Momentum Contrast for Unsupervised Visual Representation Learning

Mohan Vamsi Krishna Yanamadala, Gouthami Nadella, SivaSai Achyut Akella
Department of Computer Science
State University of New York at Binghamton

***Abstract*—Momentum Contrast (MoCo) is a self-supervised learning framework that introduces a dynamic dictionary and a momentum encoder to improve visual representation learning. This paper explores the application of MoCo on the CIFAR-10 dataset, presenting experimental results, a comprehensive analysis of its performance, and comparisons with other self-supervised methods. Our findings highlight the scalability and robustness of MoCo in contrastive learning tasks.**

***Index Terms*—Self-Supervised Learning, Contrastive Learning, Momentum Encoder, Dynamic Dictionary, CIFAR-10.**

I. Introduction

Self-supervised learning has recently emerged as a transformative approach in deep learning, particularly in natural language processing (NLP), where models like GPT [1] and BERT [2] have achieved remarkable success by leveraging vast amounts of unlabeled text data. These methods rely on tasks such as next-word prediction or masked token prediction to learn meaningful representations from unstructured data. However, the application of self-supervised learning to computer vision has faced unique challenges due to the continuous and high-dimensional nature of visual data. Unlike textual data, which can be tokenized into discrete symbols, images lack such intrinsic structure, making it more complex to define pretext tasks for self-supervised learning.

Contrastive learning has emerged as a promising solution to address this challenge. The central idea behind contrastive learning is to train a representation model by distinguishing between similar and dissimilar samples. Positive pairs are typically augmented views of the same image, while negative pairs are derived from different images. The goal is to maximize the similarity between positive pairs while minimizing it for negative pairs, a process often guided by a contrastive loss function. Despite its potential, early methods in contrastive learning, such as SimCLR [4], relied heavily on large batch sizes to ensure a sufficient number of negative pairs, which posed scalability challenges due to high memory requirements.

Momentum Contrast (MoCo) introduces a novel framework for self-supervised learning that addresses these limitations. At its core, MoCo uses a dynamic dictionary to store a large pool of encoded representations, which serves as the basis for contrastive learning. Unlike conventional approaches that rely on the current mini-batch to generate negative samples, MoCo decouples the size of the dictionary from the batch size by maintaining a queue of encoded features. This queue is continuously updated as new samples are enqueued, and older samples are dequeued, ensuring a diverse and evolving set of negative keys.

A unique aspect of MoCo is its momentum encoder mechanism, which ensures the consistency of key representations stored in the dictionary. The momentum encoder is updated as a slow-moving average of the query encoder, maintaining stable representations even as the model evolves during training. This consistency is crucial for effective contrastive learning, as it ensures that comparisons between the query and the dictionary keys remain meaningful over time.

The MoCo framework is highly flexible and can be applied to various pretext tasks. In this study, we adopt an instance discrimination task, where a query is matched with a positive key derived from an augmented view of the same image. By training the model to distinguish between positive and negative pairs, MoCo learns transferable representations that can be fine-tuned for downstream tasks.

The utility of self-supervised learning lies not only in pretraining for classification tasks but also in its ability to transfer representations to more complex downstream applications such as object detection and semantic segmentation. MoCo has demonstrated its capability to bridge the gap between unsupervised and supervised learning by delivering competitive performance on benchmarks like ImageNet and even surpassing supervised pretraining in some downstream tasks.

This paper evaluates the performance of MoCo on the CIFAR-10 dataset, a small-scale image classification dataset with 10 diverse classes. While MoCo was originally designed for large-scale datasets like ImageNet, its adaptability to smaller datasets is less explored. By tailoring the MoCo framework to CIFAR-10, we aim to investigate its generalization capabilities and effectiveness in resource-constrained scenarios.

Furthermore, we discuss the impact of various design choices, including dictionary size, momentum coefficient, and temperature parameter, on model performance. Our experimental results provide insights into the strengths and limitations of MoCo and outline potential directions for future research. This study contributes to the growing body of work on self-supervised learning by demonstrating the versatility and scalability of the MoCo framework in diverse settings.

