2026 NIST GenAI Watermark Evaluation Plan for Image Watermark Embedders

Released: 2026-30-10

Updated: 2026-30-10

GenAI Eval Team

National Institute of Standards and Technology
100 Bureau Dr, Gaithersburg, MD 20899

Contact: genai-poc@nist.gov

Disclaimers

Certain commercial equipment, instruments, software, or materials are identified in this document to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by NIST, nor necessarily the best available for the purpose. The descriptions and views contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of NIST or the U.S. Government.

Updates

2026-10-30 Initial Release

Table of Contents

1 Introduction....4
2 Tasks....5
2.1 Protocol and Rules....6
2.1.1 Rules....6
2.1.2 Image Watermarking Protocol....6
3 Data Resources....7
4 Submission Instructions....7
4.1 Watermark Embedding Instructions....7
4.2 Results Submission Guidelines....9
4.3 Submission Validation....12
4.4 Containerized Submission....13
4.5 System Decoding Instructions....14
4.6 Participation Agreement....14
4.7 System Descriptions....15
5 Evaluation Under Natural Transformations....15
6 Performance Metrics for Watermarking Outputs....16
6.1 Perceptual Similarity....16
6.2 Generation Diversity (Prompt-Based Methods Only)....16
6.3 Robustness Under Natural Transformations....17
Appendix A Metrics Definitions and Purpose....17

1 Introduction

In recent years, digital content from generative Super Intelligence (SI)[1], including deepfakes, has seen unprecedented growth and proliferation across multiple modalities, including image, video, audio, and text. While these advances enable powerful new forms of content creation, they also introduce significant challenges related to authenticity, provenance, and trust. As SI-generated content is becoming ever more indistinguishable from human-created media, there is a growing need for robust mechanisms to allow content consumers to understand the origins of what they are consuming. Digital watermarking has emerged as a key approach to address these challenges by embedding information within media that can support provenance, attribution, and tamper detection, even under potential adversarial conditions.

As part of the NIST Generative SI (GenAI) program, we invite participation from academia, industry, and research laboratories to advance the state of the art in watermarking technologies. This evaluation provides a platform to systematically assess the performance of both watermark embedding systems and adversarial techniques designed to manipulate or remove such watermarks. The evaluation framework reflects two complementary roles: watermark embedding teams and adversary teams, representing opposing objectives within a shared evaluation environment.

Watermark Embedding (W) teams will be evaluated on their ability to insert robust, secure, and imperceptible watermarks into digital content. These teams should aim to maintain watermark integrity under a range of transformations and potential adversarial attacks while preserving the perceptual quality and utility of the content.

Adversary (A) teams will be evaluated on their ability to detect, remove, or manipulate embedded watermarks without significantly degrading the perceptual quality of the content. Their goal is to challenge the robustness and security of watermarking approaches by simulating adversarial conditions under which watermark integrity may be compromised, or under which false watermarks could be introduced.

Teams are required to indicate whether they are participating as a watermark embedding team, an adversary team, or both. Insights gained from both approaches are expected to support standards development and future research directions for trustworthy and responsible use of generative SI technologies by revealing the strengths and limitations of current techniques.

The upcoming GenAI watermarking evaluation will initially focus on the image modality. The objective of the watermark embedding task is to automatically embed watermarks into images while satisfying requirements on robustness and perceptual quality. The adversarial task focuses on applying transformations or attacks on watermarked images to remove or manipulate the watermarks. This will help with the evaluation of the resilience of the embedded signals and the difficulty of forging them. Evaluation scenarios will reflect realistic usage conditions and adversarial settings. Additional details on task specifications, datasets, and evaluation protocols will be provided in subsequent sections of this document.

This document describes the task specification for watermark embedding teams. Evaluation materials and conditions will be released across multiple rounds by the NIST GenAI team. Any questions or comments concerning the GenAI evaluation series should be sent to genai-poc@nist.gov

[1] Per the Sept. 29, 2026, Executive Order 14434: Inaugurating the Era of Super Intelligence, NIST is working to update its communications to incorporate the term "super intelligence" as directed.

