

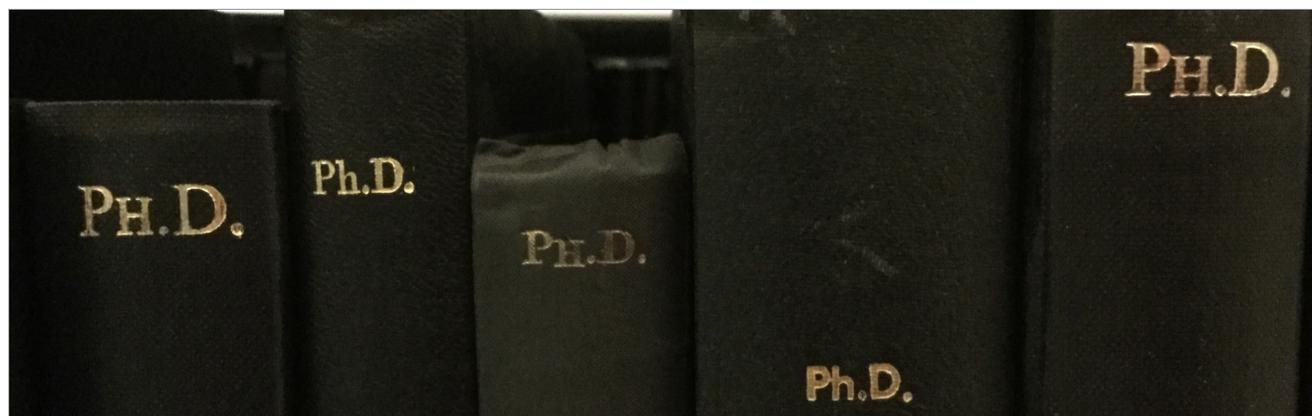


A Survival Guide to a PhD

Sep 7, 2016

This guide is patterned after my [“Doing well in your courses”](#), a post I wrote a long time ago on some of the tips/tricks I’ve developed during my undergrad. I’ve received nice comments about that guide, so in the same spirit, now that my PhD has come to an end I wanted to compile a similar retrospective document in hopes that it might be helpful to some. Unlike the undergraduate guide, this one was much more difficult to write because there is significantly more variation in how one can traverse the PhD experience. Therefore, many things are likely contentious and a good fraction will be specific to what I’m familiar with (Computer Science / Machine Learning / Computer Vision research). But disclaimers are boring, lets get to it!

Preliminaries



First, should you want to get a PhD? I was in a fortunate position of knowing since young age that I really wanted a PhD. Unfortunately it wasn’t for any very well-thought-through considerations: First, I really liked school and learning things and I wanted to learn as much as possible, and second, I really wanted to be like [Gordon Freeman](#) from the game Half-Life (who has a PhD from MIT in theoretical physics). I loved that game. But what if you’re more sensible in making your life’s decisions? Should you want to do a PhD? There’s a very nice [Quora thread](#) and in the summary of considerations that follows I’ll borrow/restate several from Justin/Ben/others there. I’ll assume that the second option you are considering is joining a medium-large company (which is likely most common). Ask yourself if you find the following

properties appealing:

Freedom. A PhD will offer you a lot of freedom in the topics you wish to pursue and learn about. You're in charge. Of course, you'll have an adviser who will impose some constraints but in general you'll have much more freedom than you might find elsewhere.

Ownership. The research you produce will be yours as an individual. Your accomplishments will have your name attached to them. In contrast, it is much more common to "blend in" inside a larger company. A common feeling here is becoming a "cog in a wheel".

Exclusivity. There are very few people who make it to the top PhD programs. You'd be joining a group of a few hundred distinguished individuals in contrast to a few tens of thousands (?) that will join some company.

Status. Regardless of whether it should be or not, working towards and eventually getting a PhD degree is culturally revered and recognized as an impressive achievement. You also get to be a Doctor; that's awesome.

Personal freedom. As a PhD student you're your own boss. Want to sleep in today? Sure. Want to skip a day and go on a vacation? Sure. All that matters is your final output and no one will force you to clock in from 9am to 5pm. Of course, some advisers might be more or less flexible about it and some companies might be as well, but it's a true first order statement.

Maximizing future choice. Joining a PhD program doesn't close any doors or eliminate future employment/lifestyle options. You can go one way (PhD -> anywhere else) but not the other (anywhere else -> PhD -> academia/research; it is statistically less likely). Additionally (although this might be quite specific to applied ML), you're strictly more hireable as a PhD graduate or even as a PhD dropout and many companies might be willing to put you in a more interesting position or with a higher starting salary. More generally, maximizing choice for the future you is a good heuristic to follow.

Maximizing variance. You're young and there's really no need to rush. Once you graduate from a PhD you can spend the next ~50 years of your life in some company. Opt for more variance in your experiences.

Personal growth. PhD is an intense experience of rapid growth (you learn a lot) and personal self-discovery (you'll become a master of managing your own psychology). PhD programs (especially if you can make it into a good one) also offer a *high density* of exceptionally bright people who will become your best friends forever.

