Research Methods in Machine Learning

Tom Dietterich
Distinguished Professor (Emeritus)
Oregon State University
Corvallis, OR USA

Oregon State University

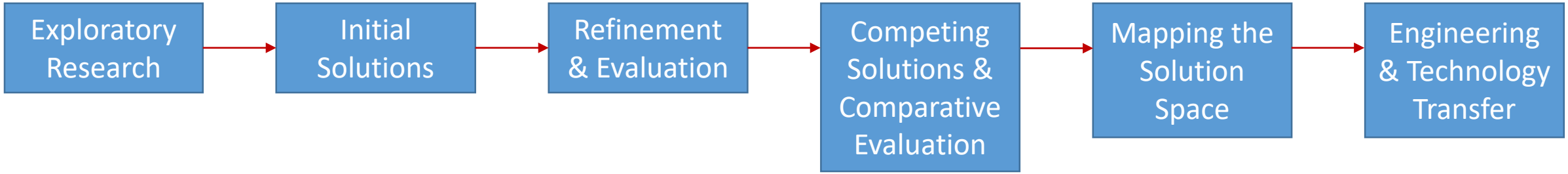
Plan for Today

- Choosing and Solving a Research Problem
  - Research Life Cycle
    - Exercise 1: What is the position of your project in the life cycle?
  - Corresponding Skills
    - Exercise 2: Skills inventory
- Write a Successful (NeurIPS) Paper
  - Process
  - Structure
  - Analysis of an Example paper
    - Exercise 3: Parsing it into the provided structure
  - Writing tips
- Wrap up

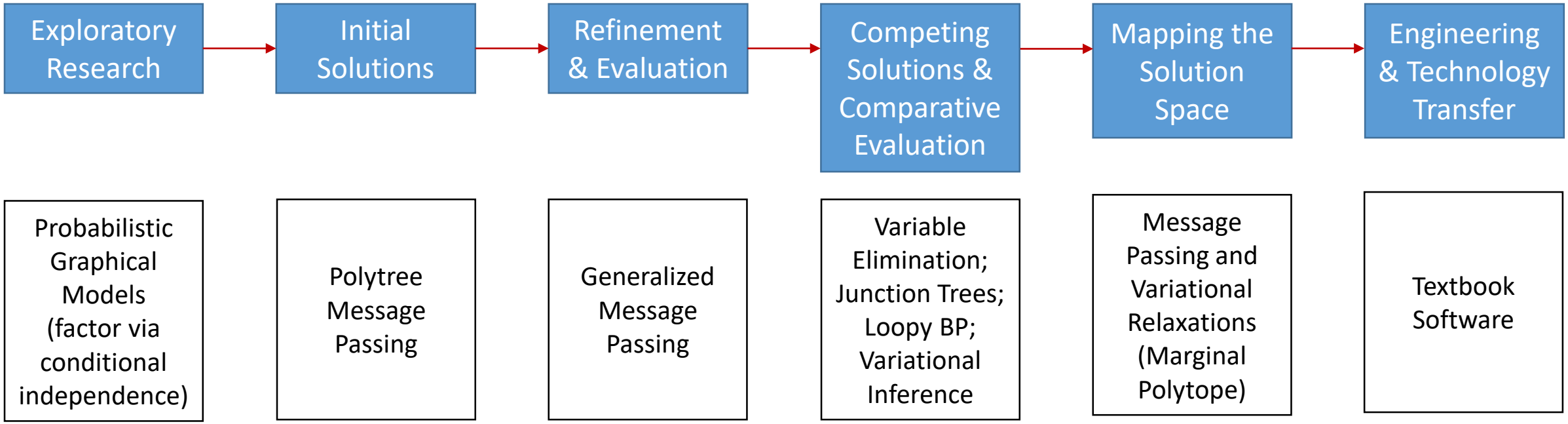
Download

- These slides: http://web.engr.oregonstate.edu/~tgd/talks/new-in-ml-2019.pdf
- This paper: https://arxiv.org/abs/1809.03113

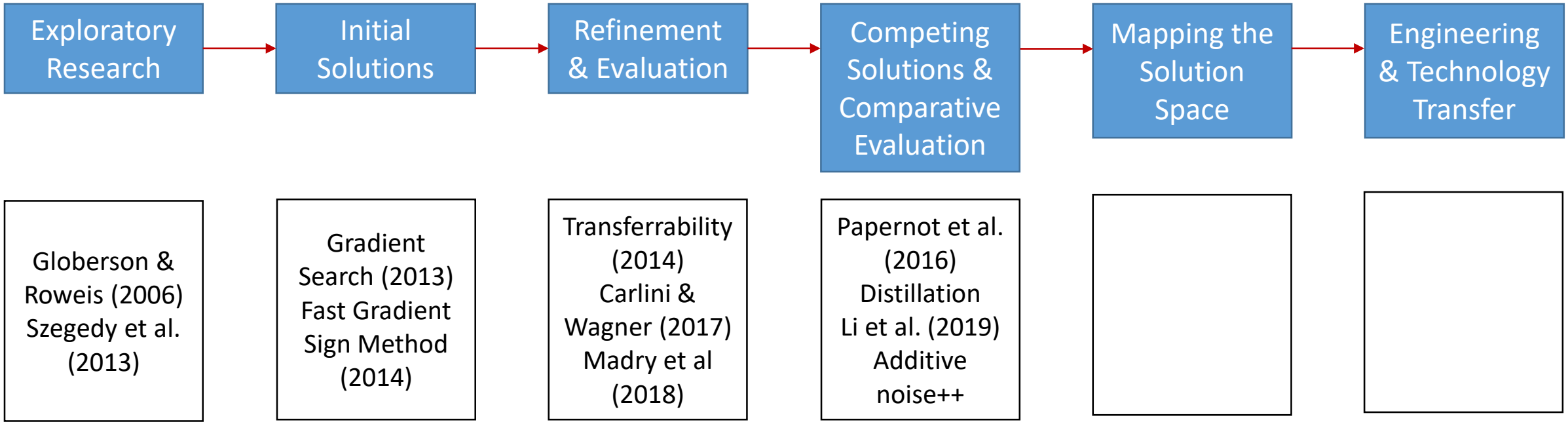
Choosing and Solving a Research Problem
Research Life Cycle