2 Tasks

The primary goal of this watermarking evaluation is to benchmark and measure the robustness of imperceptible watermarks while assessing their impact on content quality. The evaluation campaign is structured in two phases:

- **Phase 1: W-Teams (Watermark Embedding):**
  Participating teams will embed watermarks into a provided dataset of images and/or generate watermarked images from given text and/or image prompts. The objective is to produce watermarked content that maintains high perceptual similarity to the original image while ensuring that the embedded watermark is robust to subsequent natural transformations, corruptions, and adversarial attacks.
- **Phase 2: A-Teams (Adversary):**
  Participating teams will attempt to compromise or forge the watermarks embedded by W-teams. This includes removing the watermarks from a dataset of watermarked images, altering or corrupting any payloads embedded in such images, or forging a W-team's watermark on an unwatermarked image. Phase 2 will be conducted over three rounds, with each round targeting a specific use-case scenario designed to evaluate different aspects of watermark robustness under realistic and adversarial conditions.

The rest of this document discusses the evaluation plan and specification for Phase 1. The evaluation plan for Phase 2 will be provided separately.

Phase 1: Watermark Embedding (W-Teams):
In this phase, teams will have the choice between two primary categories: (1) post-hoc watermarking on provided images, or (2) watermark embedding during image generation (e.g., in the latent space of a diffusion model) using provided text or image prompts (reference images). Teams may join both categories if they wish.

NIST will provide: (a) a testing dataset of images, (b) a set of text prompts, and (c) a specified payload for each sample (either zero-bit or multi-bit). Teams will embed their watermarks into images according to the task specifications, including embedding the specified payload. However, if a watermarking method has a maximum payload capacity lower than the specified amount (e.g., 64 bits instead of 256), teams should research ways (be innovative) to fit the required payload using their system (e.g. coding or compression techniques).

For **post-hoc watermarking**, teams will generate a modified version of each provided image that embeds a watermark and (if applicable) the payload while preserving perceptual quality.

For **generation-based watermarking**, teams are required to submit results for either of the following approaches. Should a team want to attempt both, they must create two separate submissions:

(a) **Text-prompt-based generation:** For each provided text prompt, generate a maximum of N new images with the watermark embedded during the generation process, including the assigned payload if applicable. Additionally, for each watermarked image generated, generate a corresponding unwatermarked image.

(b) **Image-prompt-based generation:** For each provided input image, regenerate a maximum of N new images that are identical or near-identical to the input image, while embedding the watermark during the generation process, again including the assigned payload if applicable. The fidelity of the resulting watermarked images will be assessed using perceptual metrics such as PSNR and SSIM.

The provided evaluation data will include a diverse set of images, text prompts, and payload configurations designed to cover a broad range of visual characteristics and testing conditions that may impact the robustness, fidelity, and overall performance of watermarking algorithms. In Phase 1, the NIST GenAI team will evaluate watermark imperceptibility, generation diversity for prompt-based generation systems, and watermark robustness under a range of natural corruptions and non-adversarial transformations applied to watermarked images. Robustness evaluation will include watermark detectability as well as payload extraction and integrity, where applicable. The table below provides an overview of the number of images, prompts, and payloads included in the evaluation data.

Number of topics	Number of Images	Number of Text-Prompts	Number of Text-Prompts targeting diversity Prompts	Payload Options
Post-Hoc 1800 topics	600			<0, 64, 256>
In-Generation (Text-Prompt) - 912 topics		300	12	<0, 64, 256>
In-Generation (Image-Prompt) 1800 topics	600			<0, 64, 256>

2.1 Protocol and Rules

2.1.1 Rules

Please refer to the rules section of the GenAI program at https://ai-challenges.nist.gov/rules for more information about participation eligibility, conduct, handling data and materials, processing of runs and submissions, and public statements about your participation.

2.1.2 Image Watermarking Protocol

Throughout this document, the terms **"submission"** and **"run"** are used interchangeably to mean submitting results for all required testing dataset inputs. Each submission must include outputs (e.g., watermarked images with any applicable payload) for ALL input samples (images or text prompts). Therefore, a submission should not miss or ignore an input sample.

Teams are allowed to submit multiple (up to 5 for post-hoc and image/text-prompt, and 50 for diversity topics) watermarked images per input sample. This allows teams to explore variations in their watermarking approach, including different parameter settings and control strategies, while embedding the watermark. The only exception to this is a subset of provided topics applied to generation-based **text-prompt** methods, as their output images will be used to measure the diversity of generation. Therefore, these methods should NOT

change any settings, but keep all parameters fixed (See section 6.2 Generation Diversity (Prompt-Based Methods Only) for more information).

Each submission must only contain 1 run through the data. A team will be allowed to submit multiple submissions as per the scoreboard submission guidelines. All submitted images must:

- Be in webP format.
- Have the appropriate resolution. For generation-based methods, watermarked images should have the **maximum** supported resolution that the model can generate. For post-hoc methods, image resolution should stay the same as the provided image.
- Leave their EXIF metadata information unchanged.

