/video input):** At the start of the prompt, try asking the model to describe the image(s) or video before providing the task instruction, or try asking the model to refer to what's in the image.
- **To troubleshoot which part failed:** Ask the model to describe the image, or ask the model to explain its reasoning, to gauge the model's initial understanding.
- **If your prompt results in hallucinated content:** Try dialing down the temperature setting or asking the model for shorter descriptions so that it's less likely to extrapolate additional details.
- **Tuning the sampling parameters:** Experiment with different temperature settings and top-k selections to adjust the model's creativity.

Prompt design fundamentals

This section expands on the best practices listed in the previous section.

Note: Model results from the following Gemini examples can vary from run to run.

Be specific in your instructions

Prompts have the most success when they are clear and detailed. If you have a specific output in mind, it's better to include that requirement in the prompt to ensure you get the output you want.

Sometimes, a prompt's intent might seem clear to the person who wrote it, but it is actually under-specified. To customize model behavior, consider how your prompt could be (mis)interpreted, and ensure that the instructions you give are specific and clear. Remember to leave minimal room for misinterpretation

For this image of an airport board, asking the model to just "describe this image" could generate a general description. If you need the model to parse the time and city from the image, you can include that request directly in your prompt.

Prompt	Model response
--------	----------------



The image shows an airport arrivals and departures board.

Describe this image.

Updated prompt



Parse the time and city from the airport board shown in this image into a list.

Improved response

- 10:50 Moscow
- 11:05 Edinburgh
- 11:05 London
- 11:10 Bucharest
- 11:30 Kiev
- 11:35 Dublin
- 11:45 East Midlands
- 12:15 Sofia
- 12:30 London
- 12:30 Newcastle
- 12:40 St Petersburg
- 12:40 London
- 12:45 Manchester

Add a few examples

The Gemini model can accept multiple inputs which it can use as examples to understand the output you want. Adding these examples can help the model identify the patterns and apply the relationship between the given images and responses to the new example. This is also called "few-shot" learning.

In the example below, the initial output is written in sentence form, and also contains the country (Brazil). Suppose you want the output in a different format or style, and you want only the city, not the country. Adding few-shot examples to your prompt can steer the model to respond in the way you want.

Prompt

Determine the city along with the landmark.

Model response

The landmark is the Christ the Redeemer statue in Rio de Janeiro, Brazil.

Updated prompt

Improved response

Determine the city along with the landmark.

city: Rome, landmark: the Colosseum.



city: Beijing, landmark: Forbidden City



city: Rio de Janeiro, landmark: Christ the Redeemer statue

Break it down step-by-step

For complex tasks like the ones that require both visual understanding and reasoning, it can be helpful to split the task into smaller, more straightforward steps. Alternatively, it could also be effective if you directly ask the model to “think step by step” in your prompt.

Prompt

When will I run out of toilet paper?

Model response

Soon, you only have 3 rolls left.

Updated prompt

1. First, count how many toilet paper rolls are in this picture.
2. Then, determine how much toilet paper a typical person uses per day.
3. Calculate how long these rolls of toilet paper will last.

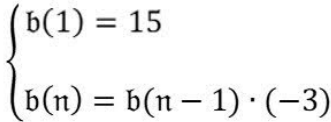
Improved response

1. There are 3 rolls of toilet paper in this picture.
2. A typical person uses about 20 sheets of toilet paper per day.
3. If each roll contains 200 sheets, then each roll will last for about 10 days. Therefore, the 3 rolls will last for about a month.

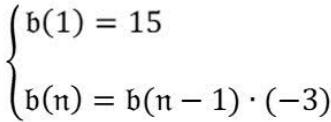
Math problems or other types of word problems are great candidates for asking the model to think step-by-step.

Prompt	Response
	-135
What is the 4th term in the sequence?	

The response from the model is incorrect. Some ways to improve this is to ask the model to parse the problem before answering the prompt.

Prompt	Response
	<i>Click to expand the result</i>
Parse the formula in the image first.	
Then based on the formula, answer "what's the 4th term in the sequence?".	

This model response is correct - now to make the response even more streamlined, you can ask the model to to think step-by-step.

Updated prompt	Improved response
	The first term is 15.
What is the 4th term in the sequence? Think step by step.	The second term is $b(1) * (-3) = 15 * (-3) = -45$.
	The third term is $b(2) * (-3) = -45 * (-3) = 135$.
	The fourth term is $b(3) * (-3) = 135 * (-3) = -405$.