Example 1: Representation and Algorithms for Probabilistic Graphical Models



Example 2: Adversarial Test Queries

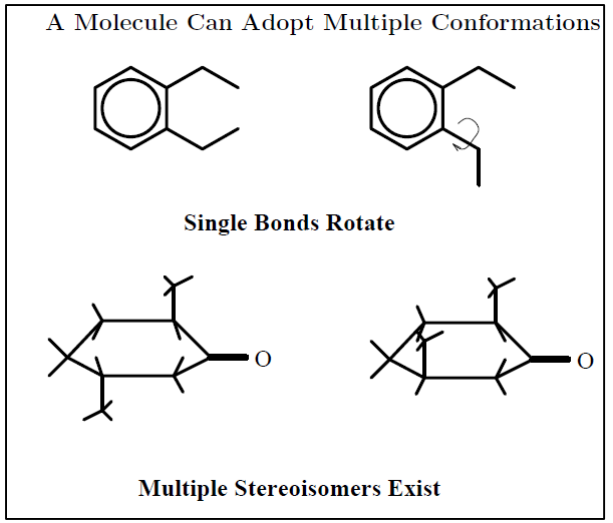


Exploratory Research

- Defining new problems, new constraints, new opportunities, new approaches
  - Example:
    - Multiple-Instance Learning: Labeled bags of instances
    - Adversarial examples
- Break out of established paradigms by changing the problem definition
  - Examples:
    - Transfer learning and domain adaptation: multiple, related learning problems
    - Feed forward neural networks: beyond traditional statistical models
- Risks:
  - It might not be an important problem
  - Need to convince readers it is an important problem
  - “Science advances funeral by funeral” (Paul Samuelson gisting Max Planck)
- Benefits
  - It is a critical path to major progress in a field

Drivers of Exploratory Research

- Novel applications
  - Multiple instance learning grew out of attempting to apply ML to drug design
- Mathematical advances and insights
  - Support vector machines combined two previous directions
    - mathematical programming to classification (Mangasarian et al)
    - Vapnik's insight that the hinge loss is convex
    - Kimeldorf & Wahba: representer theorem for spline kernels
  - Random Forests grew out of Breiman's intuitions concerning the bias-variance tradeoff and stabilization methods
- Importing ideas from other fields
  - Convex optimization
  - Variational methods from physics
  - Mathematics: theory of statistics, information theory, control theory, ODEs, real analysis, functional analysis, etc.
- Frustration
  - AutoML grew out of the pain of tweaking hyper-parameters



Initial Solutions

- Provide an initial solution to a problem
- Often very narrow or overly complex
- Examples:
  - First paper on PAC learning (Valiant, 1984) proved a result for very limited and impractical cases: k-CNF and monotone DNF
  - First paper on multiple instance learning (Dietterich et al, 1997) presented a very baroque algorithm that combined kernel density estimation with axis-parallel rectangles
  - First paper on Bayesian networks (Pearl 1985) described simple message passing for tree-structured networks
- Notes
  - It is often difficult to propose a new problem definition without also proposing an initial solution
    - Exception: Adversarial examples
  - “Nothing stimulates good research like a bad paper about an interesting problem” (Dietterich)