Teams must provide a watermark detection and decoding component (see Containerized Submission section for detailed instructions) corresponding to their submitted watermarking method. This component will be used by NIST to evaluate watermark presence and, where applicable, to extract the embedded payload from submitted images after the application of natural corruptions and transformations.

The provided detection/decoding component must be self-contained, fully functional, and use a standardized entry-point for automatic execution; otherwise, the specific entry-point command must be provided as a parameter during submission. It must operate without any additional training, fine-tuning, or external data access at evaluation time. The interface and input/output specifications must be consistent across all submitted samples. Failure to provide a functional detection/decoding component will result in the exclusion of the submission from robustness evaluation.

For more details, please check the specifications and requirements of this detection/decoding component and how teams will make it available to NIST. After all evaluation phases are concluded, NIST will publish a report summarizing system performance with anonymized team names. Teams may publish or otherwise disseminate these results unaltered and with appropriate reference to the original NIST report results.

3 Data Resources

NIST will make all necessary data resources available to watermark embedding teams. Each team will receive access to data resources upon completion of the required data agreement forms and based on the published release dates for each task. Refer to the published schedule for data release dates.

For generation-based watermarking methods, we plan to evaluate systems using **two fixed image generation models**: Stable Diffusion XL and FLUX.2 Dev. Using the same set of generation models across all submissions will help isolate the effect of the watermarking algorithms themselves. This allows the evaluation to measure how each watermarking method affects robustness and other performance metrics, while minimizing variation caused by differences among the underlying image generation models. These options were selected to accommodate watermark paradigms that are dependent on the generator's architecture.

4 Submission Instructions

4.1 Watermark Embedding Instructions

A set of topics encapsulating input samples to cover a diverse set of attributes and categories will be released.

Each input sample is defined as a <topic> element with the following required fields:

- <num>: Unique identifier for the topic.
- <num_images_min>: Minimum number of images to be watermarked
- <num_images_max>: Maximum number of images to be watermarked
- <eval_diversity>: T (true) or F (false) if this topic will be used/not used in generation diversity.
- <prompt>: Text prompt describing the desired image content (empty for post-hoc topics).
- <prompt_id>: The id of the textual prompt as found in the file prompts.json
- <image>: Filename of the reference image (usage takes the value of "**post-hoc"** or "**image-based-generation"**).
- <payload>: Watermark payload specification. Specifies the type and length in bits **(0, 64, 256).**

NOTE: For diversity evaluation, **a selected subset** of topics will require a high number of generated images (~ 50). For these particular topics, teams will be asked to submit for each prompt a set of watermarked images as well as unwatermarked images.

All topics will be aggregated in one master XML file and released to watermark embedding teams. Below are a few examples of topics demonstrating with (multi-bit) and without payload (zero-bit).

Example of topic (text prompt-based generation, diversity-focused, with multi-bit payload):

```
<topic>
  <num>topic_5445</num>
  <num_images_min>50</num_images_min>
  <num_images_max>50</num_images_max>
  <eval_diversity>T</eval_diversity>
  <prompt usage="text-based-generation">A young man with glasses looking upward to the sky</prompt>
  <prompt_id>1</prompt_id>
  <image usage="image-based-generation"></image>
  <payload type="multi-bit" length="32">10100010101011111000100001110101</payload>
</topic>
```

Example of topic (post-hoc with multi-bit payload):

```
<topic>
  <num>topic_134</num>
  <num_images_min>1</num_images_min>
  <num_images_max>5</num_images_max>
  <eval_diversity>F</eval_diversity>
  <prompt usage="text-based-generation"></prompt>
  <prompt_id></prompt_id>
  <image usage="post-hoc">37.jpg</image>
  <payload type="multi-bit"
length="64">0000001010101111100010000111010100000010101011111000100001110101
</payload>
</topic>
```

Example of topic (text prompt-based generation with zero-bit payload):

```
<topic>
  <num>topic_5446</num>
  <num_images_min>50</num_images_min>
  <num_images_max>50</num_images_max>
  <eval_diversity>T</eval_diversity>
  <prompt usage="text-based-generation">A young man with glasses looking upward to the sky</prompt>
  <prompt_id>40</prompt_id>
  <image usage="image-based-generation"></image>
  <payload type="zero-bit" length="0"></payload>
</topic>
```

Example of topic (image prompt-based generation with zero-bit payload):

