New Researcher Handbook

A Practical Guide for Early-Career PhD Students and Researchers

Xiaolong Luo
Harvard University
School of Engineering and Applied Sciences

August 26, 2025

Contents

1 Introduction

1.1 Why This Manual Matters

Dear students,

I wish I had seen this manual when I was an undergraduate student, or when I just started my PhD journey. It would have saved me from countless detours, mistakes, and unnecessary struggles. This guide contains the lessons I learned the hard way—through trial and error, late-night debugging sessions, rejected papers, and awkward presentations.

Looking back, I realize how much time and energy I could have saved if someone had told me:

- The importance of version control (I lost weeks of work because I didn't use Git properly)
- How to structure presentations (my first conference talk was a disaster—no clear message, too much detail)
- The value of documentation (I couldn't understand my own code from 6 months earlier)
- How to ask good questions in meetings (I stayed silent for months, missing valuable learning opportunities)

These aren't just rules—they're shortcuts to success that I discovered too late:

- Clear communication skills will help you share your ideas effectively
- Rigorous research habits will make you a trusted and reliable researcher
- Professional presentation skills will open doors to opportunities
- Good documentation practices will save you countless hours in the future
- Collaborative skills will make you a valued team member anywhere you go

As a PhD candidate in my fourth year, I'm still learning these lessons myself. Every guideline in this manual comes from mistakes I've made or witnessed firsthand. When I emphasize the importance of presentation skills, it's because I've missed many opportunities and left poor impressions due to bad presentations. When I emphasize documentation, it's because I've personally struggled to recreate my own experiments from months earlier.

The habits we develop now will serve us when we're:

- Presenting at international conferences (where we have one shot to impress)
- Defending our theses (where clarity can make or break the defense)
- Interviewing for our dream jobs (where these skills set us apart)
- Eventually leading our own research teams (where we'll pass on these lessons)
- Pitching ideas to investors or stakeholders (where professionalism matters most)

My intention is simple: By sharing what I've learned so far in my PhD journey, I hope we can learn from each other's experiences. This manual isn't written from a position of mastery—it's a collection of lessons I'm still learning, mistakes I'm still making, and habits I'm still building. Let's navigate this journey together and help each other become the researchers we aspire to be.

1.2 How We Build Research Excellence

Beyond the technical skills, remember that excellence in research comes from:

- **Frequent feedback from each other**: Don't work in isolation—share your work early and often
- **Reflecting on our personal and research practices**: Take time to think about what's working and what isn't
- **Getting enough sleep so we can bring our best selves to our work**: Exhausted researchers make poor decisions and write buggy code

These principles might sound simple, but I've seen too many brilliant students burn out because they ignored them. Your best work comes when you're well-rested, connected with your community, and continuously learning from both successes and failures.

2 Core Principles

2.1 Clarity Above All

Every presentation should prioritize clarity. Your audience should understand:

- The problem you're solving
- Your approach
- Your results and their implications

2.2 Consistency Matters

Use consistent formatting, terminology, and structure throughout your presentations and reports.

2.3 Respect Your Audience

Remember that your audience's time is valuable. Be prepared, be concise, and be engaging.

3 Presentation Standards

3.1 Structure

Every presentation should follow this structure:

1. **Title Slide**: Include title, your name, date, and affiliation
2. **Outline**: Brief overview of what you'll cover
3. **Background/Motivation**: Why this work matters
4. **Problem Statement**: Clear definition of what you're solving
5. **Methodology**: Your approach
6. **Results**: What you found
7. **Discussion**: What it means
8. **Future Work**: Next steps
9. **Acknowledgments**: Credit collaborators and funding

3.2 Slide Design Guidelines

- **6×6 Rule**: Maximum 6 bullet points with 6 words each
- **Font Size**: Minimum 24pt for body text, 32pt for titles
- **Color Scheme**: Use high contrast; avoid red-green combinations
- **One Message Per Slide**: Each slide should convey one main idea

4 Experimental Results Presentation

Critical Requirements

Before presenting ANY experimental results:

1. Clearly explain the experimental setting
2. Define all variables and parameters
3. State your hypotheses explicitly

Before showing ANY figure:

1. Explain what the figure shows
2. Describe the X and Y axes clearly
3. Read the caption aloud
4. Point out key trends or findings