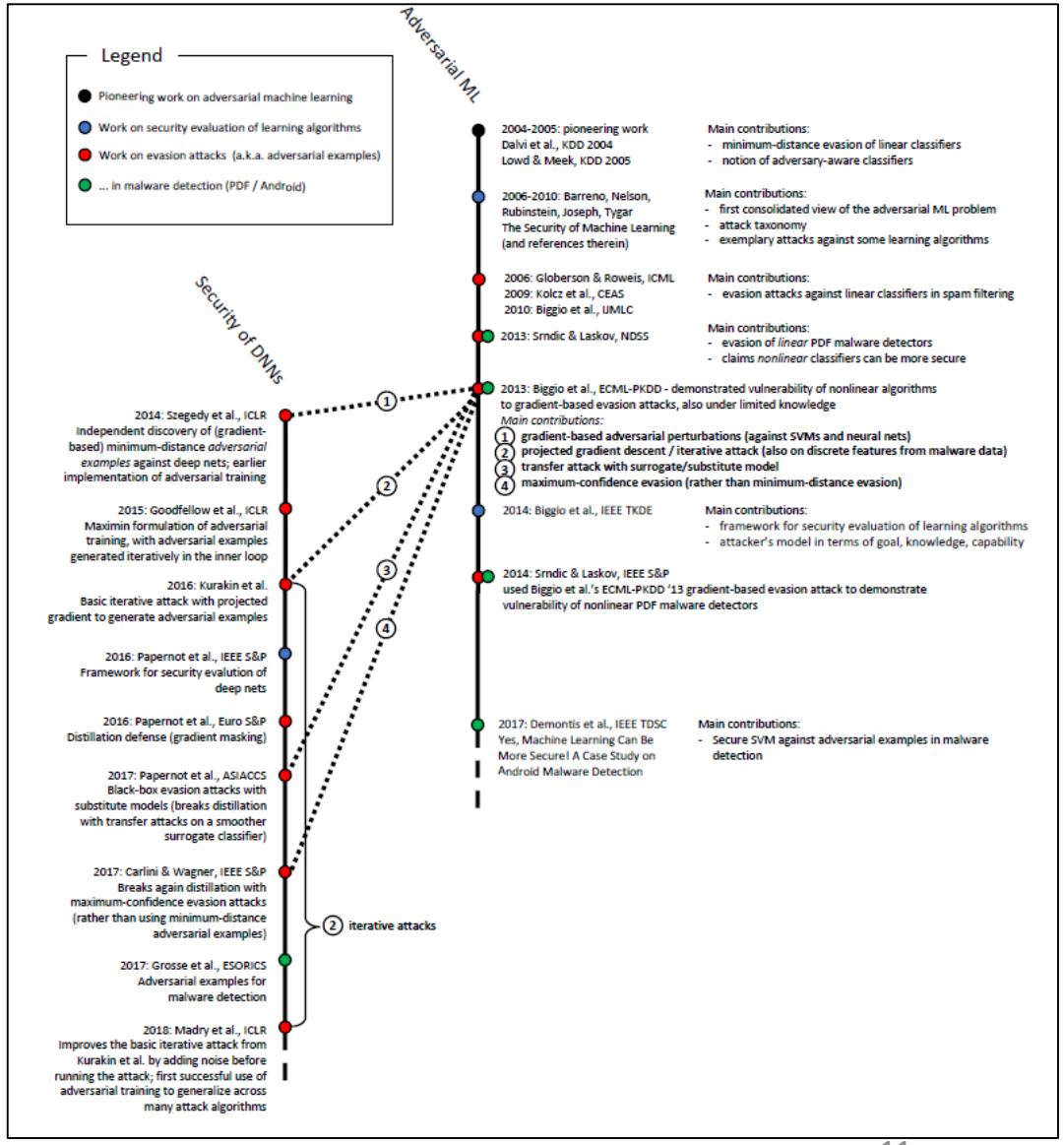
Refinement and Evaluation

- Develop refinements of the initial solution
  - Fast gradient sign method made it easier to study adversarial examples
- Study the generality and scope of the phenomenon
  - Demonstration that adversarial examples exist for many ML classifiers
    - random forests, SVMs, etc.
  - Demonstration that adversarial examples transfer across classifier types
  - Demonstration that simple defenses can easily be evaded
- Develop refinements of the initial evaluation metrics
- Notes:
  - The initial authors have a competitive advantage here, if they can grasp it
  - Otherwise, it can be a race (favors large groups, not PhD students)
  - Lots of creativity is required to ask the right questions about generality and scope

Competing Solutions and Comparative Evaluation

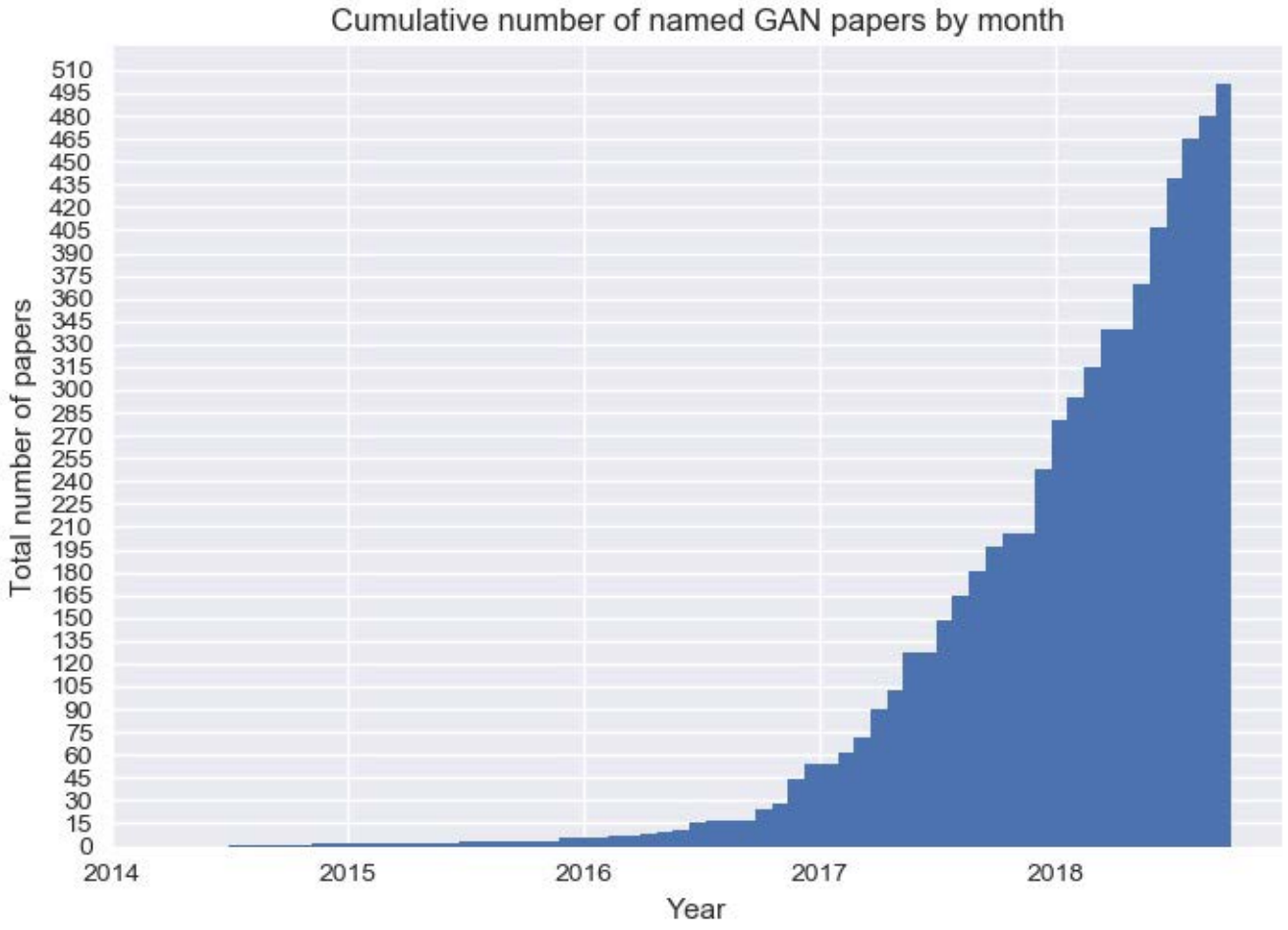
Biggio & Roli, 2018

- Sequences of improvements and alternatives are published
- Each is typically compared to previous methods
- Periodically, it is valuable to conduct a careful benchmark comparison