<topic>
<num>topic_421</num>
<num_images_min>1</num_images_min>
<num_images_max>5</num_images_max>
<eval_diversity>F</eval_diversity>
<prompt usage="image-based-generation">Using the provided input image as the visual reference, generate a new image that is identical to the input image. The goal is faithful image regeneration: same subjects, same scene, same object placement, same pose, same viewpoint, same scale, same lighting, same colors, same background, and same overall style. Do not introduce new objects, change the composition, modify identities, alter expressions, change clothing, rewrite or distort visible text, or make the image more artistic than the original. The generated image should be identical or near-identical to the input image, with only unavoidable minor generation differences.
</prompt>
<prompt_id>0</prompt_id>
<image usage="image-based-generation">356.webp</image>
<payload type="zero-bit" length="0"></payload>
</topic>

Topics will be distributed by NIST. Only GenAI watermark embedding teams who have completed and submitted all required data agreement forms will be allowed access. Check the published schedule online for testing data release dates.

4.2 Results Submission Guidelines

- Each team may submit up to 10 runs per evaluation round. Each run should include no more than the specified maximum number of images per topic, as specified in each topic xml. A submission run should be in the format of a single compressed file (i.e., Docker image) to include the run results xml file (see below for specifications) and all images as specified in Containerized Submission section. **Multiple zip files are not allowed per run**.

- Each run submission should be based on a single watermarking method (e.g., if you are applying the Tree-Ring method, then all images in a single run should be submitted using Tree-Ring). Within the fixed watermarking method, teams can experiment with various settings and hyperparameters in their watermarked images, provided these adjustments do not fundamentally alter the underlying algorithm.

- Any submitted run should be either based on a post-hoc watermarking method, given the post-hoc topics file and input images, or a generation-based watermarking method, given the generation-based topics file. This means that a run cannot mix post-hoc and generation-based methods.

- Each run submission should include a containerized implementation of the watermarking decoding/detecting software to be used by NIST for evaluation (see next sections for specific details).

- All submitted images must follow the naming convention below to be compatible with the evaluation pipeline: `**topic_ID.imageID.version.status.extension**`

    - For teams generating watermarked images from the provided dataset (i.e., post-hoc teams):

        - The imageID is the original image filename, preceded by the word 'image' (e.g., if the xml specifies <image>20.jpg</image>, your imageID is image20).

    - For teams generating both watermarked and unwatermarked images (i.e., generation teams):

- The imageID is the prompt identifier from the XML, preceded by the word 'image' (e.g., if the xml specifies <prompt_id>7</prompt_id>, your imageID is image7).

Component	Description	Example
topic_ID	The topic identifier from the xml file.	<num>topic_2000</num> → topic_2000
imageID	See instructions above.	20.jpg → image20
Version	The version number. Use 1 for a single generation, or 2, 3... for multiple versions.	1
Status	Use either watermarked or unwatermarked.	watermarked or unwatermarked
Extention	The image file format.	.webp

- Each run should contain images for all topics; a run can not skip a topic or submit images for a subset of the topics.

- All images must be submitted as webp files, and follow the guidelines for aspect ratios and/or resolution allowed (see guidelines above for allowed values). EXIF metadata must be preserved in the images.

- For both generation methods (text-based and image-based), all image content submitted should be free from offensive visuals or inappropriate remarks. Prohibited images include, but are not limited to, any image with explicit nudity, sexual content, hate symbols, child exploitation, or non-consensual images. NIST has the right to exclude any image or whole runs if it determines the content to be inappropriate for the general public. The GenAI team will have a process to automatically detect image content, and any flagged images will be reviewed by a team of NIST personnel for final recommendation.

- Each run should include high-level metadata to characterize the watermarking embedder system as requested by the below run format and DTD file. As explained in the DTD file, teams need to provide some required information/parameters, such as:

    - teamName: The name of the team as registered on the NIST GenAI website.

    - trainingData: Name of training dataset or collection of different datasets or source data.

    - priority: The priority of the submitted run (the lower the number, the higher the priority). For any required manual review of submissions, NIST may need to limit its effort to only the highest-priority runs. Priority should be an integer with a range between 1 and 10.

    - trained: A boolean (T or F) to indicate if the run was the output of a trained system by the team specifically for this task (T) or the output of an already existing system that the team used to generate the outputs (F).

    - desc: Name of the base model/technique used by the system.

    - link: A link to the model or technique used to generate the run (e.g., GitHub, etc).

- model: The model name used (for generation-based approaches, it has to match the model(s) provided by NIST). For post-hoc approaches, it can be the name of the method/technique applied.

- topic: The topic ID (the "**num**" field in the topic .xml file).

