Learn more (https://developers.googleblog.com/en/the-next-chapter-of-the-gemini-era-for-developers/)

Grounding with Google Search

Important: We're launching Grounding with Google Search! Review the updated Gemini API Additional Terms of Service (/gemini-api/terms), which include new feature terms and updates for clarity.

✓ Python Node.js REST

Try a Colab notebook (https://colab.research.google.com/github/google-gemini/cookbook/blob/main/quickstarts/Search_Grounding.ipynb)

View notebook on GitHub (https://github.com/google-gemini/cookbook/blob/main/quickstarts/Search_Groundin

The Grounding with Google Search feature in the Gemini API and AI Studio can be used to improve the accuracy and recency of responses from the model. In addition to more factual responses, when Grounding with Google Search is enabled, the Gemini API returns grounding sources (in-line supporting links) and Google Search Suggestions (#search-suggestions) along with the response content. The Search Suggestions point users to the search results corresponding to the grounded response.

Grounding with Google Search only supports text prompts. It doesn't support multimodal (text-and-image, text-and-audio, etc.) prompts. Grounding with Google Search supports all of the available languages (/gemini-api/docs/models/gemini#available-languages) for Gemini models.

This guide will help you get started with Grounding with Google Search using one of the Gemini API SDKs or the REST API.

Configure a model to use Google Search

Tip: Before running the example code, make sure that you've followed the installation and setup instructions in the quickstart (/gemini-api/docs/quickstart).

There are two ways to configure a model to use Grounding with Google Search: using a string (#configure-by-string) and using a dictionary (#configure-by-dictionary).

Configure by string

```
model = genai.GenerativeModel('models/gemini-1.5-pro-002')
response = model.generate_content(contents="Who won Wimbledon this year?",
                                  tools='google_search_retrieval')
print(response)
```

Configure by dictionary

```
model = genai.GenerativeModel('models/gemini-1.5-pro-002')
response = model.generate_content(
    contents="Who won Wimbledon this year?",
    tools={"google_search_retrieval": {
        "dynamic_retrieval_config": {
            "mode": "unspecified",
            "dynamic_threshold": 0.06}}})
print(response)
```

For a dictionary implementation, you don't have to pass in key-value pairs for **`mode`** or **`dynamic_threshold`**. You can omit them and let the model use default values, as in the following example:

```
model = genai.GenerativeModel('models/gemini-1.5-pro-002')
response = model.generate_content(contents="Who won Wimbledon this year?",
                                  tools={"google_search_retrieval": {}})
print(response)
```

The **`mode`** and **`dynamic_threshold`** settings let you control the behavior of dynamic retrieval (#dynamic-retrieval), giving you additional control over when to use Grounding with Google Search.

Why is Grounding with Google Search useful?

In generative AI, *grounding* refers to the process of connecting the model to verifiable sources of information. These sources might provide real-world workplace information or other specific context. Grounding helps with improving the accuracy, reliability, and usefulness of AI outputs.

Grounding is particularly important for prompts that require up-to-date information from the web. Using grounding, the model can access information beyond its knowledge cutoff date, get sources for the information, and answer questions that it couldn't have answered accurately otherwise.

Using Google AI Studio or the Gemini API, you can ground model output to Google Search. Grounding with Google Search provides the following benefits:

- Allows model responses that are tethered to specific content.
- Reduces model hallucinations, which are instances where the model generates content that isn't factual.
- Anchors model responses to sources a user can click through and open.
- Enhances the trustworthiness and applicability of the generated content.