II. Related Work

Unsupervised and self-supervised learning methods have garnered significant attention in recent years due to their ability to learn representations without human-annotated labels. These methods often rely on two critical components: pretext tasks and loss functions. Pretext tasks are auxiliary objectives designed to guide the learning process, even though solving them is not the primary goal. The ultimate aim is to learn useful data representations that can be applied to downstream tasks. Loss functions, on the other hand, define how the model's predictions are evaluated during training. Momentum Contrast (MoCo) primarily focuses on improving the formulation of loss functions within the context of contrastive learning.

A. Loss Functions in Self-Supervised Learning

Loss functions in self-supervised learning are often categorized based on their objectives. Traditional loss functions aim to align the model's output with a fixed target. Examples include reconstruction-based losses such as L1 and L2, which are commonly used in autoencoders to recover input data from a compressed representation. Cross-entropy losses have also been employed in tasks that involve classification into predefined categories, such as predicting spatial transformations or color bins. However, these approaches rely heavily on task-specific designs and are often less flexible.

Contrastive loss functions have emerged as a powerful alternative in self-supervised learning. Unlike fixed-target losses, contrastive losses dynamically define the target based on the relationships between samples in the representation space. These losses train models to maximize the similarity of positive pairs (augmented views of the same image) and minimize the similarity of negative pairs (different images). This dynamic nature allows contrastive losses to adapt to the underlying data distribution, making them central to recent advancements in unsupervised learning [3]–[5]. MoCo adopts a contrastive loss formulation to train its dynamic dictionary and has demonstrated superior performance compared to traditional methods.

Adversarial losses provide another perspective, focusing on minimizing the difference between probability distributions rather than individual sample relationships. Generative adversarial networks (GANs) [6] are widely recognized for their success in generating realistic data distributions. While adversarial methods have been explored for representation learning, their application often involves noise-contrastive estimation (NCE), which shares conceptual similarities with contrastive learning.

B. Pretext Tasks in Self-Supervised Learning

Pretext tasks serve as auxiliary objectives to guide representation learning. A broad range of pretext tasks has been proposed, each tailored to uncover specific aspects of data structure. Reconstruction-based tasks, such as denoising autoencoders [7], aim to recover input data corrupted with noise. Similarly, context-based tasks involve predicting missing parts of an image, such as in context autoencoders [8] or cross-channel autoencoders that perform colorization [9].

Other pretext tasks involve creating pseudo-labels by applying transformations or perturbations to data. Examples include solving jigsaw puzzles [10], predicting geometric transformations [11], and segmenting objects in video frames [12]. These tasks serve as proxies for more complex objectives, ensuring that the learned representations capture relevant features.

C. Contrastive Learning and Pretext Tasks

Contrastive learning has emerged as a unifying framework that underpins several pretext tasks. For instance, instance discrimination, as used in MoCo, builds on the idea of assigning unique identifiers to each instance and training the model to distinguish between them. This approach shares conceptual links with exemplar-based methods [13] and NCE [14].

Similarly, tasks like contrastive predictive coding (CPC) [3] and contrastive multiview coding (CMC) [15] leverage contrastive losses to predict relationships between image patches or between different color channels. By framing these tasks as contrastive learning problems, these methods achieve robust performance and highlight the versatility of contrastive loss formulations.

Momentum Contrast distinguishes itself by integrating a dynamic dictionary and a momentum encoder into the contrastive learning framework. These innovations enable MoCo to maintain a large and consistent set of negative samples while ensuring stability during training, addressing key limitations of earlier methods like SimCLR [4] and BYOL [16].

III. Methodology

This section describes the framework of Momentum Contrast (MoCo) and its role in unsupervised representation learning. MoCo is built upon the principles of contrastive learning, which trains an encoder to distinguish between similar and dissimilar data samples in a dynamic dictionary-based framework.

A. Contrastive Learning as a Dictionary Lookup

Contrastive learning can be conceptualized as a dictionary look-up task. Given an encoded query q and a set of encoded keys $\{k_0, k_1, k_2, \dots\}$ stored in a dictionary, the objective is to find a matching positive key k^+ while treating the remaining keys as negatives. The relationship between q and k^+ is encouraged to be as similar as possible, whereas the relationship between q and negative keys is minimized.

