GiMATE Coder

User Guide

Table of Contents

1. System Requirements and Compatibility

Recommended system

Item	Details
Operation System	Windows11
GPU	NVIDIA GPU only (VRAM ≥ 8 GB)
DRAM	Greater than or equal to 16 GB
GPU driver	Version 573.26 or later
CUDA	Greater than or equal to 12.8

2. Installation, Update, and Uninstallation

Installation

GiMATE Coder is bundled with the GiMATE main program; therefore, no separate installation is required. Currently, the software does not support standalone installation outside of GiMATE. You can enable the GiMATE Coder service through the Apps section in GiMATE.

Update

In addition to viewing version information within GiMATE Coder, users can also check the version in the Live Update page of the GiMATE UI. If an update is available, a notification will appear on the Live Update page. Updates must be performed through GiMATE.

Uninstallation

Users can remove GiMATE Coder via Installed Apps in Windows settings.

3. Interface Overview

3.1 Main Interface
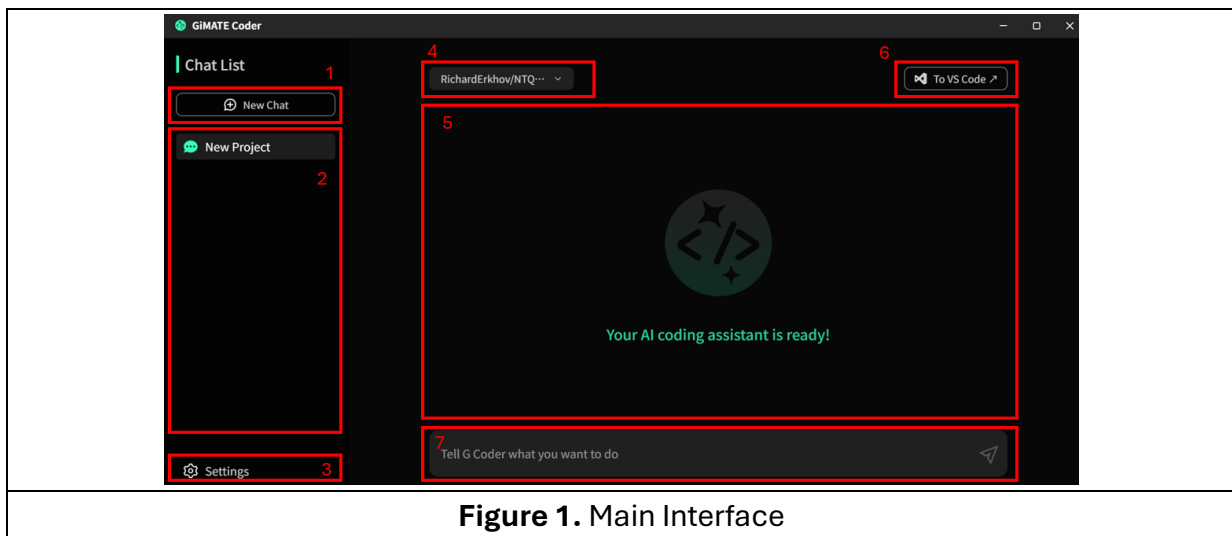
Figure 1. Main Interface

Figure 1 shows the main interface.

NO.	Name	Function / Purpose
1	New Chat	Each conversation unit records the interactions between the user and the model independently. The units are separate from one another and do not affect each other.
2	Session list	This list displays all conversation units that have been created and not yet deleted.
3	Settings	The Settings page integrates functions such as model download, resource monitoring, and system support, making it easier for users to configure and manage.
4	Model select list	Users can select a downloaded model from this menu to start a conversation. The list only displays models that have been fully downloaded.
5	Chat	Displays the user's input and the responses generated by the model.
6	[*To VS Code*] Button	This button connects the GiMATE Coder service to the **Continue** extension in Visual Studio Code, enabling quick integration and efficient development workflows.
7	Input Chat Bar	Allows the user to enter content for interaction with the model. By clicking the button on the right, the input is sent. The model then generates a response, which is displayed in real time in the conversation window.

3.2 To VS Code Page

When the user clicks the **To VS Code** button, the page navigates to the *To VS Code* settings page, as shown in Figure 2.

