SPAI Manual

Version 0.0.01

SPAI Group, IIT Kharagpur

March 30, 2024

Contents

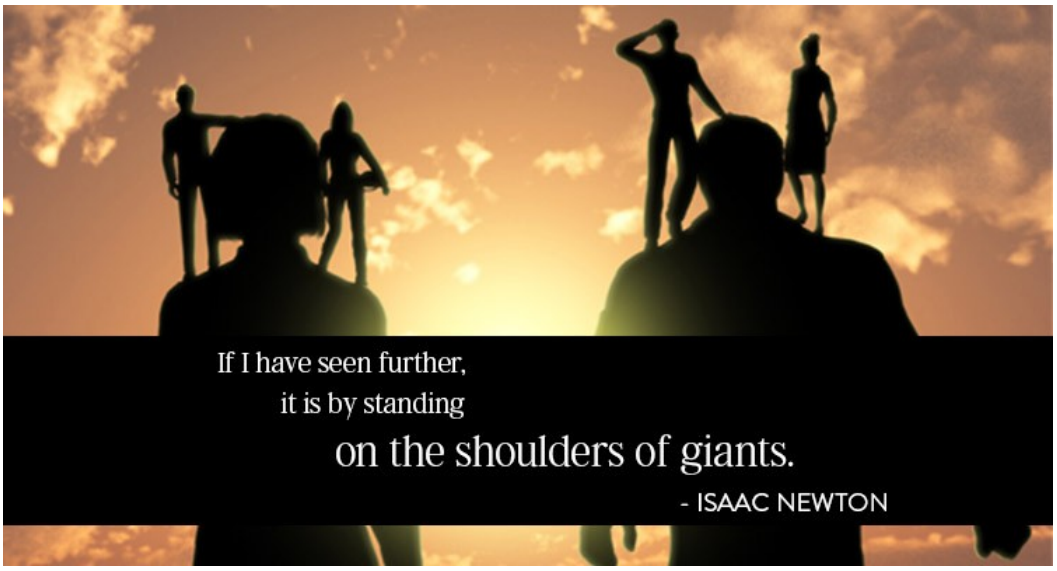

Figure 1: Motivation of the SPAI Manual

1 Introduction

Dear student, we welcome you to the Signal Processing and Artificial Intelligence (SPAI) Group! This document is a manual created by SPAI Group using LaTeX. The main purpose of this document is to help you with understanding the SPAI Group expectations. Further this helps you to *not* reinvent the wheel our SPAI predecessors had already found out, but to stand on their shoulders to see further, and explore new research territories.

Let us start by remembering SPAI Group's Vision!

SPAI Group's Vision: " *As a young citizen of India, armed with technology, knowledge and love for my nation, I realize, small aim is a crime.*"– A P J Abdul Kalam, People's President **AND** "*Science is a collaborative effort. The combined results of several people working together is often much more effective than could be that of an individual scientist working alone.*"– John Bardeen, Nobel Prize in both 1956 and 1972.

2 SPAI Group Expectations

Figure 2: An Ideal Mentor-Student Relationship: John Bardeen (right), two-times Nobel Laureate and Nick Holonayak (left), the inventor of LED, who was his research student and later joined as a Professor in the Univ of Illinois.

SPAI Group's aim is World-class Research. We follow the clarion of Prof. Nick Holonayak's group in UIUC (pioneers of LED): "*We are a small group out here .... We have to beat the 200-person, large budget group in top places. We can do it as long as we work hard and fast. A small group with good ideas and hard work, we can do better than these large groups in well-funded places.*"

We insist a healthy and professional environment within and outside the group. A scientific advancement of one member should be an advancement of the entire group. More important, getting motivation by comparing with others who did well is fine, but spoiling your self-esteem by comparisons is *not at all* acceptable. Ultimately, each SPAI member should develop an instinct for invention, an ability to focus on the problem at hand, the skillfulness to juggle multiple approaches, and a fierce determination to pursue that problem to a successful conclusion. We want your tenure in the SPAI group as one of your fondly remembered time period in your life – where you can revisit yourself growing *stronger* Scientifically, Morally, and Mentally!