The Incremental Improvement Space can be Very Crowded

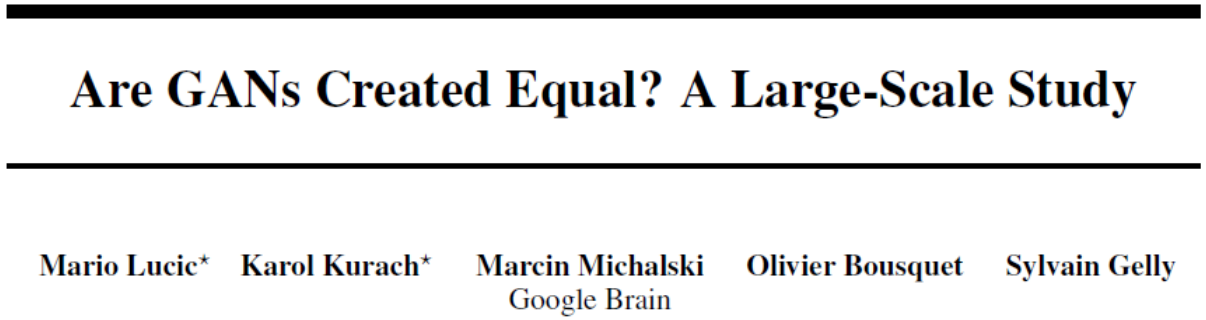
- Example: Generative Adversarial Networks
- Risks:
  - Small improvements are rarely worthwhile (unless they also provide some general insight)
  - Depends heavily on metrics which may not reflect real applications (AUC, BLEU)
  - Can get scooped easily
  - Favors large teams (not PhD students)
- Advantages:
  - It is easy
  - It feels like we are making progress
- Notes:
  - Improvements should be guided by principles: Don't search in the space of mechanisms



https://github.com/hindupuravinash/the-gan-zoo

The Illusion of Progress

- Evaluation metrics for GANs are notoriously “soft”
- It seems that a lot of effort was expended for relatively little gain



“We find that most models can reach similar scores with enough hyperparameter optimization and random restarts. This suggests that improvements can arise from a higher computational budget and tuning more than fundamental algorithmic changes.”

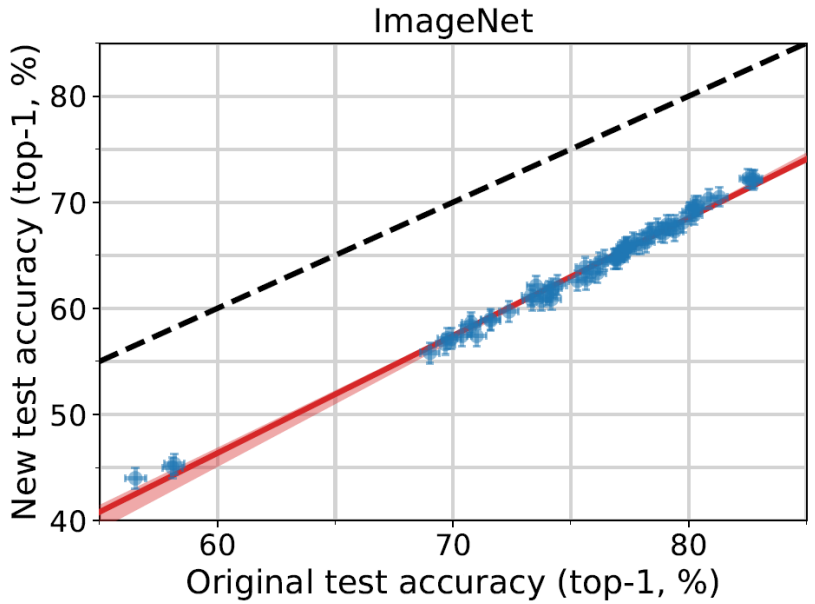
But Progress Can Also Be Real

- Recht et al. constructed new test sets for ImageNet and evaluated a wide range of published networks
- Performance was not as good as on the original test set, but this is probably due to the new test sets being more difficult

Do ImageNet Classifiers Generalize to ImageNet?

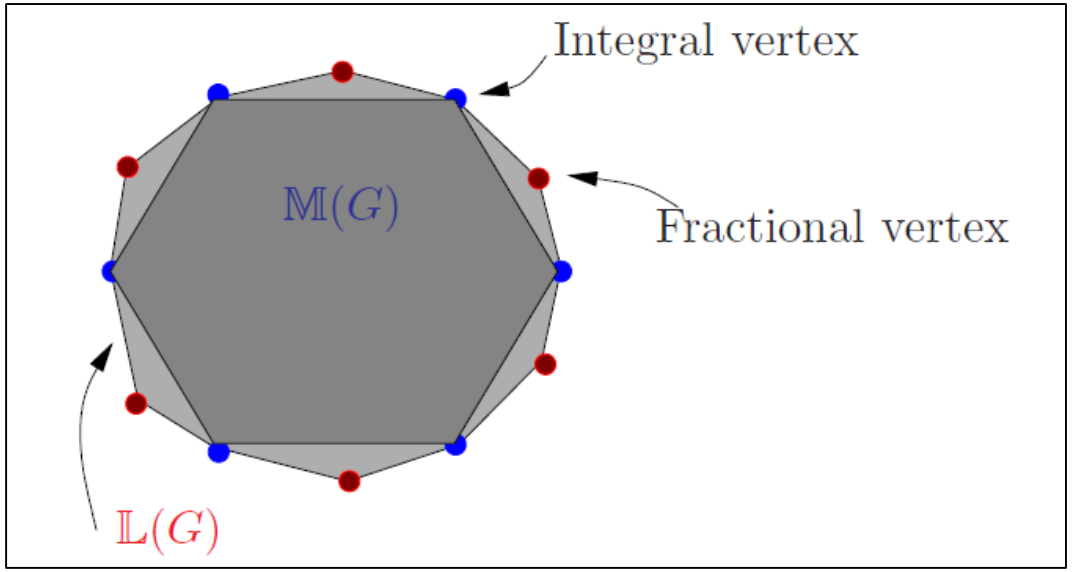
Benjamin Recht* UC Berkeley
Rebecca Roelofs UC Berkeley
Ludwig Schmidt UC Berkeley
Vaishaal Shankar UC Berkeley

"accuracy gains on the original test sets translate to larger gains on the new test sets"