- elapsedTimeSec: The processing time per topic to generate (watermark) the image after the topic was presented to it.

- approach: Should specify the approach of the watermarking system and take one of these values: "**text-based-generation**", "**image-based-generation**", or "**post-hoc.**"

- eval_diversity: T(true) or F(false) if this topic will be used in diversity measurement (as set in the topic .xml file).

- prompt: The prompt used to generate the watermarked image. For post-hoc and image-based generation topics, this should be an empty string. For text-based generation topics, this should be a string that contains the full prompt.

- payloadType: Must conform with the topic provided (either multi–bit or zero-bit).

- comment: Any relevant information that played a role in watermarking this particular image (can be useful for tracking image-level metrics).

- payloadLength: Must be 0 for zero-bit or exactly as provided in the topic for multi-bit.

- payloadValue: Must be 0 for zero-bit, or exactly as provided in the topic for multi-bit

- filename: The name of the **watermarked** image file. Generally, images should be named as follows: topic_1.image1.1.watermarked.webp, topic_1.image1.2.watermarked.webp, etc. See the instructions above for the naming instructions.

- filename_unwatermarked: The name of the **unwatermarked** image file. Generated images should be named as follows: topic_1.image1.1.unwatermarked.webp, topic_1.image1.2.unwatermarked.webp, etc. Otherwise, use the original image name from the dataset. See the instructions above for the naming instructions.

Example of a sample run:

```
<!DOCTYPE EmbedderResults SYSTEM "EmbedderResult.dtd">
<EmbedderResults teamName="participant_1">
<EmbedderRunResult trainingData="OpenAI" priority="1" trained="T" desc="Top secret x-component" link="TBD"
model="FLUX.2 Dev">
   <EmbedderTopicResult topic="topic_1" elapsedTimeSec="5" approach="text-based-generation" eval_diversity="T"
    prompt="This is an example of the full prompt">
     <Image filename="topic_1.image1.1.watermarked.webp" approach="text-based-generation" payloadType="multi-bit"
       comment=""  payloadLength="64"
payloadValue="1010001010101111100010000111010110100010101011111000100001110101"
        filename_unwatermarked="topic_1.image1.1.unwatermarked.webp"/>
     <Image filename="topic_1.image1.2.watermarked.webp" approach="text-based-generation" payloadType="multi-bit"
       comment="" payloadLength="64"
payloadValue="1010001010101111100010000111010110100010101011111000100001110101"
       filename_unwatermarked="topic_1.image1.2.unwatermarked.webp"/>
     <Image filename="topic_1.image1.3.watermarked.webp" approach="text-based-generation" payloadType="multi-bit"
```

```
        comment="" payloadLength="64"
payloadValue="10100010101011111000100001110101101000101010111110001000011101 01"
        filename_unwatermarked="topic_1.image1.3.unwatermarked.webp"/>
      <!-- more images -->
      <!-- more images -->
    </EmbedderTopicResult>

    <EmbedderTopicResult topic="topic_2" elapsedTimeSec="5" approach="text-based-generation" eval_diversity="F"
     prompt="This is an example of the full prompt">
      <Image filename="topic_2.image2.1.watermarked.webp" approach="text-based-generation" payloadType="zero-bit"
        comment="" payloadLength="0" payloadValue="0"
        filename_unwatermarked="topic_2.image2.1.unwatermarked.webp"/>
      <Image filename="topic_2.image2.2.watermarked.webp" approach="text-based-generation" payloadType="zero-bit"
        comment="" payloadLength="0" payloadValue="0"
        filename_unwatermarked="topic_2.image2.2.unwatermarked.webp"/>
      <Image filename="topic_2.image2.3.watermarked.webp" approach="text-based-generation" payloadType="zero-bit"
        comment="" payloadLength="0" payloadValue="0"
        filename_unwatermarked="topic_2.image2.3.unwatermarked.webp"/>
      <!-- more images -->
      <!-- more images -->
    </EmbedderTopicResult>
    <!-- ... -->

  <EmbedderTopicResult topic="topic_40" elapsedTimeSec="5" approach="text-based-generation" eval_diversity="T"
    prompt="This is an example of the full prompt">
      <Image filename="topic_40.image40.1.watermarked.webp" approach="text-based-generation" payloadType="multi-bit"
        comment="" payloadLength="64"
payloadValue="10100010101011111000100001110101101000101010111110001000011101 01"
        filename_unwatermarked="topic_40.image40.1.unwatermarked.webp"/>
      <Image filename="topic_40.image40.2.watermarked.webp" approach="text-based-generation" payloadType="multi-bit"
        comment="" payloadLength="64"
payloadValue="10100010101011111000100001110101101000101010111110001000011101 01"
        filename_unwatermarked="topic_40.image40.2.unwatermarked.webp"/>
      <Image filename="topic_40.image40.3.watermarked.webp" approach="text-based-generation" payloadType="multi-bit"
        comment="" payloadLength="64"
payloadValue="10100010101011111000100001110101101000101010111110001000011101 01"
        filename_unwatermarked="topic_40.image40.3.unwatermarked.webp"/>
      <!-- more images -->
      <!-- more images -->
    </EmbedderTopicResult>
  </EmbedderRunResult>
</EmbedderResults>
```

