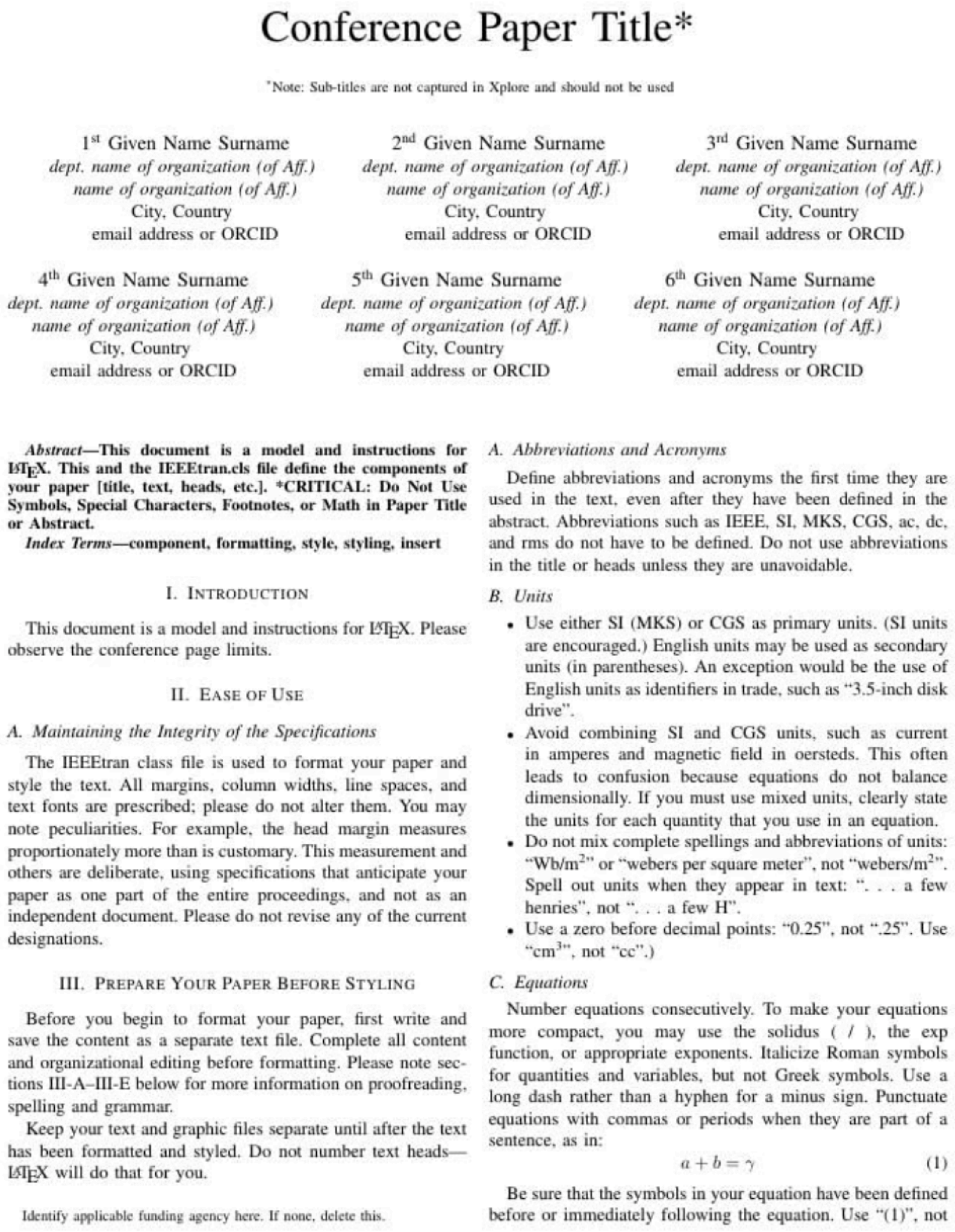
Paper Structure

Mar 22nd

Common Structure

- Abstract
- Introduction
- Preliminary Description
- Methodology
- Experiments and Results
- Related Work
- Conclusion and Future Work



Conference Paper Title*

*Note: Sub-titles are not captured in Xplore and should not be used

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

2nd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

3rd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

4th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

5th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

6th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Abstract*—This document is a model and instructions for LaTeX. This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. *CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

This document is a model and instructions for LaTeX. Please observe the conference page limits.

II. Ease of Use

A. Maintaining the Integrity of the Specifications

The IEEEtran class file is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

III. Prepare Your Paper Before Styling

Before you begin to format your paper, first write and save the content as a separate text file. Complete all content and organizational editing before formatting. Please note sections III-A–III-E below for more information on proofreading, spelling and grammar.

Keep your text and graphic files separate until after the text has been formatted and styled. Do not number text heads—LaTeX will do that for you.

A. Abbreviations and Acronyms

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, ac, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

B. Units

- Use either SI (MKS) or CGS as primary units. (SI units are encouraged.) English units may be used as secondary units (in parentheses). An exception would be the use of English units as identifiers in trade, such as "3.5-inch disk drive".
- Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.
- Do not mix complete spellings and abbreviations of units: "Wb/m^2" or "webers per square meter", not "webers/m^2". Spell out units when they appear in text: ". . . a few henries", not ". . . a few H".
- Use a zero before decimal points: "0.25", not ".25". Use "cm^3", not "cc".)

C. Equations

Number equations consecutively. To make your equations more compact, you may use the solidus (/), the exp function, or appropriate exponents. Italicize Roman symbols for quantities and variables, but not Greek symbols. Use a long dash rather than a hyphen for a minus sign. Punctuate equations with commas or periods when they are part of a sentence, as in:

