"We Rock the Hizzle and Stuff" hints on how to write a nice research essay

Michele Rossi[†], Jacopo Pegoraro[†]

Abstract—**Future vehicular communication networks call for new solutions to support their capacity demands, by leveraging the potential of the millimeter-wave (mm-wave) spectrum. Mobility, in particular, poses severe challenges in their design, and as such shall be accounted for. A key question in mm-wave vehicular networks is how to optimize the trade-off between directive Data Transmission (DT) and directional Beam Training (BT), which enables it. In this paper, learning tools are investigated to optimize this trade-off. In the proposed scenario, a Base Station (BS) uses BT to establish a mm-wave directive link towards a Mobile User (MU) moving along a road. To control the BT/DT trade-off, a Partially Observable (PO) Markov Decision Process (MDP) is formulated, where the system state corresponds to the position of the MU within the road link. The goal is to maximize the number of bits delivered by the BS to the MU over the communication session, under a power constraint. The resulting optimal policies reveal that adaptive BT/DT procedures significantly outperform common-sense heuristic schemes, and that specific mobility features, such as user position estimates, can be effectively used to enhance the overall system performance and optimize the available system resources.**

This is an example abstract. It is 204 words long, I would say an abstract should not be longer than 250 words and some Transactions journals of the IEEE are currently putting a strict limit of 200 words. Here, you should briefly state:

1) **the technical scenario/field of research and its timeliness/relevance in general (one sentence),**
2) **what you do in the report/paper and why it is important, how it advances the state of the art in its field (a few sentences),**
3) **summarize the main and best results of your study/proposal/method (one or two sentences),**
4) **(optional) how others could benefit from your results for further research, or within commercial products (one sentence).**

The abstract is one of the most important parts of the paper/report. You have roughly one minute to catch the reader's attention. A poor abstract may already move you towards the rejection side in the reviewer's decision process. In the abstract, 1) establish the context, 2) motivate the problem, 3) briefly describe the solution, and 4) present the main results of your work. Ideally, use one (short) sentence for each of the previously mentioned items to keep your abstract short. Overall, this should be a short summary of the whole content of your paper, including your results.

See the abstract as a personal challenge for each of your papers. Finally, the abstract should contain the main message about your work, so that the reader will now what she/he can find even without reading it (as it is the case most of the times).

The abstract is a mini-paper on its own and, as such, it is a major endeavor to write.

I suggest to write the Abstract as the very last thing. You may sketch it at the beginning, but then always finalize it at the end.

Index Terms—**Unsupervised Learning, Optimization, Neural Networks, Convolutional Neural Networks, Variational Autoencoders. A list of keywords defining the tools and the scenario. I would not go beyond *six* keywords.**

I. Introduction

Remark I.1. Paper contribution: First and foremost decide on what precisely is the contribution of your paper over the state of the art. If you think you have several contributions, *focus on the most important one*. It may be that you can add one or two contributions as side topics, but in general you should focus on the most important one so as *to keep your paper focused*. As a side note: If you think you have several contributions for a single paper, you should probably invest in researching state of the art and related work more thoroughly. When you know your contribution, think of a good title.

Remark I.2. Title: Find a short and precise title for you paper exactly matching the content. It's worth investing time into this matter, as *the title will be that part of the paper by which it will be referenced* (in case it gets published).

Remark I.3. How to organize your writeup with Latex: I find it useful, especially for a paper with multiple authors, to split your manuscript into multiple source Latex files. This is very easily accomplished by having single *root* file, e.g., `main.tex`, where you define the article class, all the user-defined commands, the paper formatting options (e.g., one vs two columns, the page margins, the text size and the fonts, etc.). For this project, the root is called `template.tex`. From the root you then call, in sequential order, a number of other latex source files through the command `\input{filename.tex}`. It is often convenient to have one of such files for each section of your paper. This facilitates editing multiple sections in parallel and keeping your project synchronized with some versioning and revision control system such as or `git` or `overleaf` (highly recommended).

Remark I.4. Compiling trick: For each source Latex file that you call from the main, I recommend you copy the following command in the very first line

[†]Department of Information Engineering, University of Padova, email: `{name.surname}@unipd.it`

Special thanks / acknowledgement go here.

`% !TEX root = template.tex`
where `template.tex` is the name of the root Latex file. This command tells the Latex compiler that your main root file is `template.tex`, and it makes it possible to compile the paper from any of the sub-files. Example: imagine you are editing `introduction.tex` and have that open in a window of your preferred Latex editor. With this command, you can compile from the local `introduction.tex` window, without having to open and switch every time to the root file window and compile from it. This will save you hours.