4.3 Submission Validation

- NIST will provide, prior to submission dates, a validator script to teams to validate their output XML file format as well as content specific to the task guidelines (e.g. topic ids, required attributes, etc). All embedder teams should validate their runs before submitting them to NIST. Example of available DTD validators (via a shell script): xmllint --valid simple_sample.xml

- **Instructions for validator:** to run the validator (validator assumes submission xml is located under the **outputs** folder), the usage is:

  usage: submission_validator.py [-h] --topics TOPICS [--format_only]
  Options:
    -h, --help show this help message and exit
    --topics TOPICS XML file of topics

--format_only Only validate the XML formatting. If omitted, also validate that all topics are included in the submission.

For example, to validate that the formatting is correct, the images are in the right places, and all topics are complete, using post-hoc topics, run:
python submission_validator.py --topics …/phase1_topics_post-hoc_methods.xml

To validate that the formatting of an XML submission is correct and the images are all in the right place, but the overall run is still being worked on, run:
python submission_validator.py --topics …/phase1_topics_post-hoc_methods.xml --format_only

- **Submission instructions:** Submissions will be open on the GenAI website, as specified in the published schedule, for teams to submit their data outputs in the specified format. Make sure to follow the schedule and submit on time, as extending the submission dates may not be possible.

- Upon submission, NIST will validate the data outputs uploaded and report any errors to the submitter.

4.4 Containerized Submission

Each team will receive a tarball `**wm_submission.tar.gz**` containing the resources needed to implement their decoding environment, including all necessary information for their watermarks, such as the text prompt and topic files. The provided workspace will resemble the following:

```
├── configs/
│   ├── pipeline_config.yml
│   ├── submission_validator.py
│   ├── xml_parser.py
│   └── xml_utils.py
├── datasets/
│   ├── watermarked/
│   └── unwatermarked/
├── inputs/
│   └── instructions.xml
├── outputs/                  # To be populated by participant
│   ├── submission.xml
│   └── results.csv
├── watermarks/
│   ├── watermark.py          # Example embedding script to be created by participant
│   ├── decode.py             # Example decoding script to be created by participant
│   └── key.pt                # Example key to be created by participant (if applicable)
└── Dockerfile
```

The **datasets/watermarked/** folder should be populated for all watermarking method types. The **datasets/unwatermarked/** folder may be empty for post-hoc and image-based generation, but must be populated for text-based generation. The watermarks/watermark.py and watermarks/decode.py files shown above are examples demonstrating how to interact with the `**watermarks**` module. Teams may adapt, replace, or restructure this module. NIST will mount output files to the scoring container. Therefore, teams are required to store all generated outputs, including submission.xml and results.csv, in the **outputs/** directory so they can be read without errors.

For the final submission, teams must **upload a single compressed folder via the web platform**. This folder must contain the following two components:

1. A datasets/ folder containing the images, organized according to the directory structure mentioned above (e.g., datasets/watermarked and/or datasets/unwatermarked).
2. The Docker image tarball, which must be named exactly `**watermark-decode.tar.gz**`.

The submission container must execute non-interactively, reading images from the mounted dataset volume as input and writing the results (e.g., results.csv) directly to the mounted output directory. More information on the expected structure of the output .csv file is provided in Section 4.5.

The decoding environment must be entirely self-contained. Ensure all dependencies, libraries, and model weights are installed during the Docker build process. Since the image will not have internet access at runtime, any models usually downloaded via Python must be pre-embedded in the image.

4.5 System Decoding Instructions

Each submitted run should be accompanied by a containerized implementation of the watermarking decoding/detecting software. NIST will run this decoding/detection software on the testing images after applying the natural transformations as discussed in Section Evaluation Under Natural Transformations.