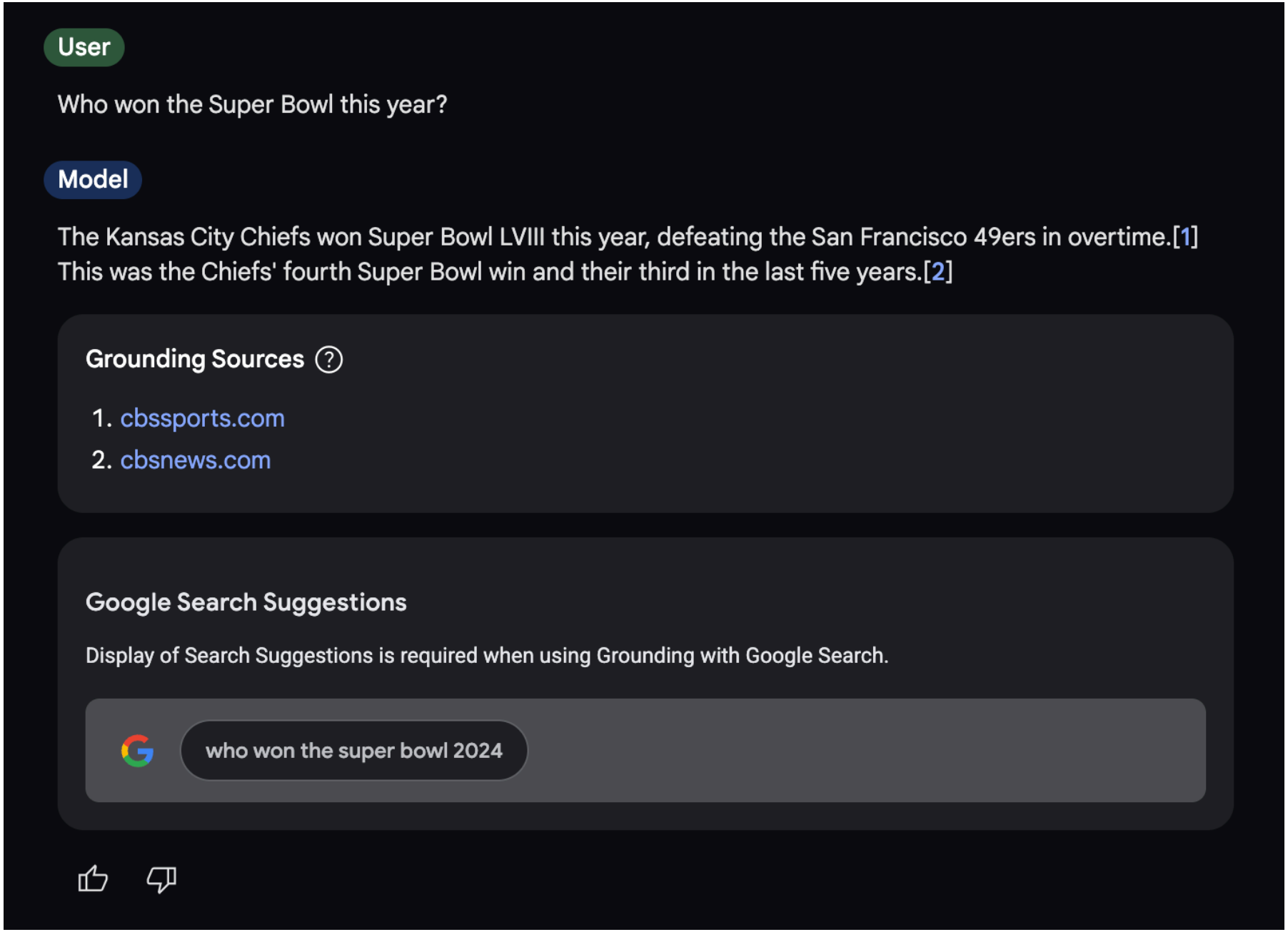
When you use Grounding with Google Search, you're effectively connecting the model to reliable Search results from the internet. Since non-grounded model responses are based on learned patterns, you might not get factual responses to prompts about current events (for example, asking for a weather forecast or the final score of a recent football game). Since the internet provides access to new information, a grounded prompt can generate more up-to-date responses, with sources cited.

Here's an example comparing a non-grounded response and a grounded response generated using the API. (The responses were generated in October 2024.)

Ungrounded Gemini	Grounding with Google Search
Prompt: Who won the Super Bowl this year? **Response:** The Kansas City Chiefs won Super Bowl LVII this year (2023).	**Prompt:** Who won the Super Bowl this year? **Response:** The Kansas City Chiefs won Super Bowl LVIII this year, defeating the San Francisco 49ers in overtime with a score of 25 to 22.

In the ungrounded response, the model refers to the Kansas City Chiefs' 2023 Super Bowl win. In the grounded response, the model correctly references their more recent 2024 win.

The following image shows how a grounded response looks in AI Studio.


User
Who won the Super Bowl this year?
Model
The Kansas City Chiefs won Super Bowl LVIII this year, defeating the San Francisco 49ers in overtime.[1] This was the Chiefs' fourth Super Bowl win and their third in the last five years.[2]
Grounding Sources
1. cbssports.com
2. cbsnews.com
Google Search Suggestions
Display of Search Suggestions is required when using Grounding with Google Search.
who won the super bowl 2024

Google Search Suggestions

To use Grounding with Google Search, you have to display Google Search Suggestions, which are suggested queries included in the metadata of the grounded response. To learn more about the display requirements, see Use Google Search Suggestions (/gemini-api/docs/grounding/search-suggestions).

Dynamic retrieval

Some queries are likely to benefit more from Grounding with Google Search than others. The *dynamic retrieval* feature gives you additional control over when to use Grounding with Google Search.