Maximum length for the whole report is 10 pages. Abstract, introduction and related work should take max two pages.

Recommended structure for the intro: you may use the following structure.

- **General (short) intro:** One paragraph to introduce your work, describing the scenario *at large*, its relevance, to prepare the reader to what follows and convince her/him that the paper focuses on an important setup/problem. Please keep this part short (I usually do five to eight lines), as this part is rather standard, **but** at the same time it has to be there.
- **Put the problem into perspective:** A second paragraph where you immediately delve into the specific problem that you are tackling, starting to detail your contribution. Here, you describe the importance of such problem, providing examples (citing papers from the literature, possibly recent ones) of previous solutions attempts, and of why these failed *to provide a complete answer*. This second paragraph should not be too long, as otherwise the reader will get bored and will abandon your paper... It should be concisely written, something like 5 to 10 lines.
- **Present the paper contribution:** A third paragraph were you state what you do in the paper, this should also be concisely written. A good rule of thumb is to make it max 10/15 lines. Here, you should state up front:
  1) **problem**: the problem at stake,
  2) **relevance**: the relevance and timeliness of what you propose,
  3) **approach**: the technique/approach you use, possibly underlying its novelty, efficiency,
  4) **value**: underline the value/novelty of your proposal referencing (recent) papers from the literature,
  5) **applicability**: tell the reader how she/he can take advantage of your work, e.g., how your work/results can be reused/exploited to achieve further scientific, technical or practical (integrated into products?) goals.
- **Summary of contributions:** After this, you may want to provide an itemized list to summarize the paper contributions. Rule of thumb: from three to six items, from three to four lines each.
- **Closing (paper structure):** You finish up by detailing the paper structure, this should be three to four lines. It is customary to do so, although I admit it may be of little use. It usually goes like: *"This report (paper) is structured as follows. In Section II we describe the state of the art, the system and data models are respectively presented in Sections III and IV. The proposed signal processing technique is detailed in Section V and its performance evaluation is carried out in Section VI. Concluding remarks are provided in Section VII."*

Remark I.5. Lately, I tend to write introduction plus abstract within a single page. This forces me to focus on the important messages that I want to deliver about the paper, leaving out all the "blah blah". **Remember:** 1) *less is more*, 2) writing a compact (*snappy*) piece of technical text is much more difficult than writing lengthy stuff with no space constraints.

II. Related Work

Some hints:

- **Goal:** The goal of this section is to describe what has been done so far in *the* literature. You should focus on and briefly describe the work done in the best papers that you have read.
- **Length:** One full column is fine but often this takes one column and a half. It is very easy to use a full page, although this may just due to your sloppiness... if you carefully go through the one page long version, you often find it possible to compact it in one column and a half. In any event, I would make this section no longer than one page, this leads to an overall *two pages* including abstract, introduction and related work. I believe this is a fair amount of space in most cases.
- **Approach:** For each you should comment on the paper's contribution, on the good and important findings of such paper and also, **1)** on why these findings are not enough and **2)** how these findings are improved upon/extended by the work that you present here. At the end of the section, you may recap the main paper contributions (maybe one or two, the most important ones) and how these extend/improve upon previous work.
- **References:** please follow this *religiously*. It will help you a lot. Use the Latex `Bibtex` tool to manage the bibliography. A `Bibtex` example file, named `biblio.bib` is provided with this template.

- **Citing conference/workshop papers:** I recommend to always include the following information into the corresponding `bibitem` entry:
  1) author names,
  2) paper title,
  3) conference / workshop name,
  4) conference / workshop address,
  5) month,
  6) year.

  Examples of this are: [1] [2].

- **Citing journal papers:** I recommend to always include the following information into the corresponding `bibitem` entry:
  1) author names,
  2) paper title,
  3) full journal name,
  4) volume (if available),
  5) number (if available),
  6) month,
  7) pages (if available),
  8) year.

  Examples of this are: [3] [4] [5].

- **Citing books:** I recommend to always include the following information into the corresponding `bibitem` entry:
  1) author names,
  2) book title,
  3) editor,
  4) edition,
  5) year.

Remark II.1. Note that some of the above fields may not be shown when you compile the Latex file, but this depends on the bibliography settings (dictated by the specific Latex style that you load at the beginning of the document). You may decide to include additional pieces of information in a given bibliographic entry, but please, **be consistent** across all the entries, i.e., use the same fields for the same publication type. Note that some of the fields may not be available (e.g., the paper *volume*, *number* or the *pages*).