Mapping the Solution Space

- Can we understand the design space for a problem?
  - Place all algorithms into a single framework
  - What are the key design decisions?
  - What are lower bounds on the best any method can do?
- Example:
  - Wainwright developed a comprehensive theory of the geometry of message passing in Bayesian networks
  - Related to LP solutions on inner and outer approximations of the marginal polytope
  - Can be applied to understand *any* message passing method for probabilistic inference

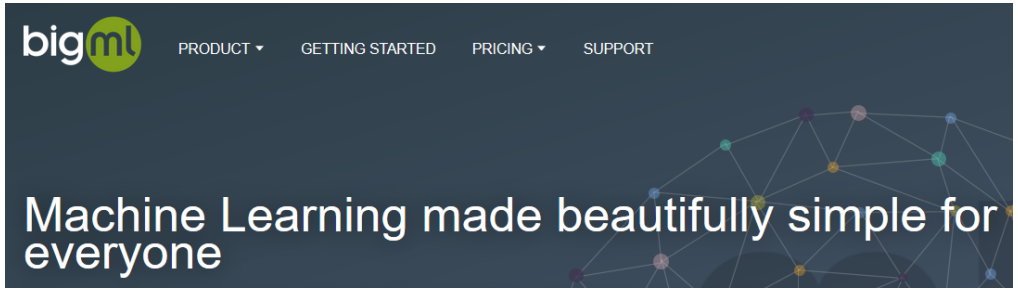


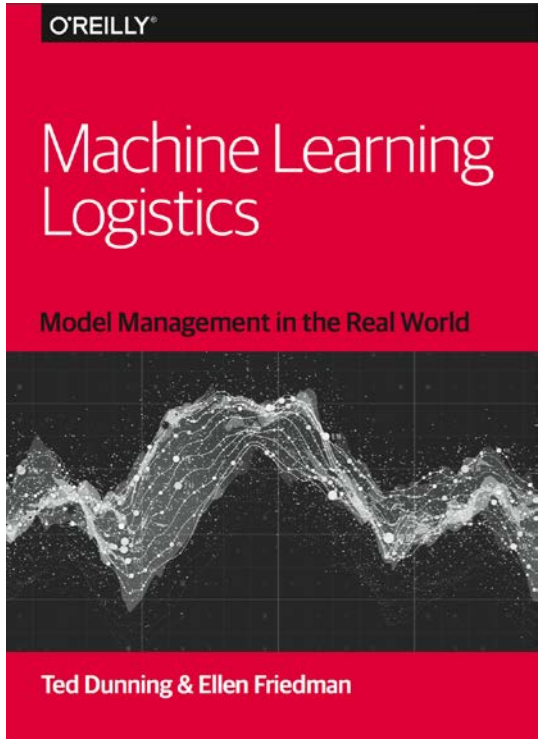
Wainwright et al (2008)
http://www.maths.dur.ac.uk/lms/087/Talks/wainwright.pdf

Engineering and Technology Transfer

Rules of Machine Learning: Best Practices for ML Engineering

Martin Zinkevich

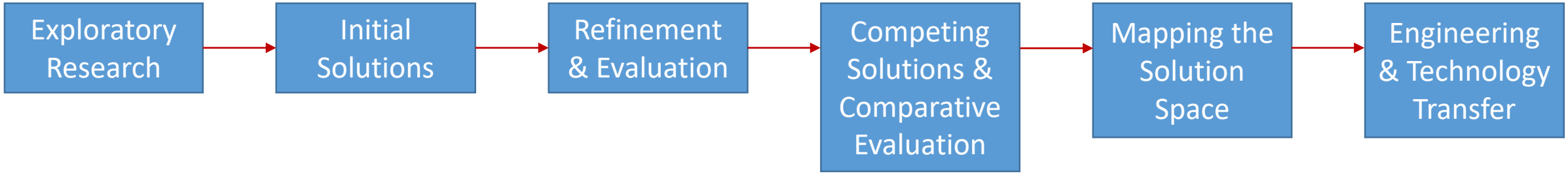




TensorFlow:
Large-Scale Machine Learning on Heterogeneous Distributed Systems
(Preliminary White Paper, November 9, 2015)

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng
Google Research*

Exercise 1: Life Cycle Position



- Form into groups of 2-3 people
- Briefly discuss one of your research projects and determine which life cycle phase best describes it

Different Life Cycle Phases Require Different Skills

	Exploratory Research	Initial Solutions	Refinement & Evaluation	Competing Solutions	Mapping Solution Space	Engineering & Deployment
Reading Literature	X	X	X	X	X	
Analysis Techniques			X	X	X	
Theorem Formulation			X	X	X	
Algorithm design	X	X	X	X		
Coding & Testing		X	X	X		X
Coding in DL Frameworks		X	X	X		X
Experiment design		X	X	X		
Story Telling	X	X	X	X	X	X
English Skills	X	X	X	X	X	X
Giving Talks	X	X	X	X	X	X

Exercise 2: Skills Inventory

SKILLS
Reading Literature
Analysis Techniques
Theorem Formulation
Algorithm design
Coding & Testing
Coding in DL Frameworks
Experiment design
Story Telling
English Skills
Giving Talks

- Working alone (or in groups) list the skills you need for your project
- These can be more specific than my list
- Assess your skill level for each of them
- Today (or, more likely, later) develop a plan for addressing any skill gaps
  - Taking classes (math background, story telling, English skills)
  - Studying examples (theorems, proof techniques, code on github)
  - Many universities provide tutoring with writing
  - Practice (giving talks)

Part 2: Writing Good Papers

- You have chosen an important problem that matches your interests and skill set
- You have results
- Time to publish!