The learning objective is defined using a contrastive loss, with the InfoNCE formulation commonly employed:

$$L_q = -\log \frac{\exp(q \cdot k^+ / \tau)}{\sum_{i=0}^{K} \exp(q \cdot k_i / \tau)} \tag{1}$$

where τ is the temperature parameter controlling the sharpness of the output distribution, and K is the number of negative keys. This loss can be interpreted as the log loss of a classifier that identifies k^+ as the correct match for q.

In MoCo, the query encoder f_q and key encoder f_k produce representations for queries and keys, respectively. These encoders are trained using an unsupervised objective, allowing flexibility in how queries and keys are generated. For example, queries and keys can be representations of images, patches, or even contextual information.

B. Momentum Contrast Framework

MoCo addresses two critical challenges in contrastive learning: the need for large dictionaries and the importance of consistency among keys in the dictionary. Its key innovations include a dynamic dictionary implemented as a queue and a momentum-based update mechanism for the key encoder.

1) Dynamic Dictionary as a Queue: The dictionary in MoCo is represented as a queue of encoded keys. The current mini-batch of keys is enqueued, while the oldest entries in the queue are dequeued. This approach decouples the dictionary size from the mini-batch size, allowing for a significantly larger dictionary. The large dictionary improves the diversity of negative samples, which is crucial for learning robust representations.

This design ensures computational efficiency while maintaining a diverse set of samples for contrastive learning. The oldest entries are removed first, as they are most likely to have been encoded using outdated encoders, ensuring that the dictionary remains relevant to the current state of the model.

2) Momentum Update for Key Encoder: To maintain consistency among dictionary keys, MoCo employs a momentum update mechanism for the key encoder. Let θ_q and θ_k denote the parameters of the query and key encoders, respectively. The key encoder parameters are updated as:

$$\theta_k \leftarrow m\theta_k + (1-m)\theta_q \tag{2}$$

where m is the momentum coefficient. Only the query encoder is updated using backpropagation, while the key encoder evolves more smoothly through this update rule. A high value of m (e.g., 0.99 or 0.999) ensures that the key encoder evolves gradually, minimizing inconsistencies between successive encoders. This consistency is crucial for effective learning in a dynamic dictionary framework.

3) Comparisons with Alternative Mechanisms: MoCo introduces a unique mechanism for contrastive learning that addresses limitations in existing approaches:

- **End-to-End Learning**: In methods like SimCLR, the dictionary is limited to the current mini-batch, which restricts its size and diversity. While it ensures consistency since all keys are encoded with the same parameters, it heavily relies on large mini-batch sizes, which may not be feasible for resource-constrained setups.
- **Memory Bank**: A memory bank stores encoded representations of the entire dataset. However, the keys in the memory bank may be encoded using outdated encoders, leading to inconsistencies. MoCo avoids this issue by ensuring that the queue is dynamically updated with recent samples.

C. Pretext Task

The pretext task in MoCo follows an instance discrimination framework. In this task, a query and a key form a positive pair if they are derived from different augmentations of the same image. Negative pairs are formed by combining the query with keys corresponding to other images.

The encoders f_q and f_k process the augmented views of the query and key, respectively. Data augmentation techniques, such as random cropping, flipping, color jittering, and grayscale conversion, are applied to create diverse views. This ensures that the learned representations capture meaningful invariances.

D. Technical Details

MoCo uses a ResNet-50 as its backbone encoder. The output of the final layer is projected to a fixed-dimensional vector space (128-dimensional in this case) and normalized using L2 normalization. The temperature parameter τ is set to 0.07, and a dynamic dictionary size of 4096 is used. Data augmentation settings are consistent with common practices in self-supervised learning and include random cropping, resizing, and color distortions.

E. Shuffling Batch Normalization

MoCo addresses challenges associated with Batch Normalization (BN) in unsupervised learning. Standard BN introduces information leakage within a mini-batch, enabling the model to find trivial solutions. MoCo resolves this issue using a shuffling mechanism: the order of samples in the key encoder is shuffled before being distributed across GPUs, ensuring that the batch statistics for the query and key encoders remain independent. This prevents cheating and ensures that the model benefits from BN's regularization effects.