4.1 Experimental Setup Checklist

When presenting experiments, always include:

- **Dataset**: Size, source, preprocessing steps
- **Baselines**: What you're comparing against
- **Metrics**: How you measure success
- **Hyperparameters**: All relevant settings
- **Statistical Significance**: Error bars, p-values when applicable

4.2 Figure Guidelines

4.2.1 Essential Elements

Every figure MUST have:

- Clear, descriptive title and Labeled axes with units
- Legend (if multiple data series)
- Caption explaining what's shown
- Appropriate scale and range

- Use vector graphics (PDF, SVG) when possible
- Include error bars or confidence intervals

4.2.2 Scientific Color Schemes

Professional Color Palettes

Recommended Scientific Color Schemes:

1. Sequential (Single Hue): For ordered data

- **Blues:**
- **Viridis:**
- **Cividis:**

2. Diverging: For data with meaningful midpoint

- **RdBu (Red-Blue):**
- **PuOr (Purple-Orange):**
- **BrBG (Brown-Green):**

3. Categorical: For discrete categories

- **Set1:**
- **Okabe-Ito (Colorblind-safe):**
- **Dark2:**

Color Selection Resource:

- Best Color Palettes for Scientific Figures and Data Visualizations - Comprehensive guide for scientific color selection

5 Research Habits for CS & AI

5.1 Daily Practices

5.1.1 Code and Experiment Management

- **Version Control**: Commit code daily with meaningful messages
- **Documentation**: Document code as you write it, not after
- **Reproducibility**: Always set random seeds and log all parameters
- **Backup**: Use git and cloud storage; never trust a single copy

5.1.2 Time Management

- **Deep Work Blocks**: Reserve 2-4 hour blocks for focused research
- **Experimentation**: Run experiments overnight/weekends when possible
- **Writing**: Write a little every day, even just notes

5.2 Research Methodology

5.2.1 Literature Review

- **Systematic Search**: Use Google Scholar, arXiv, and conference proceedings
- **Latest Papers Hub**: Hugging Face Papers - Excellent source for discovering the latest high-quality ML papers with community discussions
- **Quality Assessment**: When selecting papers, always note:
  - **Institution**: Which university or research lab produced the work
  - **Publication Year**: How recent is the research (prioritize recent work for current trends)
  - **Venue**: Where was it published (top-tier conferences/journals have rigorous peer review)
  - **Impact**: Citation count and influence in the field
- **Prioritize High-Quality Sources**:
  - Top conferences: NeurIPS, ICML, ICLR, ACL, CVPR, AAAI, IJCAI
  - Top journals: Nature, Science, JMLR, IEEE TPAMI, TACL
  - Reputable institutions with strong track records in your area
- **Note-Taking**: Summarize key ideas, methods, and limitations
- **Stay Current**: Follow top conferences (NeurIPS, ICML, ACL, CVPR, etc.)

5.2.2 Experiment Design

- **Start Simple**: Begin with baseline implementations
- **One Variable**: Change one thing at a time
- **Hypothesis-Driven**: Always have a clear hypothesis before running
- **Negative Results**: Document what doesn't work—it's valuable!

5.3 Collaboration Best Practices

5.3.1 Communication

- **Regular Updates**: Weekly progress reports to advisor
- **Ask Questions Early**: Don't struggle alone for days
- **Share Drafts**: Get feedback on writing early and often
- **Meeting Prep**: Come with agenda and specific questions

5.3.2 Code Collaboration

- **Code Reviews**: Request reviews for significant changes
- **Documentation**: Write README files for every project
- **Modular Design**: Write reusable, testable code
- **Shared Standards**: Follow team coding conventions

5.3.3 Active Participation in Group Meetings

Why Active Participation Matters

Asking questions in group meetings is not just encouraged—it's essential for your growth as a researcher. Here's why:

Benefits for You:

- **Deeper Understanding**: Questions help clarify concepts you might have misunderstood
- **Build Connections**: Your questions can spark collaborations and discussions
- **Develop Critical Thinking**: Learning to ask good questions is a crucial research skill
- **Show Engagement**: Active participation demonstrates your interest and commitment
- **Learn from Others**: Often, your peers have the same questions but hesitate to ask
- **Practice Communication**: Formulating clear questions improves your ability to communicate research ideas