2.1 Desirable Research Directions/Problems

We urge all works of SPAI Group to be of basic research[1], with strong theories backed by rigorous experiments. Basic (aka fundamental or pure) research is driven by a scientist's curiosity or interest in a scientific question (as opposed to Applied research, where research is driven by a specific industrial application). What is then a good Scientific Question for conducting basic research? According to two-times Nobel Laureate John Bardeen[2], a good scientific question must have three important requirements:

1. "First of all", look at "whether there is a *technological basis*" for the work. "If you think something in some theory but it can never be realized because there is no technology there, you are working in empty space."
2. Second, the problem needed to be *challenging*, "because if it's so simple that you can do it on the back of an envelope, well, the project is over."
3. Third, the research should have *applications potential*. "That's what most people in basic science overlook. If you do something and you want it to have importance, it has to mean something to the people."

Bardeen's scientific legacy is extraordinary for its breadth and depth. According to him, it is very important to theorists to immerse themselves in experimental data and be guided by that. Some students have more aptitude in theoretical analysis, whereas others like more of experimental analysis. We insist both types of students to reinforce each others' research. Ultimately, we encourage everyone to find out for themselves derive mathematical insights, design an experiment, and interpret results. Bardeen's nine-fold way[3] to solve research problems is:

i Focus first on the experimental results through reading and personal contact –(Carefully analyse experimental results and identify gaps).

ii Develop a phenomenological description that ties different experimental results together – (Reason out how and why this happen).

iii Explore alternative physical pictures and mathematical descriptions *without* becoming wedded to any particular one – (Do not insist that a theory in some paper is right or your theory is right).

[1] https://www.nature.com/articles/028510a0
[2] https://www.currentscience.ac.in/Volumes/85/11/1636.pdf
[3] https://www.nasonline.org/publications/biographical-memoirs/memoir-pdfs/bardeen-john.pdf

iv Macroscopic arguments or higher-level view have precedence over microscopic calculations or intricate views – (A theory or logic should first carry a purpose or goal, then second, a reason for why it is true. After that only, a detailed mathematical proof or validity matters).

v Focus on physical understanding or truth, *not* mathematical elegance, and use the simplest possible mathematical description of system – (Try to use simpler mathematics to prove theory or validate logic).

vi Keep up with new theoretical techniques or new solutions —for one of them may prove useful – (Always stay updated on your research area).

vii Decide on a specific model or method as the penultimate, not the first, step toward a solution – (Finalize a Theory or Logic after experiments).

viii Choose the right collaborators – (Collaborators need to have complementary strength and willingness to work).

ix **DON'T GIVE UP**: Stay with the problem until it is solved. – (Difficulties are natural in research. If you change problem, you may keep changing your problems when difficulty comes and no problems bear fruit.)

2.2 Desirable Target Venues for Publishing

If you publish in top tier venues, then only your work gets better visibility. Getting one top tier publication is equivalent to getting three low impact publications; at the same time no one appreciates the latter. Top tier publications require solid idea, extensive experimentation that unambiguously validate your idea, and good paper presentation. The acceptance percentage is usually around 17%, which means 83 out of 100 papers will be rejected! To show the strength of SPAI Group, aiming for nothing less than the following venues in Tables 1-2 are acceptable (remember SPAI's Vision in Sec. 1).

We prioritize conference submission as it has less and definite time for the paper to be published, if accepted (as compared to Journals). If rejected, we can revise and resubmit to utmost 2 more conferences; if still the paper is struggling, then we submit the revised version to a Journal.

Desirable Conference Venue List		
Conference Name	h5-index (↑)	Deadline
Conference on Computer Vision and Pattern Recognition (**CVPR**)	422	Mid of Nov
Neural Information Processing Systems (**NeurIPS**)	309	Mid of May
International Conference on Learning Representations (**ICLR**)	303	Mid of March
International Conference on Machine Learning (**ICML**)	254	End of January
International/European Conference on Computer Vision (**ICCV/ECCV**)	228/238	Mid of March
AAAI Conference on Artificial Intelligence (**AAAI**)	212	Mid of August

Table 1: Venues arranged according to h5-index. An h-5 index of 200 means that that conference has published 200 articles in the previous 5 years that have 200 or more citations each.