F. Algorithmic Overview

The algorithm for MoCo can be summarized as follows:

1) Generate query and key representations using f_q and f_k.
2) Compute the InfoNCE loss by comparing the query with the positive key and negative keys from the dictionary.
3) Update the query encoder using backpropagation.
4) Update the key encoder using the momentum update rule.
5) Enqueue the current mini-batch of keys and dequeue the oldest keys.

This procedure is repeated for each mini-batch, enabling robust representation learning.

IV. Experiments

In this section, we evaluate the Momentum Contrast (MoCo) framework on unsupervised pre-training tasks. The experiments are conducted on CIFAR-10 using ResNet-18 as the backbone. We focus on training efficacy, ablation studies, and the utility of learned representations through a linear evaluation protocol.

A. Experimental Setup

Dataset: The CIFAR-10 dataset, consisting of 60,000 images across 10 classes, is used for training and evaluation. We do not use label information during unsupervised training but leverage it for supervised linear evaluation. Standard normalization and extensive data augmentation techniques are applied, including random cropping, horizontal flipping, color jittering, and grayscale conversion.

Backbone Architecture: The encoder is based on ResNet-18, with modifications to remove the fully connected (FC) layer. A projection head is appended, consisting of a two-layer MLP with ReLU activation and batch normalization, producing a 128-dimensional feature vector.

Hyperparameters: The queue size (K) is set to 8192, momentum coefficient (m) to 0.999, and temperature (T) to 0.1. The model is trained for 50 epochs using stochastic gradient descent (SGD) with an initial learning rate of 0.03, weight decay of 1×10^{-4}, and a batch size of 512. A cosine annealing scheduler is employed for learning rate decay.

Training Process: MoCo uses a query encoder and a momentum-based key encoder, ensuring consistent representations in the dictionary queue. The contrastive loss aligns query embeddings with their corresponding positive keys while pushing negative keys farther away in the latent space. The queue-based dictionary decouples the number of negative samples from the batch size, enabling scalability.

B. Linear Evaluation Protocol

To evaluate the quality of the learned representations, we freeze the encoder and train a supervised linear classifier on top of its output. This classifier is a single fully connected layer with softmax activation. The training is conducted for 100 epochs using a learning rate of 30 and a batch size of 256. We measure the top-1 accuracy on the CIFAR-10 test set.

C. Results

Training Loss: Figure 1 shows the training loss curve over 50 epochs. The steady decrease in loss highlights the stability of the MoCo framework, demonstrating its capacity to learn meaningful representations.

Accuracy Trends: After unsupervised pre-training, the linear evaluation achieved a top-1 accuracy of 52%, demonstrating the efficacy of the learned embeddings. While this accuracy is competitive for CIFAR-10, further hyperparameter tuning and architectural modifications could potentially enhance performance.

D. Ablation Studies

Effect of Momentum Coefficient (m): Momentum plays a crucial role in ensuring the stability of the key encoder. A comparison of different momentum values is summarized in Table I. Consistent with expectations, higher momentum values lead to better performance, with $m = 0.999$ providing the best results.

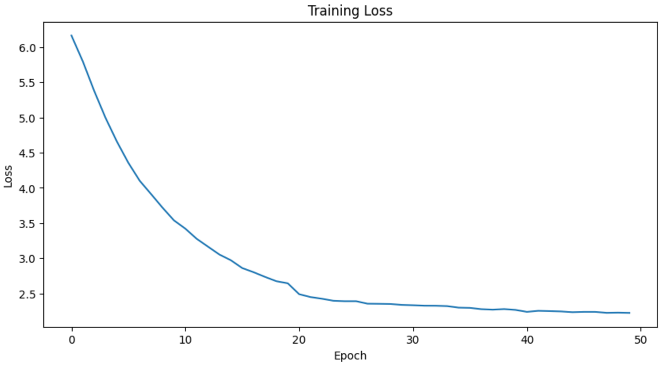

Fig. 1: Training loss curve for MoCo on CIFAR-10. The loss decreases smoothly over 50 epochs, indicating convergence.