If the dynamic retrieval mode is unspecified, Grounding with Google Search is always triggered. If the mode is set to dynamic, the model decides when to use grounding based on a threshold that you can configure. The threshold is a floating-point value in the range [0,1] and defaults to 0.3. If the threshold value is 0, the response is always grounded with Google Search; if it's 1, it never is.

How dynamic retrieval works

You can use dynamic retrieval in your request to choose when to turn on Grounding with Google Search. This is useful when the prompt doesn't require an answer grounded in Google Search and the model can provide an answer based on its own knowledge without grounding. This helps you manage latency, quality, and cost more effectively.

Before you invoke the dynamic retrieval configuration in your request, understand the following terminology:

- **Prediction score**: When you request a grounded answer, Gemini assigns a *prediction score* to the prompt. The prediction score is a floating point value in the range [0,1]. Its value depends on whether the prompt can benefit from grounding the answer with the most

up-to-date information from Google Search. Thus, if a prompt requires an answer grounded in the most recent facts on the web, it has a higher prediction score. A prompt for which a model-generated answer is sufficient has a lower prediction score.

Here are examples of some prompts and their prediction scores.

★ **Note:** The prediction scores are assigned by Gemini and can vary over time depending on several factors.

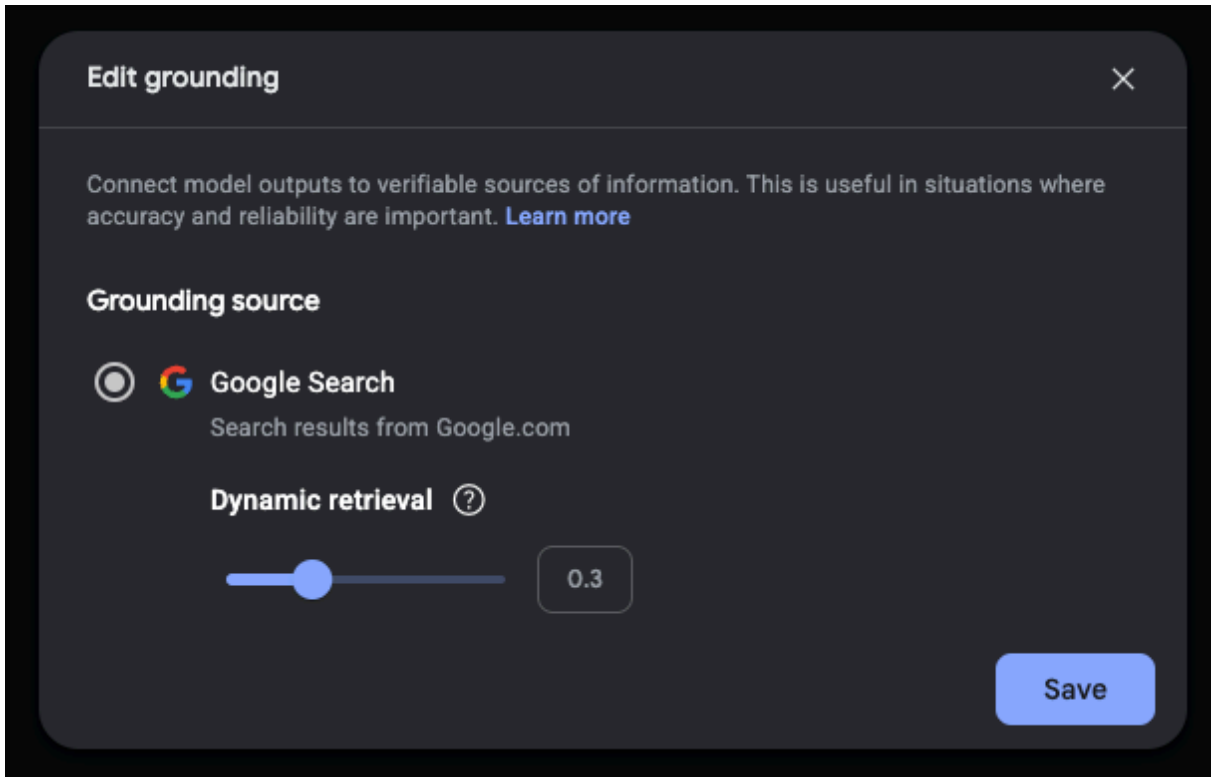
Prompt	Prediction score	Comment
"Write a poem about peonies"	0.13	The model can rely on its knowledge and the answer doesn't need grounding.
"Suggest a toy for a 2yo child"	0.36	The model can rely on its knowledge and the answer doesn't need grounding.
"Can you give a recipe for an asian-inspired guacamole?"	0.55	Google Search can give a grounded answer, but grounding isn't strictly required; the model knowledge might be sufficient.
"What's Agent Builder? How is grounding billed in Agent Builder?"	0.72	Requires Google Search to generate a well-grounded answer.
"Who won the latest F1 grand prix?"	0.97	Requires Google Search to generate a well-grounded answer.