Common Types of Questions to Ask:

5.4 Professional Development

5.4.1 Skills to Develop

- **Programming**: Python, PyTorch/TensorFlow, Git
- **Mathematics**: Linear algebra, probability, optimization
- **Writing**: Technical writing, LaTeX
- **Presentation**: Public speaking, visualization

6 Report Writing Standards

6.1 Structure

Follow the standard scientific paper structure:

1. Abstract (150-250 words)
2. Introduction
3. Related Work
4. Methodology
5. Experiments
6. Results and Discussion
7. Conclusion
8. References

Question Type	Example Questions
Clarification Questions	• "Could you explain what you mean by [technical term]?" • "How does this differ from [related concept]?" • "Can you walk through that derivation/algorithm again?"
Methodology Questions	• "Why did you choose this particular approach?" • "Have you considered [alternative method]?" • "How did you handle [specific challenge]?" • "What are the computational requirements?"
Results and Interpretation	• "What's the statistical significance of these results?" • "How do these results compare to the baseline?" • "What do you think explains this unexpected outcome?" • "Have you tested this on [different dataset/scenario]?"
Broader Impact Questions	• "How does this relate to [other work in the field]?" • "What are the practical applications of this research?" • "What are the limitations of this approach?" • "What ethical considerations should we keep in mind?"
Future Work Questions	• "What are the next steps for this project?" • "How could this be extended to [related problem]?" • "What would you do differently if starting over?" • "What's the biggest challenge you foresee?"

Table 1: Common types of questions to ask during group meetings

6.2 Common Mistakes to Avoid

Don't Do This!

1. **Wall of Text**: Never fill slides with paragraphs
2. **Unclear Axes**: Always label axes before discussing graphs
3. **Missing Context**: Never jump to results without setup
4. **Unreadable Fonts**: Test visibility from the back of the room
5. **No Rehearsal**: Always practice your timing
6. **Ignoring Questions**: Listen carefully and answer what was asked

7 Weekly Update Format

7.1 Structure

Your weekly updates should include:

1. **Progress**: What you accomplished
2. **Challenges**: What difficulties you encountered
3. **Plans**: What you'll do next week
4. **Questions**: What help you need

7.2 Best Practices

- Keep updates concise (5-10 slides)
- Include visual evidence of progress
- Be honest about challenges
- Come prepared with specific questions

8 How to be a Productive PhD Student

8.1 Strategic Research Framework

The Foundation of PhD Success

Becoming a productive PhD student requires more than just hard work—it demands strategic thinking, systematic organization, and deliberate practice. This section synthesizes best practices from successful researchers worldwide.

8.1.1 Clear Goals and Structured Output Paths

Define Your North Star:

- **Long-term Vision**: What impact do you want to make in your field?
- **3-Year Plan**: Break down your PhD into yearly milestones
- **Quarterly Goals**: Concrete deliverables (papers, code, experiments)
- **Weekly Targets**: Specific tasks that move you toward quarterly goals

Output-Driven Approach:

1. **Papers**: Target 2-3 quality publications per year
2. **Code**: Build reusable research tools and frameworks
3. **Presentations**: Regular practice at lab meetings and conferences
4. **Collaborations**: Actively engage in 1-2 collaborative projects

8.1.2 Daily Action System

The PhD Daily Routine

Morning (Deep Work Block):

- 2-4 hours of uninterrupted research/coding
- No email, no meetings, no distractions
- Focus on your most important task

Afternoon (Collaborative Work):

- Meetings, discussions, code reviews
- Email and administrative tasks
- Reading papers and staying current

Evening (Reflection and Planning):

- Document daily progress in research log
- Plan next day's priorities
- Light reading or skill development

Notion-Based Management System:

- **Research Dashboard**: Track all projects, deadlines, and progress
- **Literature Database**: Organize papers with tags, notes, and connections
- **Experiment Log**: Document all experiments with parameters and results
- **Idea Capture**: Quick notes for research ideas and insights
- **Weekly Reviews**: Reflect on progress and adjust plans