Figure 2. To VS Code Settings Page
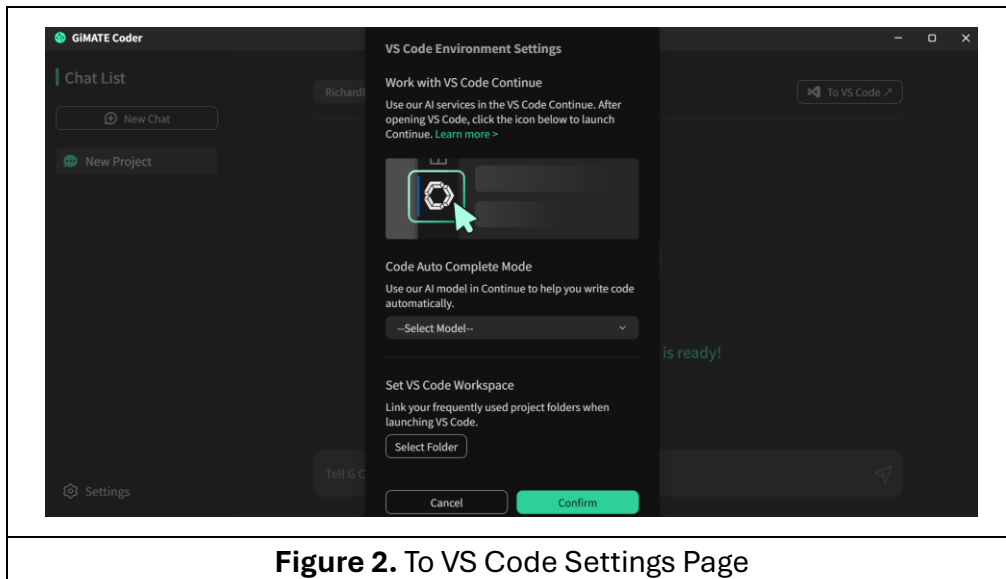
This page contains two settings. After configuration, GiMATE Coder will automatically launch VS Code, activate the Continue extension, and enable the auto-completion feature.

a. **Configure Auto Code Completion Model (Code Auto Complete Model):**

GiMATE Coder not only provides a chat-based service but also supports the auto code completion feature through VS Code. This feature allows the model to automatically generate, and complete code based on the user's coding context. Before enabling this feature, users must select an auto-completion model in the **To VS Code** settings page to ensure proper operation.

b. **Configure VS Code Workspace:**

Users can select a specific workspace here. When VS Code launches, GiMATE Coder will automatically apply the settings associated with the selected workspace. If no workspace is specified, VS Code will still open, but without any predefined settings applied.

3.3 Settings Page

3.3.1 Model Installation
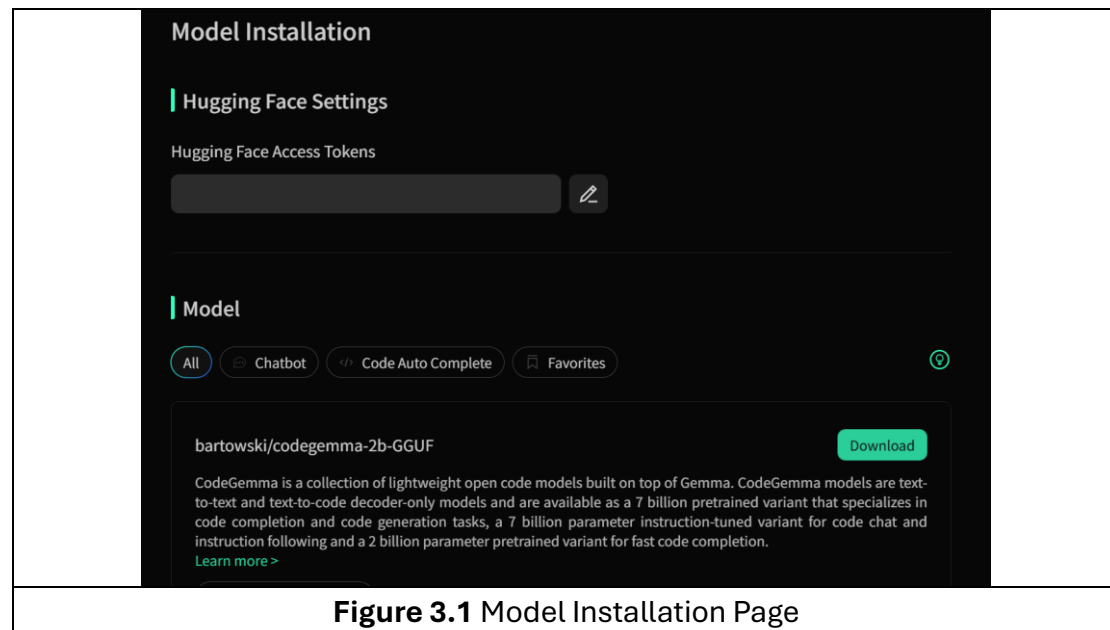
Figure 3.1 Model Installation Page

This page is primarily used for downloading and managing models.