Expertise. PhD is probably your only opportunity in life to really drill deep into a topic and become a recognized leading expert *in the world* at something. You're exploring the edge of our knowledge as a species, without the burden of lesser distractions or constraints. There's something beautiful about that and if you disagree, it could be a sign that PhD is not for you.

The disclaimer. I wanted to also add a few words on some of the potential downsides and failure modes. The PhD is a very specific kind of experience that deserves a large disclaimer. You will inevitably find yourself working very hard (especially before paper deadlines). You need to be okay with the suffering and have enough mental stamina and determination to deal with the pressure. At some points you will lose track of what day of the week it is and go on a diet of leftover food from the microkitchens. You'll sit exhausted and alone in the lab on a beautiful, sunny Saturday scrolling through Facebook pictures of your friends having fun on exotic trips, paid for by their 5-10x larger salaries. You will have to throw away 3 months of your work while somehow keeping your mental health intact. You'll struggle with the realization that months of your work were spent on a paper with a few citations while your friends do exciting startups with TechCrunch articles or push products to millions of people. You'll experience identity crises during which you'll question your life decisions and wonder what you're doing with some of the best years of your life. As a result, you should be quite certain that you can thrive in an unstructured environment in the pursuit research and discovery for science. If you're unsure you should lean slightly negative by default. Ideally you should consider getting a taste of research as an undergraduate on a summer research program before before you decide to commit. In fact, one of the primary reasons that research experience is so desirable during the PhD hiring process is not the research itself, but the fact that the student is more likely to know what they're getting themselves into.

I should clarify explicitly that this post is not about convincing anyone to do a PhD, I've merely tried to enumerate some of the common considerations above. The majority of this post focuses on some tips/tricks for navigating the experience once if you decide to go for it (which we'll see shortly, below).

Lastly, as a random thought I heard it said that you should only do a PhD if you want to go into academia. In light of all of the above I'd argue that a PhD has strong intrinsic value - it's an end by itself, not just a means to some end (e.g. academic job).

Getting into a PhD program: references, references, references. Great, you've decided to go for it. Now how do you get into a good PhD program? The first order approximation is quite simple - by far most important component are strong reference letters. The ideal scenario is that a well-known professor writes you a letter along the lines of: "Blah is in top 5 of students I've ever worked with. She takes initiative, comes up with her own ideas, and gets them to work." The worst letter is along the lines of: "Blah took my class. She did well." A research

publication under your belt from a summer research program is a very strong bonus, but not absolutely required provided you have strong letters. In particular note: grades are quite irrelevant but you generally don't want them to be too low. This was not obvious to me as an undergrad and I spent a lot of energy on getting good grades. This time should have instead been directed towards research (or at the very least personal projects), as much and as early as possible, and if possible under supervision of multiple people (you'll need 3+ letters!). As a last point, what won't help you too much is pestering your potential advisers out of the blue. They are often incredibly busy people and if you try to approach them too aggressively in an effort to impress them somehow in conferences or over email this may agitate them.

Picking the school. Once you get into some PhD programs, how do you pick the school? It's easy, join Stanford! Just kidding. More seriously, your dream school should 1) be a top school (not because it looks good on your resume/CV but because of feedback loops; top schools attract other top people, many of whom you will get to know and work with) 2) have a few potential advisers you would want to work with. I really do mean the "few" part - this is very important and provides a safety cushion for you if things don't work out with your top choice for any one of hundreds of reasons - things in many cases outside of your control, e.g. your dream professor leaves, moves, or spontaneously disappears, and 3) be in a good environment physically. I don't think new admits appreciate this enough: you will spend 5+ years of your really good years living near the school campus. Trust me, this is a long time and your life will consist of much more than just research.

Adviser



Image credit: [PhD comics](http://www.phdcomics.com).

Student adviser relationship. The adviser is an extremely important person who will exercise a lot of influence over your PhD experience. It's important to understand the nature of the

relationship: the adviser-student relationship is a symbiosis; you have your own goals and want something out of your PhD, but they also have their own goals, constraints and they're building their own career. Therefore, it is very helpful to understand your adviser's incentive structures: how the tenure process works, how they are evaluated, how they get funding, how they fund you, what department politics they might be embedded in, how they win awards, how academia in general works and specifically how they gain recognition and respect of their colleagues. This alone will help you avoid or mitigate a large fraction of student-adviser friction points and allow you to plan appropriately. I also don't want to make the relationship sound too much like a business transaction. The advisor-student relationship, more often than not, ends up developing into a lasting one, predicated on much more than just career advancement.

