9 May 2025

OSRF Policy on the Use of Generative Tools (“Generative AI”) in Contributions

Introduction

Tools that use generative technology, also known as “Generative AI”, “GenAI”, and “GAI”, have exploded in both capability and popularity. Their use by contributors to open-source projects is not just inevitable, but already occurring, and it is generally accepted that the use of such tools in software development, document writing, website creation, and other related tasks of open-source development will become commonplace.

This document provides the official policy as well as general guidance from the Open Source Robotics Foundation (OSRF) on the use of generative tools to produce any contribution to an OSRF project, both in whole and in part.

Definitions

For the purposes of this document, “Generative Tools” refers to tools that are used by a human contributor to automatically create some or all of the content of a contribution, where the tool’s generation method relies on “training” of a “model” using existing content previously created by one or more humans.

For the avoidance of doubt, “Generative Tools” does not refer to tools that automatically create content using well-known algorithmic methods, such as procedural texture generators, and tools that rely on (appropriately licensed) libraries of content, such as model-to-code generators.

Policy

The OSRF permits the use of generative tools in producing contributions to its projects, with some qualifications:

- Any contribution may consist, in whole or in part, of the output of one or more generative tools.
- Any use of generative tools in a contribution must be disclosed at the time of making the contribution.
- The disclosure must be recorded in a way that ensures it has the same or greater lifetime as the contribution itself.

Below we provide additional guidance that all contributors must take into account when deciding whether to use generative tools in the production of their contributions.

9 May 2025

Guidance - Responsibilities of Contributors

Understand the tools

Each Generative Tool has its own distinct capabilities and limitations. Aside from the category of contribution (source code, graphical, etc.), one tool may be more appropriate for the contribution you are making than other tools.

It is important to understand the capabilities and limitations of the tools you are using and be aware of what to look for when vetting and verifying the output. This understanding will also aid you in making the most of the tool without jeopardizing the acceptability of your contribution.

Understand how the tools were trained

Knowing what data, or at least what type of data, a tool was trained on will help you understand if it is going to produce suitable and usable output. It will also help you to understand where to stop using the tool and apply your own creative ability in order to produce the best and most appropriate contribution you can in a reasonable time. For example, if a source code generating tool was primarily trained on Python and Rust, with a much smaller number of examples of C++, it is unlikely to produce high quality C++ code.

Understand the relevant copyright law landscape

You should understand what it means to own copyright over all or part of a contribution, and what rights that gives you with respect to making the contribution to an open source project.

You should also be aware that, at the time of writing, **copyright cannot be held on the output of generative tools**, including by the user and by the provider of the tool, but also that this does not prevent you holding copyright on the parts of a contribution you created yourself.

Review and understand terms of use, privacy policy, and other documents

Read and understand the terms of use of the tools you are using. They may place limits on your use of the output of the tools. It is important to ensure that the terms of use do not conflict with your intended use, especially when it involves contributing that output to an open source project.

You should also understand the privacy policy of the tool's provider, as well as any other documents relevant to your use of the output of the generative tools.

Understand what your inputs are used for

Generative tools are trained on an enormous quantity of human-produced creative content. In some tools, this may include the inputs from users of those tools to prompt a desired output. For example, a source code generating tool may use all or part of the source file currently being edited as input to generate the remainder of a line, function, class, or similar.

Be judicious about what input you provide to any generative tool, and understand what input it may take without obviously asking the user. This particularly applies when working with data that may be confidential, proprietary, or have privacy implications. Examples include a source file not intended for release outside of your employer, or the personally identifying information on a group of people.

Make use of available tools for vetting output

There is a growing market of tools for checking the output of generative tools for potential intellectual property problems, including plagiarism, exact duplication of part of the training data, and other copyright infringement. Other tools can detect clearly poor-quality code or prose, or common security flaws.

Some generative tool providers may provide such tools themselves. Other providers of these tools may provide them as add-ons to generative tools from other companies. Be aware of what tools are available and make use of them to reduce the risk of making a contribution that infringes someone's copyright or otherwise degrades the project you are contributing to.

Verify accuracy and appropriateness of output

Do not assume content created by generative tools is perfect. In fact it deserves greater scrutiny than that produced solely by humans due to the well-known problem in generative tools of "hallucination". This can cause the tool to produce flawed output with the appearance of accuracy.

As the contributor, it is your responsibility to ensure that the whole of the contribution is accurate to the intended use within the project, including parts generated by tools. Some actions you may need to take include (but are not limited to):

- Perform all the normal verification processes that you would for any contribution.
- Perform thorough code reviews, ensure the contribution is fully tested, with all tests (including pre-existing tests) passing.
- Perform an appropriate security audit.
- Proof-read comments and documentation.
- Verify the readability and understandability of diagrams and figures.
- Check for intellectual property problems.

Be transparent

As with any open-source contribution, the safe and effective use of generative tools relies on all contributors being transparent about how they produced their contributions. Clearly state your sources for your contribution whenever possible. Track and disclose the use of all generative tools in all parts of your contribution, and ensure that their use is clearly documented for the people who will review your contribution and for future traceability - especially important should problems (legal or otherwise) with those tools be identified after your contribution has been accepted. Follow the guidance in this document for how to indicate when your contribution uses generative tools.

9 May 2025

Use common disclosure statement formats

To support automated methods of tracing and locating contributions that use generative tools, all contributors have a responsibility to use common formatting of their disclosure statements.

The method of disclosure depends on the nature of the contribution being made.