8.2 Top Conferences by Field

8.2.1 AI/Machine Learning Conferences

Conference	Full Name	Focus Area	Website
Core Machine Learning (Tier 1)			
NeurIPS	Neural Information Processing Systems	General ML	neurips.cc
ICML	International Conference on Machine Learning	ML Theory	icml.cc
ICLR	International Conference on Learning Representations	Deep Learning	iclr.cc
Computer Vision			
CVPR	Computer Vision and Pattern Recognition	Vision	cvpr.thecvf.com
ICCV	International Conference on Computer Vision	Vision	thecvf.com
ECCV	European Conference on Computer Vision	Vision	eccv.ecva.net
Natural Language Processing			
ACL	Association for Computational Linguistics	NLP	aclanthology.org
EMNLP	Empirical Methods in NLP	NLP	emnlp.org
NAACL	North American Chapter of ACL	NLP	naacl.org
General AI			
AAAI	Association for Advancement of AI	General AI	aaai.org
IJCAI	International Joint Conference on AI	General AI	ijcai.org
Data Mining and Web			
KDD	Knowledge Discovery and Data Mining	Data Mining	kdd.org
WWW	The Web Conference	Web/IR	thewebconf.org

Table 2: Top-tier AI/ML conferences with submission deadlines typically 4-6 months before the conference

8.2.2 Computer Science Conferences

Conference	Full Name	Area	Website
Databases			
SIGMOD	Special Interest Group on Management of Data	Databases	sigmod.org
VLDB	Very Large Data Bases	Databases	vldb.org
Systems			
OSDI	Operating Systems Design and Implementation	Systems	usenix.org
SOSP	Symposium on Operating Systems Principles	Systems	sosp.org
Graphics			
SIGGRAPH	Special Interest Group on Graphics	Graphics	siggraph.org
Human-Computer Interaction			
CHI	Conference on Human Factors in Computing	HCI	chi.acm.org
UIST	User Interface Software and Technology	HCI	uist.acm.org
Security			
S&P	IEEE Symposium on Security and Privacy	Security	ieee-security.org
CCS	Computer and Communications Security	Security	sigsac.org
USENIX	USENIX Security Symposium	Security	usenix.org
Programming Languages			
PLDI	Programming Language Design and Implementation	PL	pldi.sigplan.org
POPL	Principles of Programming Languages	PL	popl.mpi-sws.org

Table 3: Top-tier Computer Science conferences across different research areas

8.2.3 AI for Science Conferences and Venues

The Rising Field of AI4Science

AI for Science is an emerging interdisciplinary field where machine learning meets traditional sciences. These venues offer unique opportunities to publish work that bridges AI and scientific domains.

Venue	Type	Website/Info
Workshops at Major ML Conferences		
AI4Science @ NeurIPS	Workshop	ai4sciencecommunity.github.io
AI4Science @ ICLR	Workshop	ai4sciencecommunity.github.io
AI4Science @ ICML	Workshop	ai4sciencecommunity.github.io
ML4Physical Sciences @ NeurIPS	Workshop	Annual at NeurIPS
Computational Biology @ ICML	Workshop	Annual at ICML
Journals		
Nature Machine Intelligence	Journal	nature.com/natmachintell
Science Robotics	Journal	science.org
Digital Discovery	Journal	RSC Publishing
npj Computational Materials	Journal	nature.com
Physical Review X	Journal	aps.org
Specialized Conferences		
MLCB	Machine Learning in Comp Bio	mlsb.io
ISMB/ECCB	Bioinformatics	iscb.org
AIChE	Chemical Engineering + AI	aiche.org

Table 4: AI for Science publication venues combining machine learning with scientific disciplines

8.3 Publication Strategy

8.3.1 Conference Timeline Management

Critical: Plan Your Submissions 6 Months Ahead

Most top conferences have deadlines 5-6 months before the conference date. Missing a deadline means waiting a full year for the next opportunity!

Typical Timeline for a Paper Submission:

- **T-6 months**: Start experiments and writing
- **T-3 months**: Complete main experiments
- **T-1 month**: Finish first draft, get feedback
- **T-1 week**: Polish, proofread, prepare supplementary
- **T-0**: Submit and celebrate!