Desirable Journal Venue List		
Journal Name	h5-index (↑)	SJR (↑)
IEEE Transactions on Pattern Analysis and Machine Intelligence (**TPAMI**)	397	4.447
Expert Systems with Applications (**ESA**)	249	1.873
Journal of Machine Learning Research (**JMLR**)	239	2.281
IEEE Transactions on Neural Networks & Learning Systems (**TNLS**)	234	3.447

Table 2: Journal Venues arranged according to the h-5 index and SCImago Journal Rank (SJR). The SJR indicator is a measure of the prestige of scholarly journals that accounts for both the number of citations received by a journal and the prestige of the journals where the citations come from.

2.3 Desirable Target Aims for Different Group

We expect the results in Table 3 after the completion of your tenure in SPAI Group. To accomplish this requires sincerity, hard work, and tenacity.

Role	Requirement	Desirable Target Aim
PhD	More than 3 first-author top-tier publications	Postdoc in top foreign Univ (MIT, Stanford, UIUC, etc)
BTP/MTP	More than 1 co-author top-tier publication	Direct PhD in top foreign Univ (NOT include MS Program)

Table 3: Desirable Target Aims for different Group.

3 GPU Server Tips and Tricks

Pl read this if you are assigned a GPU server credential in the SPAI group (this usually happens once a student completes the literature study).

3.1 How to Create a User Account in Server (only for Admin)

An example of how to create a new user with the preferred location. Note that this needs Admin privileges.

```
$ pwd # See whether you are in the preferred dir
$ mkdir /SPAI_ons  # This is a comment
$ sudo useradd -m -d /SPAI_ons SPAI_ons
$ sudo passwd SPAI_ons
$ spai_password # Provide password (you cannot see the characters)
```

An example of how to delete the new user. Note that this needs Admin privileges.

```
$ pwd # See whether you are in the preferred dir
$ sudo userdel SPAI_ons
```

(To avoid cluttering, please create various user accounts clubbed based on projects.)

3.2 How to Access a User Account in Server

An example of how to access a user account. Suppose the IP of the system is 10.21.9.14. Open a terminal and type:

```
$ ssh SPAI_ons@10.21.9.14
$ spai_password # Provide password (you cannot see the characters)
```

If you are accessing your allotted User Account for the first time, create a directory/folder with your name. All your data, codes, packages and operations must confine to your directory (specifically when deleting something).

```
$ mkdir my_name
$ cd my_name # Change working dir to yours
$ pwd # Print working dir; use this to verify you are in your dir
```

If needed, one can check the free space available and GPU spec as follows:

```
$ df -Ph . | tail -1 | awk '{print $4}' # Check free space available
$ nvidia-smi # Check number of GPUs, their utilization, and specs
```

3.3 How to Transfer files from One's Computer to Server

Use the command scp which stands for Secure copy (Linux/Unix) or pscp (Windows) to securely move folders or files from a local computer onto remote server. For pscp, one needs to install PuTTy software in Windows. Below is a demonstration of scp (change scp to pscp in Windows machine).

1. Transfer files from local machine to remote host.

```
$ scp -R /sourcepath/source_dir_or_file
↪  remote_user@10.13.13.11:/path/to/remote/directory
```

2. Transfer files from remote host to local machine.

```
$ scp -R remote_user@10.13.13.11:/path/to/remote/directory
↪  /sourcepath/source_dir_or_file
```

3.4 How to Create Project-Specific Virtual Environments

Suppose you are working on a research project that requires Pytorch 1.1, and its dependencies, while another environment associated with a finished project has Pytorch 2.1 (perhaps because version 1.1 was the most current

version at the time the project finished). If you upgraded to V2.1, your previous project may throw errors due to some feature in V1.1 got superseded.