- **Threshold**: In your API request, you can specify a dynamic retrieval configuration with a threshold. The threshold is a floating point value in the range [0,1] and defaults to 0.3. If the threshold value is zero, the response is always grounded with Google Search. For all other values of threshold, the following is applicable:

- If the prediction score is greater than or equal to the threshold, the answer is grounded with Google Search. A lower threshold implies that more prompts have responses that are generated using Grounding with Google Search.
- If the prediction score is less than the threshold, the model might still generate the answer, but it isn't grounded with Google Search.

To learn how to set the dynamic retrieval threshold using an SDK or the REST API, see the appropriate code example (#configure-grounding).

If you're using AI Studio, you can set the dynamic retrieval threshold by clicking **Edit grounding**.



To find a good threshold that suits your business needs, you can create a representative set of queries that you expect to encounter. Then you can sort the queries according to the prediction score in the response and select a good threshold for your use case.

A grounded response

If your prompt successfully grounds to Google Search, the response will include **`groundingMetadata`**. A grounded response might look something like this (parts of the response have been omitted for brevity):


```
{
  "candidates": [
    {
      "content": {
        "parts": [
          {
            "text": "Carlos Alcaraz won the Gentlemen's Singles title at the 2024 Wimbledon Championships. He defeate
          }
        ],
        "role": "model"
      },
      ...
      "groundingMetadata": {
        "searchEntryPoint": {
          "renderedContent": "\u003cstyle\u003e\n.container {\n  align-items: center;\n  border-radius: 8px;\n  displ
        },
        "groundingChunks": [
          {
            "web": {
              "uri": "https://vertexaisearch.cloud.google.com/grounding-api-redirect/AWhgh4whET1ta3sDETZvcicd8FeNe4z0
```

```
        "title": "wikipedia.org"
      }
    },
    {
      "web": {
        "uri": "https://vertexaisearch.cloud.google.com/grounding-api-redirect/AWhgh4wR1M-9-yMPUr_KdHlnoAmQ8ZX9
        "title": "wikipedia.org"
      }
    },
    {
      "web": {
        "uri": "https://vertexaisearch.cloud.google.com/grounding-api-redirect/AWhgh4wsDmROzbP-tmt8GdwCW_pqISTZ
        "title": "cbssports.com"
      }
    },
    {
      "web": {
        "uri": "https://vertexaisearch.cloud.google.com/grounding-api-redirect/AWhgh4yzjLkorHiUKjhOPkWaZ9b4cO-c
        "title": "jagranjosh.com"
      }
    },
    {
      "web": {
        "uri": "https://vertexaisearch.cloud.google.com/grounding-api-redirect/AWhgh4y9L4oeNGWCatFz63b9PpP3ys-W
        "title": "apnews.com"
      }
    }
  ],
  "groundingSupports": [
    {
      "segment": {
        "endIndex": 85,
```

```
      "text": "Carlos Alcaraz won the Gentlemen's Singles title at the 2024 Wimbledon Championships."
    },
    "groundingChunkIndices": [
      0,
      1,
      2,
      3
    ],
    "confidenceScores": [
      0.97380733,
      0.97380733,
      0.97380733,
      0.97380733
    ]
  },
  {
    "segment": {
      "startIndex": 86,
      "endIndex": 210,
      "text": "He defeated Novak Djokovic in the final, winning his second consecutive Wimbledon title and fo
    },
    "groundingChunkIndices": [
      1,
      0,
      4
    ],
    "confidenceScores": [
      0.96145374,
      0.96145374,
      0.96145374
    ]
  }
```

```
        ],
        "webSearchQueries": [
          "who won wimbledon 2024"
        ]
      }
    }
  ],
  ...
}
```

If the response doesn't include `groundingMetadata`, this means the response wasn't successfully grounded. There are several reasons this could happen, including low source relevance or incomplete information within the model response.

When a grounded result is generated, the metadata contains URIs that redirect to the publishers of the content that was used to generate the grounded result. These URIs contain the `vertexaisearch` subdomain, as in this truncated example: `https://vertexaisearch.cloud.google.com/grounding-api-redirect/...`. The metadata also contains the publishers' domains. The provided URIs remain accessible for 30 days after the grounded result is generated.

Important: The provided URIs must be directly accessible by the end users and must not be queried programmatically through automated means. If automated access is detected, the grounded answer generation service might stop providing the redirection URIs.

The `renderedContent` field within `searchEntryPoint` is the provided code for implementing Google Search Suggestions. See Use Google Search Suggestions (/gemini-api/docs/grounding/search-suggestions) to learn more.

Last updated 2024-12-19 UTC.