8.3.2 Quality vs. Quantity Balance

The Publication Pyramid:

1. **Flagship Paper** (1 per year): Your best work targeting top venues (NeurIPS, ICML, CVPR)
2. **Solid Contributions** (1-2 per year): Good work at specialized conferences
3. **Workshop Papers** (2-3 per year): Early-stage ideas, work-in-progress

8.3.3 Collaboration Guidelines

Authorship Best Practices:

- **First Author**: Lead the project, write majority of paper
- **Contributing Author**: Significant experiments or sections
- **Advisory Author**: Guidance and feedback
- Always discuss authorship order early in the project
- Document contributions for each paper

8.3.4 Strategic Paper Planning: Key Points to Think Through

Before You Start Writing

A successful paper isn't just about good results—it's about clear thinking. Before writing a single word, ensure you can answer these strategic questions about your work.

I. Paper Core Elements
1. Main Contribution:

- What is my core innovation? Is it theoretical innovation, algorithm design, system optimization, application scenario, or analysis/unification of existing methods?

- Can I explain in one sentence: "We are the first to solve X important problem"?
- Do I have multi-level contributions? (theory + practice; new method + new task; new analysis + new benchmark)

2. Results Presentation Strategy:

- Which experiments are **must-have** core comparisons?
- Which are **supporting** ablation studies / case studies / visualizations?
- Do I need multiple benchmarks (across vision/NLP) to show generality?
- Can I achieve SOTA with fewer resources or higher efficiency, or significantly outperform on specific data/tasks?

II. Pre-Writing Strategic Questions
1. Topic Positioning:

- Is my topic hot at this conference this year? Is it oversaturated?
- Does it align with emerging trends? (e.g., "Multimodal LLM", "efficient finetuning", "long context modeling", "AI4Science")
- Does it match past hot sessions/workshops/invited talks at the venue?

2. Innovation Assessment:

- How is my work clearly different from the past year's related work? (not just a simple tweak)
- If it's combinatorial innovation (A+B), is it natural, effective, and insightful?
- Do I have a unique perspective or counter-intuitive but effective discovery?

3. Method vs Insight Framing:

- Is my method just a "reasonable trick" or does it reveal a new phenomenon/structure/principle?
- Which framing is better: a new model? a new phenomenon? a more general framework?
- Is it worth abstracting as "xxx is all you need", "rethinking xxx", or "beyond xxx"?

4. Dataset and Task Design:

- Am I proposing a representative new benchmark or task?
- If using existing data, have I chosen the right entry point/task definition/evaluation metrics?
- Have I proposed a compelling motivation from the application perspective? (e.g., "current models still fail in scenario X")

5. Reproducibility and Generalizability:

- Can others easily reproduce my work? (crucial for NeurIPS/ICLR)
- Can the method generalize to other domains?

- Have I done sufficient general analysis or interpretability experiments to support my claims?

III. Strategic and Risk Considerations
1. Maturity vs First-Mover:

- Is my work better suited for "careful polishing and mature submission" or "submit first round, iterate quickly"?
- Should I post on arXiv first to establish timeline, or wait for a complete version?

2. Submission Roadmap:

- Should I use workshop/spotlight/poster/oral strategy to familiarize with the field community first?
- If rejected this round, what's my backup venue? (ICML → NeurIPS → ICLR → domain-specific conferences)
- How does this paper fit into my overall PhD publication strategy?

Quick Checklist Before Starting:

- ☐ Can I explain my contribution in one sentence?
- ☐ Do I have a clear experimental validation plan?
- ☐ Have I checked recent papers (last 6 months) to ensure novelty?
- ☐ Is my framing aligned with conference trends?
- ☐ Do I have a backup plan if rejected?

8.4 Productivity Systems and Tools

8.4.1 Research Pipeline Management

```
Idea Generation -> Literature Review -> Hypothesis Formation
        |                                       |
        v                                       v
Experiment Design <- - - - - - - - - ->  Initial Results
        |                                       |
        v                                       v
Full Experiments -> Analysis -> Writing -> Submission
        |              |           |           |
        v              v           v           v
    [Track]        [Track]     [Track]    [Track]
```

Tools for Each Stage:

- **Idea Generation**: Notion, Obsidian, Research diary
- **Literature Review**: Zotero, Mendeley, Connected Papers
- **Experiments**: Weights & Biases, MLflow, Sacred
- **Writing**: LaTeX + Git, Overleaf for collaboration
- **Project Management**: GitHub Projects, Trello, Asana