III. Processing Pipeline

Remark III.1. On tailoring the paper structure to your needs: The structure recommended for the previous sections is rather standard and could work for different papers with differing technical content, the structure and the paper content from here on highly depends on the type of paper, possibilities are: mostly based on theoretical analysis, showing experimental design/activity, proposing a new technique and analyzing its performance via experiments or simulation. For the HDA course we deal with machine learning and, in detail, with training and testing neural network architectures to perform some specific inference or classification task. The following structure and comments are specifically addressing this type of technical content.

Remark III.2. Why having this section: With this section, we start the technical description with a *high level* introduction of your work (e.g., processing pipeline). Here, you do not have to necessarily go into the technical details of every block/algorithm of your design, this will be done later as the paper develops. What I would like to see here is a high level description of the approach, i.e., which processing blocks you used, what they do (in words) and how these were combined, etc. This section should introduce the reader to your design, explain the different parts/blocks, how they interact and why. You should not delve into technical details for each block, but you should rather explain the big picture. Besides a well written explanation, I often use a nice diagram containing the various blocks and detailing their interrelations.

Writing tips: Sections, Figures and Tables are usually shortened as Sec., Fig., Tab.

- **Cross referencing:** In Latex, cross referencing is easy. You need to label an object through the `\label{labelid}` command and referencing it where you need it through the `\ref{labelid}` command.
- **Suggestion:** I suggest to cross reference a table using `Tab.~\ref{tab:tableid}`, the same holds for figures and sections, by just replacing "`Tab.`" with "`Fig.`" and "`Sec.`". Of course when defining the table, you need to add a Latex command `\label{tab:tableid}` in the right place inside the table Latex environment to cross-link it. The tag `tab:tableid` is user defined and is the identifier that you associate with the table in question. You could call it `pippo` of you wish, but I recommend to use something like "`tab:tableid`" for tables, "`eq:eqid`" for equations, "`sec:secid`" for sections and so forth, where `tableid` has to be unique for each table in the document. The same applies to figures and all other objects. I guess you got the idea. This will lead to a neat Latex code and will facilitate cross referencing while avoiding duplicate labels, especially in large Latex documents (think of a book for example).
- **But what about the tilde?** This is a nice trick I have learned from a friend (many years ago from Prof. Frank Fitzek, now at TU Dresden). When you write `Tab.~\ref{tab:tableid}` Latex knows that `Tab.` and the corresponding table number `\ref{tab:tableid}` must be displayed within the same line, i.e., they can never be broken across lines. This is nice and desirable I believe. I always use it for all referenced material, including citations; example: "As done in~`\cite{suppa-wu-2019}`."
- **More about breaking stuff across lines** often times you have composed words, in line equations, etc. and for some reason you would like Latex to never break them across lines. Example: the Latex command `\mbox{Neural Networks (NN)}` is processed by Latex so that "Neural Networks (NN)" is never broken across lines. I use this very often, also for inline equations that I do not want Latex to split across subsequent lines.

IV. Dataset

In this section, we start by introducing the dataset used for our experiments, then explain how we processed it to use with our model, and finally how we made it an i.i.d. dataset.

1) Input data

The dataset we used for the training and evaluation is ModelNet10 from https://modelnet.cs.princeton.edu/.

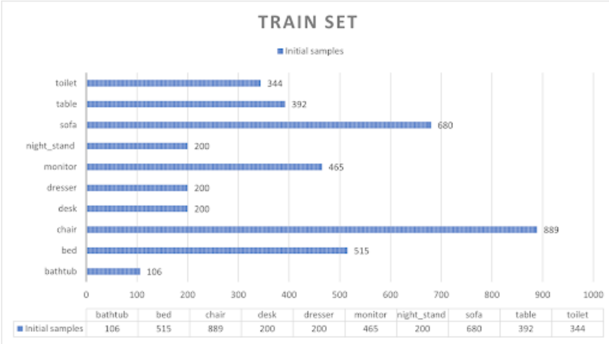

Fig. 1: Initial Train Set Distribution [TODO with ordby name]

It contains 4899 3D objects divided into 10 different categories and has an initial splitting of the data into 80% for training and 20% for testing.
In Figure 1 we can see the initial distribution of the training samples.

2) Pre-processing data

The CAD models, or Meshes, are in Object File Format (OFF), but our model, described in the section V, requires 3D boolean vectors with dimensions of $32 \times 32 \times 32$.