$$a + b = \gamma \tag{1}$$

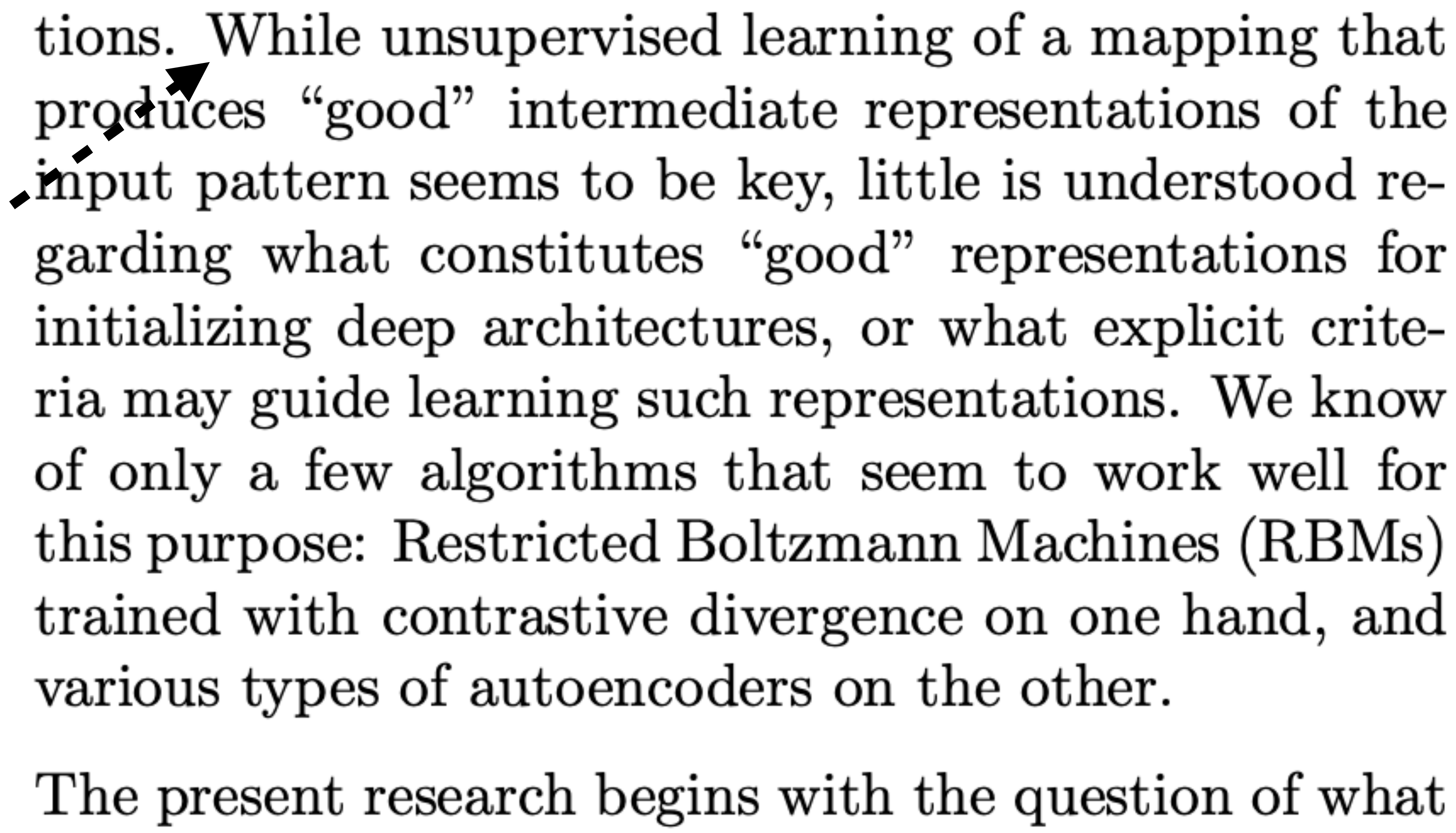
Be sure that the symbols in your equation have been defined before or immediately following the equation. Use "(1)", not

Identify applicable funding agency here. If none, delete this.

The IEEE Template

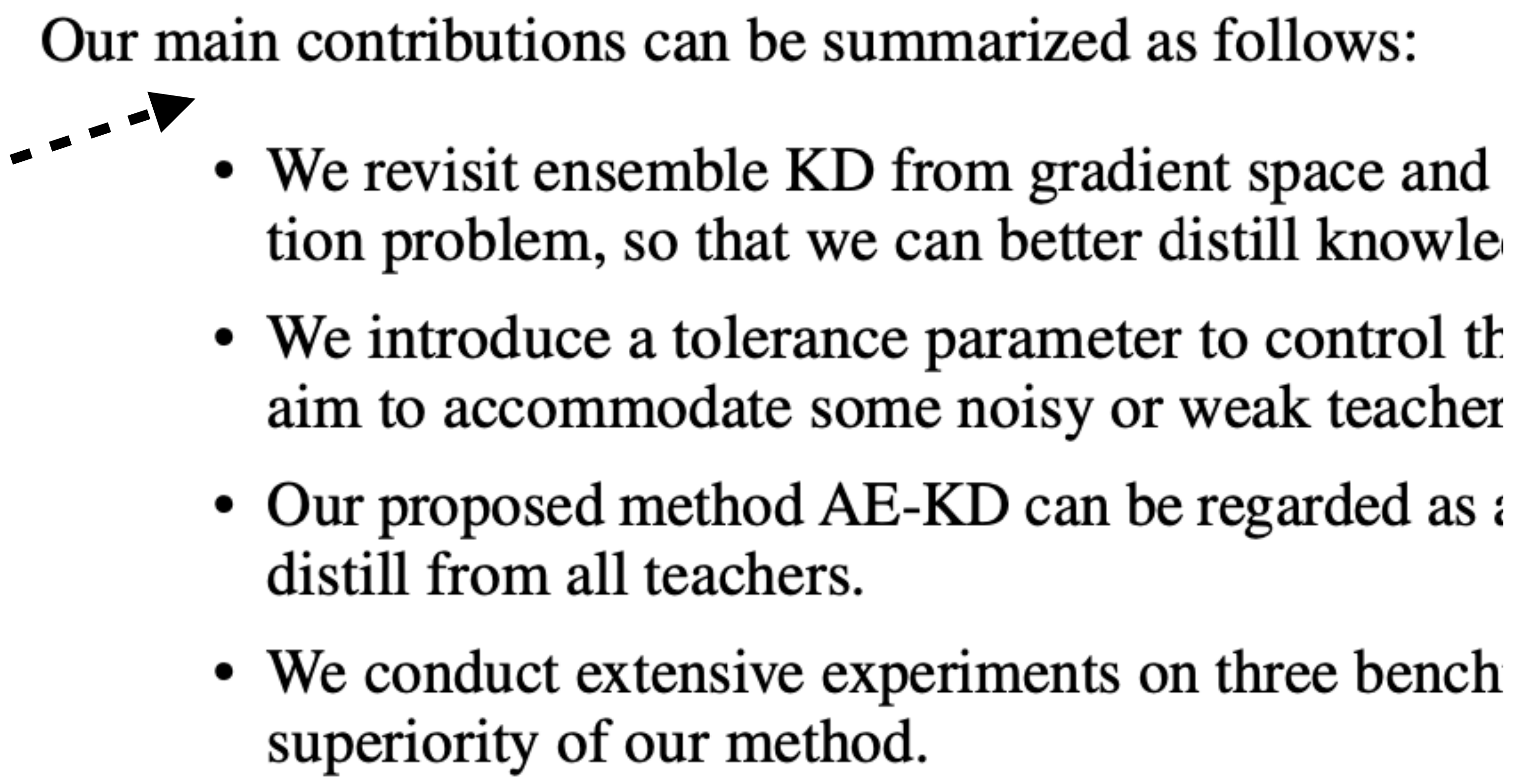
Introduction

- Presents the topic
- Why is it interesting?
- Issues from existing solutions
- Includes some references
- Could highlight interesting results
- Establishes the paper contributions