8.4.2 Time Management Strategies

The 80/20 Rule for PhD Productivity

Focus 80% of your time on:

- Core research that directly contributes to papers
- Writing and improving your communication skills
- Building deep expertise in your specific area

Limit to 20% of your time:

- Attending talks outside your direct area
- Administrative tasks and service
- Exploring tangentially related topics

8.4.3 Mental Health and Sustainability

Maintaining Long-term Productivity:

- **Regular Breaks**: Take weekends off, use vacation time
- **Physical Health**: Exercise 3-4 times per week
- **Social Connections**: Maintain relationships outside research
- **Hobbies**: Keep non-academic interests alive
- **Support Network**: Build relationships with peers and mentors

Remember: PhD is a Marathon, Not a Sprint

Sustainable productivity beats intense bursts followed by burnout. Build systems that support consistent, long-term progress rather than relying on heroic efforts.

8.5 Academic Social Networking

8.5.1 Building Your Research Network

Conference Networking Strategy:

1. **Before the Conference**:
   - Review accepted papers and identify interesting authors
   - Prepare your elevator pitch (30 seconds about your research)
   - Schedule meetings with researchers you want to meet
2. **During the Conference**:
   - Attend poster sessions—easiest place to meet people
   - Ask thoughtful questions during Q&A
   - Join social events and workshops

   - Exchange contact information and follow up

3. **After the Conference**:

   - Send follow-up emails within one week
   - Share relevant papers or resources
   - Consider collaboration opportunities
   - Stay connected via social media/email

8.5.2 Online Presence

Essential Platforms for Researchers:

- **Google Scholar**: Track citations and create your profile
- **Twitter/X**: Follow researchers, share papers, join discussions
- **LinkedIn**: Professional networking and job opportunities
- **GitHub**: Share code and collaborate on projects
- **Personal Website**: Portfolio of your work and research statement

8.6 Essential Reading for PhD Students

Must-Read Books and Articles

These readings provide invaluable insights into the PhD journey, research life, and academic career development. Each has helped thousands of students navigate their doctoral studies more effectively.

8.6.1 Core PhD Experience

- **The PhD Grind by Philip Guo**
  - A candid memoir of one student's six-year journey through a computer science PhD at Stanford
  - Provides honest insights into the struggles, failures, and eventual successes
  - Essential reading for understanding the emotional and practical realities of PhD life
  - Available free online at `http://linyun.info/phd-grinding.pdf`

8.6.2 Career and Professional Development

- **A PhD Is Not Enough!** by Peter J. Feibelman
  - Guide to building a successful scientific career after PhD
  - Covers networking, job hunting, grant writing, and career planning
  - Practical advice on transitioning from student to professional researcher
- **Getting What You Came For** by Robert Peters
  - Comprehensive guide to choosing and succeeding in graduate school
  - Covers everything from selecting advisors to defending your dissertation

    - Includes strategies for time management and maintaining mental health

- **Deep Work** by Cal Newport
    - Strategies for focused work in a distracted world
    - Particularly relevant for research and writing
    - Techniques for maximizing cognitive capabilities
- **The 7 Habits of Highly Effective People** by Stephen Covey
    - Time management and personal effectiveness
    - Principles applicable to research and collaboration
    - Framework for long-term success and life balance

8.6.3 Online Resources and Blogs

- **The Illustrated Guide to a PhD** by Matt Might
    - Visual explanation of what a PhD really means
    - Helps maintain perspective during challenging times
    - Widely shared and appreciated by PhD students worldwide
- **PhD Comics** by Jorge Cham
    - Humorous take on graduate student life
    - Provides comic relief and community feeling
    - Addresses common PhD experiences and challenges
- **Nature Career Columns**
    - Regular articles on research careers and PhD life
    - Advice from established researchers
    - Current perspectives on academic job market
- **Academia Stack Exchange**
    - Q&A platform for academic life questions
    - Community wisdom on common PhD challenges
    - Practical solutions to specific problems

Reading Recommendation

Start with **The PhD Grind**—it's short, free, and will give you realistic expectations about the PhD journey. Read it before starting your PhD or during your first year to better understand what lies ahead.