Try specifying the output format

A common problem is the need for model output to be in a certain format (e.g. markdown, JSON, HTML), especially if the model output needs to be ingested by a downstream task. You can try to produce output in that format by instructing the model to do so within the prompt. Below are two examples:

Prompt

Subject	Status	Time	Start date	Due on
Math	Done	1.5 hrs	11/29/2023	12/1/2023
Geography	Done	30 min	12/2/2023	12/4/2023
Physics	In progress	2 hrs	12/3/2023	12/5/2023
Biology	Not started	6 hrs	11/2/2016	12/6/2023

Parse the table in this image into markdown format

Model response

| Subject | Status | Time | Start date | Due on |


| ----- | ----- | --- | ----- | ----- |

| Math | Done | 1.5 hrs | 11/29/2023 | 12/1/2023 |

| Geography | Done | 30 min | 12/2/2023 | 12/4/2023 |

| Physics | In progress | 2 hrs | 12/3/2023 | 12/5/2023 |

| Biology | Not started | 6 hrs | 11/2/2016 | 12/6/2023 |

Prompt	Model response
 <p>Provide a list of all the following attributes: ingredients, type of cuisine, vegetarian or not, in JSON format</p>	<pre>```json``` { "ingredients": ["rice", "shrimp", "clams", "mussels", "peas", "tomatoes", "onions", "garlic", "olive oil",], }</pre>


```
"paprika",  
"salt",  
"pepper"  
],  
"type of cuisine": "Spanish",  
"vegetarian": false
```

Put your image first for single-image prompts


While Gemini can interpret image and text in any order within a prompt, placing a single image before the text prompt might lead to better results. In the following samples, you'll notice the image or video comes first before the text in each prompt.

Troubleshooting your multimodal prompt

You might need to troubleshoot your prompt if you are not getting a helpful response. Here are a few strategies you could try.

If the model is not drawing information from the relevant part of the image

To get a more specific response, you can point out which aspects of the image you want the prompt to draw information from to inform its response.

Prompt	Model response
	The diapers will last for 198 days before they run out.
How many days will these diapers last a baby?	

Updated prompt

Improved response

The box contains 198 diapers. The box states the diapers are for a child weighing 22-37 lbs.

The average child in this weight range is likely between 1 and 2 years old.

A 2-year-old typically goes through 6 diapers per day, therefore, the diapers will last around 33 days (198/6).

How long will these diapers last before I run out?

Use the weight shown on the box to determine the child's age, and use the total number of diapers in the box.

Divide the total number by how many diapers the child goes through per day.

If the model output is too generic and not tailored enough to the image or video input

To help the model tailor its response to the image(s), try asking it to describe the images before performing its reasoning task.

Prompt**Model response**

All four images are examples of landscape photography.



What is in common between these images?

Updated prompt

Improved response



The first image shows a person packing a suitcase. The second image shows an empty hallway in a shopping mall. The third image shows a roll of toilet paper. The fourth image shows a dark and stormy sea.

The commonality between these images is that they are all examples of liminal spaces. Liminal spaces are places or situations that are in between two other places or situations. They are often characterized by a sense of uncertainty or ambiguity.

The first image shows the person in between being at home and being on a trip. The second image shows the hallway in between two stores. The third image is of toilet paper, which can represent a bathroom. A bathroom is a transitional space between private and public realms. It's a place where we prepare ourselves to enter or leave the outside world.

The fourth image shows the sea in between being calm and being stormy.

Liminal spaces can be both exciting and unsettling. They can be a place of new beginning or a place of endings. They can be a place of reflection or a place of action.

First, describe what's in each image in detail. What's in common between these images?

You can also try explicitly asking the model to refer to what's in the image.

Updated prompt



Improved response


All of the images are related to travel. The first image shows a person packing a suitcase, the second image shows an empty hallway in a shopping mall, the third image shows a stack of toilet paper, and the fourth image shows a dark and stormy sea.

These images are all related to travel because they are all things that people need or use when they travel. The first image shows a person packing a suitcase, which is something that people do when they are going on a trip. The second image shows an empty hallway in a shopping mall, which is a place where people often go to buy things for their trips. The third image shows a stack of toilet paper, which is something that people often use when they are traveling. The fourth image shows a dark and stormy sea, which is something that people often encounter when they are traveling.



What is in common between these images? Refer to what's in the images in your response.

The same tips work for creative output from images or videos. In the following example with the rabbit, explicitly asking the model to include aspects from the image can improve the output.