Pre-vs-post tenure. Every adviser is different so it's helpful to understand the axes of variations and their repercussions on your PhD experience. As one rule of thumb (and keep in mind there are many exceptions), it's important to keep track of whether a potential adviser is pre-tenure or post-tenure. The younger faculty members will usually be around more (they are working hard to get tenure) and will usually be more low-level, have stronger opinions on what you should be working on, they'll do math with you, pitch concrete ideas, or even look at (or contribute to) your code. This is a much more hands-on and possibly intense experience because the adviser will need a strong publication record to get tenure and they are incentivised to push you to work just as hard. In contrast, more senior faculty members may have larger labs and tend to have many other commitments (e.g. committees, talks, travel) other than research, which means that they can only afford to stay on a higher level of abstraction both in the area of their research and in the level of supervision for their students. To caricature, it's a difference between "you're missing a second term in that equation" and "you may want to read up more in this area, talk to this or that person, and sell your work this or that way". In the latter case, the low-level advice can still come from the senior PhD students in the lab or the postdocs.

Axes of variation. There are many other axes to be aware of. Some advisers are fluffy and some prefer to keep your relationship very professional. Some will try to exercise a lot of influence on the details of your work and some are much more hands off. Some will have a focus on specific models and their applications to various tasks while some will focus on tasks and more indifference towards any particular modeling approach. In terms of more managerial properties, some will meet you every week (or day!) multiple times and some you won't see for months. Some advisers answer emails right away and some don't answer email for a week (or ever, haha). Some advisers make demands about your work schedule (e.g. you better work long hours or weekends) and some won't. Some advisers generously support their students with equipment and some think laptops or old computers are mostly fine. Some advisers will fund you to go to a conferences even if you don't have a paper there

and some won't. Some advisers are entrepreneurial or applied and some lean more towards theoretical work. Some will let you do summer internships and some will consider internships just a distraction.

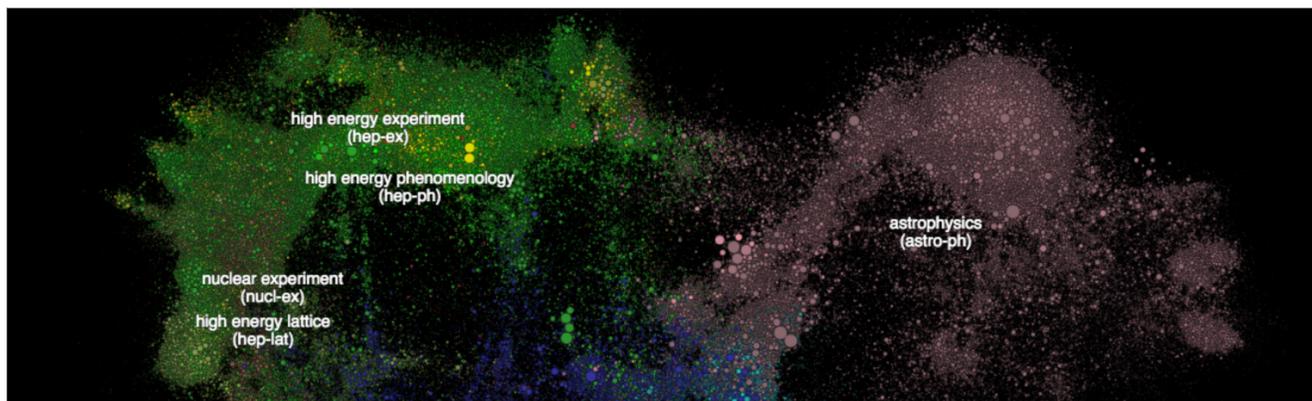
Finding an adviser. So how do you pick an adviser? The first stop, of course, is to talk to them in person. The student-adviser relationship is sometimes referred to as a marriage and you should make sure that there is a good fit. Of course, first you want to make sure that you can talk with them and that you get along personally, but it's also important to get an idea of what area of "professor space" they occupy with respect to the aforementioned axes, and especially whether there is an intellectual resonance between the two of you in terms of the problems you are interested in. This can be just as important as their management style.

Collecting references. You should also collect references on your potential adviser. One good strategy is to talk to their students. If you want to get actual information this shouldn't be done in a very formal way or setting but in a relaxed environment or mood (e.g. a party). In many cases the students might still avoid saying bad things about the adviser if asked in a general manner, but they will usually answer truthfully when you ask specific questions, e.g. "how often do you meet?", or "how hands on are they?". Another strategy is to look at where their previous students ended up (you can usually find this on the website under an alumni section), which of course also statistically informs your own eventual outcome.

Impressing an adviser. The adviser-student matching process is sometimes compared to a marriage - you pick them but they also pick you. The ideal student from their perspective is someone with interest and passion, someone who doesn't need too much hand-holding, and someone who takes initiative - who shows up a week later having done not just what the adviser suggested, but who went beyond it; improved on it in unexpected ways.

Consider the entire lab. Another important point to realize is that you'll be seeing your adviser maybe once a week but you'll be seeing most of their students every single day in the lab and they will go on to become your closest friends. In most cases you will also end up collaborating with some of the senior PhD students or postdocs and they will play a role very similar to that of your adviser. The postdocs, in particular, are professors-in-training and they will likely be eager to work with you as they are trying to gain advising experience they can point to for their academic job search. Therefore, you want to make sure the entire group has people you can get along with, people you respect and who you can work with closely on research projects.