A Presentation Checklist

Before every presentation, verify:

- ☐ Title slide has all required information
- ☐ Outline matches actual content
- ☐ All figures have labeled axes
- ☐ All figures have descriptive captions
- ☐ Experimental setup is clearly explained
- ☐ Results are connected to hypotheses
- ☐ Acknowledgments are included
- ☐ Presentation is within time limit
- ☐ Slides are numbered
- ☐ References are properly cited

B Figure Checklist

For every figure, ensure:

- ☐ Title clearly states what's shown
- ☐ X-axis is labeled with units
- ☐ Y-axis is labeled with units
- ☐ Legend explains all symbols/colors
- ☐ Font size is readable when projected
- ☐ Color scheme is accessible
- ☐ Error bars/confidence intervals shown
- ☐ Caption explains key findings

C Research Tools and Resources

C.1 Recommended Software

Category	**Tool**	**Purpose**
Version Control	Git, GitHub	Code management
Writing	LaTeX (Overleaf), Markdown	Documents and papers
Data Visualization	Matplotlib, Seaborn, Plotly	Creating figures
Diagram Tools	Mermaid, PlantUML	Flowcharts and diagrams
Note Taking	Notion, Obsidian, Roam	Research notes

Table 5: Essential research tools for CS and AI researchers

C.2 Terminal Setup Recommendation

Upgrade Your Terminal Experience: iTerm2 + Oh My Zsh

For a significantly improved development experience, we strongly recommend replacing the default macOS Terminal with:

- **iTerm2**: A powerful terminal emulator
- **zsh**: Modern shell (default on macOS)
- **Oh My Zsh**: Framework for managing zsh configuration
- **Powerlevel10k**: Beautiful and informative theme

Key Advantages:

- The terminal interface looks amazing — it shows path, time, git status, conda environment, etc. all by default
- Supports tabs and pane splits, so you don't need to open multiple windows
- Smooth syntax highlighting and auto-completion — feels super nice

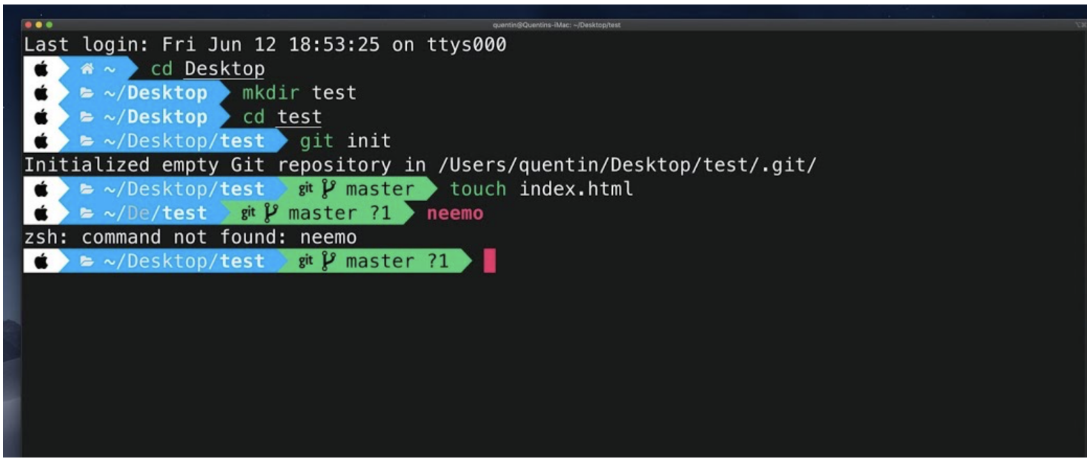

Figure 1: iTerm2 with Powerlevel10k theme showing git status, conda environment, and more

C.3 Visualization and Diagramming Tools

Advanced Tools for Research Visualization

C.3.1 Excalidraw

Website: https://excalidraw.com/

Excalidraw is a virtual whiteboard tool that enables you to create beautiful hand-drawn style diagrams effortlessly:

- **Perfect for:** Quick architectural diagrams, flowcharts, and conceptual illustrations
- **Key Features:**
  - Hand-drawn aesthetic that's perfect for informal presentations
  - Real-time collaboration for team brainstorming
  - Export to PNG, SVG, and clipboard

- Works entirely in the browser — no installation needed

- **Research Applications:**
  - System architecture sketches during design phase
  - Algorithm flowcharts for presentations
  - Brainstorming session visualizations
  - Quick diagrams for group meetings