The detection software needs to output a .csv file with the following columns:

1. Filepath as a string variable
2. Extracted payload as string variable (zero-bit teams return empty strings "")
3. Watermark detection score as a float value (between 0.0 and 1.0, where higher values indicate a greater probability that a watermark is present)
4. Payload extraction score (between 0.0 and 1.0, where higher values indicate greater probability that the extracted payload is correct). Should be an empty string for zero-bit watermark.

The decoding software should be designed to accept images as input and return a single .csv file containing the aforementioned contents and appropriate column headers, as shown in the example below. After applying the transformations, corrupted images will be transferred to the participant's containerized implementation, and NIST will use the decoding script to extract the information necessary to proceed with the evaluation. Teams are expected to provide instructions on how to run the decoding script. For consistency, name the .csv file **`results.csv`.**

Filepath	Extracted Payload	Watermark Detection Score	Payload Extraction Score
"datasets/topic_1…"	"0111000..."	0.825	0.6
"datasets/topic_2…"	"1001010..."	0.783	0.75

4.6 Participation Agreement

All watermark embedding teams submitting data outputs will be required to complete and sign a Participation Agreement before uploading their data outputs.

4.7 System Descriptions

The NIST GenAI team will request a system description for the top-performing systems (Pareto-optimal) in their submissions. Documenting a system is vital to interpreting evaluation results. Make sure you document this information while developing your system and submitting your results. A system description should include, but is not limited to, the following information:

Section 1. Submission Identifier(s)

List the submission IDs for which system outputs were submitted; the GenAI team can help identify Submission IDs as needed.

Section 2. System Description

A brief technical description of your system and the system model used.

Section 3. System Hardware Description and Runtime Computation

Describe the computing hardware setup(s) and report the number of CPU and GPU cores. A hardware setup is the aggregate of all computational components used.

Section 4. Training Data and Knowledge Sources

List the resources used for system development and runtime knowledge sources, if any.

Section 5. References

List pertinent references, if any.

5 Evaluation Under Natural Transformations

In Phase 1, the NIST GenAI team will evaluate the robustness of submitted watermarked images by applying a set of **natural transformations** representative of common real-world processing. These transformations are not designed to be adversarial; rather, they reflect typical operations that images undergo during storage, sharing, editing, and display.

Each submitted watermarked image will be subjected to one or more transformations, including different strength levels, drawn from standardized categories, which may include (but are not limited to):

- **Compression and re-encoding** (e.g., JPEG/WebP at varying quality levels)
- **Resizing and resampling** (downscaling, upscaling, interpolation changes)
- **Geometric modifications** (cropping, minor rotation, padding)
- **Photometric adjustments** (brightness, contrast, gamma, color shifts)
- **Noise and filtering** (Gaussian noise, blur, denoising)
- **Image filters** (social media enhancers)
- **Format and pipeline effects** (color space conversion, re-saving workflows)

In addition to individual transformations, selected **multi-step pipelines** (e.g., resize → compress → crop) may be applied to better approximate realistic processing scenarios such as social media sharing or image editing workflows.

Following the transformations, watermark robustness will be assessed using the results from the participant-provided detection and decoding components. Evaluation will measure:

- **Imperceptibility** (watermarked image quality)
- **Watermark detectability** (presence/absence)

- **Payload recovery accuracy** (for multi-bit watermarking)
- **Payload integrity** (correctness of recovered bits)

All transformations, including parameter ranges and combinations, will be predefined and applied consistently across all submissions to ensure fairness and reproducibility. The same transformation protocols will be used for all teams within a given evaluation track.

This phase is intended to quantify the resilience of watermarking methods under **non-adversarial, real-world conditions**, providing insight into how well embedded signals survive typical image handling and distribution processes. The next section will discuss in more detail the performance metrics planned to be applied to measure perceptual similarity, robustness, diversity of generation, and payload decodability.

6 Performance Metrics for Watermarking Outputs

The evaluation of watermark embedding systems in Phase 1 will focus on three primary dimensions: **perceptual similarity**, **generation quality and diversity** (for prompt-based methods), and **robustness under natural transformations**. All metrics will be computed using standardized protocols. See Metrics Definitions and purpose for a summary of the metrics definitions.

6.1 Perceptual Similarity

Perceptual similarity measures the visual quality of watermarked images relative to their original or reference images. For **post-hoc watermarking methods** (i.e., watermarking applied to provided images), **textual prompt-based generation** (generating pairs of watermarked and unwatermarked images) and **image-based regeneration** (systems that regenerate a provided image while embedding the watermark), imperceptibility will be evaluated by comparing the submitted image to the provided reference image using:

- Peak Signal-to-Noise Ratio (PSNR)
- Structural Similarity Index Measure (SSIM)
- Learned Perceptual Image Patch Similarity (LPIPS and R-LPIPS)

These metrics quantify pixel-level fidelity and perceptual similarity between the original and watermarked images.