tions. While unsupervised learning of a mapping that produces "good" intermediate representations of the input pattern seems to be key, little is understood regarding what constitutes "good" representations for initializing deep architectures, or what explicit criteria may guide learning such representations. We know of only a few algorithms that seem to work well for this purpose: Restricted Boltzmann Machines (RBMs) trained with contrastive divergence on one hand, and various types of autoencoders on the other.

The present research begins with the question of what



Our main contributions can be summarized as follows:

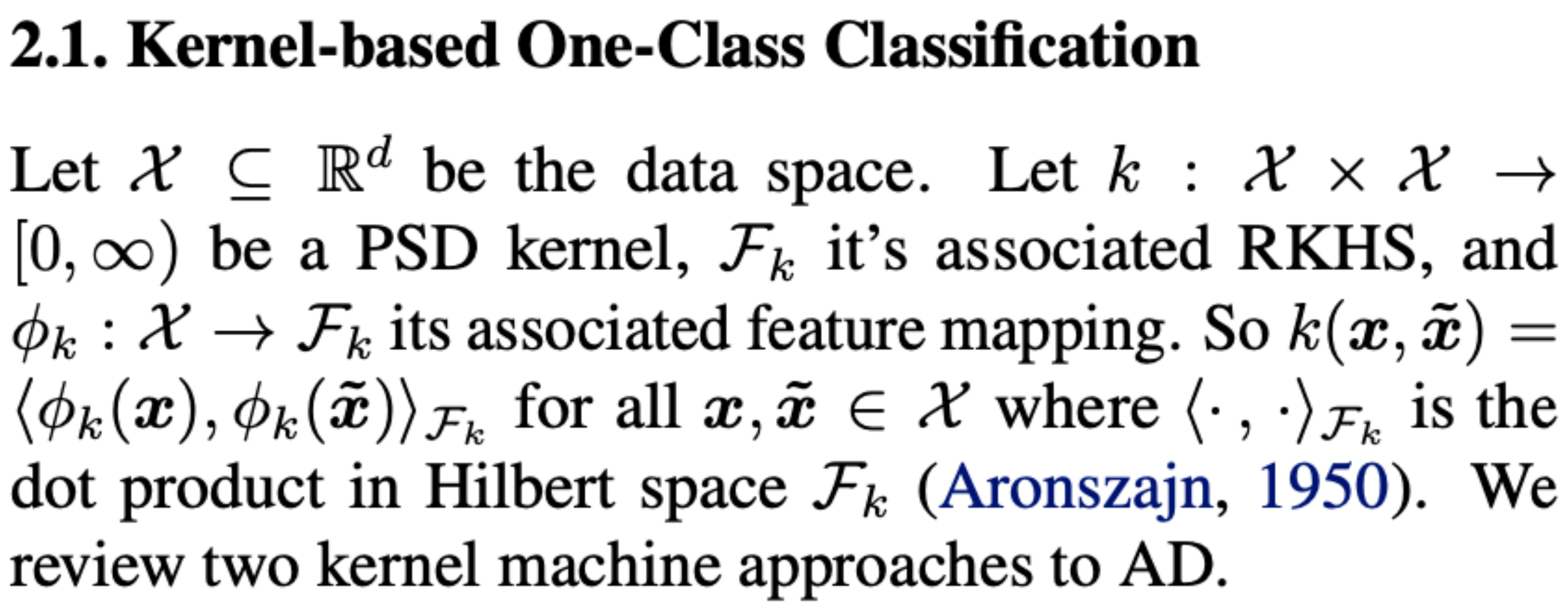
- We revisit ensemble KD from gradient space and tion problem, so that we can better distill knowle
- We introduce a tolerance parameter to control th aim to accommodate some noisy or weak teacher
- Our proposed method AE-KD can be regarded as a distill from all teachers.
- We conduct extensive experiments on three bench superiority of our method.

Introduction - Contributions

- Small construction tools classifier
    - Renting companies, monitoring use, service check
    - Business value
- Broadcast less data, run model in low power device
    - More complex than the threshold
    - Evaluation of different methods
- Manipulation of data, group labels
    - Better predictions