Virtual environments (like Conda) keep these dependencies in separate "sandboxes" so you can switch between both applications easily and get them running. Conda is an open source package management system and environment management system that quickly installs, runs and updates packages and their dependencies. Conda comes in two broad forms: *Anaconda* with large with lots of programs in it already, and *Miniconda* that is more lightweight and then we can install just what we want. We recommend independent Miniconda for each user (Sec. 3.4.1). With this, a user can create as many project-specific Conda environment as needed (Sec. 3.4.1).

3.4.1 How to Install Miniconda Dedicatedly for a User

A dedicated Miniconda for each user avoids accidental modification/deletion of other users' Conda environments. Each user is supposed to work using the Miniconda installed on one's own Directory. One can install Miniconda dedicatedly for oneself as

```
$ cd my_name # Change working dir to yours
$ mkdir -p ./miniconda3  # Create a subdir to install minicaonda
$ wget
 ↪  https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
 ↪  -O ./miniconda3/miniconda.sh # Download latest miniconda version
$ bash ./miniconda3/miniconda.sh -b -u -p ./miniconda3 # run the
 ↪  install script
$ rm -rf ./miniconda3/miniconda.sh  # delete the install script
$ ./miniconda3/bin/conda list  # If success, you will see a list.
```

3.4.2 How to Create a Project-Specific Conda Environment

Suppose that you want to create a environment for your current project, say "CvnnProject".

```
$ ./miniconda3/bin/conda create --name CvnnProject # Create Env
$ ./miniconda3/bin/conda env list # Verify Env is created
$ source ./miniconda3/bin/activate ./miniconda3/envs/CvnnProject #
 ↪  If activated the Env, you may see (CvnnProject) in Terminal.
```

Suppose the current project require Pytorch GPU (note that default Pytorch installation is CPU based). Do an internet search 'how to install pytorch gpu in conda'. For example, staying inside the CvnnProject environment,

```
$ conda install conda-forge::pytorch-gpu # Install Pytorch-gpu
↪  inside CvnnProject
```

Suppose one wants to install Torchvision in CvnnProject environment:

```
$ conda install -c conda-forge libpng
$ conda install -c conda-forge libjpeg
$ conda install torchvision -c pytorch
```

Once you installed the packages/dependencies for the CvnnProject, now we can run the codes we want.

```
$ python my_pytorch_training.py
```

If one want to deactivate the current environment, and activate a different environment (already created before):

```
$ conda deactivate
$ ./miniconda3/bin/conda env list # See available envs and identify
↪  the required env
$ ./miniconda3/bin/conda create --name required_env
```

If one wants to *permanently delete* an environment

```
$ ./miniconda3/bin/conda env remove --name unwanted_env
```

3.5 How to use Screen for Training

Deep Learning training usually takes long hours; however closing your ssh terminal (Sec. 3.2) in your local computer terminates that particular session and hence your training program. Screen solves this issue. Screen allows to run a long-running task in screen, detach, disconnect. The job will still be running in screen and one can come back later, reattach, and check its progress. In addition, Screen allows to run multiple tasks without making multiple ssh connections to a remote server.

To create a screen named MyName (do this before activating your env),

```
$ screen -S MyName # Create screen
$ echo $STY  # Verify if you are in Screen (MyName will be shown)
```

To detach a screen, type ctrl-a + d key combinations. To attach an old screen again,

```
$ screen -ls # List all existing screens, & identify old Screen name
$ screen -r OldScreen # Attach using the selected Screen name
```

To quit any screen, first detach and type

```
$ screen -X -S MyName quit
```

3.6 How to use VS Code as a Graphical Interface of Server

The Visual Studio Code Remote - SSH extension allows you to open a remote folder and edit files on any remote GPU server and run codes. Once connected to a server, you can edit program files and folders in an interactive fashion anywhere on the remote file system, using full advantage of VS Code's feature set. It appears as if you are working with VS Code in your Laptop, but the changes you made and code execution take place in the server. No source code needs to be on your local machine to gain these benefits since the extension runs commands and other extensions directly on the remote machine. Follow the following steps to use VS code (See here for a nice detailed illustration[4]):

1. Install VS Code from Microsoft in your system (if it is not present).
2. Open VS Code, and Install the Remote - SSH extension.
3. Click "Connect to" and give ssh details (see Sec. 3.2).
4. Click "Open Folder" and navigate to your particular folder.
5. Now you can find/edit your codes, and even open VSCode's terminal which is now nothing but the remote GPU Server terminal.