Research topics



t-SNE visualization of a small subset of human knowledge (from [paperscape](#)). Each circle is an arXiv paper and size indicates the number of citations.

So you've entered a PhD program and found an adviser. Now what do you work on?

An exercise in the outer loop. First note the nature of the experience. A PhD is simultaneously a fun and frustrating experience because you're constantly operating on a meta problem level. You're not just solving problems - that's merely the simple inner loop. You spend most of your time on the outer loop, figuring out what problems are worth solving and what problems are ripe for solving. You're constantly imagining yourself solving hypothetical problems and asking yourself where that puts you, what it could unlock, or if anyone cares. If you're like me this can sometimes drive you a little crazy because you're spending long hours working on things and you're not even sure if they are the correct things to work on or if a solution exists.

Developing taste. When it comes to choosing problems you'll hear academics talk about a mystical sense of "taste". It's a real thing. When you pitch a potential problem to your adviser you'll either see their face contort, their eyes rolling, and their attention drift, or you'll sense the excitement in their eyes as they contemplate the uncharted territory ripe for exploration. In that split second a lot happens: an evaluation of the problem's importance, difficulty, its *sexiness*, its historical context (and possibly also its fit to their active grants). In other words, your adviser is likely to be a master of the outer loop and will have a highly developed sense of *taste* for problems. During your PhD you'll get to acquire this sense yourself.

In particular, I think I had a terrible taste coming in to the PhD. I can see this from the notes I took in my early PhD years. A lot of the problems I was excited about at the time were in retrospect poorly conceived, intractable, or irrelevant. I'd like to think I refined the sense by the end through practice and apprenticeship.

Let me now try to serialize a few thoughts on what goes into this sense of taste, and what makes a problem interesting to work on.

A fertile ground. First, recognize that during your PhD you will dive deeply into one area and your papers will very likely chain on top of each other to create a body of work (which becomes your thesis). Therefore, you should always be thinking several steps ahead when choosing a problem. It's impossible to predict how things will unfold but you can often get a sense of how much room there could be for additional work.

Plays to your adviser's interests and strengths. You will want to operate in the realm of your adviser's interest. Some advisers may allow you to work on slightly tangential areas but you would not be taking full advantage of their knowledge and you are making them less likely to want to help you with your project or promote your work. For instance, (and this goes to my previous point of understanding your adviser's job) every adviser has a "default talk" slide deck on their research that they give all the time and if your work can add new exciting cutting edge work slides to this deck then you'll find them much more invested, helpful and involved in your research. Additionally, their talks will promote and publicize your work.

Be ambitious: the sublinear scaling of hardness. People have a strange bug built into psychology: a 10x more important or impactful problem intuitively *feels* 10x harder (or 10x less likely) to achieve. This is a fallacy - in my experience a 10x more important problem is at most 2-3x harder to achieve. In fact, in some cases a 10x harder problem may be easier to achieve. How is this? It's because thinking 10x forces you out of the box, to confront the real limitations of an approach, to think from first principles, to change the strategy completely, to innovate. If you aspire to improve something by 10% and work hard then you will. But if you aspire to improve it by 100% you are still quite likely to, but you will do it very differently.

Ambitious but with an attack. At this point it's also important to point out that there are plenty of important problems that don't make great projects. I recommend reading [You and Your Research](#) by Richard Hamming, where this point is expanded on:

If you do not work on an important problem, it's unlikely you'll do important work. It's perfectly obvious. Great scientists have thought through, in a careful way, a number of important problems in their field, and they keep an eye on wondering how to attack them. Let me warn you, 'important problem' must be phrased carefully. The three outstanding problems in physics, in a certain sense, were never worked on while I was at Bell Labs. By important I mean guaranteed a Nobel Prize and any sum of money you want to mention. We didn't work on (1) time travel, (2) teleportation, and (3) antigravity. They are not important problems because we do not have an attack. It's not the consequence that makes a problem important, it is that you have a reasonable attack. That is what makes a problem important.

The person who did X. Ultimately, the goal of a PhD is to not only develop a deep expertise in a field but to also make your mark upon it. To steer it, shape it. The ideal scenario is that by the end of the PhD you own some part of an important area, preferably one that is also easy and fast to describe. You want people to say things like “she’s the person who did X”. If you can fill in a blank there you’ll be successful.

Valuable skills. Recognize that during your PhD you will become an expert at the area of your choosing (as fun aside, note that $[5 \text{ years}] \times [260 \text{ working days}] \times [8 \text{ hours per day}]$ is 10,400 hours; if you believe Gladwell then a PhD is exactly the amount of time to become an expert). So imagine yourself 5 years later being a world expert in this area (the 10,000 hours will ensure that regardless of the academic impact of your work). Are these skills exciting or potentially valuable to your future endeavors?