Source code

For source code contributions, you should add a disclosure statement in the commit message for all commits where some portion of the source code was generated. This statement must list the tools used. For example:

```
Fix a memory leak when deleting nodes early.

Generated-by: GitHub Copilot v3.2; Amazon CodeWhisperer
2024/10 release.
```

Note the fully-qualified name of the tools, including the provider and version/release information. Note also that the disclosure statement is after the first line of the commit message, as it does not need to be constantly visible in commit logs, just capable of being found when necessary. Including the disclosure in the commit message is considered the most robust method of disclosure as it is permanently stored in the repository and so will live as long as, or longer than, the generated code itself.

You should also include the same statement in the pull request description for the pull request containing those commits. Although considered inferior in strength to the commit message statements, the pull request description statement has the benefit of greater visibility to reviewers. The pull request description, including the disclosure statement, may look like the following example.

> This pull request adds the capability to do amazing things to nodes. Closes #42.
>
> Some portions of this pull request were generated using GitHub Copilot v3.2 and Amazon CodeWhisperer 2024/10 release.

You may also add a comment in each source code file that includes generated tool output. However this is not mandatory and is considered inferior to including the disclosure statement in the commit message and the pull request description, as it has relatively weak traceability over time. For example:

```
// Generated-by: GitHub Copilot v3.2; Amazon CodeWhisperer
2024/10 release
```

Documentation

For documentation that is stored in a Git repository, you should add the disclosure statement to the relevant commits and the pull request description, as with source code contributions. If the documentation format supports comments, you may also add the disclosure statement as a comment in each relevant documentation file.

For documentation not stored in a Git repository, the disclosure statement should be included in the meta-data of the document in some way, such as using a review comment in Microsoft Word or Google Docs, or adding a comment to the document's description.

Translations of documentation and of strings included in source code

Follow the same approach for including the disclosure statement as for source code and documentation.

Graphical works

Each graphical work must include a copyright statement and license assignment when contributed. If a generative tool was used to create all or part of the graphical work, the disclosure must be included in that statement.

Keep up to date with this guidance

The field of generative tools is rapidly progressing and both the capabilities and legal aspects of them are likewise evolving rapidly. New laws may be produced in one nation or another, case law may be created in one of the many on-going legal battles around the nature of these tools, the wider open-source community may alter its stance on the tools, and the tools themselves may become capable of producing output that only meets user-specified licensing.

Changes such as these will require revisions to this guidance. Although the OSRF will endeavor to keep this guidance up to date with respect to the state of the art, and announce any changes publicly, it is your responsibility as a committer to keep up to date with the guidance and policy in this document and follow the most recent version available when you make a contribution.

9 May 2025

Frequently Asked Questions

Why do I need to disclose the use of a Generative Tool in a contribution?

The legality of the output of generative tools, with particular regard to copyright law but also other aspects such as the ability to be licensed, is not yet known. Rather than outright deny all contributions that include the output of generative tools, we prefer to allow them, but make it possible to find these contributions at a later date. Should a particular tool's output be found at a later date to be legally problematic, the use of disclosure allows all relevant contributions to be found and evaluated. Without disclosure, the finding of one tool to be legally problematic could put the entire code base under suspicion.

How can I disclose the use of a Generative Tool in a contribution?

Please see the guidance for examples of disclosure methods for different types of contributions.

What sort of contributions does this guidance apply to?

This guidance applies to any and all contributions made to OSRF projects, including those administered by the OSRA under an open-source model. This includes source code, documentation, graphical user interface designs, website layouts, translations, and any other content that may be created through the use of one or more Generative Tools.

What sorts of generated outputs are suitable for inclusion in a contribution?

In general, any generated output that is usable as part of a contribution, in the way that contribution is intended to be made and intended to be used within the target project, is suitable. However, this is limited by the need to comply with copyright and intellectual property laws, and with the license of the target project. Any generated content to be included in a contribution in whole or part must meet at least one of the following requirements.

1. The output is not copyrightable subject matter, even if produced by a human.
2. The output does not include any third party materials.
3. The output does include third party materials but those materials are used with permission (e.g. under a compatible open-source license) of the third party copyright holders and in compliance with the relevant license terms.

Examples of "third party materials" include:

- verbatim copies of source material,
- large code snippets that are substantially similar to existing copyrighted code,
- text passages that are copied verbatim from copyrighted sources, and
- images or other media that are derived from copyrighted works without permission.

It is your responsibility as the contributor to ensure that one or more of the above is met through any suitable means, including but not limited to awareness and documentation of the copyright status of all training data used in creating the tool, and running the output through suitable tools for detecting intellectual property problems.

How do I know if a Generative Tool was trained on copyrighted content?

It may not be possible to know what content was used to train a Generative Tool, due to the proprietary nature of many of the available tools and the highly competitive market for these tools. When using outputs of tools where the content and copyright status of the training data is unknown, make use of tools for identifying plagiarism in the output and identifying when output matches training data, and follow the guidance in this document to correctly disclose the use of all Generative Tools involved.

What should I do if I made a contribution using a Generative Tool that was accepted, and then later found to have included copyrighted material in the generated portions?

The same process should be followed in these situations as for any other contribution where plagiarism or other copyright infringement has been identified.

Does the OSRF have a list of approved or recommended Generative Tools?

The OSRF is not able to evaluate all available Generative Tools, nor is it in the OSRF's interests to limit or otherwise tell contributors what tools they should use to produce their contributions. You may use any tools you see fit, provided that the outputs of those tools and how you contribute those outputs follow the guidance in this document and respect copyright.