[4] https://carleton.ca/scs/2023/vscode-remote-access-and-code-editing/

3.7 Selecting Specific GPU(s) in a Server for Training

To select specific GPUs to run a training code, say `YourScript.py`

```
$ CUDA_VISIBLE_DEVICES=0 python YourScript.py # Select first GPU
$ CUDA_VISIBLE_DEVICES=1 python YourScript.py # Select second GPU
$ CUDA_VISIBLE_DEVICES=0,1 python YourScript.py # Select 2 GPUs (if
↪  Pytorch code has multi-gpu training instructions.)
```

One has to explicitly code in Pytorch for multi-gpu training[5]. Even if a code has multi-gpu training instructions, one can train using a single GPU by selecting a single GPU. However, if a code does *not* have multi-gpu training instructions, one *cannot* use multiple GPUs even if one selects multiple GPUs (here the training happens in the first GPU in the list).

[5] https://pytorch.org/tutorials/beginner/blitz/data_parallel_tutorial.html

4 Pytorch Tips and Tricks

4.1 Assignment for Beginners

If you are a beginner, please understand completely the following Pytorch Tutorial[6]. If you got assigned a SPAI GPU Server, you should try executing the same in your assigned Server and collect similar results (following Sec. 3). First run in CPU and then run in GPU. Train the code using Screen (Sec. 3.5).

```
$ python my_pytorch_code.py  # Run in One's Project-specific Conda
↪  Environment
```

While the code is running, create a different Screen to find the GPU memory utilization and Volatile GPU utility. If your code runs in GPU, you should be able to find in the Terminal, your code in the processes section, the GPU where your code is running, and corresponding memory usage.

```
$ nvidia-smi -l
```

If there exist free GPU memory, you may run additional codes in the same GPU (no need to care about GPU utility even if it is 100% (or 30%) as this tells your data loader works faster (or slower) than GPU processes the data).

[6] https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html

5 LaTeX Tips and Tricks

Pl read this if you are ready to write a paper (this usually happens once a student completes the exploratory and/or experimental study). First of all, there are 3 rules to be followed while writing mathematics expressions: see Mermin's description[7]. Second, we recommend reading Freeman's advice[8] on how to write papers to avoid paper rejections due to some silly reasons (this doc is from pre-DeepLearning era, but it is true now as well).

5.1 How to Start an Overleaf to Write Papers

Two advantages of TeX over text editors and word processors: high-quality typesetting (especially for math) and the ability to automate formatting. LaTeX easily produce PDFs with hyperlinks, table of contents, indices, etc, with an excellent referencing system to cite papers.

Top tier conferences (ICLR, NeurIPS, CVPR, ICCV, ECCV, etc) and Joutnals (PAMI, TIP, TMLR, etc) provide their paper templates in zip file. To start writing paper, you can create a project in Overleaf and upload the zip file in your project (as illustrated here[9]). With Overleaf, all the formatting and numbering is handled automatically according to the template you've chosen. Overleaf allows you to share this with your collaborators, who can also write in parallel some other sections of the paper.

5.2 How to include Figures

First you have to upload the image file from your computer using the upload link in the file-tree menu. Then use the `includegraphics` command to include it in your document. Use the figure environment and the caption command to add a number and a caption to your figure. See the code for Figure 3 in this section for an example.