Paper = Claim + Evidence + Story

- Introduction:
  - What problem are you attacking?
  - Why is it important?
  - What is known already? (summary)
  - What aspects are still unsolved? What are the shortcomings of existing solutions? (summary)
  - What claim(s) are you making?
  - What evidence will you present?
  - What conclusions do you draw? “No suspense”
- Current state of knowledge about the problem
  - Review of existing work
  - Existing solutions and their shortcomings

Body: Theoretical Claims and Evidence

- Notation and definitions
- Previous results you will be using
- Qualitative analysis: What kind of result can we expect for this kind of problem?
- Statement of result (theorem)
- Sketch of proof (usually put full proof in appendix)
- Discussion of assumptions and limitations of the result
- Comparison with related results, especially if they are not directly comparable

Body: Algorithm/Method Paper

- Definitions and notation
- Qualitative analysis: What kind of result can we expect for this kind of problem?
- Description of previous algorithm ideas that you will be using
- Overview of your approach: what is the key insight?
- Description of the algorithm with pseudo-code
- Discussion of configuration and hyper-parameter tuning
- Discussion of asymptotic computational complexity (if it is non-obvious or looks like it might raise scaling issues)
- Discussion of assumptions and limitations of the approach

Body: Experimental Evidence

- Goal of the experiment (e.g., what are the research questions you the experiments try to answer?)
- Experiment structure
  - Data sets
  - Algorithms being compared (including baselines and oracles)
  - Manipulations (independent variables being manipulated)
  - Evaluation metrics
  - Analytical plan (e.g., statistical testing)
- Results of the experiments
  - Always include an assessment of uncertainty (confidence intervals, posterior distribution, statistical tests)
- Discussion of the results
  - Explain the relationship between the results and the research questions and claims of the paper

Concluding Remarks

- The actual conclusions should be in the introduction
- Concluding remarks can discuss the broader significance of the claims as well as open problems

Telling a Story

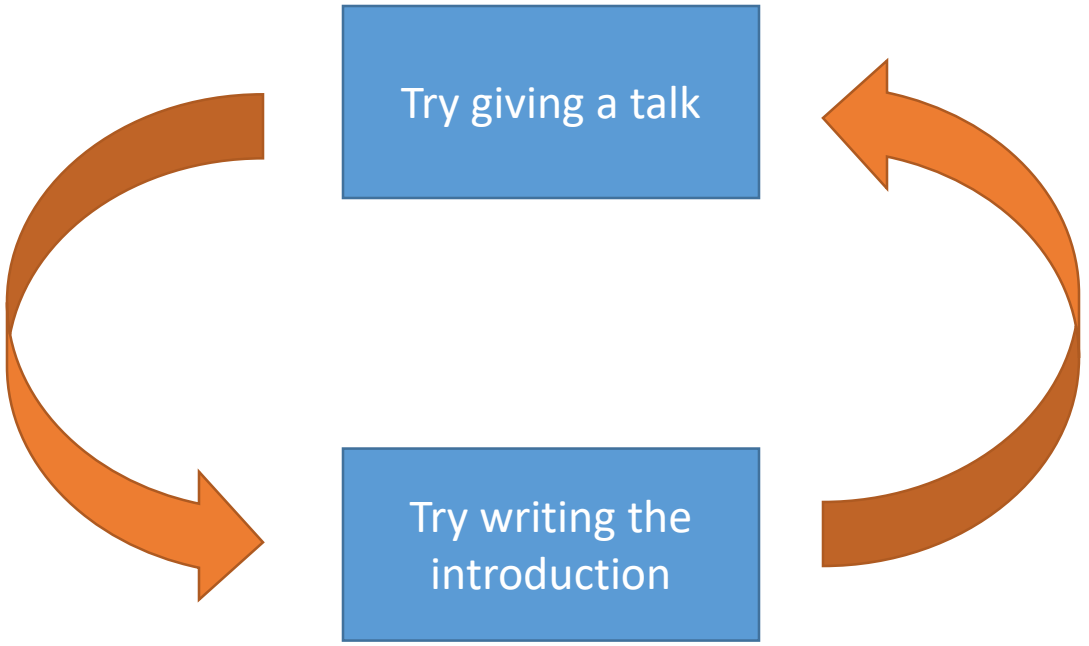
- For a complex theorem or a complex algorithm, you will want to "build it" incrementally
- Example:
  - Describe a clean, simplified algorithm
    - may only work for special cases
    - may not be computationally tractable
  - Then introduce refinements and approximations
    - how to handle more complex cases
    - approximations that make it feasible

The writing process

- Early in the research process, it can be useful to imagine the final paper
  - claims
  - evidence
  - tables and graphs
- Work backwards from this to design the experiments

Developing the Story

- It is often difficult to structure the story
  - When multiple claims and experiments are interdependent, you need to find a sequential order in which to present them
- I find it useful to try giving a talk
  - Forces a sequential order
- I interleave this with trying to write the introduction
- Alternatively create a poster and then explain it to five different people
  - Helps find holes, figure out what questions people have
- Note: The story is NOT about the sequence in which the research was done



The Abstract and Introduction

- These are the first things you write...

  …and the last things you write

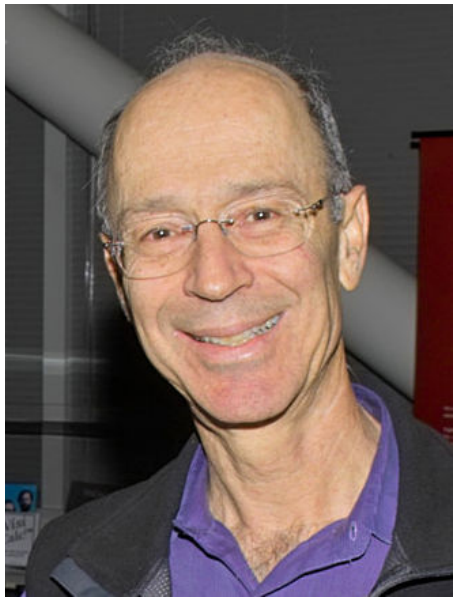
- Write in the present tense
  - “This paper describes an improved method for securing data sets from tampering...”
  - “The new method improves top-5 accuracy from 89% to 95% at no additional cost”

Mistakes to avoid

- Popularity is not a good reason to work on a topic.
  - “Adversarial examples have received a lot of attention lately” NO!
  - “Adversarial examples demonstrate the vulnerability of machine learning system to cyberattack” YES!
- Don’t hype the novelty – State the result
  - “We show here, for the first time, that ...” NO!
  - “Our method provides a non-trivial robustness guarantee on Imagenet, which has been beyond the capability of previous methods...” YES!