NOTE: For non-diversity topics, text-prompt-based generation methods must use the same generation seed across all required watermarked/unwatermarked images for each topic. The prompt and other generation settings must remain fixed, while the watermarking factors designated for evaluation are varied. Generation randomness should be controlled separately from any watermark-specific randomness. This helps reduce the influence of generation randomness, allowing the effects of watermarking factors on the generated images to be assessed more clearly. Topics designated for diversity evaluation must comply with the seed policy outlined in the diversity evaluation section.

6.2 Generation Diversity (Prompt-Based Methods Only)

For methods that generate multiple images per input prompt, diversity will be evaluated to measure variation across generated outputs while maintaining semantic consistency. A subset of topics will be selected for diversity evaluation, and the topics and .xml file will indicate which topic and how many images are required to

be generated. Teams should generate the same number of images as watermarked and as unwatermarked (i.e., two sets of images using the same exact prompt).

For each topic, teams must use a predetermined sequence of distinct generation seeds. **The same seed must be used for each corresponding watermarked–unwatermarked image pair, while different pairs must use different seeds.** The same seed sequence must therefore be used for both sets. Please follow the file naming convention while saving the files in the appropriate folders as indicated in the Results Submission Guidelines section.

- CLIP-based diversity metrics will be used to quantify variation among generated images for a given prompt. Results will be compared between the watermarked and unwatermarked sets.
- Other metrics as appropriate.

All generation parameters other than the seed must remain fixed for each topic across both sets, except for changes required by the watermarking method. Variation across generated images should arise primarily from the controlled changes in the generation seed.

6.3 Robustness Under Natural Transformations

To evaluate robustness, a standardized set of natural (non-adversarial) transformations will be applied to the submitted watermarked images. Following the transformations, each image will be processed using the participant-provided watermark detector and decoder. Robustness evaluation includes:

- **Watermark Detectability**
  - True Positive Rate (TPR)
  - False Positive Rate (FPR) - utilizing unwatermarked images
  - Receiver Operating Characteristic Area Under Curve (ROC-AUC)
  - Brier Score
- **Payload Decodability** (for multi-bit watermarking)
  - Bit Error Rate (BER)
  - Bit Accuracy

Payload decodability is evaluated when a watermark is detected and applies to multi-bit watermarking methods.

Appendix A Metrics Definitions and Purpose

Metrics	Purpose
Bit Error Rate (BER)	**Extraction** Quantifies the ratio of incorrectly decoded bits after an attack. (Lower scores are better for watermarks. Higher scores are better for attackers.)
Bit Accuracy (BA)	**Extraction** Quantifies the ratio of correctly decoded bits after an attack. BA = 1 - BER

	(Higher scores are better for watermarks. Lower scores are better for attackers.)
TPR @ 1% FPR	**Detection** Quantifies the proportion of actual positive cases successfully caught when the model is tuned to only allow 1% of negative cases to be incorrectly flagged.
AUC-ROC	**Detection** Measures how well a watermark embedder's decoding algorithm can distinguish between unwatermarked and watermarked images for all possible decision thresholds.
Brier Score	**Detection** Measures the accuracy of probabilistic prediction by computing the Mean Squared Error (MSE) between the predicted probability and the real output as a binary label [0,1].
Peak Signal-to-Noise Ratio (PSNR)	**Imperceptibility** Computes the ratio between the maximum possible value of a signal (original image) and the power of the distorting noise that affects the quality of its representation (attacked image).
Structural Similarity Index (SSIM)	**Imperceptibility** Compares the similarity between two images by evaluating luminance, contrast, and structural information.
Learned Perceptual Image Patch Similarity (LPIPS)	**Imperceptibility** Measures perceptual similarity based on deep network feature embeddings ("aligns closely with human judgment").
Contrastive Language-Image Pre-Training (CLIP)	**Diversity** Measures the semantic consistency between the image and its corresponding textual description via the multimodal CLIP model. CLIP Score/Alignment: Measures image-text alignment. CLIP IQA/Quality: Measures perceptual image quality.
DreamSim	**Diversity** Measures perceptual similarity using pairwise image distance, computed as the cosine distance between concatenated backbone model features ("aligns closely with human judgment").

You can find more details about the metrics on the NIST GenAI website (Metrics).