**Tasks for Prospective GSoC 2025 Applicants for the**

***CMS and End-to-End Deep Learning Projects @ ML4SCI Umbrella Organization***

Below are the tasks that we will use to evaluate prospective students for the E2E projects. After completing the common task, please thoroughly complete the specific task for your project of interest and (optionally) other tasks if you would like to also be considered for additional E2E projects at the same time (this may increase your chances of success, but make sure you don't do it at the expense of the specific project you are interested in)

**Note:** please work in your own github branch (i.e. NO PRs should be made). Send us a link to your code when you are finished and we will evaluate it.

Note: We encourage you to **upload the solutions on your github page** and **submit the github link to the solutions and your CV** at least 1 week before the GSoC Proposal Submission deadline, or earlier, so that you have enough time to write the proposal.

**NOTE: YOU MUST ALSO SUBMIT A PROPOSAL THROUGH THE GOOGLE SUMMER OF CODE PORTAL TO BE CONSIDERED.**

***!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!***

***—-------------------------------------------------------------------------------------------------------------------***

***Test Submission Instructions***

***—--------------------------------------------------------------------------------------------------------------------***

**Submission Instructions:** Please send us your CV and a link to all your completed work (github repo, Jupyter notebook + pdf of Jupyter notebook with output) to [ml4-sci@cern.ch](mailto:ml4-sci@cern.ch) with Evaluation Test: E2E/CMS in the title.

***!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!***

[Note] Tasks last updated on 3rd March 2025.

**Common Task 1. Electron/photon classification**

Datasets:

<https://cernbox.cern.ch/index.php/s/AtBT8y4MiQYFcgc> (photons)

<https://cernbox.cern.ch/index.php/s/FbXw3V4XNyYB3oA> (electrons)

**Description:** 32x32 matrices with two channels: hit energy and time for two types of

particles, electrons and photons, hitting the detector.

Please use a **Resnet-15 (you are free to play around with the architecture) like model**, to achieve the highest possible

classification score on this image dataset. Please provide a **Jupyter notebook** that shows your solution along with the model weights. Preferably, only use **PyTorch** or Keras in your solutions.

Please train your model on 80% of the data and evaluate on the remaining 20%. Please make sure not to overfit on the test dataset - it will be checked with an independent sample.

**Specific tasks for CMS Projects**

***Specific Tasks 2a***

* If applying for the [Event Classification with Masked Transformer Autoencoders](https://ml4sci.org/gsoc/2025/proposal_CMS1.html)

**Description:**

* Train a Transformer Autoencoder model of your choice on the dataset below using only the first 21 features and only the first 1.1million events. The last 100k items are to be used as your test set.
* Train a decoder of your choice which uses the latent space outputs of the Transformer encoder layer as inputs.
* Evaluate the performance of the classifier and present a ROC-AUC score for your final classifier.
* For a sense of good performance you can check the original reference paper. <https://arxiv.org/pdf/1402.4735.pdf>
* Discuss choices made in model selection and optimization.

**Dataset:**

* + [**https://archive.ics.uci.edu/dataset/280/higgs**](https://archive.ics.uci.edu/dataset/280/higgs) **(First 1.1 million events)**

***Specific Tasks 2b***

* If applying for the [Super resolution at the CMS detector](https://ml4sci.org/gsoc/2025/proposal_CMS2.html)

**Dataset:** [**https://cernbox.cern.ch/s/EYgmOkI9BjwxNqy**](https://cernbox.cern.ch/s/EYgmOkI9BjwxNqy)

**Description:**

* Dataset contains125x125 matrices of low (X\_jets\_LR) and high (X\_jets) resolution in three-channel images for two classes of particles, quarks and gluons, impinging on a calorimeter.
* Train any Generative Adversarial Network for super-resolution task.
* Discuss choices made in model selection and optimization.

**End-to-end (E2E) Deep Learning projects**

***Please first complete the Common Tasks #1, same as for CMS project. Then please complete the specific tasks below for individual projects.***

***Specific Tasks 2c***

* If applying for the [End-to-End event classification with sparse autoencoders](https://ml4sci.org/gsoc/2025/proposal_E2E2.html)
* Train a variational autoencoder architecture of your choice on the given unlabelled dataset.
* Use the labelled dataset to fine tune the encoder part for classification of the given binary logits.
* Prune the model and report the variation of performance with pruning ratio. Plot the results as a line plot with FLOPS on the X axis and error on the Y axis.
* [Bonus Task] Try only if you have managed to complete the above tasks. Train a sparse autoencoder architecture of your choice and benchmark it against the baseline architecture. Plot the results of both the Baseline and Sparse Network as a line plot with FLOPS on the X axis and error on the Y axis.

