How to make your contribution?

Do something, and write a paper

Pascal Sun

UWA NLP-TLP Group

The University of Western Australia

May 7, 2025

Table of Contents

Where is the gap?

Types of Papers in Machine Learning

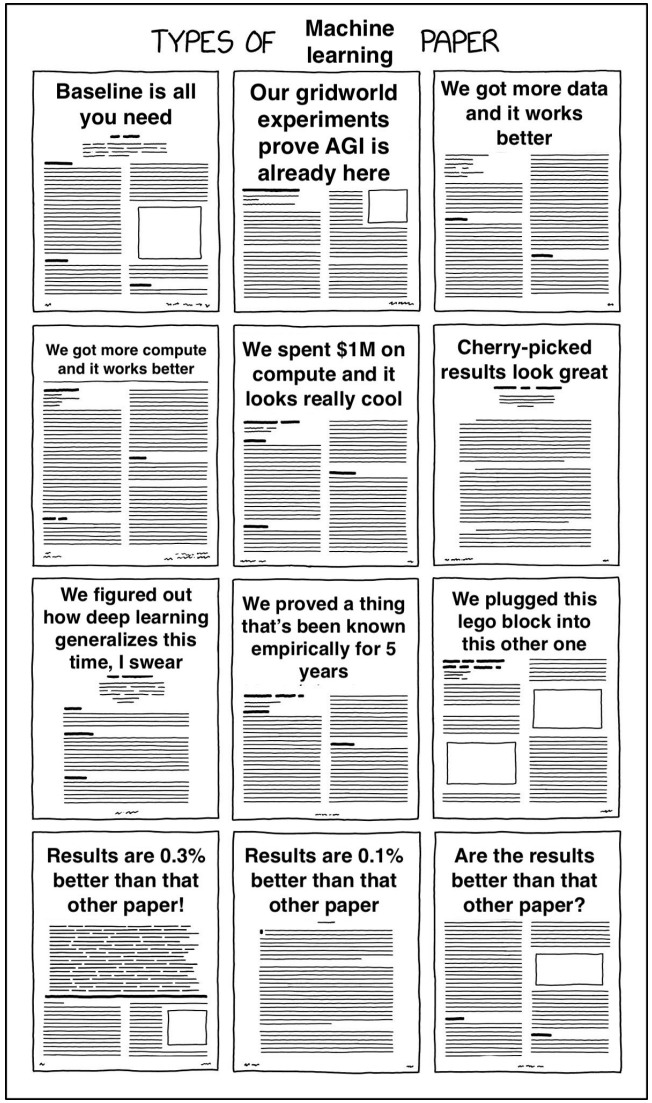

Figure: A joke summary from Reddit[1]

Classification of Contributions

Quote from the NeurIPS 2020 Reviewer Guidelines:

> *"There are many examples of contributions that warrant publication at NeurIPS. These contributions may be theoretical, methodological, algorithmic, empirical, connecting ideas in disparate fields ('bridge papers'), or providing a critical analysis (e.g., principled justifications of why the community is going after the wrong outcome or using the wrong types of approaches)."* [2]

[2] `https://neurips.cc/Conferences/2020/PaperInformation/ReviewerGuidelines`

Wobbrock's 7 Contribution Types (Overview) [3]

- Empirical
- Artifacts
- Methodological
- Theoretical
- Datasets
- Survey
- Opinion

[3] https://www.cs.umd.edu/hcil/chi/ResearchContributionTypes.pdf

1. Empirical Contributions

Definition:

- New findings based on systematically observed data (can be quantitative, qualitative, or mixed)
- Typically derived from studies such as lab experiments, field observations, or pilot deployments

Examples:

- Large-scale experimentation comparing deep learning architectures on standard benchmarks
- Ablation studies or hyperparameter sweeps to isolate key factors in model performance
- Real-world deployment evaluations in areas like robotics or autonomous driving

2. Artifacts (Tools or Systemic Inventions)

Definition:

- Novel systems, architectures, tools, techniques, or designs that reveal new opportunities
- Enable new outcomes, insights, or practical applications

Examples:

- A new text processing package that excels on sepcific NLP tasks
- An interpretability tool or library for visualizing model decisions
- A specialized software framework or hardware accelerator for faster ML training
- Sometimes, it works as it works, you can say it from your intuition, you do not well explain why it works if you can not.

