Advising Student Theses

Dirk Hartmann*

April 2026

Contents

Student theses are important milestones in your academic career. Choosing a thesis advisor is therefore an important decision, whether it is a bachelor. master, or doctoral thesis. The advisor should not only be an expert in the field you are interested in, but a mutual understanding of expectations is equally important. For everyone interested in writing a thesis with me, I have summarized the elements that matter most to me, including the advisor role, the thesis document itself, and useful tools. These reflections are based on my experience supervising bachelor, master, and doctoral students over the last decade. In addition to my own experience, I have also drawn on recommendations from some of my colleagues [4, 6]. This summary applies equally to bachelor, master, and doctoral students — the main difference being the scope.

If you believe I could be a good match as a thesis supervisor, please send me an email describing your field of interest and a brief motivation for why you would like to write a thesis with me. Please also include some information about yourself, such as your study program, previous experiences, and current grades.

1 The Scope

The scope of any thesis is to demonstrate that you can work independently and systematically on an academic problem. This distinguishes it from a student project, whose focus is primarily to

*TU Darmstadt, Schlossgartenstr. 8, 64289 Darmstadt, Germany; dirk.hartmann@tu-darmstadt.de

produce something (new). A thesis, however, requires you to position and compare your results against the academic state of the art, i.e., to embed them in a broader context. Thereby, the main difference between a bachelor, master, and doctoral thesis lies in the scope.

A **bachelor thesis** shows that you can work with academic sources, understand a topic, and answer a focused question in a structured way. You need to demonstrate that you can understand and apply academic work. However, you are not expected to produce new results; it is sufficient to reproduce existing work or incrementally improve it (e.g., an incremental extension of existing software).

A **master thesis** should make a clear scholarly contribution. That contribution does not need to be groundbreaking[1], but it should go beyond a simple summary. You should critically engage with the literature, understand current limitations, and reach the academic frontier (e.g., apply an existing method to a new domain). You should demonstrate that you can independently analyze and develop academic work, but you do not need to produce entirely new results,

A **doctoral thesis** is expected to make an original contribution to academic knowledge. You should demonstrate that you can create new results that add something genuinely new to the field through independent, advanced research at a professional level.

2 The Advisor

The role of any academic thesis is to demonstrate your capability to work independently. That is, you should demonstrate that you independently understand and apply academic work (bachelor thesis), independently analyze and develop academic work (master thesis), or independently create new academic knowledge (doctoral thesis). You are therefore expected to take ownership of the project. Learning to take independent ownership is likely one of the most valuable things you will take away from writing a thesis. Since you may well switch fields at some point after your thesis, the benefits of specific subject expertise may not last long, but the ability to take independent ownership is something you will benefit from for the rest of your life.

Therefore, the role of the advisor is that of an academic guide, helping you to stay academically sound, focused, and within a manageable workload. This is why I prefer the term *advisor* over *supervisor*. Advisors will often also assign an assistant advisor, e.g., such as a PostDoc or PhD student, to keep the workload manageable. A good (assistant) advisor should:

- help to narrow and shape a viable research question;
- advise on methodology, structure, and academic standards;
- give feedback on outlines and drafts;
- check that the project is feasible in scope and timeline;
- support you when running into conceptual or practical problems;
- ensure the thesis meets the formal expectations.

An (assistant) advisor should **not**:

- choose the exact topic for you;
- give feedback on every intermediate result;
- request weekly progress updates;
- correct every language mistake line by line;
- guarantee a specific grade.

[1] Although this is often the ambition, the strict timelines of theses usually make it an unrealistic expectation. That said, it is still worth striving for.

Since the advisor is not your supervisor or manager, weekly progress update meetings are and cannot be expected. To maintain appropriate continuity, I consider monthly 30-minute meetings ideal. Of course, ad hoc meetings are always possible if advice is needed at short notice.