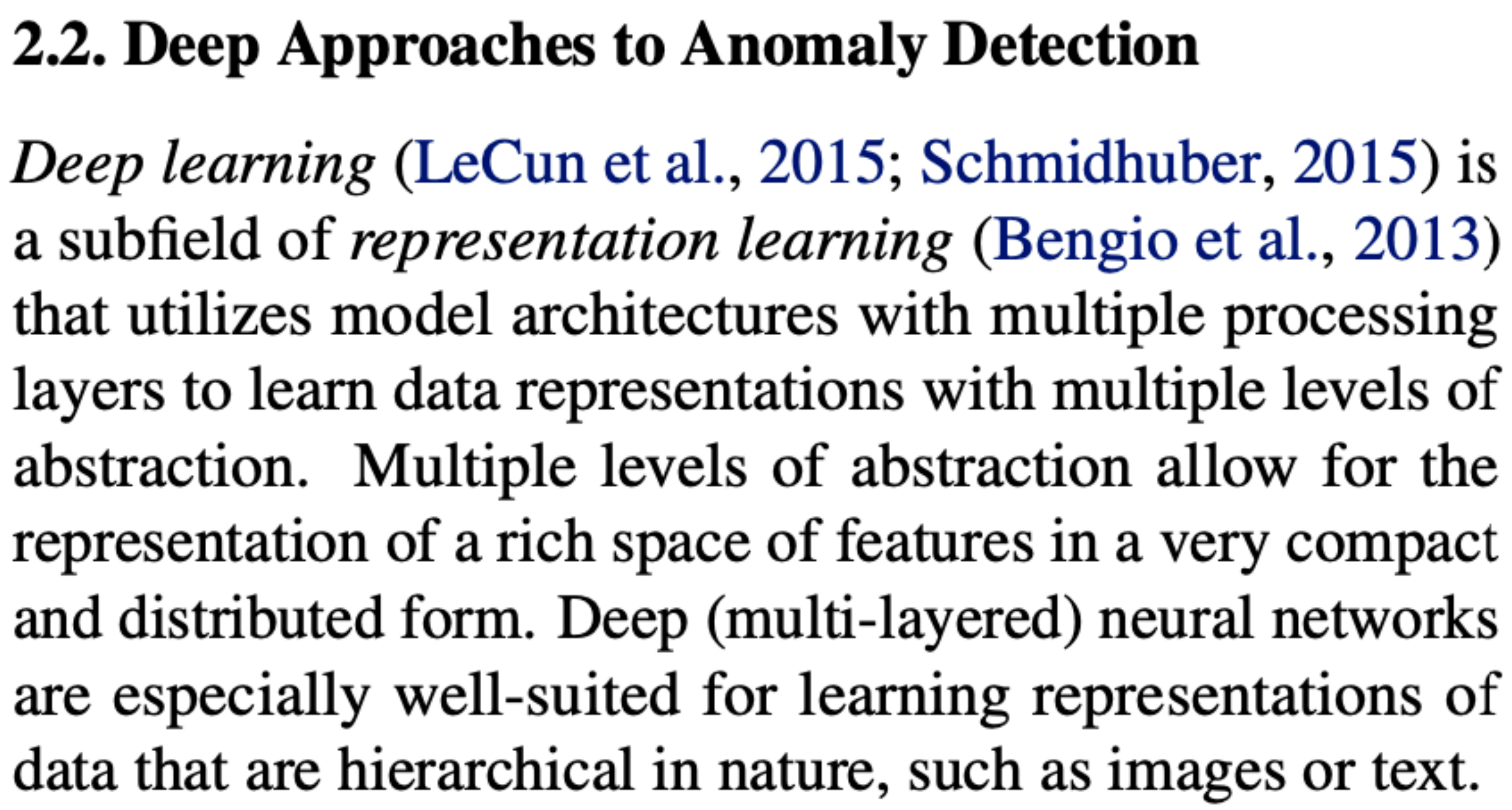
Related Work

- How the models or tools are used in other scenarios?
  - How SVM/Random Forest is used? Is the data normalized?
  - Human activities
- What other people do for the same or a similar problem?
  - Big machines, RNN, LSTM



2.1. Kernel-based One-Class Classification

Let $\mathcal{X} \subseteq \mathbb{R}^d$ be the data space. Let $k : \mathcal{X} \times \mathcal{X} \to [0, \infty)$ be a PSD kernel, $\mathcal{F}_k$ it's associated RKHS, and $\phi_k : \mathcal{X} \to \mathcal{F}_k$ its associated feature mapping. So $k(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \langle \phi_k(\boldsymbol{x}), \phi_k(\tilde{\boldsymbol{x}}) \rangle_{\mathcal{F}_k}$ for all $\boldsymbol{x}, \tilde{\boldsymbol{x}} \in \mathcal{X}$ where $\langle \cdot\,, \cdot \rangle_{\mathcal{F}_k}$ is the dot product in Hilbert space $\mathcal{F}_k$ (Aronszajn, 1950). We review two kernel machine approaches to AD.