3. Methodological Contributions

Definition:

- New or refined research methods to enhance how ML/AI work is done
- Empower scientists to discover new findings and developers to build more robust solutions
- This part normally slightly driven by mathematics, you will need to explain why you do this, and how it works.

Examples:

- A novel optimization algorithm with faster convergence on large datasets
- Advanced hyperparameter tuning strategies (e.g., Bayesian optimization, population-based training)
- Best-practice protocols ensuring reproducible ML experiments

4. Theoretical Contributions

Definition:

- New models, principles, concepts, or frameworks for understanding ML/AI
- Can be quantitative or qualitative, but structured for future exploration

Examples:

- Novel convergence proofs or generalization bounds in deep learning
- Formalizing the interpretability or causality within neural networks
- Foundational models or proofs explaining why certain architectures generalize well

5. Dataset Contributions

Definition:

- Providing a new and useful corpus or benchmark for the ML/AI community
- Often accompanied by in-depth analysis of the dataset's characteristics

Examples:

- A large-scale multilingual speech dataset covering underrepresented languages
- Synthetic adversarial images to test model robustness
- A curated medical image collection with expert annotations for disease diagnosis research

6. Survey Contributions

Definition:

- Review and synthesize prior work in a research field, highlighting trends and gaps
- Organize existing literature to reflect on its significance and open challenges

Examples:

- A comprehensive survey of adversarial attack/defense strategies in computer vision
- A review analyzing reinforcement learning algorithms across standardized tasks
- A large-scale meta-study of transformer-based architectures in NLP

7. Opinion Contributions

Definition:

- Persuasive argumentation to influence readers' perspectives
- Often draws on findings from other contribution types to support the viewpoint

Examples:

- Advocating for transparent ML model reporting to address bias and ethical concerns
- Critical discussion on the alignment problem between benchmark goals and real-world needs
- Highlighting overlooked issues in large-scale language models, e.g., privacy or misinformation risks

Artifacts vs. Methodological vs. Theoretical Contributions

Aspect	Artifact	Methodological	Theoretical
Focus	Create a usable system, tool, or model	Propose a new way to conduct research or experimentation	Develop new models, principles, or conceptual frameworks
Output Type	System, architecture, framework, or tool	Procedure, training method, evaluation strategy	Mathematical theory, formal framework, abstract model
Purpose	Enable new capabilities or applications	Improve the process of doing ML research or practice	Explain or predict ML phenomena at a deeper level
Examples	SHAP, PyTorch	Bayesian optimization, robustness protocol	Generalization bounds, convergence proof, causal models

What Makes a Scientific Contribution?

To claim a scientific contribution (Publication), your work should be:

- **Novel**
  Have you come up with something that is *new* compared to prior research?
- **Significant**
  Have you solved a *relevant and non-trivial problem* with real-world impact?
- **Valid**
  Have you used *appropriate methods* (e.g., data collection, analysis, technology development)?

Some arugment points in Modern ML/AI

- **Explainability**
  - Domain explainability: Making results interpretable to domain experts
  - Model explainability: Understanding internal model decisions
- **Responsibility**
  - Measuring and mitigating data-introduced bias
  - Privacy protection and data minimization
- **Robustness**
  - Resilience against adversarial attacks
  - Stable performance under domain shift
- **Domain-Agnostic Methods**
  - Generalization across multiple domains/tasks
  - Transfer learning with minimal adaptation
- **Scalability**
  - Linear/sublinear growth in computational requirements
  - Efficient inference on resource-constrained devices