Doctoral Thesis Mentor

Only very few universities foresee the role of a thesis mentor. However, I personally find this role extremely helpful in the context of doctoral theses. There will likely be many moments during a doctoral thesis where you feel frustrated. Creating new knowledge is harder than most people expect and is particularly difficult to plan. Carrying that responsibility can feel heavy at times, even though it is often also what makes a thesis rewarding. Having a trusted person alongside your thesis advisor can therefore be genuinely helpful. A good mentor should be someone you trust, feel comfortable sharing frustrations with, and who understands what it means to write a doctoral thesis.

3 The Document

Your main objective is to write the actual document. All software development, research, literature review, and so on are additional, but very necessary, steps toward that objective. Therefore, everything should be focused on the document, because this is what will be graded at the end. Although writing a master thesis often involves significant effort in implementation and experimentation. Reviewers will not assess the code directly; only what is documented can be graded.[2]

When writing the thesis, please keep the reader in mind. The thesis is not only a document for your advisor and the reviewers. A good choice for an anticipated reader would be yourself at the very start of the thesis. Theses are often an entry point for others into a specific research topic and should therefore not be written exclusively for experts. This differentiates a thesis from a scientific papers. Along these lines, it is important to provide sufficient motivation (there is a storytelling aspect) and sufficient detail so that a non-expert can engage with the topic (there is a textbook aspect). Please also keep reproducibility in mind: the reader should be able to reproduce all your results from the thesis alone. Last but not least, every reader will appreciate consistent, clear language and well-formatted formulae. For detailed formatting recommendations, see [4].

The structure outlined below is only a recommendation[3]. As long as there are good reasons for adopting a different structure and these can be clearly rationalized, it is entirely appropriate to adapt it.

Introduction

The introduction should, in addition to providing a high-level overview of the topic, state the specific scientific question to be addressed in the thesis. Formulating the research question explicitly tends to provide clarity and focus. It is therefore highly recommended to draft the introduction early in the process in order to benefit from this clarity throughout. It is natural for the ambition of a thesis to exceed what can be achieved within the given time frame. The specific research question may evolve or may not be fully answered - this is expected. However, this does not remove the requirement to formulate an initial scope and research question.

[2]Though the quality of the code, if you decide to share it will also play a significant role in shaping your reputation as a researcher. See Section on *Coding* further below.

[3]Theses are a means to train academic work, and independent thinking is a core part of this process - including decisions about the structure of the thesis.

Beyond the research question itself, the introduction should highlight the motivation for addressing it and the potential impact of answering it. Regardless of whether it is a bachelor, master, or doctoral thesis, it is beneficial to keep a non-expert but scientifically curious reader in mind.

State-of-the-Art

An important part of any thesis is developing a solid overview of the relevant field. Every thesis should therefore include a well-structured overview with sufficient references to current research literature. It should identify and compare similar and alternative approaches that have been used to address the research question or closely related problems. Thus, significant time should be allocated early on to reviewing the state-of-the-art[4], even though this section will typically continue to evolve over time.

Beyond demonstrating the ability to enter a new research field, this section is crucial for identifying competing approaches for later comparison. For example, implementations should be compared against existing state-of-the-art methods. Choosing appropriate baselines is particularly important when developing novel computational approaches [10].

Theory

While the introduction and state-of-the-art sections provide a broad overview, the theory section is expected to describe in detail the methods that have been implemented. In particular, concrete formulae are expected, in any computational science and engineering thesis.

Implementation

If the thesis is implementation-centric, it is recommended to include a dedicated section describing the implementation. This enables others to reproduce the results, with reproducibility being a key aspect of scientific research[5]. In addition to implementation details, it is important to describe how the code has been verified and validated, e.g., through unit testing. Careful verification and validation significantly increase the credibility of the results. The correctness of the implementation remains the responsibility of the author.

Results

The results section should demonstrate the performance and significance of the achieved outcomes. While presenting a compelling application or use case is valuable, scientific rigor requires a detailed quantitative analysis. Such an analysis may take various forms, ranging from classical evaluations (e.g., runtime measurements and convergence studies) to user studies assessing usability [2]. Ideally, multiple aspects should be covered rather than focusing on a single dimension.