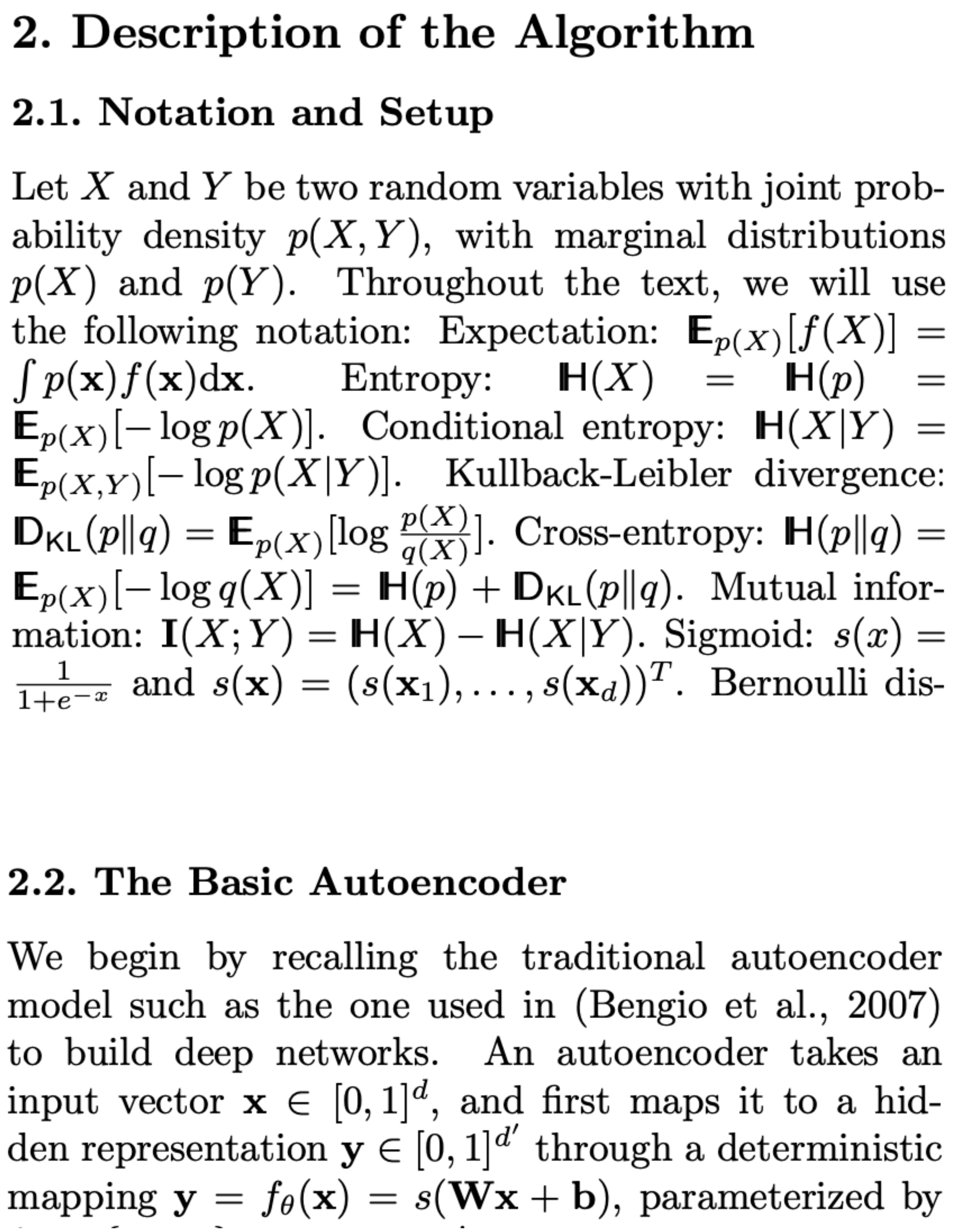


2.2. Deep Approaches to Anomaly Detection

Deep learning (LeCun et al., 2015; Schmidhuber, 2015) is a subfield of *representation learning* (Bengio et al., 2013) that utilizes model architectures with multiple processing layers to learn data representations with multiple levels of abstraction. Multiple levels of abstraction allow for the representation of a rich space of features in a very compact and distributed form. Deep (multi-layered) neural networks are especially well-suited for learning representations of data that are hierarchical in nature, such as images or text.

Preliminary Description

- Introduces notation
- Define necessary concepts - SVM?
  - How time series are defined and used?
  - Special structures or operations
    - Accelerometer data, structure
- Can be merged in other sections



2. Description of the Algorithm

2.1. Notation and Setup

Let X and Y be two random variables with joint probability density $p(X, Y)$, with marginal distributions $p(X)$ and $p(Y)$. Throughout the text, we will use the following notation: Expectation: $\mathbb{E}_{p(X)}[f(X)] = \int p(\mathbf{x}) f(\mathbf{x}) \mathrm{d}\mathbf{x}$. Entropy: $\mathbb{H}(X) = \mathbb{H}(p) = \mathbb{E}_{p(X)}[-\log p(X)]$. Conditional entropy: $\mathbb{H}(X|Y) = \mathbb{E}_{p(X,Y)}[-\log p(X|Y)]$. Kullback-Leibler divergence: $\mathbb{D}_{\mathsf{KL}}(p\|q) = \mathbb{E}_{p(X)}[\log \frac{p(X)}{q(X)}]$. Cross-entropy: $\mathbb{H}(p\|q) = \mathbb{E}_{p(X)}[-\log q(X)] = \mathbb{H}(p) + \mathbb{D}_{\mathsf{KL}}(p\|q)$. Mutual information: $\mathbf{I}(X; Y) = \mathbb{H}(X) - \mathbb{H}(X|Y)$. Sigmoid: $s(x) = \frac{1}{1+e^{-x}}$ and $s(\mathbf{x}) = (s(\mathbf{x}_1), \ldots, s(\mathbf{x}_d))^T$. Bernoulli dis-

2.2. The Basic Autoencoder

We begin by recalling the traditional autoencoder model such as the one used in (Bengio et al., 2007) to build deep networks. An autoencoder takes an input vector $\mathbf{x} \in [0, 1]^d$, and first maps it to a hidden representation $\mathbf{y} \in [0, 1]^{d'}$ through a deterministic mapping $\mathbf{y} = f_\theta(\mathbf{x}) = s(\mathbf{W}\mathbf{x} + \mathbf{b})$, parameterized by

Methodology

- A comprehensive review of the paper contribution
- A detailed description of the method
    - Supported by figures, equations, and pseudocode
    - Mostly technical details
- How all the components are integrated?
    - Justify their use