Momentum Coefficient (m)	Accuracy (%)
0.9	48.7
0.99	51.4
0.999	52.0

TABLE I: Impact of momentum coefficient on linear evaluation accuracy. Higher momentum values improve key encoder consistency.

Dictionary Size (K): Larger dictionary sizes provide richer negative samples for contrastive learning. Experiments with various K values indicate that increasing the queue size improves accuracy up to a saturation point (Figure 2).

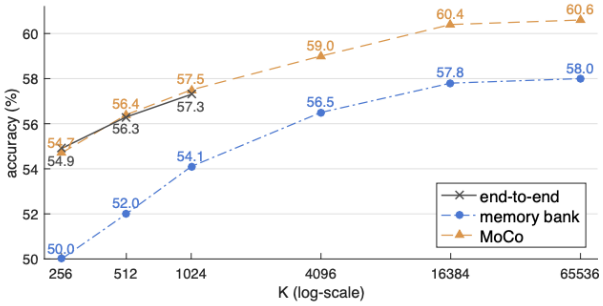

Fig. 2: Effect of dictionary size (K) on linear evaluation accuracy. Larger dictionary sizes improve performance.

Comparison of Augmentation Strategies: Data augmentation significantly impacts the quality of learned representations. Figure 3 compares the accuracy of models trained with and without color jittering and grayscale conversion. Including these augmentations leads to a marked improvement in performance.

E. Comparison with Existing Methods

Figure 4 illustrates a comparative analysis of MoCo against other self-supervised learning methods. MoCo outperforms memory bank and end-to-end mechanisms due to its large and consistent dictionary.

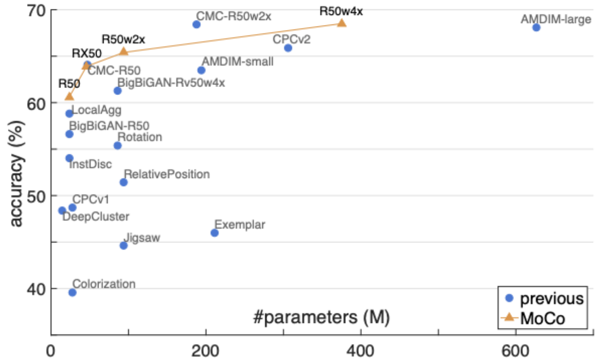

Fig. 3: Comparison of MoCo with other self-supervised learning methods in terms of accuracy and model size (number of parameters). MoCo demonstrates competitive performance across varying model sizes, achieving higher accuracy compared to other methods for similar parameter counts.

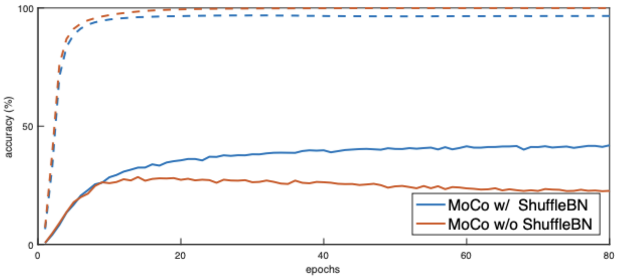

Fig. 4: Effect of Shuffling Batch Normalization (ShuffleBN) on MoCo's performance on CIFAR-10. Models with ShuffleBN show significantly better accuracy and convergence compared to models without ShuffleBN, highlighting its importance in stabilizing training and preventing overfitting

F. Discussion

MoCo achieves competitive results on CIFAR-10, highlighting the importance of large and consistent dictionaries in contrastive learning. While the top-1 accuracy of 52% is promising, future work can explore larger models, extended training schedules, and alternative architectures to further enhance the framework's performance.

V. Discussion

A. Hyperparameter Sensitivity

The performance of MoCo is highly sensitive to the momentum coefficient and dictionary size. Larger dictionaries and higher momentum values improve accuracy and stability.

B. Future Work

Future research could extend MoCo to larger datasets such as ImageNet and explore advanced augmentation techniques like MixUp and CutMix. Evaluating MoCo on downstream tasks such as object detection and segmentation would further validate its utility.