C.3.2 PlotNeuralNet

Repository: https://github.com/HarisIqbal88/PlotNeuralNet

PlotNeuralNet is a LaTeX-based tool specifically designed for creating publication-quality neural network architecture diagrams:

- **Perfect for:** Research papers, thesis figures, and technical presentations
- **Key Features:**
  - Creates professional 3D neural network visualizations
  - Fully customizable layers and connections
  - Consistent styling across all diagrams
  - Generates LaTeX/TikZ code for easy integration
- **Research Applications:**
  - CNN/RNN/Transformer architecture illustrations
  - Paper and thesis figures that meet publication standards
  - Technical documentation of model architectures
  - Poster presentations at conferences
- **Pro Tip:** Start with the provided examples and modify them to match your specific architecture — it's much easier than building from scratch!

C.4 System Resource Monitoring and Performance Optimization

Essential Skills for Efficient Development

As researchers working with data processing and deep learning, monitoring system resources and utilizing parallel processing are critical skills that can dramatically improve your productivity and code efficiency.

C.4.1 System Resource Monitoring

Why Monitor Resources?

- Identify bottlenecks in data processing pipelines
- Detect memory leaks and inefficient code early
- Optimize GPU utilization for deep learning tasks
- Prevent system crashes from resource exhaustion
- Validate that parallel processing is working effectively

Essential Monitoring Tools:

- **htop**: Interactive process viewer showing real-time CPU usage per core
- **nvtop**: GPU monitoring for NVIDIA cards (GPU utilization, memory, temperature)
- **gpustat**: Simple command-line GPU status viewer
- **btop**: Modern resource monitor with beautiful graphs and real-time updates
- **nvidia-smi**: Built-in NVIDIA GPU monitoring tool

C.4.2 Parallel Processing for Efficiency

Serial vs Parallel Processing Comparison:

```
Serial Processing:      [Task 1] -> [Task 2] -> [Task 3] -> [Task 4]
                       Time: ================================

Parallel Processing:    [Task 1] [Task 2]
                       [Task 3] [Task 4]
                       Time: ================

Result: ~4x speed improvement (ideal case)
```

Performance Comparison Example:

Task	Serial Time	Parallel Time (4 cores)	Speed-up
Data preprocessing	100s	28s	3.6x
Model training (CPU)	240s	65s	3.7x
Batch inference	60s	18s	3.3x
Image processing	180s	48s	3.8x

Table 6: Typical performance gains from parallelization

Practical Implementation Examples:

- **Python**: Use `multiprocessing`, `concurrent.futures`, or `joblib`
- **NumPy/Pandas**: Vectorized operations instead of loops
- **Deep Learning**: Batch processing, DataLoader with `num_workers > 0`
- **Data Processing**: Transform `map()` → `parallel_map()`

C.4.3 Best Practices and Monitoring Guidelines

What to Monitor During Development:

- **CPU Usage**: Aim for ¿80% utilization across all cores when parallel processing
- **Memory Usage**: Watch for memory leaks; avoid swapping to disk
- **GPU Utilization**: Should be ¿90% during deep learning training
- **I/O Wait**: High I/O wait indicates data loading bottlenecks

Common Parallel Processing Pitfalls:

- I/O bottlenecks limiting parallel gains
- Python's GIL (use multiprocessing instead of threading for CPU tasks)
- Overhead from creating too many small parallel tasks
- Insufficient memory causing thrashing

Quick Reference Commands:

```
# Monitor CPU usage
htop                    # Interactive CPU monitor
top -u                 # Alternative CPU monitor

# Monitor GPU (NVIDIA)
nvidia-smi -l 1        # Updates every 1 second
nvtop                  # Interactive GPU monitor
gpustat -i 1           # Simple GPU status

# Python parallel example
from multiprocessing import Pool
with Pool() as p:
    results = p.map(process_func, data_list)
```

Pro Tips:

- Always benchmark before and after implementing parallelization
- Monitor resources during the ENTIRE training process, not just at the start
- For deep learning: use `DataLoader(num_workers=4-8)` for faster data loading
- Check if your bottleneck is CPU, GPU, or I/O before optimizing

AI Disclosure

Some text and links in this document were compiled with AI assistance. Please forgive any broken links.