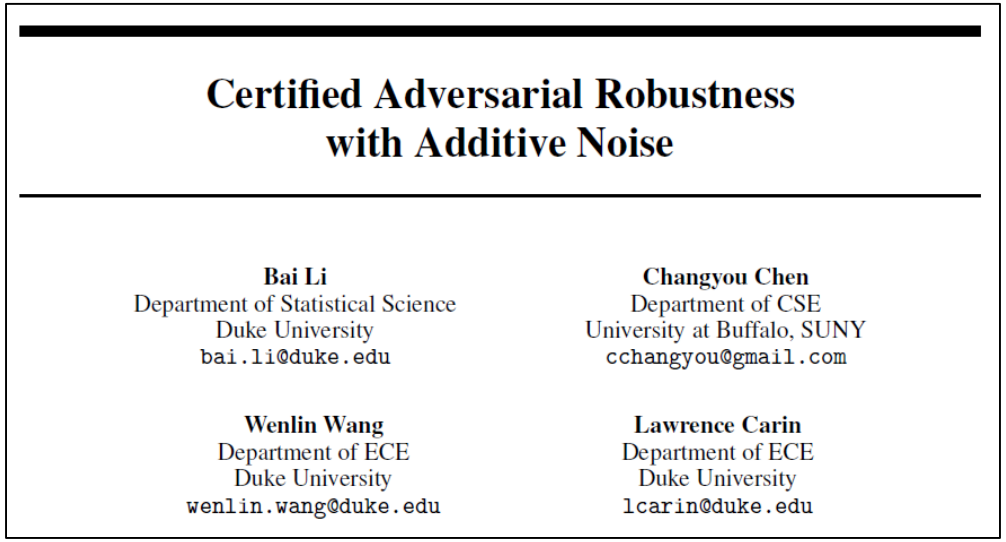
Paper-Driven Revision of the Method

- Advice I once received from Peter Hart:
  - Sometimes as you are writing the paper, you realize it would be a lot easier to write if the algorithm (or the experiments) had been slightly different
  - If so, fix the algorithm (redo-the experiments) so it is easier to describe



Exercise 3: Analyzing a Paper

- Download https://arxiv.org/abs/1809.03113
- Skim the paper and fill out the following page



Certified Adversarial Robustness with Additive Noise

Bai Li
Department of Statistical Science
Duke University
bai.li@duke.edu

Changyou Chen
Department of CSE
University at Buffalo, SUNY
cchangyou@gmail.com

Wenlin Wang
Department of ECE
Duke University
wenlin.wang@duke.edu

Lawrence Carin
Department of ECE
Duke University
lcarin@duke.edu

Paper Analysis

Facet	Notes
Problem:	
Importance:	
Claims:	
State of Knowledge:	
Evidence: Theoretical	
Evidence: Empirical	
Story Structure	

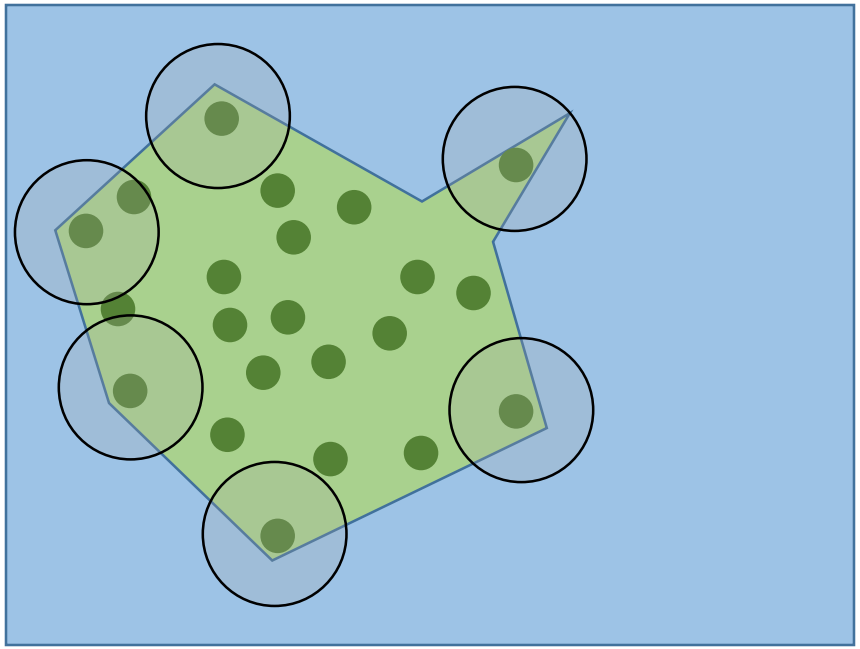
Test time robustness

- Given a query x_q
- Run it through the network M times, each time adding a different Gaussian perturbation δ_m and observe the resulting prediction y_m
- Predict the most common prediction

$$p_j = \frac{1}{M}\sum_{m=1}^{M} \mathbb{I}\left[\hat{f}\left(x_q + \delta_m\right) = j\right] \quad \delta_m \sim N(0, \sigma^2 I)$$

$$\hat{f}_{stab}\left(x_q\right) = \arg\max_j p_j$$

- This smooths the decision boundary



Training Time Robustness: Stability Training with Noise

- Find θ to minimize

$$\sum_i L\big(y_i, f(x_i, \theta)\big) + \lambda D\big(f(x_i), f(x_i + \delta_i)\big) \quad \text{where } \delta_i \sim N(0, \sigma^2 I)$$

- where

$$D\big(f(x_i), f(x_i + \delta_i)\big) = \sum_j P(\hat{y} = j | x_i) \log P(\hat{y} = j | x_i + \delta_i)$$

- is the cross-entropy loss
- Encourages smoothness of f in the neighborhood of each training example x_i.