<u>Hugging Face Settings:</u>

Currently, GiMATE Coder only supports downloading models from the Hugging Face platform. Before downloading, users must enter a valid Hugging Face Token in the Settings page. For instructions on obtaining and configuring the token, refer to the Appendix.

<u>Model:</u>

About the Model List

This list displays all models currently supported by GiMATE Coder, including chat models and auto code completion models. Models can be viewed by category using different tabs.

   a. **Model Download:** Users can click Download in the list (see Figure 3.2) to download the corresponding model.
   b. **Favorite Model Setting:** After downloading, users can mark a model as a Favorite. Favorited models are automatically loaded when GiMATE Coder starts. When marking a model, the system checks available resources. If the simultaneous load limit is exceeded, a warning message will appear.
   c. **Model Deletion:** Users can delete unused models directly from the list (see Figure 3.3).
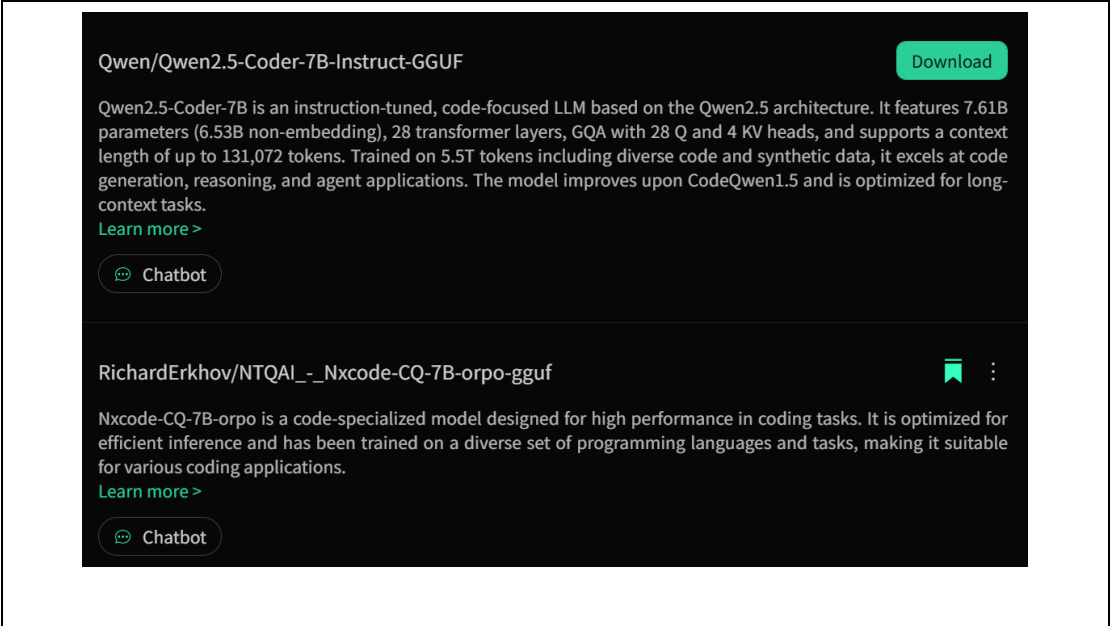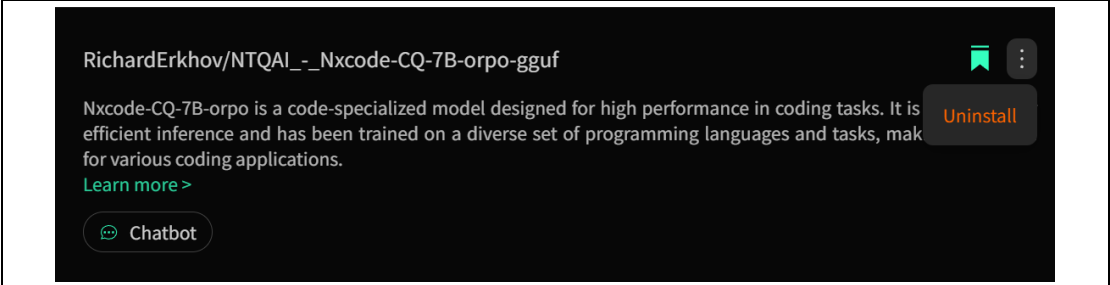
Figure 3.2 Model Download and Favorite Model