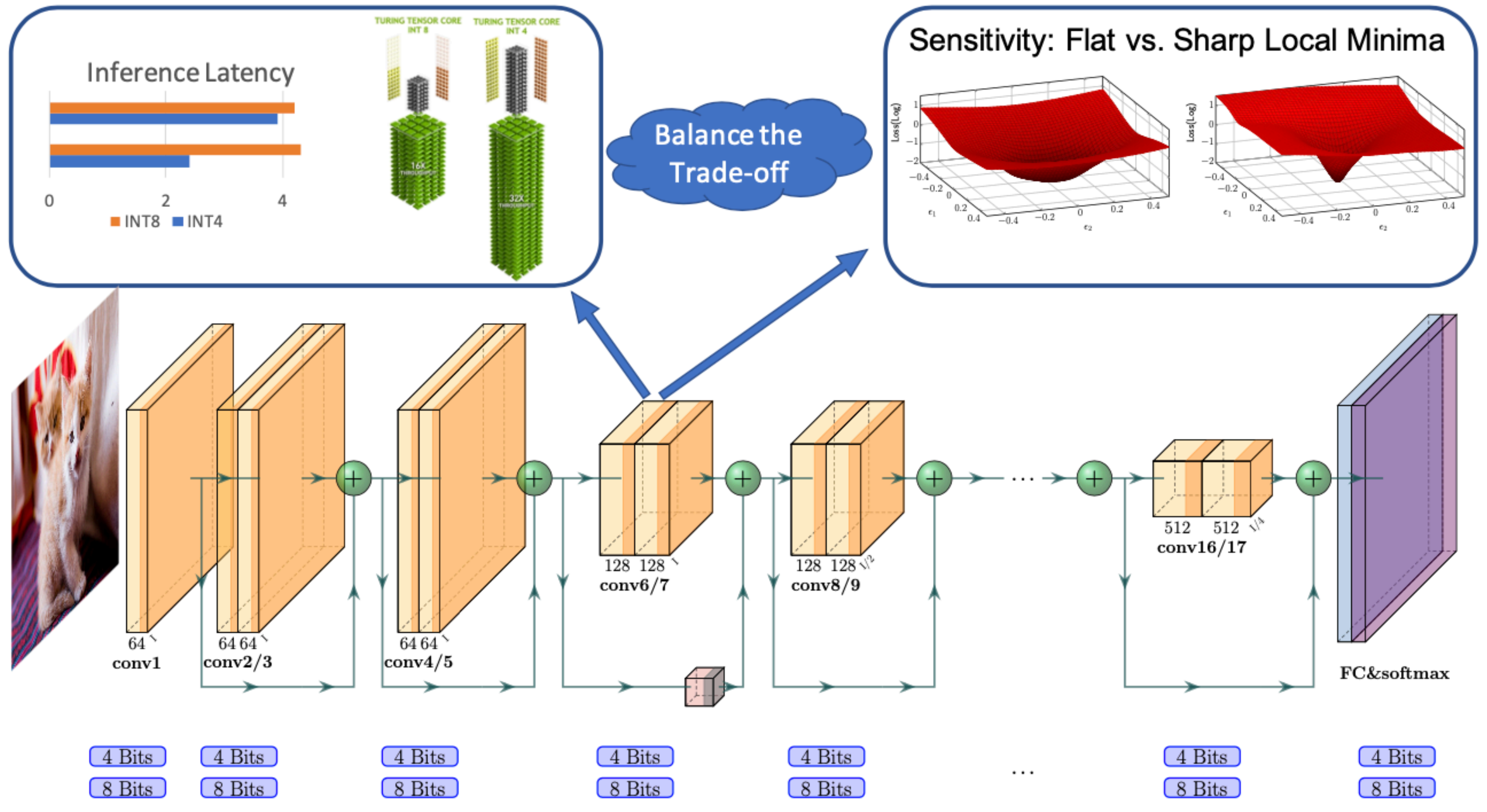


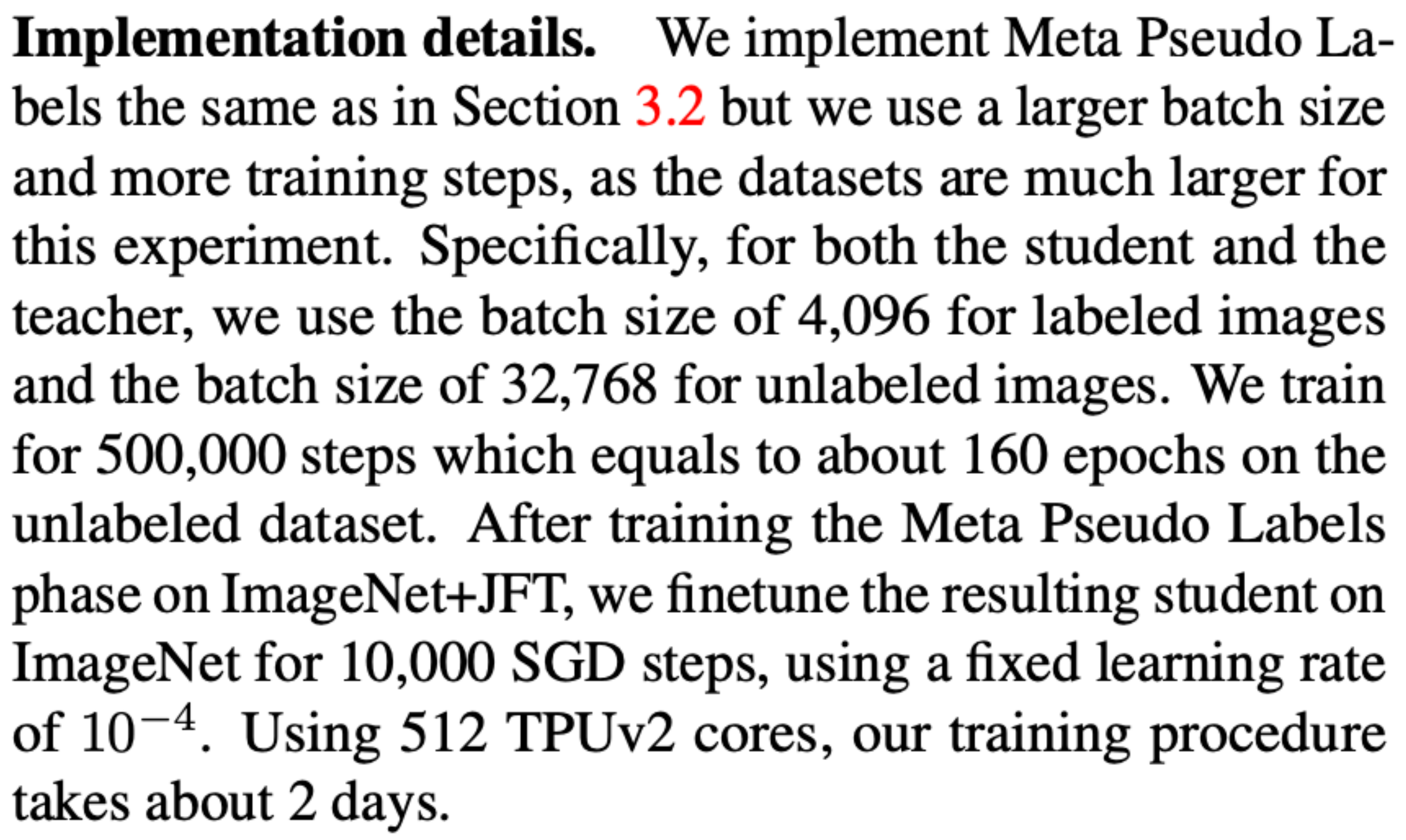
Figure 3. Illustration of inference speed and generalization performance trade-off of ResNet18. For each layer, we need to consider the speedup of INT4 vs INT8 and the sensitivity based on the second order (Hessian) sharpness (Dong et al., 2020) of this layer.