Datasets:

* Unlabelled (Use wget to download them directly): [Index of /cfs/m4392/G25](https://portal.nersc.gov/cfs/m4392/G25/)
* Labelled (Use wget to download them directly): [Index of /cfs/m4392/G25](https://portal.nersc.gov/cfs/m4392/G25/)

***Specific Tasks 2d***

* If applying for the [Diffusion models for fast and accurate simulations of low level CMS experiment data](https://ml4sci.org/gsoc/2025/proposal_E2E3.html)
* Train a generative model using the unlabelled dataset using the principle from the diffusion model paper. The architecture need not to be extremely deep or computationally expensive. Proof of concept is what we are looking for.
* Use the diffusion model to generate new samples from the given datasets and compare the generated samples with the training dataset. Develop your own statistical tests (you are allowed to be creative here) and give appropriate justification for the same to compare the generated samples with the training samples
* [Bonus Task] Try only if you have managed to complete the above tasks. Train a GAN and a VAE with similar architecture and compare the samples generated from each method using the statistical tests that were developed.
* [Bonus Task] Try only if you have managed to complete the above tasks. Demonstrate the issue of complexity bias of VAEs and mode collapse of GANs.

Datasets:

* Unlabelled (Use wget to download them directly): [Index of /cfs/m4392/G25](https://portal.nersc.gov/cfs/m4392/G25/)

***Specific Tasks 2e and 2f***

* If applying for the [Deep Learning Inference for mass regression](https://ml4sci.org/gsoc/2025/proposal_E2E4.html)

***Specific Tasks 2e***

Please train a model to estimate (regress) the mass of the particle based on particle images using the provided dataset.

**DataSet Description:** 125x125 image matrices with name of variables: ieta and iphi, with 4 channels called X\_jet (Track pT, DZ and D0, ECAL). Please use at least ECAL and Track pT channels and ‘am‘ as the target feature. If there are more than 4 channels in the dataset then you should use X\_jet (Track pT, DZ and D0, ECAL) only. Please train your model on 80% of the data and evaluate on the remaining 20%. Please make sure not to overfit on the test dataset - it will be checked with an independent sample.

**Datasets:** <https://cernbox.cern.ch/s/zUvpkKhXIp0MJ0g>

***Specific Task 2f Inference within CMSSW***

**Setup CMS Software Framework (CMSSW) on your local machine (with docker) and check the inference timing using the steps below.**

**Hint:**

Use CERN CentOS7 + CMSSW docker images from here:

<https://hub.docker.com/r/clelange/cmssw> (standalone image, tag 10\_6\_8\_patch1 suggested) or <https://hub.docker.com/r/clelange/cc7-cmssw-cvmfs> (required packages downloaded on-the-fly from CERN’s SW repository cvmfs)

Some information about the CMSSW docker images can be found here:

<https://github.com/clelange/cmssw-docker/blob/master/README.md>

(especially the section “Running containers”)

In the latter case (cvmfs-based) you can setup a CMSSW development area with, e.g.:

*cmsrel CMSSW\_11\_0\_1;*

*cd CMSSW\_11\_0\_1/src/;*

*cmsenv*

Additional CMSSW packages can be downloaded using, e.g.:

*git cms-addpkg DataFormats/TestObjects*

*git clone https://github.com/rchudasa/RecoE2E.git*

Code can be compiled typing:

*scram b -j8*

You can download the root file to run the inference from this link: E/gamma:

<https://cernbox.cern.ch/s/Yp3oZl8cUU6JoFC>

Convert the model from Task 1 in ONNX format and use it in the inference

**Inference can be run as follows**

*cmsRun RecoE2E/EGTagger/python/EGInference\_cfg.py inputFiles=file:****SIM\_DoubleGammaPt50\_Pythia8\_1000Ev.root*** *maxEvents=-1 EGModelName=sample.onnx*

***Specific Tasks 2g***

* If applying for the [Next generation vision transformers for end to end mass regression and classification](https://ml4sci.org/gsoc/2025/proposal_E2E5.html) then perform the following specific task

**Specific Task 2g. Next generation vision transformers for end to end mass regression and classification.**

**Description:**

* Train a Resnet-15 model using Self-Supervised Learning using a custom loss/training scheme defined in any of the papers ([MoCo](https://arxiv.org/abs/1911.05722), [SimCLR](https://arxiv.org/abs/2002.05709), [OBoW](https://arxiv.org/abs/2012.11552), [Barlow Twins](https://arxiv.org/abs/2103.03230) or [VICReg](https://arxiv.org/abs/2105.04906)) on the provided unlabelled dataset.
* Finetune the model for both regression and classification using the low learning rate on the provided labelled dataset and compare the results with a model trained from scratch.