Figure 3.3 Delete Model

3.3.2 Resource Manager

Figure 3.4 Resource Manager

Figure 3.5 Uninstalling an Active Model
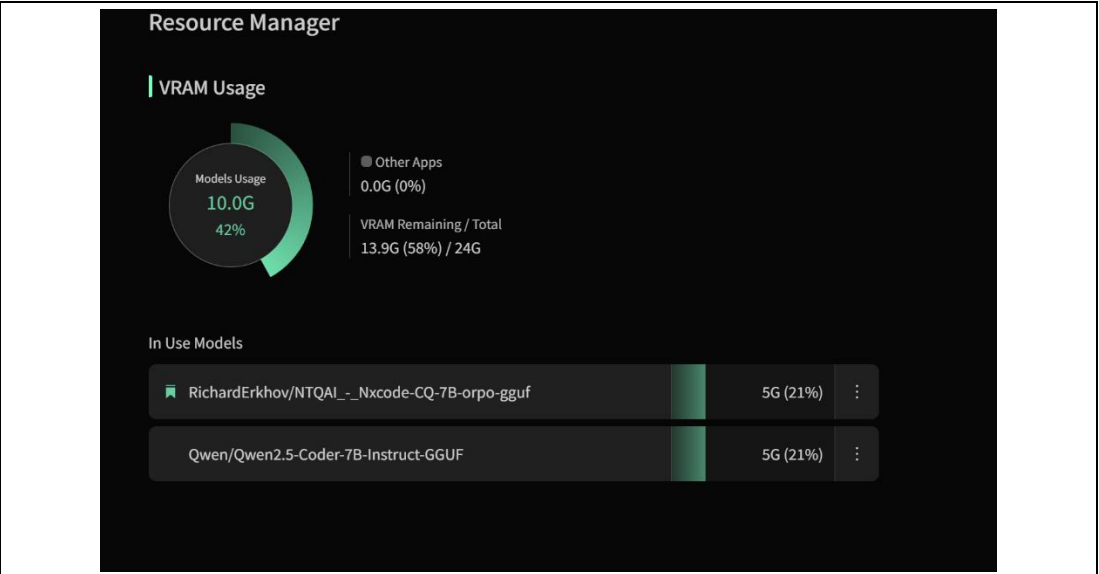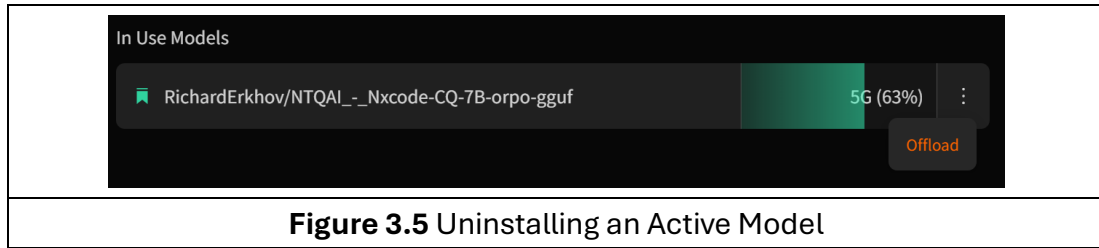
This page uses a dashboard to display GiMATE Coder's real-time VRAM usage, as shown in Figure 3.4.

- **In Use Models List:** Displays all models currently loaded into VRAM along with their usage.

- **VRAM Release:** Allows users to unload models that are no longer needed from VRAM to free up memory space (see Figure 3.5).

4. Main Features of GiMATE Coder

4.1 Model Download and Management

GiMATE Coder provides users with quick model download and deletion functions, as described in Section 3.3.1. It also offers tools for managing running models, as described in Section 3.3.2.

