

## D User Guide

This guide provides the basics to build, install, and run the FESTS framework for generating spatio-temporal data based on the Woven Perception dataset.

### D.1 The Formally Explainable Spatio-Temporal Scenes Framework

This framework can be found under the `fests/` folder provided in the supplementary materials. To use this tool, you must first have completed the following pre-requisites:

1. Install the STREAM tool (v0.1.1)
2. Install the Python interpreter (v3.10)
3. Install the FESTS library (v0.1.0)

**Note:** For the following subsections below (Apps. [D.1.1](#) to [D.1.3](#)), all commands are ran from the root project directory as the working directory—the directory where these instructions reside.

#### D.1.1 Installing the STREAM Tool

The STREAM tool is used to generate the formally verifiable match results from the perception data provided to be appended in the creation of the FESTS dataset.

**Pre-Requisites** The STREAM tool is a Rust-based command-line interface tool which relies on the Rust compiler and toolchain to build and install, correctly. Therefore, installation of the toolchain is required, this can be done by running the following command:

```
$ curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
```

**Installation** To install the STREAM tool to your system to be ran as a command-line tool, run the following command from the root directory, accordingly:

```
$ cargo install --path strem/
```

#### D.1.2 Installing the Python Interpreter

In this demonstration, a Linux-based environment is assumed for proper installation of the correct Python interpreter. For this guide, we will assume an Ubuntu-based system. To install the correct Python version, run the following command from the root directory, accordingly:

```
$ sudo apt update && \
sudo apt install software-properties-common -y

$ sudo add-apt-repository ppa:deadsnakes/ppa && \
sudo apt update

$ sudo apt install python3.10 python3.10-venv python3.10-dev
```

#### D.1.3 Install the FESTS Library

To install the FESTS library for FESTS-based dataset generation, run the following command from the root directory, accordingly:

```
$ pip install feststools/
```

To verify successful installation, run the following commands:

```
$ strem --version && \
python --version && \
python feststools/scripts/process.py --help
```

### D.1.4 Running the FESTS Framework Tool

To run the tool and generate a uniformly sampled FESTS dataset, run the following command from the root directory, accordingly:

```
$ python festools/scripts/process.py \
--output="output/" \
--recursive \
--jobs=32 \
--context="festools/data/" \
festools/data/woven/
```

After running the command above, the results will be saved to the output/data/ folder. From this, the format of such a file may look as follows:

```
{
  "input": {
    "input": "You identify video scenes matching a natural language query
    → using frame-level object detections.\nInput XML
    → structure:\n<root>\n  <query>Natural language scene
    → description.</query>\n  <data>\n
    → frame,identifier,label,score,xmin,ymin,xmax,ymax\n
    → 0,AB,pedestrian,1.0,1254,603,269,101\n
    → 1,AC,car,0.9,1300,600,280,110\n    ... \n  </data>\n</root>\nOutput
    → format:\n-Matched frames as lists of consecutive indices in
    → <result> tags.\n-Brief explanation inside <reasoning> tags.\n-If no
    → match, output: <result>[]</result><reasoning></reasoning>\nExample
    → output:\n<result>[[1,2,3],[7,8]]</result>\n<reasoning>Frames 1-3
    → and 7-8 matched due to presence of pedestrians crossing the
    → road.</reasoning>\nNo extra text outside <result> and <reasoning>
    → tags.---\n<root>\n  <query>Find all instances where the area of a
    → car is greater than 5000 pixels for one or two frames.</query>\n  <d
    → ata>\nindex,identifier,class,xmin,ymin,xmax,ymax\n23,aa,car,232,53
    → 8,307,571\n23,ba,car,323,504,518,643\n23,ca,car,558,508,741,672\n2
    → 3,da,car,488,517,570,579\n23,ea,car,893,517,1011,554\n23,fa,car,28
    → 5,525,366,578\n23,ga,car,480,521,537,562\n23,ha,car,265,526,407,60
    → 4\n24,ga,car,485,521,540,561\n24,ia,car,39,528,258,623\n24,da,car,
    → 497,517,574,576\n24,ca,car,564,507,736,662\n24,ha,car,281,526,415,
    → 600\n24,aa,car,217,538,293,570\n24,ba,car,343,505,523,636\n24,fa,c
    → ar,293,525,366,576\n24,ea,car,893,516,1011,554\n</data>\n</root>\n"
  },
  "output": [
    [
      23,
      24
    ]
  ],
  "explanations": [
    "From index 23 to 24, area of the bounding box of a car is greater than
    → 5000."
  ]
}
```

In addition, you may view the output/stats.json to view a set of overall and per-query dataset statistics such as the elapsed time, seed, number of files processed, percentage of files that have non-empty matches, etc.

## D.2 Resources

This section highlights some resources that are anonymously linked outside of the supplementary materials due to storage limits. You can find a list of them below:

- For the fine-tuned model weights, see [here](#).
- For the FESTS dataset based on the Woven Perception dataset ground truth labels used for fine-tuning, see [here](#).

The following links listed above are first and foremost anonymized to uphold the double-blind requirements. However, upon notice, publicized versions will be released with proper DOI information, etc. to retain data integrity.

### D.2.1 Important

The table below highlights some important resources provided in the supplementary materials and their purposes for inspection:

Resource	Details
<code>strem/</code>	The STREAM tool source code with modifications
<code>fests/</code>	The FESTS framework source code
<code>fests/data/woven/</code>	A sample from the Woven Perception dataset
<code>fests/data/prompt.txt</code>	The LLM prompt used for fine-tuning the LLM model
<code>fests/data/queries.json</code>	The set of SpRE and NL queries fine-tuned with
<code>dataset/fests/</code>	A sample FESTS dataset generated from the Woven dataset

Table 4: A set of important resources and locations.