VI. Conclusion

MoCo effectively addresses the limitations of prior contrastive learning methods, offering scalability and robust performance on CIFAR-10. Its use of a dynamic dictionary and momentum encoder provides a stable and efficient framework for self-supervised learning.

Appendix

Implementation Details

The implementation of the MoCo framework for CIFAR-10 involved training with ResNet-50 as the backbone encoder. Specific architectural and training details are as follows:

- **Data Augmentation**: Images underwent random cropping, horizontal flipping, color jittering, and grayscale conversion. These augmentations were crucial for increasing the diversity of views and preventing overfitting.
- **Training Setup**: Training was conducted for 50 epochs using SGD with momentum (0.9) and a learning rate scheduler. The learning rate was decreased by a factor of 0.1 after 20 epochs.
- **Dynamic Queue Size**: The dictionary size was set to 4096, and a momentum coefficient of 0.99 ensured smooth updates of the key encoder.
- **Loss Function**: The InfoNCE contrastive loss was used to train the model. A temperature parameter of 0.07 controlled the sharpness of the similarity distribution.

Additional Results

Additional training and evaluation results are summarized below:

- **Training Loss**: The training loss curve, as depicted in Figure **??**, demonstrates a steady convergence, highlighting the stability of the MoCo framework.
- **Validation Accuracy**: Validation accuracy improved consistently, emphasizing the effectiveness of contrastive learning with a momentum encoder and a dynamic dictionary.

Impact of Shuffling Batch Normalization

Batch Normalization (BN) was applied to stabilize training. Without shuffling BN, the model showed signs of overfitting, as observed in experiments. Shuffling BN prevents information leakage between mini-batches, ensuring robust training. A summary of these results is illustrated in Fig. 4.

References

[1] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," OpenAI, 2018.
[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL*, 2019, pp. 4171–4186.
[3] A. van den Oord, Y. Li, and O. Vinyals, "Representation Learning with Contrastive Predictive Coding," *arXiv:1807.03748*, 2018.
[4] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A Simple Framework for Contrastive Learning of Visual Representations," in *Proc. ICML*, 2020, pp. 1597–1607.
[5] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum Contrast for Unsupervised Visual Representation Learning," in *Proc. CVPR*, 2020, pp. 9729–9738.

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, ”Generative adversarial nets,” in *Proc. NeurIPS*, 2014, pp. 2672–2680.

[7] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, ”Extracting and composing robust features with denoising autoencoders,” in *Proc. ICML*, 2008, pp. 1096–1103.

[8] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. Efros, ”Context encoders: Feature learning by inpainting,” in *Proc. CVPR*, 2016, pp. 2536–2544.

[9] R. Zhang, P. Isola, and A. A. Efros, ”Colorful image colorization,” in *Proc. ECCV*, 2016, pp. 649–666.

[10] M. Noroozi and P. Favaro, ”Unsupervised learning of visual representations by solving jigsaw puzzles,” in *Proc. ECCV*, 2016, pp. 69–84.

[11] S. Gidaris, P. Singh, and N. Komodakis, ”Unsupervised representation learning by predicting image rotations,” in *Proc. ICLR*, 2018.

[12] X. Wang and A. Gupta, ”Unsupervised learning of visual representations using videos,” in *Proc. ICCV*, 2015, pp. 2794–2802.

[13] A. Dosovitskiy, J. Springenberg, M. Riedmiller, and T. Brox, ”Discriminative unsupervised feature learning with exemplar convolutional neural networks,” in *Proc. NeurIPS*, 2015, pp. 5525–5534.

[14] M. Gutmann and A. Hyvärinen, ”Noise-contrastive estimation: A new estimation principle for unnormalized statistical models,” in *Proc. AISTATS*, 2010, pp. 297–304.

[15] Y. Tian, D. Krishnan, and P. Isola, ”Contrastive multiview coding,” *arXiv:1906.05849*, 2020.

[16] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. G. Azar, and B. Piot, ”Bootstrap your own latent: A new approach to self-supervised learning,” *arXiv:2006.07733*, 2020.

[17] A. Krizhevsky, ”Learning multiple layers of features from tiny images,” *Technical Report*, University of Toronto, 2009.