Problem Formulation

Your work like a fish

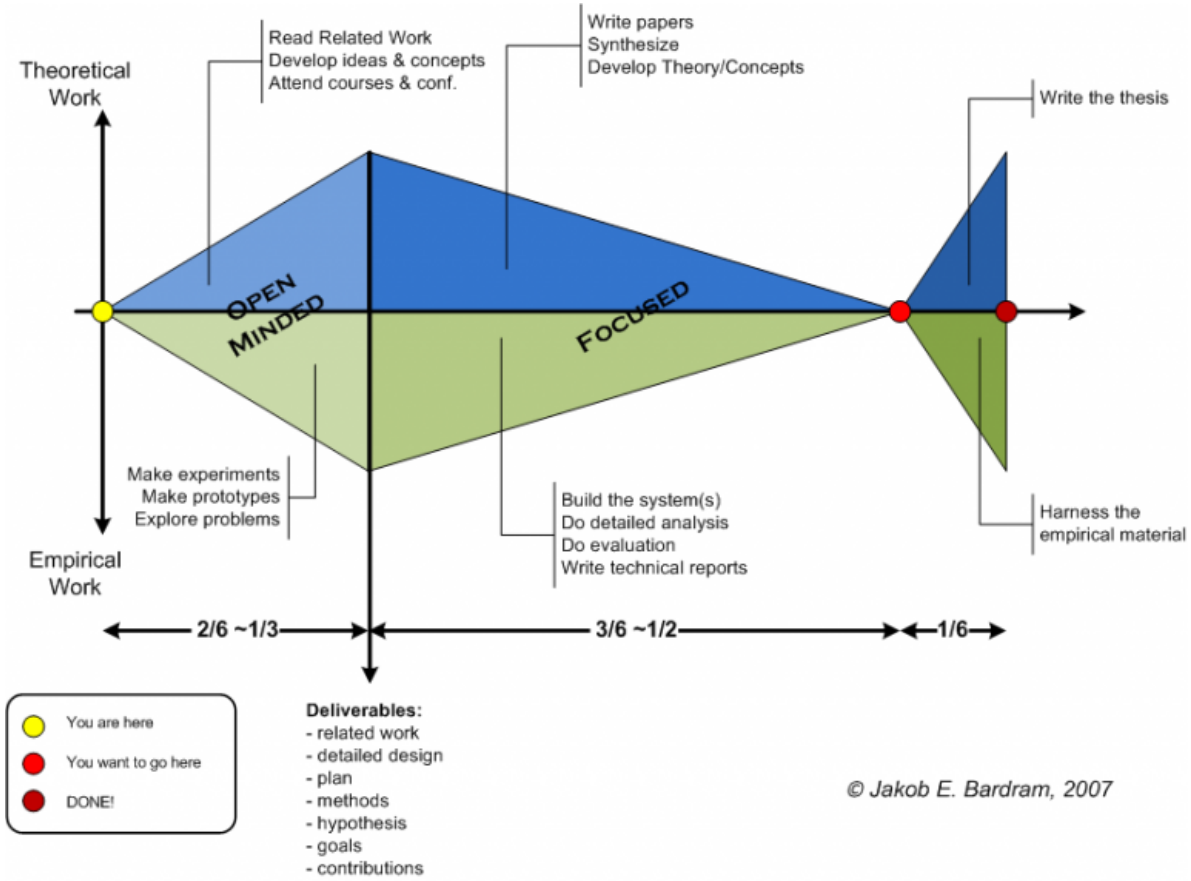

Figure: Illustration regarding the work you need to do [4]

[4] https://www.cs.umd.edu/hcil/chi/ResearchContributionTypes.pdf

What is Problem Formulation?

Problem formulation is the process of transforming a real-world challenge into a precise, well-defined research task.

It is a critical step because:

- It defines the **scope** and **goals** of your research.
- It identifies the **key factors** involved — inputs, outputs, variables, constraints.
- It leads directly into a formal mathematical representation.
- It informs and guides your **experimental design**.

Effective Problem Formulation: Key Elements

An effective problem formulation includes:

1. **Context** - What's the broader field and application domain?
2. **Gap identification** - What's missing in current approaches?
3. **Precise statement** - Clear, concise definition of what you're solving
4. **Variables** - What are the inputs, outputs, and parameters?
5. **Constraints** - What limitations or requirements exist?
6. **Success criteria** - How will you know if the problem is solved?

Example: Instead of "Improve speech recognition," formulate as "Develop a noise-robust speech recognition algorithm that maintains >95% accuracy in environments with SNR below 10dB."
Or reduce the computational cost by 50% while maintaining the same accuracy.

Common Pitfalls in Problem Formulation

- **Too broad** - "Improve deep learning" lacks specificity
- **Too narrow** - Overly specific problems may lack significance
- **Misaligned with evaluation** - Problem definition doesn't match how you measure success
- **Untestable** - Cannot be empirically validated
- **Disconnected from theory** - Lacks grounding in scientific principles
- **Solving the wrong problem** - Addressing symptoms rather than root causes

Key insight: A well-formulated problem is already halfway to its solution.