In order to get the required input format, we have done the following steps:

- With the help of the Open3d Python library, we converted each model in the dataset to voxel grids, 2D list of filled positions. After this step, each Mesh was converted into a voxelgrid of size $32 \times 32 \times 32$.
  As mentioned, voxel grids are good for saving space and in our case, we went from 2.17 GB to 311 MB.
- Next, we translated each voxel grid to a $32 \times 32 \times 32$ boolean array, thus making it ready as input for the model. The code is available in our Git repository under the folder: *code/preprocessing/voxelization/*.

As we can see from Figure 1, the ModelNet10 is not an Independent and Identically Distributed (i.i.d.) dataset. The assumption of I.I.D is central to almost all machine learning algorithms, and we tried to make our data as much as i.i.d in the following way:

- Firstly, we normalized the data, from the previous step, by subtracting the mean and dividing by the standard deviation. [TODO: remove?]
  normalized_voxelgrid = (voxelgrid - mean) / std
- Then, we augmented the dataset with new rotations of the voxel grids. Rotations are made by adding multiples of 30 degrees to the initial orientation. [TODO with formulas?]
- Finally, we randomly picked, per category, ~1k samples for the train set and ~300 samples for the test set. [TODO we are not precise here]

Figure 2 shows the distribution we get after these steps.

3) Partitioning data

A good practice is to divide the dataset into three sets: one set for training and two smaller sets for validation and testing.

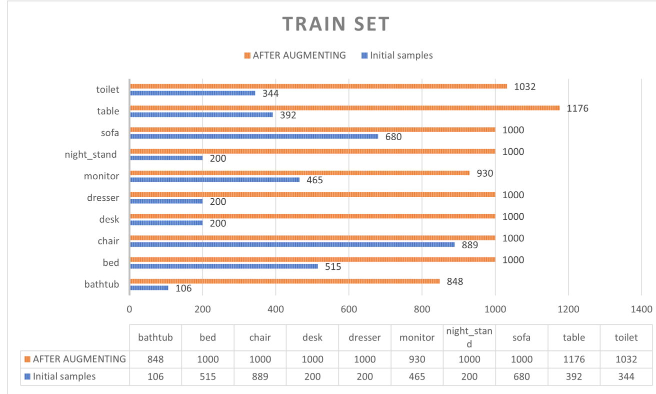

Fig. 2: Final Train Set Distribution [TODO with ordby name]

The ratio we used for splitting is 80/10/10, where 80 percent of the data is put into the training set, and the remaining percentage is equally spread across the test and validation sets.
This will be briefly recalled within the "Results" section.

V. Learning Framework

Here you finally describe the learning strategy / algorithm that you conceived and used to solve the problem at stake. A good diagram to exemplify how learning is carried out is often very useful. In this section, you should describe the learning model, its parameters, any optimization over a given parameter set, etc. You can organize this section into sub-sections. You are free to choose the most appropriate structure.

Remark V.1. Note that the diagram that you put here differs from that of Section III as here you show the details of how your learning framework, or the core of it, is built. In Section III you instead provide a high-level description of the involved processing blocks, i.e., you describe the *processing flow* and the rationale behind it.

On math typesetting: there are many Latex tricks that you should use to produce a high quality technical essay. A few are listed below, in random order:

- **Vectors and matrices:** x is a scalar, whereas $\boldsymbol{x}$ (in bold) is a vector, and $\boldsymbol{X}$ is a matrix with elements $\boldsymbol{X} = [x_{ij}]$. For bold symbols you may use the `\bm` Latex command, e.g., `\bm(x)`.
- **Operators:** such as max, min, argmax, argmin and special functions such as $\log(\cdot)$, $\exp(\cdot)$, $\sin(\cdot)$, $\cos(\cdot)$ are obtained through specific latex commands `\min`, `\max`, `\arg\!min`, `\arg\!max`, `\log`, `\exp`, `\sin`, `\cos`, etc. Use them! log, exp, min, sin, cos, etc., look ugly.
- **Sets** can be represented through calligraphic fonts, e.g., $\mathcal{S}$, $\mathcal{F}$, $\mathcal{B}$, etc., obtained using the Latex command `\mathcal{S}`, etc.
- **Equations:** for a single equation use the `equation` Latex environment. Example:

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \tag{1}$$

Now, using round brackets (and) we get

$$\sigma(x) = (\frac{1}{1+e^{-x}}), \tag{2}$$

but this looks ugly, you should use "`\left (`" for "(" and "`\right )`" for ")", obtaining

$$\sigma(x) = \left(\frac{1}{1+e^{-x}}\right). \tag{3}$$

- **Punctuation:** Displayed equations are usually considered to be part of the text and, in turn, they will get the very same punctuation as if they were inline with the text (and part of the sentence). If the sentence ends with a displayed equation, the equation gets a period "." right after it, see Eq. (1). iI the equation is instead part of a running sentence, which is continued after it, then the equation may be ended by a "," as in Eq. (2). Use the standard grammar rules and your good sense of flow to assess how equations should be punctuated, I usually read through as if they were plain text.