Note that your figure will automatically be placed in the most appropriate place for it, given the surrounding text and taking into account other figures or tables that may be close by. You can find out more about adding images to your documents in this help article on including images on Overleaf.

```
\begin{figure}
\centering
```

[7] http://www.ai.mit.edu/courses/6.899/papers/mermin.pdf
[8] https://billf.mit.edu/sites/default/files/documents/cvprPapers.pdf
[9] https://www.overleaf.com/learn/how-to/Uploading_a_project

Figure 3: This frog was uploaded via the file-tree menu.

```
\includegraphics[width=0.3\textwidth]{frog.jpg}
\caption{\label{fig:frog}This frog was uploaded via the file-tree
↪  menu.}
\end{figure}
```

5.3 How to add Tables

Use the table and tabular environments for basic tables — see Table 4, for example. For more information, please see this help article on tables.

Table 4 shows how to add a table caption and reference a table.

```
 \begin{table}
\centering
\begin{tabular}{||c c c c||}
 \hline
 Col1 & Col2 $\uparrow$ & Col2 $\downarrow$ & Col3 \\ [0.5ex]
 \hline\hline
 1 & 6 & 87837 & 787 \\
 \hline
 2 & 7 & \underline{78} & 5415 \\
 3 & 544 & 778 & 7507 \\
 4 & \textbf{541} & 18744 & \textcolor{blue}{7560} \\
 5 & 88 & 788 & 6344 \\ [1ex]
 \hline
\end{tabular}
\caption{Table to test captions and labels. $\uparrow$ indicates
↪  higher is better.}
```

Col1	Col2 ↑	Col2 ↓	Col3
1	6	87837	787
2	7	78	5415
3	544	778	7507
4	**541**	18744	7560
5	88	788	6344

Table 4: Table to test captions and labels. ↑ indicates higher is better.

```
\label{table:data}
\end{table}
```

5.4 How to add Lists

You can make lists with automatic numbering ...

```
\begin{enumerate}
\item Like this,
\item and like this.
\end{enumerate}
\dots or bullet points \dots
\begin{itemize}
\item Like this,
\item and like this.
\end{itemize}
```

1. Like this,
2. and like this.

...or bullet points ...

- Like this,
- and like this.

5.5 How to write Mathematics

```
\LaTeX{} is great at typesetting mathematics. Let $X_1, X_2, \ldots, X_n$ be a sequence of independent and identically distributed random variables with $\text{E}[X_i] = \mu$ and $\text{Var}[X_i] = \sigma^2 < \infty$, and let
\begin{equation}
S_n = \frac{X_1 + X_2 + \cdots + X_n}{n} = \frac{1}{n}\sum_{i}^{n} X_i
\end{equation}
denote their mean. Then as $n$ approaches infinity, the random variables $\sqrt{n}(S_n - \mu)$ converge in distribution to a normal $\mathcal{N}(0, \sigma^2)$.
```

LaTeX is great at typesetting mathematics. Let $X_1, X_2, \ldots, X_n$ be a sequence of independent and identically distributed random variables with $\text{E}[X_i] = \mu$ and $\text{Var}[X_i] = \sigma^2 < \infty$, and let

$$S_n = \frac{X_1 + X_2 + \cdots + X_n}{n} = \frac{1}{n}\sum_{i}^{n} X_i \tag{1}$$

denote their mean. Then as n approaches infinity, the random variables $\sqrt{n}(S_n - \mu)$ converge in distribution to a normal $\mathcal{N}(0, \sigma^2)$.

5.6 How to customize the template

You may wish to customize the template for your own style, or to meet the specific needs of your documentation. If you're already familiar with LaTeX, you can go ahead and add the packages you're familiar with to the document preamble. If you run into any problems and can't find the answers in the package documentation or in the Overleaf help library[10], the forums such as TeX StackExchange[11] and LaTeX Community[12] are a great source of answers.

Some details on how to customize a .cls file (which sets the layout and overall format of the various elements of the template) can be found at Writing your own class[13], and LaTeX2e for class and package writers[14].

[10] https://www.overleaf.com/learn
[11] https://tex.stackexchange.com/
[12] https://latex.org/forum/
[13] https://www.overleaf.com/learn/latex/Writing_your_own_class
[14] http://texdoc.net/pkg/clsguide


INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR
भारतीय प्रौद्योगिकी संस्थान खड़गपुर
19 51
योगः कर्मसु कौशलम्


SP
Ai
Signal Processing and Artificial Intelligence Group