We are releasing new models, reducing prices for GPT-3.5 Turbo, and introducing new ways for developers to manage API keys and understand API usage. The new models include:

- Two new embedding models
- An updated GPT-4 Turbo preview model
- An updated GPT-3.5 Turbo model
- An updated text moderation model

By default, data sent to the OpenAI API will not be used to train or improve OpenAI models.

New embedding models with lower pricing

We are introducing two new embedding models: a smaller and highly efficient `text-embedding-3-small` model, and a larger and more powerful `text-embedding-3-large` model.

An embedding is a sequence of numbers that represents the concepts within content such as natural language or code. Embeddings make it easy for machine learning models and other algorithms to understand the relationships between content and to perform tasks like clustering or retrieval. They power applications like knowledge retrieval in both ChatGPT and the Assistants API, and many retrieval augmented generation (RAG) developer tools.

A new small text embedding model

`text-embedding-3-small` is our new highly efficient embedding model and provides a significant upgrade over its predecessor, the `text-embedding-ada-002` model released in December 2022.

Stronger performance. Comparing `text-embedding-ada-002` to `text-embedding-3-small`, the average score on a commonly used benchmark for multi-language retrieval (MIRACL) has increased from 31.4% to 44.0%, while the average score on a commonly used benchmark for English tasks (MTEB) has increased from 61.0% to 62.3%.

Reduced price. `text-embedding-3-small` is also substantially more efficient than our previous generation `text-embedding-ada-002` model. Pricing for `text-embedding-3-small` has

therefore been reduced by 5X compared to `text-embedding-ada-002`, from a price per 1k tokens of $0.0001 to $0.00002.

We are not deprecating `text-embedding-ada-002`, so while we recommend the newer model, customers are welcome to continue using the previous generation model.

A new large text embedding model: `text-embedding-3-large`

`text-embedding-3-large` is our new next generation larger embedding model and creates embeddings with up to 3072 dimensions.

Stronger performance. `text-embedding-3-large` is our new best performing model. Comparing `text-embedding-ada-002` to `text-embedding-3-large`: on MIRACL, the average score has increased from 31.4% to 54.9%, while on MTEB, the average score has increased from 61.0% to 64.6%.

Eval benchmark	ada v2	text-embedding-3-small	text-embedding-3-large
MIRACL average	31.4	44.0	54.9
MTEB average	61.0	62.3	64.6

`text-embedding-3-large` will be priced at $0.00013 / 1k tokens.

You can learn more about using the new embedding models in our Embeddings guide.

Native support for shortening embeddings

Using larger embeddings, for example storing them in a vector store for retrieval, generally costs more and consumes more compute, memory and storage than using smaller embeddings.

Both of our new embedding models were trained with a technique that allows developers to trade-off performance and cost of using embeddings. Specifically, developers can shorten embeddings (i.e. remove some numbers from the end of the sequence) without the embedding losing its concept-representing properties by passing in the `dimensions` API parameter. For example, on the MTEB benchmark, a `text-embedding-3-large` embedding can be shortened

to a size of 256 while still outperforming an unshortened `text-embedding-ada-002` embedding with a size of 1536.

	ada v2	text-embedding-3-small		text-embedding-3-large		
Embedding size	1536	512	1536	256	1024	3072
Average MTEB score	61.0	61.6	62.3	62.0	64.1	64.6

This enables very flexible usage. For example, when using a vector data store that only supports embeddings up to 1024 dimensions long, developers can now still use our best embedding model `text-embedding-3-large` and specify a value of 1024 for the `dimensions` API parameter, which will shorten the embedding down from 3072 dimensions, trading off some accuracy in exchange for the smaller vector size.

Other new models and lower pricing

Updated GPT-3.5 Turbo model and lower pricing

Next week we are introducing a new GPT-3.5 Turbo model, `gpt-3.5-turbo-0125`, and for the third time in the past year, we will be decreasing prices on GPT-3.5 Turbo to help our customers scale. Input prices for the new model are reduced by 50% to $0.0005 /1K tokens and output prices are reduced by 25% to $0.0015 /1K tokens. This model will also have various improvements including higher accuracy at responding in requested formats and a fix for a bug which caused a text encoding issue for non-English language function calls.

Customers using the pinned `gpt-3.5-turbo` model alias will be automatically upgraded from `gpt-3.5-turbo-0613` to `gpt-3.5-turbo-0125` two weeks after this model launches.

Updated GPT-4 Turbo preview

Over 70% of requests from GPT-4 API customers have transitioned to GPT-4 Turbo since its release, as developers take advantage of its updated knowledge cutoff, larger 128k context windows, and lower prices.

Today, we are releasing an updated GPT-4 Turbo preview model, `gpt-4-0125-preview`. This model completes tasks like code generation more thoroughly than the previous preview model

and is intended to reduce cases of “laziness” where the model doesn’t complete a task. The new model also includes the fix for the bug impacting non-English UTF-8 generations.

For those who want to be automatically upgraded to new GPT-4 Turbo preview versions, we are also introducing a new `gpt-4-turbo-preview` model name alias, which will always point to our latest GPT-4 Turbo preview model.

We plan to launch GPT-4 Turbo with vision in general availability in the coming months.

Updated moderation model

The free Moderation API allows developers to identify potentially harmful text. As part of our ongoing safety work, we are releasing `text-moderation-007`, our most robust moderation model to-date. The `text-moderation-latest` and `text-moderation-stable` aliases have been updated to point to it. You can learn more about building safe AI systems through our safety best practices guide.

New ways to understand API usage and manage API keys

We are launching two platform improvements to give developers both more visibility into their usage and control over API keys.

First, developers can now assign permissions to API keys from the API keys page. For example, a key could be assigned read-only access to power an internal tracking dashboard, or restricted to only access certain endpoints.

Second, the usage dashboard and usage export function now expose metrics on an API key level after turning on tracking. This makes it simple to view usage on a per feature, team, product, or project level, simply by having separate API keys for each.

In the coming months, we plan to further improve the ability for developers to view their API usage and manage API keys, especially in larger organizations.