Negative examples. There are also some problems or types of papers that you ideally want to avoid. For instance, you’ll sometimes hear academics talk about “*incremental work*” (this is the worst adjective possible in academia). Incremental work is a paper that enhances something existing by making it more complex and gets 2% extra on some benchmark. The amusing thing about these papers is that they have a reasonably high chance of getting accepted (a reviewer can’t point to anything to kill them; they are also sometimes referred to as “*cockroach papers*”), so if you have a string of these papers accepted you can feel as though you’re being very productive, but in fact these papers won’t go on to be highly cited and you won’t go on to have a lot of impact on the field. Similarly, finding projects should ideally not include thoughts along the lines of “there’s this next logical step in the air that no one has done yet, let me do it”, or “this should be an easy poster”.

Case study: my thesis. To make some of this discussion more concrete I wanted to use the example of how my own PhD unfolded. First, fun fact: my entire thesis is based on work I did in the last 1.5 years of my PhD. i.e. it took me quite a long time to wiggle around in the metaproblem space and find a problem that I felt very excited to work on (the other ~2 years I mostly meandered on 3D things (e.g. Kinect Fusion, 3D meshes, point cloud features) and video things). Then at one point in my 3rd year I randomly stopped by Richard Socher’s office on some Saturday at 2am. We had a chat about interesting problems and I realized that some of his work on images and language was in fact getting at something very interesting (of course, the area at the intersection of images and language goes back quite a lot further than Richard as well). I couldn’t quite see all the papers that would follow but it seemed heuristically very promising: it was highly fertile (a lot of unsolved problems, a lot of interesting possibilities on grounding descriptions to images), I felt that it was very cool and important, it was easy to explain, it seemed to be at the boundary of possible (Deep Learning has just started to work), the datasets had just started to become available (Flickr8K had just come out), it fit nicely into Fei-Fei’s interests and even if I were not successful I’d at least get

lots of practice with optimizing interesting deep nets that I could reapply elsewhere. I had a strong feeling of a tsunami of checkmarks as everything clicked in place in my mind. I pitched this to Fei-Fei (my adviser) as an area to dive into the next day and, with relief, she enthusiastically approved, encouraged me, and would later go on to steer me within the space (e.g. Fei-Fei insisted that I do image to sentence generation while I was mostly content with ranking.). I'm happy with how things evolved from there. In short, I meandered around for 2 years stuck around the outer loop, finding something to dive into. Once it clicked for me what that was based on several heuristics, I dug in.

Resistance. I'd like to also mention that your adviser is by no means infallible. I've witnessed and heard of many instances in which, in retrospect, the adviser made the wrong call. If you feel this way during your phd you should have the courage to sometimes ignore your adviser. Academia generally celebrates independent thinking but the response of your specific adviser can vary depending on circumstances. I'm aware of multiple cases where the bet worked out very well and I've also personally experienced cases where it did not. For instance, I disagreed strongly with some advice Andrew Ng gave me in my very first year. I ended up working on a problem he wasn't very excited about and, surprise, he turned out to be very right and I wasted a few months. Win some lose some :)

Don't play the game. Finally, I'd like to challenge you to think of a PhD as more than just a sequence of papers. You're not a paper writer. You're a member of a research community and your goal is to push the field forward. Papers are one common way of doing that but I would encourage you to look beyond the established academic game. Think for yourself and from first principles. Do things others don't do but should. Step off the treadmill that has been put before you. I tried to do some of this myself throughout my PhD. This blog is an example - it allows me communicate things that wouldn't ordinarily go into papers. The ImageNet human reference experiments are an example - I felt strongly that it was important for the field to know the ballpark human accuracy on ILSVRC so I took a few weeks off and evaluated it. The academic search tools (e.g. arxiv-sanity) are an example - I felt continuously frustrated by the inefficiency of finding papers in the literature and I released and maintain the site in hopes that it can be useful to others. Teaching CS231n twice is an example - I put much more effort into it than is rationally advisable for a PhD student who should be doing research, but I felt that the field was held back if people couldn't efficiently learn about the topic and enter. A lot of my PhD endeavors have likely come at a cost in standard academic metrics (e.g. h-index, or number of publications in top venues) but I did them anyway, I would do it the same way again, and here I am encouraging others to as well. To add a pinch of salt and wash down the ideology a bit, based on several past discussions with my friends and colleagues I know that this view is contentious and that many would disagree.

Writing papers

JOURNAL OF L^AT_EX CLASS FILES, VOL. 11, NO. 4, DECEMBER 2012 1

How to Use the IEEEtran L^AT_EX Class

Michael Shell, *Member, IEEE*

(Invited Paper)

Abstract—This article describes how to use the IEEEtran class with L^AT_EX to produce high quality typeset papers that are suitable for submission to the Institute of Electrical and Electronics Engineers (IEEE). IEEEtran can produce conference, journal and technical note (correspondence) papers with a suitable choice of class options. This document was produced using IEEEtran in journal mode.

Index Terms—Class, IEEEtran, L^AT_EX, paper, style, template, typesetting.

with more complex usage techniques, can be found in `bare_adv.tex`.