VI. Results

In this section, you should provide the numerical results. You are free to decide the structure of this section. As a general "rule of thumb", use plots to describe your results, showing, e.g., precision, recall and F-measure as a function of the system (learning) parameters. You can also show the precision matrix.

Remark VI.1. Present the material in a progressive and logical manner, starting with simple things and adding details and explaining more complex findings as you go. Also, do not try to explain/show multiple concepts within the same sentence. Try to **address one concept at a time**, explain it properly, and only then move on to the next one.

Remark VI.2. The best results are obtained by generating the graphs using a vector type file, commonly, either `encapsulated postscript (eps)` or `pdf` formats. To plot your figures, use the Latex `\includegraphics` command. Lately, I tend to use pdf more.

Remark VI.3. If your model has hyper-parameters, show selected results for several values of these. Usually, tables are a good approach to concisely visualize the performance as hyper-parameters change. It is also good to show the results for different flavors of the learning architecture, i.e., how architectural choices affect the overall performance. An example is the use of CNN only or CNN with adversarial training, or using residual layers for CNNs, dropout for better generalization or autoencoder models. So you may obtain different models that solve the same problem, e.g., CNN, CNN+residual layers, etc.

VII. Concluding Remarks

This section should take max half a column, I personally find it difficult to come up with really useful observations, I mean ones that bring a new contribution with respect to what you have already expounded in the "Results" section. In case you have some serious stuff to write, you may also extend the section to 3/4 of a column :-).

In many papers, here you find a summary of what done. It is basically an abstract where instead of using the present tense you use the past participle, as you refer to something that you have already developed in the previous sections. While I did it myself in the past, I now find it rather useless.

What I would like to see here is:

1) a very short summary of what done,
2) some (possibly) intelligent observations on the relevance and *applicability* of your algorithms / findings,
3) what is still missing, and can be added in the future to extend your work.

The idea is that this section should be *useful* and not just a repetition of the abstract (just re-phrased and written using a different tense...).

Moreover: being a project report, I would also like to see a specific paragraph stating

4) what you have learned, and
5) any difficulties you may have encountered.

VIII. Exam Rules

What you need to do to pass the exam:

- Optional: team up with another student. Max. group size is **three students** per group;
- Identify a project to work on, devise your own neural network architecture and test it on the provided dataset;
- **Prepare a written project report** including: i) diagrams, ii) configuration pars, iii) results, iv) your discussion;
- **Prior to presenting your work**: upload i) your written report and ii) the code;
- **Present your work** using slides (max. duration is 15 minutes): take turns in presenting your work, your individual contribution to the project should clearly emerge.

Your final grade will be obtained taking into account the following criteria:

- **Project** (50 points): originality (10 pt.) - data preprocessing techniques (10 pt.) - learning architectures (20 pt.) - comparison against other/existing approaches (10 pt.)
- **Written report** (25 points): clarity of exposition (8 pt.) - completeness (9 pt.) - analysis of results (number and type of metrics used) (8 pt.)
- **Oral exposition** (25 points): duration (your talk must take max. 15 minutes, using slides) (15 pt.) - questions (10 pt.)

The final grade will be computed as

$$\text{grade} = \frac{\text{tot\_points} \times 30}{110} \tag{4}$$

References

[1] M. Zargham, A. Ribeiro, A. Ozdaglar, and A. Jadbabaie, "Accelerated dual descent for network optimization," in *American Control Conference*, (San Francisco, CA, US), June 2011.

[2] C. M. Sadler and M. Martonosi, "Data compression algorithms for energy-constrained devices in delay tolerant networks," in *ACM SenSys*, (Boulder, CO, US), Oct. 2006.

[3] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, pp. 379–423, July 1948.

[4] S. Boyd, N. Parikh, E. Chu, and B. P. J. Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends in Machine Learning*, vol. 3, pp. 1–122, Jan. 2011.

[5] D. Zordan, B. Martinez, I. Vilajosana, and M. Rossi, "On the Performance of Lossy Compression Schemes for Energy Constrained Sensor Networking," *ACM Transactions on Sensor Networks*, vol. 11, pp. 15:1–15:34, Aug. 2014.