From Research Idea to Math Representation

Steps to formalize a research problem:

1. **Identify the real-world problem**
   - e.g., Recommend products users will likely purchase
2. **Define key factors: inputs, outputs, assumptions**
   - e.g., User history x, next item y, user/item features
3. **Translate to mathematical representation**
   - Define a function $f_\theta(x) \rightarrow y$ with parameters θ
4. **Specify objective and constraints**
   - e.g., Minimize loss $\mathcal{L}(f_\theta(x), y)$
   - Add constraints (e.g., fairness, regularization, sparsity)

A clear formulation supports a reproducible, measurable, and focused experimental design.

Examples of Problem Formulations in ML/AI

1. **Classification Problem**
   - *Task*: Predict if an email is spam or not
   - *Formulation*: Given input $x \in \mathbb{R}^n$, predict label $y \in \{0, 1\}$ using a model f_θ
   - *Loss*: Cross-entropy loss between predicted and true labels
2. **Regression Problem**
   - *Task*: Predict the price of a house
   - *Formulation*: Learn a function $f_\theta(x) \to y$, where $y \in \mathbb{R}$
   - *Loss*: Mean Squared Error (MSE)
3. **Reinforcement Learning**
   - *Task*: Learn to play a game
   - *Formulation*: Maximize expected reward $\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$
   - *Includes*: States, actions, policy $\pi(a|s)$, reward r

Design your experiment

Why Experimental Design Matters

Figure: The apple

Why Experimental Design Matters

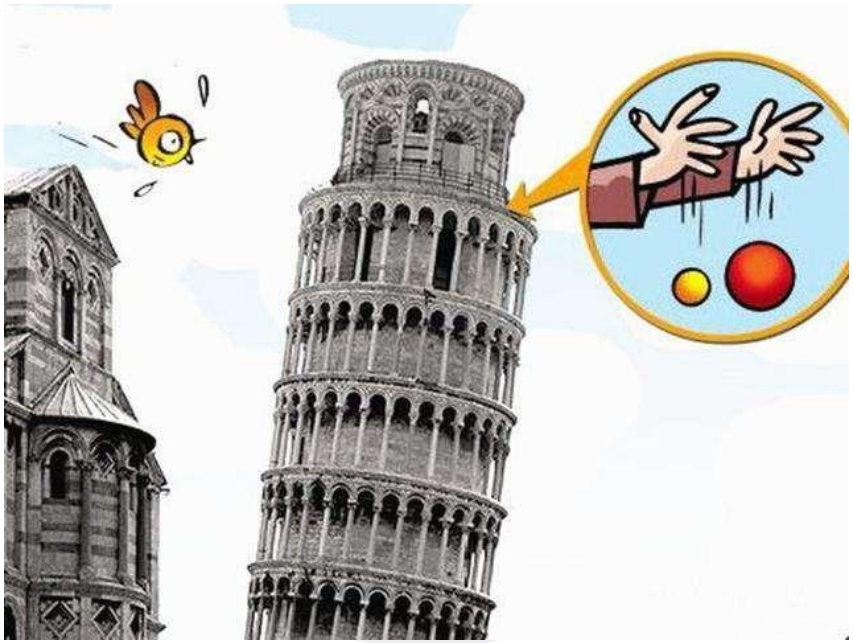

Figure: A very good experiment design example

Why Experimental Design Matters

- Experimental design is at the heart of empirical machine learning research.
- Good experiments lead to trustworthy, insightful conclusions.
- Poorly designed experiments can result in misleading claims, wasted effort, or unreproducible results.

Start with a Research Question

- Always begin with a clear hypothesis or question.
- What are you trying to evaluate, compare, or prove?
- Examples:
  - "Does model A outperform model B on task X?"
  - "Does adding module Z improve generalization?"
- Your experimental setup should be directly aligned with your research goal.

Benchmarking and Dataset Selection

- Choose datasets that are:
  - Relevant to the problem
  - Well-established (for comparability)
  - Properly partitioned (train/val/test or cross-validation)
- Be aware of data leakage, imbalance, or redundancy.
- Consider multiple datasets to validate generalizability.

Algorithm Evaluation Setup

- Use a fair and repeatable evaluation protocol:
  - Fix random seeds
  - Compare against strong baselines
  - Ensure consistent data splits
- Run multiple trials with different seeds and report mean + std
- Document all implementation details clearly

Evaluation Metrics

- Choose metrics aligned with your task and goals:
  - Classification: Accuracy, F1, AUC
  - Regression: RMSE, MAE
  - Ranking: NDCG, MAP
- Use multiple metrics when appropriate.
- Be cautious of misleading metrics (e.g., accuracy in imbalanced data).