It is assumed that the reader has at least a basic working knowledge of L^AT_EX. Those so lacking are strongly encouraged to read some of the excellent literature on the subject [4]–[6]. In particular, Tobias Oetiker’s *The Not So Short Introduction to L^AT_EX 2_ε* [5], which provides a general overview of working with L^AT_EX, and Stephan M. Moser’s *How to Typeset Equations*

Writing good papers is an essential survival skill of an academic (kind of like making fire for a caveman). In particular, it is very important to realize that papers are a specific thing: they look a certain way, they flow a certain way, they have a certain structure, language, and statistics that the other academics expect. It’s usually a painful exercise for me to look through some of my early PhD paper drafts because they are quite terrible. There is a lot to learn here.

Review papers. If you’re trying to learn to write better papers it can feel like a sensible strategy to look at many good papers and try to distill patterns. This turns out to not be the best strategy; it’s analogous to only receiving positive examples for a binary classification problem. What you really want is to also have exposure to a large number of bad papers and one way to get this is by reviewing papers. Most good conferences have an acceptance rate of about 25% so most papers you’ll review are bad, which will allow you to build a powerful binary classifier. You’ll read through a bad paper and realize how unclear it is, or how it doesn’t define its variables, how vague and abstract its intro is, or how it dives in to the details too quickly, and you’ll learn to avoid the same pitfalls in your own papers. Another related valuable experience is to attend (or form) journal clubs - you’ll see experienced researchers critique papers and get an impression for how your own papers will be analyzed by others.

Get the gestalt right. I remember being impressed with Fei-Fei (my adviser) once during a reviewing session. I had a stack of 4 papers I had reviewed over the last several hours and she picked them up, flipped through each one for 10 seconds, and said one of them was good and the other three bad. Indeed, I was accepting the one and rejecting the other three, but something that took me several hours took her seconds. Fei-Fei was relying on the *gestalt* of the papers as a powerful heuristic. Your papers, as you become a more senior

researcher take on a characteristic look. An introduction of ~1 page. A ~1 page related work section with a good density of citations - not too sparse but not too crowded. A well-designed pull figure (on page 1 or 2) and system figure (on page 3) that were not made in MS Paint. A technical section with some math symbols somewhere, results tables with lots of numbers and some of them bold, one additional cute analysis experiment, and the paper has exactly 8 pages (the page limit) and not a single line less. You'll have to learn how to endow your papers with the same gestalt because many researchers rely on it as a cognitive shortcut when they judge your work.

Identify the core contribution. Before you start writing anything it's important to identify the single core contribution that your paper makes to the field. I would especially highlight the word *single*. A paper is not a random collection of some experiments you ran that you report on. The paper sells a single thing that was not obvious or present before. You have to argue that the thing is important, that it hasn't been done before, and then you support its merit experimentally in controlled experiments. The entire paper is organized around this core contribution with surgical precision. In particular it doesn't have any additional fluff and it doesn't try to pack anything else on a side. As a concrete example, I made a mistake in one of my earlier papers on [video classification](#) where I tried to pack in two contributions: 1) a set of architectural layouts for video convnets and an unrelated 2) multi-resolution architecture which gave small improvements. I added it because I reasoned first that maybe someone could find it interesting and follow up on it later and second because I thought that contributions in a paper are additive: two contributions are better than one. Unfortunately, this is false and very wrong. The second contribution was minor/dubious and it diluted the paper, it was distracting, and no one cared. I've made a similar mistake again in my [CVPR 2014 paper](#) which presented two separate models: a ranking model and a generation model. Several good in-retrospect arguments could be made that I should have submitted two separate papers; the reason it was one is more historical than rational.

The structure. Once you've identified your core contribution there is a default recipe for writing a paper about it. The upper level structure is by default Intro, Related Work, Model, Experiments, Conclusions. When I write my intro I find that it helps to put down a coherent top-level narrative in latex comments and then fill in the text below. I like to organize each of my paragraphs around a single concrete point stated on the first sentence that is then supported in the rest of the paragraph. This structure makes it easy for a reader to skim the paper. A good flow of ideas is then along the lines of 1) X (+define X if not obvious) is an important problem 2) The core challenges are this and that. 2) Previous work on X has addressed these with Y, but the problems with this are Z. 3) In this work we do W (?). 4) This has the following appealing properties and our experiments show this and that. You can play with this structure a bit but these core points should be clearly made. Note again that the paper is surgically organized around your exact contribution. For example, when you list the

challenges you want to list exactly the things that you address later; you don't go meandering about unrelated things to what you have done (you can speculate a bit more later in conclusion). It is important to keep a sensible structure throughout your paper, not just in the intro. For example, when you explain the model each section should: 1) explain clearly what is being done in the section, 2) explain what the core challenges are 3) explain what a baseline approach is or what others have done before 4) motivate and explain what you do 5) describe it.

Break the structure. You should also feel free (and you're encouraged to!) play with these formulas to some extent and add some spice to your papers. For example, see this amusing paper from [Razavian et al. in 2014](#) that structures the introduction as a dialog between a student and the professor. It's clever and I like it. As another example, a lot of papers from [Alyosha Efron](#) have a playful tone and make great case studies in writing fun papers. As only one of many examples, see this paper he wrote with Antonio Torralba: [Unbiased look at dataset bias](#). Another possibility I've seen work well is to include an FAQ section, possibly in the appendix.