A key element for the analysis is the comparison with state-of-the-art approaches. In most cases, alternative methods exist for achieving similar objectives, and only through appropriate comparison can the quality of the contribution be assessed. It is not necessarily expected that the proposed approach outperforms existing methods, especially in bachelor or master theses. However, the results should be carefully positioned relative to existing approaches.

[4] A useful practice is to write a short summary (2 - 3 sentences) for each paper read and to document it in a draft version of the thesis. This will make assembling the final state-of-the-art section considerably easier.

[5] Ideally, the code should also be made publicly available (e.g., via GitHub). This benefits the research community and may also be advantageous for future career opportunities.

Summary and Conclusion

The conclusion should summarize the thesis and revisit the initial research question. A positive outcome is not necessarily required. For example, identifying that a method does not work for a specific application is also a valuable scientific result. As long as this is not due to faulty or careless implementation, it will not negatively affect the evaluation. In addition, new ideas and potential research directions that emerged during the thesis should be discussed.

4 The Computational Experiments

When writing a thesis with me, coding will very likely be an integral part of the work. Even though only the thesis and not the code will be graded, I strongly recommend writing clean, well-tested code [9, 1]. Ensuring the correctness of the implementation is your responsibility. I have seen too many theses where the correctness of the code was unclear, which significantly reduced the value of the work. Writing clean code and thoroughly unit-testing each sub-module are key elements for achieving this. Simple regression tests at the end are generally not sufficient on their own.

Beyond writing good code, proper planning and execution of your computational experiments is equally crucial. The flexibility and efficiency of today's development frameworks often encourages disorganized workflows that lead to irreproducible results [7]. Reproducibility matters not only for academic value but also for your individual reputation [5]. To address this, Tammy Kolda suggests seven practices: use version control, separate experiments from figures, create human-readable logs, log details of individual runs, use scripts for automation, separate and organize your project components, and share your code, data, and experimental logs. *"These seven practices are not just about making your work reproducible - they're about making it credible, collaborative, and impactful."* [7]

Finally, consider open-sourcing your code after the thesis. As Randall LeVeque notes, *"[w]hatever state it is in, the code is an important part of the scientific record and often contains a wealth of details that do not appear in the paper, no matter how well the authors describe the method used."* [8]. Beyond supporting reproducibility, sharing your code will likely lead to broader citation of your work, e.g., as others build on or compare against it, as well as increased visibility of yourself, for instance with future employers.

5 The Tools

Writing

I strongly recommend writing the thesis in LaTeX. Beyond its ease of use for mathematical equations, the results simply look more beautiful. In addition to local installations, there are many good online tools available, the most prominent being Overleaf and TU Darmstadt's ShareLaTeX.

Coding

The choice of the coding language is up to you and depends very much on the scope. Typical considerations are efficiency, readability, adoption, as well as your skills. Regular examples include C++, Python, Julia, or Matlab.

Next to a professional IDE (Integrated Development Environment), and proper versioning [3], I strongly recommend the use of modern coding tools, including AI-assisted approaches, to improve

efficiency. However, a thorough understanding of the code and appropriate validation are expected. The correctness of the implementation remains the responsibility of the author.

Versioning

Versioning is your safety net during the thesis. I strongly recommend using GitHub, GitLab, or similar tools for this purpose, unless versioning is already built into the tools you are using, e.g., Overleaf includes a versioning system in the backend.

Generative AI

The use of generative AI is encouraged to improve coding efficiency. However, its use during writing should be limited. Formulating original thoughts and ideas is a central learning objective of writing a thesis. Large language models should therefore primarily be used for language refinement, i.e., as a post-processing step after drafting the text. A focused tool such as Grammarly is preferable, as more general tools tend to inflate the text and often disrupt the original line of thought.

Any use of generative AI, including for coding, should be clearly documented in the thesis. For further guidance, the official guidelines of TU Darmstadt[6] should be consulted.