4.2 Code Editing Assistant

GiMATE Coder mainly covers **Ten core** functions:

NO.	Function Name	Function Name
1	Code Generation	Generates code based on natural language instructions.
2	Interactive Chat-Based Development	Provides code-related Q&A, error analysis, and optimization suggestions.
3	Code Correction	Automatically fixes errors, and provides security checks and testing suggestions.
4	IDE/Editor Integration	Deep integration with VS Code.
5	Documentation and Comment Generation	Automatically generates function comments, code summaries, and API documentation.
6	Code Completion	Automatically completes single lines, multiple lines, or entire functions.
7	Code Refactoring	Optimizes function structures, improves variable naming, and removes unnecessary code.
8	Multi-Model Support & Local Execution	Supports multiple LLM models and offline execution.
9	Intelligent Search & Project Understanding	Provides full project code search, logic analysis, and cross-file context understanding.
10	Flexible Coding Style	Allows users to define their preferred coding style.

4.3 VS Code Extension

GiMATE Coder provides integration with VS Code through the **Continue** extension. This allows users to access GiMATE Coder's chat functionality directly within the editor, enabling intelligent coding assistance in a single environment. In addition, the deep integration between GiMATE Coder and VS Code makes the auto-completion feature easy to use.

For more Information, check **Continue**.

5. Security and Privacy

a. GiMATE Coder adopts a fully local computing architecture. Except for model downloads, which require an internet connection, all computations are performed in the user's local environment. This design ensures that data processing does not rely on external servers, effectively reducing the risk of data leakage and providing a higher level of privacy and security.

b. When VS Code is detected, GiMATE Coder will automatically install the Continue extension to achieve deep integration between GiMATE Coder and VS Code.

6. Troubleshooting

Q: How do I use GiMATE Coder in VS Code?

A: Follow the instructions in Section 3.2 to start the service and open VS Code.

Q: How can I view detailed information about a model in the model list?

A: Go to the *Model Installation* page, where you can find a brief description of each model. For more detailed information, click **Learn More** and Coder will direct you to the model's official website.

Q: How do I switch between different models in Coder?

A: In the chat interface, you will find a **Model Dropdown Menu** listing all downloaded models. Simply select the model you want to use, and the system will switch automatically.

Q: Can I use Coder and the VS Code extension without an internet connection?

A: Yes. You can use GiMATE Coder and related model services offline, since all services run locally on your computer.

Q: How do I download a model through GiMATE Coder?

A: Go to the *Settings* page and select the *Model Installation* section. There, you will be prompted to enter your Hugging Face Token. Once entered, click **Download** to obtain the desired model.

7. Third-Party Licenses and Copyrights

Please follow the path below to check the details in GiMATE Coder App:

GiMATE Coder → Settings → Support → About → License Information

Click License Information to view the related details.

8. GiMATE Coder Usage Restrictions

1. General Usage Restrictions

i. This software only supports NVIDIA GPUs and requires at least 8GB of VRAM to operate properly.

2. Model Download Restrictions

i. The model list only includes models certified by GIGABYTE. Not all Hugging Face models are available.

ii. User-added or imported models are not supported.

iii. Users must register their own Hugging Face Token and agree to the specific model's terms of use individually before downloading GIGABYTE-certified models.

3. Model Usage Restrictions

i. Input Limit: The maximum input per request is 16K tokens (approximately 6,400 English words).

ii. When using the Auto-Completion feature, user code content will be automatically sent to the locally loaded model for inference.

iii. To ensure full functionality and smooth operation, GiMATE Coder will automatically install and enable the Continue VS Code Extension.

iv. The models integrated into this application are provided by third parties and are not developed or maintained by the company. All responses are for reference only and may be limited by the model's training data and algorithm characteristics. Accuracy, completeness, or timeliness cannot be guaranteed. Users are responsible for verifying information and assume all risks for actions

or decisions based on model outputs. The company is not liable for any direct or indirect losses arising from the use of model responses.

v. GiMATE Coder is an inference platform, and the models are neither trained nor fine-tuned by GIGABYTE. All outputs generated by the models are the result of automated processes by the AI system. This platform makes no guarantees regarding the accuracy, completeness, legality, or applicability of such outputs.

vi.Users are solely responsible for evaluating and bearing the risks associated with generated content, including but not limited to: legal compliance (such as data protection, privacy, and consumer protection), copyright and intellectual property, ethics, discrimination, or bias issues.

vii. If users apply generated content in commercial, educational, medical, financial, or other high-risk fields, they must ensure compliance with applicable laws, regulations, and industry standards. The platform bears no responsibility.

4. VS Code Extension Restrictions
The Continue extension is a third-party VS Code add-on. If users modify the extension in a way that causes integration failures or functional errors with GiMATE Coder, all related consequences are the user's sole responsibility.