Hyperparameter Tuning

- Always separate tuning from test evaluation!
- Common strategies:
  - Grid search
  - Random search
  - Bayesian optimization
- Use a validation set or cross-validation to tune.
- Apply consistent tuning procedures across models.

Ablation Studies

- Remove or modify individual components to isolate their impact.
- Example: Evaluate a model with and without attention layer.
- Useful for understanding which parts of the system matter most.
- Make it clear how each variant differs from the full model.

Fairness, Efficiency, and Scalability

- Beyond performance, consider:
  - **Fairness**: Does it work equally well for different groups?
  - **Efficiency**: How much compute/memory is required?
  - **Scalability**: Can it handle more data, longer sequences, etc.?
- These factors are critical for real-world impact.

Reproducibility Best Practices

- Keep your experiments reproducible:
  - Fix random seeds
  - Use version-controlled code and configs
  - Log all settings, metrics, and dependencies
- Package code and data for easy re-run by others (e.g., GitHub, Colab, HuggingFace Spaces)

Reporting Results

- Be transparent and thorough:
  - Report average + variance over multiple runs
  - Include baseline and ablation comparisons
  - Document experimental setup in full detail
- Avoid cherry-picking best results
- Consider reproducibility checklists from top conferences (e.g., NeurIPS, ICLR)

Conclusion and Takeaways

- Experimental design is a key skill in ML research.
- Start with clear questions and fair comparisons.
- Choose relevant datasets, metrics, and tuning strategies.
- Use ablation and analysis to deepen understanding.
- Report results honestly and reproducibly.

Good experiments lead to good science.

Demonstration

Problem Formulation

- **Temporal Knowledge Graph Embedding (TKGE)** aims to perform link prediction within a temporal context.
- **Objective**: Predict the existence of missing links between two entities at a given timestamp.
- **Input**: A temporal knowledge graph consisting of entities, relations, and timestamps
- **Output**: A score representing the likelihood of a link existing between two entities at a specific time.
- **Mathematical Formulation**: Typically grounded in literature—adjustments can be made to fit specific temporal reasoning tasks.

Dataset

- There are several research in the literature, for example this one, people tend to use ICEWS14 and ICEWS05-15, GDELT, etc.
- And we do find out a code baseline everyone working on, so we can copy and start to get the pipeline running.

Evaluation metrics

- We want to check the MRR, Hits@1, Hits@3, Hits@5, Hits@10 => Hits@K.
- Also other aspects people care about, including the time complexity, the space complexity, etc.
- Also novelity can comes from how to design your embedding space, etc. For your research, you can spot your novel contribution after the literature review.

Code baseline

- We can copy the code baseline, and start to get the pipeline running.
- We can also try to understand the code baseline, and see how to improve it.
- I actually spot a bug which cause the evaluation metrics higher than it should be, and I fixed it, this will happen in the real world. And this is why people like to open code access.
- Then do your innovation part, for us it is the temporal graph embedding model itself.
- You can try to first understand why the previous series of work working, then summarize that, get some inspiration.
- This is the small contribution part you will make to the world if you are successfully get the result a bit better from any perspective.

Let's discuss your problem

Step 1: Problem Definition

1. **Problem Type:** Is it classification, regression, generation, or something else?
2. **Input/Output:** What goes in? What comes out?
3. **Key Factors:** What are the variables, parameters, and constraints?

Step 2: Mathematical Formulation

- Define your problem in mathematical notation
- Specify the objective function you want to optimize if you have a clear one
- If you are not specifically focus on model development, define clear about what is the evaluation metrics you want to pursue.
- Clearly state any constraints or assumptions

Step 3: Literature Review & Code Setup

- **Locate similar work:** Find papers with approaches related to your problem
- **Find code baselines:** Identify repositories you can build upon
- **Set up your pipeline:** Get the basic workflow running

Don't reinvent the wheel—stand on the shoulders of giants

Step 4: Empirical Framework

- **Datasets:** Select appropriate benchmarks or create your own
- **Evaluation metrics:** Define how you'll measure success
- **Baseline comparison:** Establish strong baselines to compare against

A robust evaluation framework ensures your results are meaningful

Step 5: Innovate & Contribute

- Understand why existing approaches work (or don't)
- Identify specific areas for improvement
- Implement your novel ideas
- Conduct rigorous experiments and analysis

Your contribution, however small, moves the field forward

Q&A

Q&A

Questions?

Feel free to ask anything