Analysis

- **Problem:** Adversarial examples at test time
- **Importance:** No justification (assumes reader already agrees)
- **Claims:**
  - Claim 1: A better adversarial robustness guarantee than [18]
  - Claim 2: Training strategy inspired by the analysis that improves the bounds in practice
- **Evidence:**
  - Formal statement of robustness result with proof
  - Experimental evaluation of Stability Training with Noise (STN)

State of Knowledge

- Robustness to changing distributions
  - Criticism: "divergence between distributions is rarely used as an empirical measure of strength of adversarial attacks" is weak. Popularity is not a good scientific reason to study something
- Existing guarantees only work under narrow conditions
  - single hidden layer ReLU
  - feed-forward only
- These methods have been generalized somewhat. [18] connects adversarial robustness to differential privacy but the bound is loose
- Previous analysis has used concentration of measure; we use Renyi divergence instead

Story Structure

- Previous work
- Preliminaries
  - Notation
  - Concepts (Renyi Divergence)
  - State of the art in provable robustness
  - State of the art in empirical robustness
- Test time robustness:
  - Main theoretical result and proof
- Training time robustness:
  - Stability training with noise
- Experiments
  - Test time certificates of robustness
  - Training time experimental robustness (with and without test time robustness)

Life Cycle Stage

- Competing Solutions
  - Is it time for a comprehensive evaluation?
  - Create a web site where controlled experiments can be run?

Exercise 4: To do at home

- Outline one of your papers using this framework

Writing Hints to Study At Home

Citations in Text

- Do not treat a citation as a word in a sentence
  - WRONG: “[5] has shown that decision trees can match the accuracy of MLPs”
  - RIGHT: “Dietterich [5] has shown that decision trees can match the accuracy of MLPs”
  - This treats people as doing the research rather than papers
- Make captions self-contained so that the reader can understand a figure by reading the caption

Dietterich's Rules of English

1. Avoid "use". Try "apply", "employ", "select", "perform", "execute", "choose", "evaluate", etc.
2. "Utilize" should refer only to resources ("...fully utilize memory bandwidth...")
3. Avoid contractions.
4. Use English equivalents of Latin phrases outside of parentheses.
   - Replace "etc." with "and so on", "i.e." with "that is", "e.g." with "for example", and "vs." with "versus".
5. Obey parallel form: "The project seeks to develop new methods and to implement them."
   - Parallelize on infinitives ("to develop", "to implement"), on noun phrases ("seeks to develop new algorithms, new implementations, and new results"), on relative clauses ("a new method that will optimize productivity, that will account for computation requirements, and that will minimize communication costs.") and on prepositional phrases (see this sentence itself).

Dietterich's Rules of English: Common Word Problems

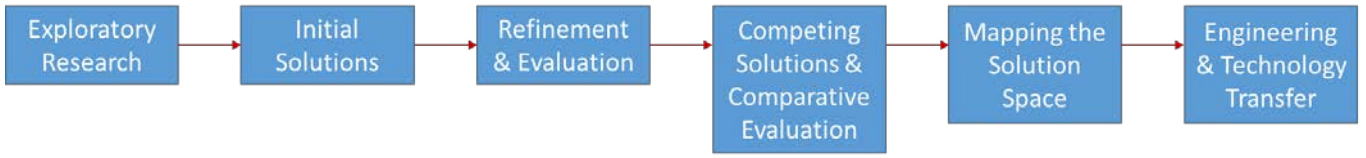
- "affect" (verb) versus "effect" (noun).
- "that" (introduces restrictive relative clause) versus "which" (introduces unrestrictive relative clause).
- Example: "The First iteration that finds a non-null element causes an error message to be displayed.". The phrase "that finds a non-null element" helps identify the iteration in question. If we omit this phrase, the meaning is lost.
- However, consider "Our Meiko CS-2, which was funded by a grant from NSF, has sixteen high-speed processors." The phrase "which was funded by a grant from NSF" tells us something incidental to the main clause. It can be deleted without creating confusion about the identity of the subject of the sentence.
- "between" (relates 2 things) versus "among" (relates >2 things).
- Possessive pronouns. Compare "it's" and "its", "who's" and "whose". The possessive forms are "its" and "whose". The others are contractions ("it's" means "it is", "who's" means "who is").
- Use "or" only when you mean it. Often "and" is clearer.
- "led" is the past tense of "lead". "lead" (pronounced like "led") is a chemical element with a rather low melting point.

Dietterich's Rules of English: Common Syntax Problems

- A colon must be preceded by a complete clause. "There are three methods: walking, running, and flying." is correct. "The three methods are: walking, running, and flying." is wrong.
- Commas separate complete clauses (typically introduced by "and", "but", "therefore", "because", "since", etc.). "This proposal shows important problems, and it presents several solutions." If the "it" is deleted, the comma preceding the "and" should be deleted also.
- Commas set off lead-in phrases. "In this proposal, we discuss...".
- Commas separate lists of three or more items. "Walking, running, and flying."
- Commas set off non-restrictive clauses. "This proposal, which was written for CS519, is excellent."
- Commas break up competing adjectives. "A large, very red car" or "Object-based, portable, programming environment."
- Semicolons. These are used to separate two closely-related complete sentences. "Processor speed must be more than a single number describing a computer; it must be a function of the work being done."
- "em" dash ("—"). These are very emphatic separators. They can separate complete sentences or just sentence fragments. "Vector units—such as the 100Mflops units on the Meiko CS-2—complicate the analysis." "It is difficult to see how to proceed—something must be done!"
- Hyphens. These are used to prevent ambiguity, especially for compound adjectives. "low-latency connection", "run-time performance", "machine-learning algorithm", and "problem-solving system" are examples. Note that when these are not used as adjectives, they are not hyphenated. "The connection has low latency." "The code is executed at run time." Beware of the word "speedup". It is never hyphenated. "Speedup" is a quantity (e.g., "a speedup of 25.") or an adjective ("speedup learning"). "Speed up" is a verb (e.g., "We must find a way to speed up this algorithm.").
- The word "each" is wonderful. It lets you switch from plural to singular to avoid ambiguity. "The computer contains 16 processors, each of which has two vector units."

Wrap Up

- Research Life Cycle



- Skills

	Exploratory Research	Initial Solutions	Refinement & Evaluation	Competing Solutions	Mapping Solution Space	Engineering & Deployment
Reading Literature	X	X	X	X	X	
Analysis Techniques			X	X	X	
Theorem Formulation			X	X	X	
Algorithm design	X	X	X	X		
Coding & Testing		X	X	X		X

- Paper = Claim + Evidence + Story