Common mistake: the laundry list. One very common mistake to avoid is the "laundry list", which looks as follows: "Here is the problem. Okay now to solve this problem first we do X, then we do Y, then we do Z, and now we do W, and here is what we get". You should try very hard to avoid this structure. Each point should be justified, motivated, explained. Why do you do X or Y? What are the alternatives? What have others done? It's okay to say things like this is common (add citation if possible). Your paper is not a report, an enumeration of what you've done, or some kind of a translation of your chronological notes and experiments into latex. It is a highly processed and very focused discussion of a problem, your approach and its context. It is supposed to teach your colleagues something and you have to justify your steps, not just describe what you did.

The language. Over time you'll develop a vocabulary of good words and bad words to use when writing papers. Speaking about machine learning or computer vision papers specifically as concrete examples, in your papers you never "study" or "investigate" (there are boring, passive, bad words); instead you "develop" or even better you "propose". And you don't present a "system" or, *shudder*, a "pipeline"; instead, you develop a "model". You don't learn "features", you learn "representations". And god forbid, you never "combine", "modify" or "expand". These are incremental, gross terms that will certainly get your paper rejected :).

An internal deadlines 2 weeks prior. Not many labs do this, but luckily Fei-Fei is quite adamant about an internal deadline 2 weeks before the due date in which you must submit at least a 5-page draft with all the final experiments (even if not with final numbers) that goes through an internal review process identical to the external one (with the same review forms

filled out, etc). I found this practice to be extremely useful because forcing yourself to lay out the full paper almost always reveals some number of critical experiments you must run for the paper to flow and for its argument flow to be coherent, consistent and convincing.

Another great resource on this topic is [Tips for Writing Technical Papers](#) from Jennifer Widom.

Writing code



A lot of your time will of course be taken up with the *execution* of your ideas, which likely involves a lot of coding. I won't dwell on this too much because it's not uniquely academic, but I would like to bring up a few points.

Release your code. It's a somewhat surprising fact but you can get away with publishing papers and not releasing your code. You will also feel a lot of incentive to not release your code: it can be a lot of work to release code can look like spaghetti since you iterate very quickly, you have to clean up a lot), it can be intimidating to think that others might judge you on your at most decent coding abilities, it is painful to maintain code and answer questions from other people about it (forever), and you might also be concerned that people could spot bugs that invalidate your results. However, it is precisely for some of these reasons that you should commit to releasing your code: it will force you to adopt better coding habits due to fear of public shaming (which will end up saving you time!), it will force you to learn better engineering practices, it will force you to be more thorough with your code (e.g. writing unit tests to make bugs much less likely), it will make others much more likely to follow up on your work (and hence lead to more citations of your papers) and of course it will be much more useful to everyone as a record of exactly what was done for posterity. When you do release your code I recommend taking advantage of [docker containers](#); this will reduce the amount of headaches people email you about when they can't get all the dependencies (and their precise versions) installed.

Think of the future you. Make sure to document all your code very well for yourself. I guarantee you that you will come back to your code base a few months later (e.g. to do a few

more experiments for the camera ready version of the paper), and you will feel *completely* lost in it. I got into the habit of creating very thorough readme.txt files in all my repos (for my personal use) as notes to future self on how the code works, how to run it, etc.

Giving talks



So, you published a paper and it's an oral! Now you get to give a few minute talk to a large audience of people - what should it look like?

The goal of a talk. First, that there's a common misconception that the goal of your talk is to tell your audience about what you did in your paper. This is incorrect, and should only be a second or third degree design criterion. The goal of your talk is to 1) get the audience really excited about the problem you worked on (they must appreciate it or they will not care about your solution otherwise!) 2) teach the audience something (ideally while giving them a taste of your insight/solution; don't be afraid to spend time on other's related work), and 3) entertain (they will start checking their Facebook otherwise). Ideally, by the end of the talk the people in your audience are thinking some mixture of "wow, I'm working in the wrong area", "I have to read this paper", and "This person has an impressive understanding of the whole area".

A few do's: There are several properties that make talks better. For instance, Do: Lots of pictures. People Love pictures. Videos and animations should be used more sparingly because they distract. Do: make the talk actionable - talk about something someone can *do* after your talk. Do: give a live demo if possible, it can make your talk more memorable. Do: develop a broader intellectual arch that your work is part of. Do: develop it into a story (people love stories). Do: cite, cite, cite - a lot! It takes very little slide space to pay credit to your colleagues. It pleases them and always reflects well on you because it shows that you're humble about your own contribution, and aware that it builds on a lot of what has

come before and what is happening in parallel. You can even cite related work published at the same conference and briefly advertise it. Do: practice the talk! First for yourself in isolation and later to your lab/friends. This almost always reveals very insightful flaws in your narrative and flow.