Methodology - Could be included

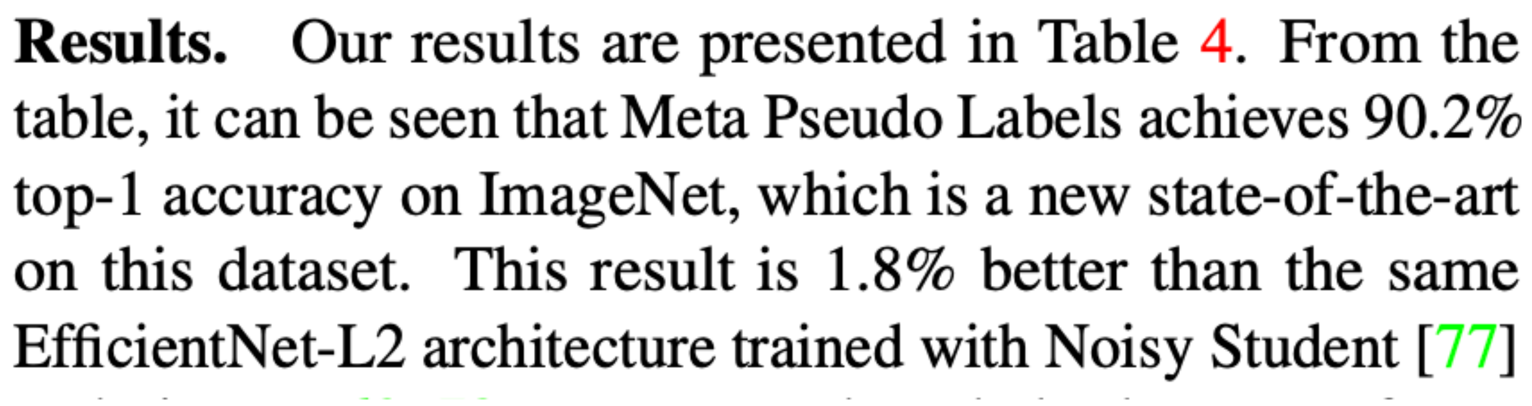
- Data processing
  - Getting data, merging, sampling, removing outliers, among other tasks
- Algorithm testing
  - Scikit-learn libraries to test suitable algorithms
- Device implementation
  - Algorithm migration to C
  - Getting and processing the data from sensors
  - Broadcasting the results

Experiments and Results

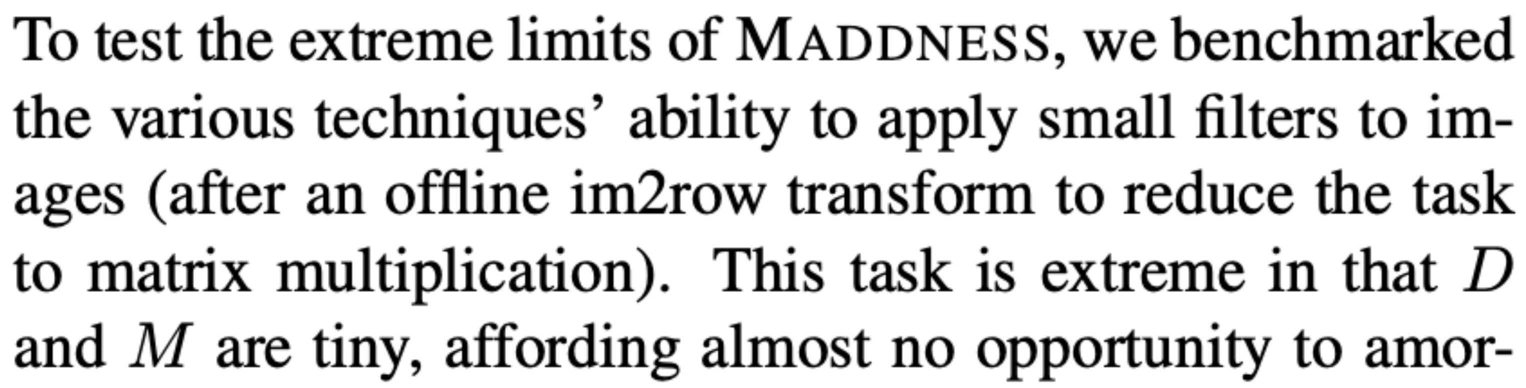
- How the experiments are designed?
  - Setup: machine, software version, data sets, parameters
  - Evaluation: metrics and baseline methods
- Comparison and analysis
  - Tables or charts showing the metrics results for all methods
- Study of other settings
  - Effect of adjustments over the proposal



Implementation details. We implement Meta Pseudo Labels the same as in Section 3.2 but we use a larger batch size and more training steps, as the datasets are much larger for this experiment. Specifically, for both the student and the teacher, we use the batch size of 4,096 for labeled images and the batch size of 32,768 for unlabeled images. We train for 500,000 steps which equals to about 160 epochs on the unlabeled dataset. After training the Meta Pseudo Labels phase on ImageNet+JFT, we finetune the resulting student on ImageNet for 10,000 SGD steps, using a fixed learning rate of 10^{-4}. Using 512 TPUv2 cores, our training procedure takes about 2 days.



Results. Our results are presented in Table 4. From the table, it can be seen that Meta Pseudo Labels achieves 90.2% top-1 accuracy on ImageNet, which is a new state-of-the-art on this dataset. This result is 1.8% better than the same EfficientNet-L2 architecture trained with Noisy Student [77]



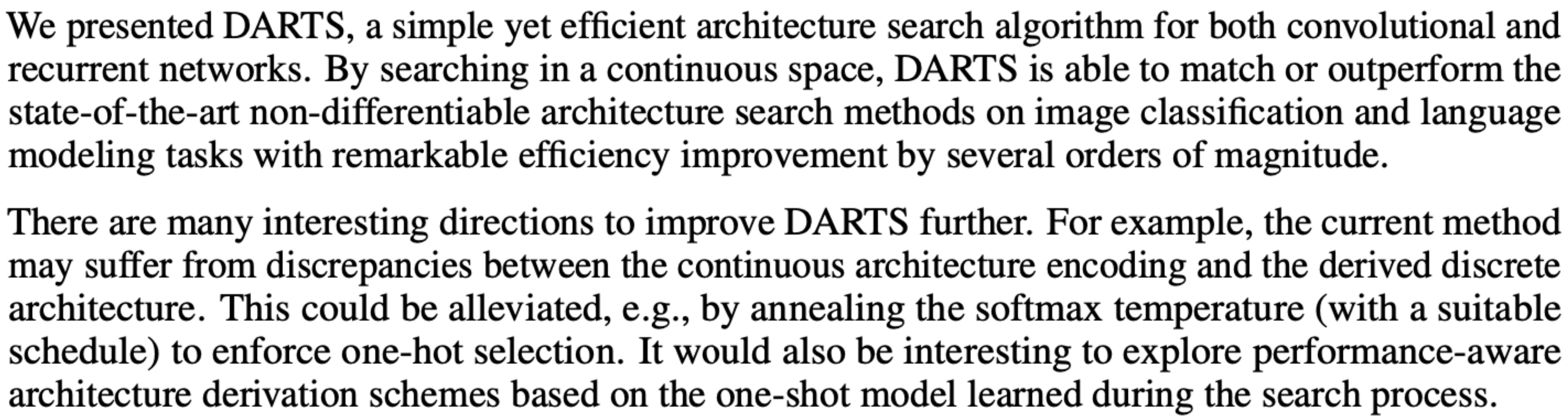
To test the extreme limits of MADDNESS, we benchmarked the various techniques' ability to apply small filters to images (after an offline im2row transform to reduce the task to matrix multiplication). This task is extreme in that D and M are tiny, affording almost no opportunity to amor-

