The Title of Your Work

Your Name
Department,
University
City, Country
YourEmail

ABSTRACT

This is the abstract. Here, you should provide a short summary of the entire work. You should include the following elements: your research problem and objectives, your methods, your key results, and your conclusion. Aim for 150 to 300 words.

KEYWORDS

Your, Keywords, Go, Here

1 INTRODUCTION

This section serves as the introduction to the work and it is likely the second thing readers will focus on, after the abstract. This section should contextualize your work, introducing the topic and the expected results. Usually, it is also in this section that authors include a brief summary of the main contributions of the work.

Additionally, you can have several subsections for topics such as the problem statement, the motivation for this work, and also any needed background information for unfamiliar readers. These can also be made into their own sections that follow the introduction. The problem statement is a short description of the problem that your work addresses, identifying the gaps between the current state and the desired future state. The motivation should sell the importance of this work and the reader should understand what is the impact of solving the identified problem.

2 ACTIVITIES AND PLAN

In this section you should present the temporal evolution of the project. Include any relevant diagrams or charts, and list the finished tasks, possibly comparing them to the list of expected tasks.

3 YOUR SOLUTION

In this section you will describe your work, both in form and function. That is, you should describe its structure/architecture, but also how it works, both internally and externally. Internally, any data structures and algorithms that you have used are relevant, especially if they were developed especially for this work. Externally, consider how an end-user would use your work.

Additionally, in this section you can also present the technical difficulties faced during the development, if and how they were overcome, and any limitations of the final product.

Artifacts such as code or API listings do not belong in this section. If they are important to the report and need to be included, they should be put in appendices at the end of the document.

4 EXPERIMENTAL EVALUATION

In this section you can describe the experimental evaluation that your performed to evaluate your project.

```
function MemoizedFunction(input)
    $hash \leftarrow$ HashFunction($input$)
    $< found, output > \leftarrow$ Lookup($hash, input$)
    if $found = False$ then
        $output \leftarrow$ OriginalFunction($input$)
        Update($hash, input, output$)
    end if
    return $output$
end function
```

Figure 1: This is the algorithm caption.

4.1 Experimental Setup

In this subsection you present the setup you used to perform your tests. The benchmarks uses, the runtime system, the hardware, the methodology, and so on. This is needed for reproducibility purposes and for readers to understand how your work is more or less relevant in their own context.

4.2 Experimental Results

In this subsection you present the achieved results and you conduct an analysis of the experiments to see if the results obtained are what was expected according to your initial assumptions. If they are not, you are expected to understand and explain why.

5 RELATED WORK

In this section you can present other work that is relevant to your own research and development. Usually a short paragraph will explain how the other approach works and what are the main differences to your work, e.g., what problems were left open that you manage to solve or how you took a different approach and what are its pros and cons.

6 CONCLUSIONS

In this section you present the main conclusions of your work in a summarized form. You can also present relevant future work on to to tackle current limitations and extend functionality.

7 USEFUL EXAMPLES

This section, which you should delete later on, has examples on how to use some of Latex's most common features and environments.

Figure 1 presents a basic algorithm using the *algorithmic* environment for pseudo-code.

This is how you make an unumbered list:

- This is the first item;
- This is the second item;
- This is the final item.

Figure 2: This is the figure caption.

```
double mw_log_double (double A) {
    // compute hash
    uint64_t A_bits = * (uint64_t*) &A;
    uint64_t hash_64_bits = A_bits;
    uint64_t hash_32_bits = (hash_64_bits ^ (hash_64_bits >> 32));
    uint64_t hash_16_bits = (hash_32_bits ^ (hash_32_bits >> 16));
    hash_16_bits = hash_16_bits & 0xffff;
    // lookup and return if hit
    if (mw_log_double_table[hash_16_bits][0] == A_bits) {
        return * (double*) &mw_log_double_table[hash_16_bits][1];
    }
    // call original and update table if miss
    double result = log(A);
    mw_log_double_table[hash_16_bits][0] = A_bits;
    mw_log_double_table[hash_16_bits][1] = * (uint64_t*) &result;
    return result;
}
```

Figure 3: This is the code caption.

Whenever you cite someone else's work, you need to include a reference to the relevant source. This is how you make a citation [2]. You can also cite multiple sources at once like this [1, 3]. The references need to be present in the *refs.bib* file.

Figure 2 shows you how to include a figure. By default, Latex uses PNG or PDF formats for figures.

If you need to include equations, you can use Latex's excellent math environments. Here is a simple example of an equation:

$$O(i, j) = c \times \log\left(1 + I(i, j)\right), \quad (1)$$

where c is a constant, I is the input, O is the output, and i and j are the image coordinates.

You can use the acronym package to help manage acronyms. The first time you use an acronym, its full form will be displayed, e.g., Domain-Specific Language (DSL) or High-Performance Computing (HPC). However, the following times, only the short version will be used, as in DSL or HPC. You can also use the plural form of acronyms, e.g., DSLs. The list of known acronyms is defined in the preamble of the document.

If you want to include code, you can also do so by using the listings package. Here is an example of how to do it. Figure 3 presents an example of how to include source code.

Finally, tables can also be a good option to present data. Table 1 gives you an example of how to generate tables. Notice that the caption appears above the table, contrary to what happens with figures.

ACKNOWLEDGMENTS

The author would like to acknowledge . . .

REFERENCES

[1] Daniel A. Connors, Hillery C. Hunter, Ben-Chung Cheng, and Wen-Mei W. Hwu. 2000. Hardware Support for Dynamic Activation of Compiler-Directed Computation Reuse. *SIGPLAN Notices* 35, 11 (Nov. 2000), 222–233.

[2] Donald Michie. 1968. "Memo" functions and machine learning. *Nature* 218, 5136 (1968), 19–22.

[3] Christopher Strachey. 2000. Fundamental Concepts in Programming Languages. *Higher-Order and Symbolic Computation* 13, 1 (Apr 2000), 11–49.

Table 1: This is the table caption.

	machine A	machine B
CPU	Intel Core i7-9700 CPU	2x Intel Xeon E5-2630 v3
CPU Frequency	3.00GHz	2.40GHz
RAM	16GB DDR4	128GB
OS	Ubuntu 20.04 LTS	Ubuntu 16.04 LTS
Compiler	GCC 9.3	GCC 7.3
libm	v2.31	v2.23
libomp	v4.5	v4.5