Bibliography Management

There are many good and free bibliography tools available, such as Zotero. I personally prefer using a simple BibTeX file. Whenever you read a paper, it is recommended to write 2–3 sentences summarizing what the paper is about and how it relates to your thesis. This will considerably ease writing the state-of-the-art section later.

6 Time Management

Time management is another challenging aspect of writing a thesis, so do not underestimate it. The breakdown below offers a recommendation for how to allocate your time. However, like software development, writing a thesis is an iterative process. Thus, do not follow the plan below in strict order; instead, take an iterative approach and switch back and forth between the different elements as needed. This flexibility can also be very helpful when you get stuck on a particular point.

Literature research should ideally account for around 10-20 percent of the time. It should be done early and thoroughly to avoid discovering after five months that someone else completed the same work ten years ago. Whenever you read a paper related to your thesis, write down 2-3 sentences in the state-of-the-art section. This will prevent you from having to reread all the papers when writing that section later.

Theoretical work and software development will constitute the main body of work in your thesis and should account for around 30-50 percent of the time. Make sure to properly document and structure this work, as it will not only help you later but also ensure reproducibility. Remember: reviewers will not assess the code directly; only what is documented can be graded. Furthermore, appropriate diligence will help you avoid errors (e.g., in coding) that would slow down progress. Plan sufficient time for continuous testing throughout.

Testing and benchmarking is a key element of a thesis and should take around 10–30 percent of the time. You should clearly demonstrate how your approach compares to other methods, based on appropriate quantitative studies, whether numerical performance, usability [2], or similar

[6] https://www.hda.tu-darmstadt.de/hochschuldidaktik

metrics. Insufficiently validated baseline results, which often lead to overoptimism, are a common criticism in research [10].

Writing the document should be your main objective. It should therefore account for 20-30 percent of the time. All software development, research, and literature review are additional, but very necessary, steps toward that objective. Write as early as you can, as often as you can, and especially when you have clear thoughts that can be put down quickly. Planning only the final two weeks for writing typically leads to a significantly worse grade.

References

[1] Beck, Kent: *Test-driven development: by example*. Addison-Wesley Professional, 2003

[2] Brooke, John u. a.: SUS-A quick and dirty usability scale. In: *Usability evaluation in industry* 189 (1996), Nr. 194, S. 4–7

[3] Chacon, Scott ; Straub, Ben: *Pro git*. Springer Nature, 2014

[4] Dietrich, Felix: *How to write a thesis*. Online Document. 2025. – URL `https://fd-research.com/wp-content/uploads/2025/05/writing-a-thesis-1.pdf`

[5] Dubner, Stephen J.: *Why Is There So Much Fraud in Academia?* Freakonomics Radio Podcast. 1 2024. – URL `https://freakonomics.com/podcast/why-is-there-so-much-fraud-in-academia/`. – Episode 572

[6] Jürgens, Elmar: *Thesis Guide*. 2018. – URL `https://thesisguide.org`

[7] Kolda, Tamara G.: Taming the Chaos of Computational Experiments. In: *Siam News* 58 (2025), Nr. 7. – URL `https://www.siam.org/publications/siam-news/articles/taming-the-chaos-of-computational-experiments/`

[8] LeVeque, Randall J.: Top ten reasons to not share your code (and why you should anyway). In: *Siam News* 46 (2013), Nr. 3. – URL `https://www.siam.org/publications/siam-news/articles/top-ten-reasons-to-not-share-your-code-and-why-you-should-anyway`

[9] Martin, Robert C.: *Clean code: a handbook of agile software craftsmanship*. Pearson Education, 2009. – URL `https://github.com/Gatjuat-Wicteat-Riek/clean-code-book`

[10] McGreivy, Nick ; Hakim, Ammar: Weak baselines and reporting biases lead to overoptimism in machine learning for fluid-related partial differential equations. In: *Nature machine intelligence* 6 (2024), Nr. 10, S. 1256–1269