Prompt	Model response
 <p>Can you write me a poem about this image?</p>	<p>Little bunny, so soft and brown, Hopping through the field, up and down. Your ears so long, your tail so small, You're the cutest creature of them all.</p> <p>You nibble on the grass, so green and sweet, And hop away when you see me. But I'll always be your friend, Little bunny, until the very end.</p>
Updated prompt	Improved response



Can you write me a descriptive and dramatic poem about this image and include the location?

Little bunny on the green,
Ears so long, so keen.
What do you see out there,
In the distance, where you stare?

Is it the ocean blue,
Sparkling in the sun's hue?
Or the waves that crash and roar,
A rhythm that you adore?

Little bunny, tell me true,
What captures your attention so?
Is it the salty breeze,
Or the seagulls' cries that please?

Perhaps it's the vastness of it all,
A world so big and small.
A place where dreams can roam,
And adventures can be sown.

Little bunny, I wonder what you dream,
As you sit on the grass, so serene.
Do you yearn to explore the deep,
Or stay on land, where you can leap?

Whatever it may be, little bunny,
Keep that spark of wonder burning brightly.
For in your dreams and aspirations,
Lies a world waiting for your creation.

Troubleshooting which part of the prompt failed

It can be hard to know whether a prompt failed because the model didn't **understand the image** to begin with, or if it did understand the image but did not perform the correct **reasoning steps** afterward.

To disambiguate between those reasons, ask the model to describe what's in the image.

In this example below, if the model responds with a snack that seems surprising when paired with tea (e.g. popcorn), you can first troubleshoot to determine whether the model correctly recognized that the image contains tea.

Prompt

Prompt for troubleshooting



What's a snack I can make in 1 minute that would go well with this?

Describe what's in this image.

Another strategy is to ask the model to explain its reasoning. That can help you narrow down which part of the reasoning broke down, if any.

Prompt

Prompt for troubleshooting



What's a snack I can make in 1 minute that would go well with this?

What's a snack I can make in 1 minute that would go well with this? Please explain why.

Tuning the sampling parameters

In each request, you send not only the multimodal prompt but a set of sampling parameters to the model. The model can generate different results for different parameter values. Experiment with the different parameters to get the best values for the task. The most commonly adjusted parameters are the following:

- Temperature
- top-P

- top-K

Temperature

Temperature is used for sampling during response generation, which occurs when top-P and top-K are applied. Temperature controls the degree of randomness in token selection. Lower temperatures are good for prompts that require a more deterministic and less open-ended or creative response, while higher temperatures can lead to more diverse or creative results. A temperature of 0 is deterministic, meaning that the highest probability response is always selected.

For most use cases, try starting with a temperature of 0.4. If you need more creative results, try increasing the temperature. If you observe clear hallucinations, try reducing the temperature.

Top-K

Top-K changes how the model selects tokens for output. A top-K of 1 means the next selected token is the most probable among all tokens in the model's vocabulary (also called greedy decoding), while a top-K of 3 means that the next token is selected from among the three most probable tokens by using temperature.

For each token selection step, the top-K tokens with the highest probabilities are sampled. Then tokens are further filtered based on top-P with the final token selected using temperature sampling.

Specify a lower value for less random responses and a higher value for more random responses. The default value of top-K is 32.

Top-P

Top-P changes how the model selects tokens for output. Tokens are selected from the most (see top-K) to least probable until the sum of their probabilities equals the top-P value. For example, if tokens A, B, and C have a probability of 0.6, 0.3, 0.1 and the top-P value is 0.9, then the model will select either A or B as the next token by using temperature and excludes C as a candidate.

Specify a lower value for less random responses and a higher value for more random responses. The default value of top-P is 1.0.

What's next

- Try a quickstart tutorial using [Vertex AI Studio](#) (/vertex-ai/generative-ai/docs/start/quickstarts/quickstart) or the [Vertex AI API](#)

(/vertex-ai/generative-ai/docs/start/quickstarts/quickstart-multimodal).

- To get started using the Gemini API, see the [Gemini API quickstart](#) (/vertex-ai/generative-ai/docs/start/quickstarts/quickstart-multimodal).

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (https://creativecommons.org/licenses/by/4.0/), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (https://www.apache.org/licenses/LICENSE-2.0). For details, see the [Google Developers Site Policies](https://developers.google.com/site-policies) (https://developers.google.com/site-policies). Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2025-06-06 UTC.