Please train your finetuned and scratch model on 80% of the data (Labelled) and evaluate on the remaining 20%. Please make sure not to overfit on the test dataset - it will be checked with an independent sample. Please provide a **Jupyter notebook** that shows your solution along with the model weights.

**Datasets (Unlabelled/Pretraining Stage and Labelled/Finetuning Stage):**

[**https://cernbox.cern.ch/s/e3pqxcIznqdYyRv**](https://cernbox.cern.ch/s/e3pqxcIznqdYyRv)

* [Index of /cfs/m4392/G25](https://portal.nersc.gov/cfs/m4392/G25/)
* Use *Dataset\_Specific\_Unlabelled.h5 for training*
* Use *Dataset\_Specific\_labelled\_full\_only\_for\_2i.h5* for fine tuning

***Specific Tasks 2h***

* If applying for the [End-to-End particle collision track reconstruction](https://ml4sci.org/gsoc/2025/proposal_E2E6.html)

**Datasets:** <https://cernbox.cern.ch/s/oolDBdQegsITFcv>

**Description:**

* Choose 2 Graph-based architectures of your choice to classify quarks or gluon jets. Provide a description of how you have converted the point-cloud dataset to a set of interconnected nodes and edges.
* Discuss the performance of the 2 chosen architectures.

***Specific Tasks 2i***

* If applying for the [Foundation models for End-to-End event reconstruction](https://ml4sci.org/gsoc/2025/proposal_E2E7.html)
* Using Self-Supervised Learning techniques from computer vision build an encoder to map the given dataset samples to latent vectors.
* Using the latent vectors from the labelled dataset, build a classification model for the logit “y” and regression models for the “pT” and “m”. Keep the depth of the downstream models as shallow as possible. For classification ROC curves are to be plotted with AUC-ROC as the metric and for the regression tasks True vs Predicted plots are to be plotted with MSE and F1 score as the metric.
* [Bonus Task] Try only if you have managed to complete the above tasks. Build a Mixture of Expert setup using different models such as VAE, Contrastive Models and Supervised Models. Distill the information of the ensemble into an encoder such that it has higher metrics for all these different tasks.

Datasets:

* Unlabelled (Use wget to download them directly): [Index of /cfs/m4392/G25](https://portal.nersc.gov/cfs/m4392/G25/)

***Specific Tasks 2j***

* If applying for the [Discovery of hidden symmetries and conservation laws](https://ml4sci.org/gsoc/2025/proposal_E2E8.html) or [Semi-supervised Symmetry Discovery](https://ml4sci.org/gsoc/projects/2025/project_SYMMETRY.html)

**Task 1:**

* **Dataset Preparation:** Use the vanilla MNIST dataset for this purpose. **Rotate every sample in steps of 30 degrees** and store them in a data format of your choice. Only use the digits 1 and 2 from the dataset if the computational budget is limited.
* **Latent Space Creation:** Build an Variational Auto-Encoder of your choice and train it using the dataset prepared in the previous step.

**Task 2:**

* Supervised Symmetry Discovery: On the latent space using a MLP and by simply rotating the samples in steps of 30 degrees learn the transform (using MLP on the latent space) that maps every vector to a rotated version of it in the image space.

**Task 3:**

* **Unsupervised Symmetry Discovery:** Using the latent space now and referring to the work in [paper, discover](https://arxiv.org/abs/2302.00806) the symmetries in the MNIST dataset created earlier that preserve the logit. Rotation should be one of the discovered symmetries.

**Bonus Task [Very Hard] :**

* **Rotation Invariant Network:** Build a rotation invariant network using the discovered symmetries earlier. We discourage the contributor from spending too much time on this task given its difficulty.

Note: This project is advanced in nature and we are expecting to recruit multiple contributors for the same who will work on similar topics but will explore different approaches. Some of these tasks might not result in perfect symmetry discoveries or results. We encourage the contributors to submit irrespective of the quality of symmetry operators they discover.