Don't: texttexttext. Don't crowd your slides with text. There should be very few or no bullet points - speakers sometimes try to use these as a crutch to remind themselves what they should be talking about but the slides are not for you they are for the audience. These should be in your speaker notes. On the topic of crowding the slides, also avoid complex diagrams as much as you can - your audience has a fixed bit bandwidth and I guarantee that your own very familiar and "simple" diagram is not as simple or interpretable to someone seeing it for the first time.

Careful with: result tables: Don't include dense tables of results showing that your method works better. You got a paper, I'm sure your results were decent. I always find these parts boring and unnecessary unless the numbers show something interesting (other than your method works better), or of course unless there is a large gap that you're very proud of. If you do include results or graphs build them up slowly with transitions, don't post them all at once and spend 3 minutes on one slide.

Pitfall: the thin band between bored/confused. It's actually quite tricky to design talks where a good portion of your audience *learns* something. A common failure case (as an audience member) is to see talks where I'm painfully bored during the first half and completely confused during the second half, learning nothing by the end. This can occur in talks that have a very general (too general) overview followed by a technical (too technical) second portion. Try to identify when your talk is in danger of having this property.

Pitfall: running out of time. Many speakers spend too much time on the early intro parts (that can often be somewhat boring) and then frantically speed through all the last few slides that contain the most interesting results, analysis or demos. Don't be that person.

Pitfall: formulaic talks. I might be a special case but I'm always a fan of non-formulaic talks that challenge conventions. For instance, I *despise* the outline slide. It makes the talk so boring, it's like saying: "This movie is about a ring of power. In the first chapter we'll see a hobbit come into possession of the ring. In the second we'll see him travel to Mordor. In the third he'll cast the ring into Mount Doom and destroy it. I will start with chapter 1" - Come on! I use outline slides for much longer talks to keep the audience anchored if they zone out (at 30min+ they inevitably will a few times), but it should be used sparingly.

Observe and learn. Ultimately, the best way to become better at giving talks (as it is with writing papers too) is to make conscious effort to pay attention to what great (and not so

great) speakers do and build a binary classifier in your mind. Don't just enjoy talks; analyze them, break them down, learn from them. Additionally, pay close attention to the audience and their reactions. Sometimes a speaker will put up a complex table with many numbers and you will notice half of the audience immediately look down on their phone and open Facebook. Build an internal classifier of the events that cause this to happen and avoid them in your talks.

Attending conferences



On the subject of conferences:

Go. It's very important that you go to conferences, especially the 1-2 top conferences in your area. If your adviser lacks funds and does not want to pay for your travel expenses (e.g. if you don't have a paper) then you should be willing to pay for yourself (usually about \$2000 for travel, accommodation, registration and food). This is important because you want to become part of the academic community and get a chance to meet more people in the area and gossip about research topics. Science might have this image of a few brilliant lone wolfs working in isolation, but the truth is that research is predominantly a highly social endeavor - you stand on the shoulders of many people, you're working on problems in parallel with other people, and it is these people that you're also writing papers to. Additionally, it's unfortunate but each field has knowledge that doesn't get serialized into papers but is instead spread across a shared understanding of the community; things such as what are the next important topics to work on, what papers are most interesting, what is the inside scoop on papers, how they developed historically, what methods work (not just on paper, in reality), etcetc. It is very valuable (and fun!) to become part of the community and get direct access to the hivemind - to learn from it first, and to hopefully influence it later.

Talks: choose by speaker. One conference trick I've developed is that if you're choosing which talks to attend it can be better to look at the speakers instead of the topics. Some people give better talks than others (it's a skill, and you'll discover these people in time) and in my experience I find that it often pays off to see them speak even if it is on a topic that isn't exactly connected to your area of research.

The real action is in the hallways. The speed of innovation (especially in Machine Learning) now works at timescales much faster than conferences so most of the relevant papers you'll see at the conference are in fact old news. Therefore, conferences are primarily a social event. Instead of attending a talk I encourage you to view the hallway as one of the main events that doesn't appear on the schedule. It can also be valuable to stroll the poster session and discover some interesting papers and ideas that you may have missed.

It is said that there are three stages to a PhD. In the first stage you look at a related paper's reference section and you haven't read most of the papers. In the second stage you recognize all the papers. In the third stage you've shared a beer with all the first authors of all the papers.

Closing thoughts

I can't find the quote anymore but I heard Sam Altman of YC say that there are no shortcuts or cheats when it comes to building a startup. You can't expect to win in the long run by somehow gaming the system or putting up false appearances. I think that the same applies in academia. Ultimately you're trying to do good research and push the field forward and if you try to game any of the proxy metrics you won't be successful in the long run. This is especially so because academia is in fact surprisingly small and highly interconnected, so anything shady you try to do to pad your academic resume (e.g. self-citing a lot, publishing the same idea multiple times with small remixes, resubmitting the same rejected paper over and over again with no changes, conveniently trying to leave out some baselines etc.) will eventually catch up with you and you will not be successful.

So at the end of the day it's quite simple. Do good work, communicate it properly, people will notice and good things will happen. Have a fun ride!

EDIT: [HN discussion link](#).