Experiments and Results

- Metrics: accuracy, running time, memory consumption
- Methods: Threshold, Random Forest, LSTM, other libraries
  - Limitations for the implementation on the device
- Results: Table, chart - How the model works with respect to others?
  - Justify the choice. Why a methods could be the best one for the setting?
- Other experiments
  - For example: sampling of ten? Test sampling of 5 and 20 as well

Conclusion and Future Work

- Briefly summarised the paper contribution and findings
- Proposes possible direction for future research

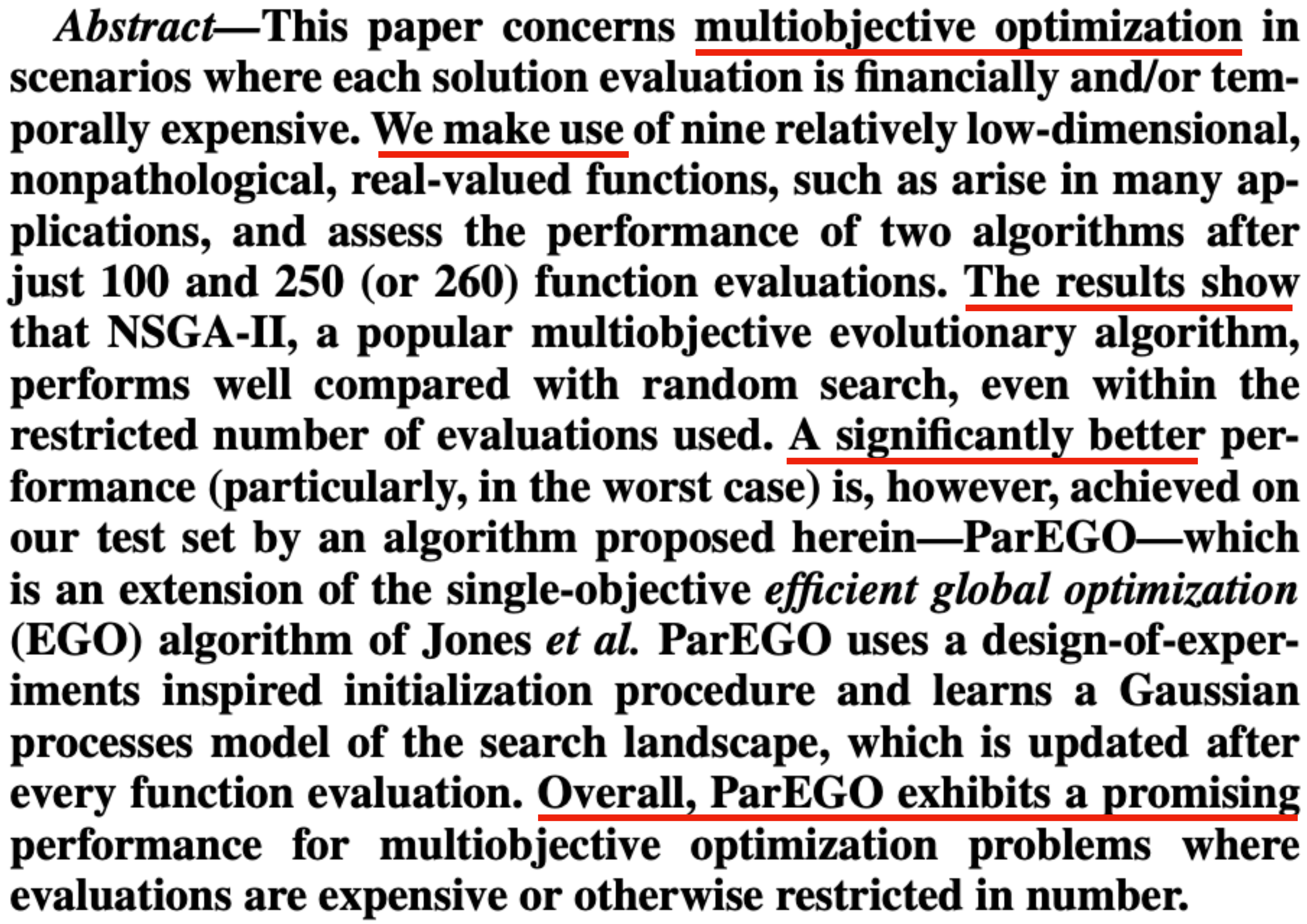


We presented DARTS, a simple yet efficient architecture search algorithm for both convolutional and recurrent networks. By searching in a continuous space, DARTS is able to match or outperform the state-of-the-art non-differentiable architecture search methods on image classification and language modeling tasks with remarkable efficiency improvement by several orders of magnitude.

There are many interesting directions to improve DARTS further. For example, the current method may suffer from discrepancies between the continuous architecture encoding and the derived discrete architecture. This could be alleviated, e.g., by annealing the softmax temperature (with a suitable schedule) to enforce one-hot selection. It would also be interesting to explore performance-aware architecture derivation schemes based on the one-shot model learned during the search process.

Abstract

- Summarises the overall contribution
- Usually around 200 words
- Written after completing most of the paper



Abstract—This paper concerns multiobjective optimization in scenarios where each solution evaluation is financially and/or temporally expensive. We make use of nine relatively low-dimensional, nonpathological, real-valued functions, such as arise in many applications, and assess the performance of two algorithms after just 100 and 250 (or 260) function evaluations. The results show that NSGA-II, a popular multiobjective evolutionary algorithm, performs well compared with random search, even within the restricted number of evaluations used. A significantly better performance (particularly, in the worst case) is, however, achieved on our test set by an algorithm proposed herein—ParEGO—which is an extension of the single-objective *efficient global optimization* (EGO) algorithm of Jones *et al.* ParEGO uses a design-of-experiments inspired initialization procedure and learns a Gaussian processes model of the search landscape, which is updated after every function evaluation. Overall, ParEGO exhibits a promising performance for multiobjective optimization problems where evaluations are expensive or otherwise restricted in number.