hyperrefHyper figures OFFhyperrefLink nesting OFFhyperrefHyper index ONhyperrefPlain pages OFFhyperrefBackreferencing OFF hyperrefImplicit mode ON; LaTeX internals redefined hyperrefBookmarks ON

hyperrefHyper figures OFFhyperrefLink nesting OFFhyperrefHyper index ONhyperrefbackreferencing OFFhyperrefLink coloring OFFhyperrefLink coloring with OCG OFFhyperrefPDF/A mode OFF

hyperrefDriver (autodetected): hpdftex rerunfilecheckFeature \pdfmdfivesum is not available(e.g. pdfTeX or LuaTeX with package 'pdftexcmds').Therefore file contents cannot be checked efficientlyand the loading of the package is aborted

hyperrefLink coloring OFF

Semi-Supervised GANs for MNIST Classification with 100 Labels: An Experimental Study of the K+1 Discriminator and Feature Matching

Manel LOUNISSI :
Université Paris Cité
manel2.lounissi@gmail.com & $Sandeep - SinghNIRMAL$:
Université Paris Cité
nirmalsinghsandeep@gmail.com & $BriceSAILLARD$:
Université Paris Cité
brice.saillard.bs@gmail.com & $HamadyGACKOU$
Université Paris Cité
hamady.gackou@etu.u-paris.fr

December 12, 2025

Abstract

We investigate semi-supervised classification on MNIST using only 100 labeled samples, combining a supervised CNN baseline with a Semi-Supervised GAN (SGAN) framework using a K+1 classifier and feature matching. This template provides the structure of the report; the placeholder text should be replaced by the project team.

1 Introduction

This section introduces the low-label learning problem on MNIST, the motivation for semi-supervised GANs, and an overview of the approach. Discuss the challenge of using only 100 labels, existing semi-supervised methods, and why SGAN is relevant.

2 Baseline Method

Describe the supervised CNN baseline used for comparison. Include the architecture (conv–relu–pool), training setup (epochs, optimizer, augmentation), and initial accuracy. This baseline serves as the reference to quantify SGAN improvements.

3 SGAN Methodology

The Semi-Supervised GAN extends the discriminator into a $(K + 1)$-class classifier, where the extra class corresponds to generated samples. The generator is trained using feature matching for stability.

3.1 K+1 Classifier

Explain the decomposition:

$$p_D(y|x) = \begin{cases} p(\text{digit} = k|x), & k \in \{1, \dots, 10\} \\ p(\text{fake}|x), & k = 11. \end{cases}$$

Discuss supervised loss on labeled data and unsupervised loss on unlabeled data.

3.2 Feature Matching

Describe the generator loss based on matching intermediate feature statistics instead of fooling the discriminator directly:

$$\mathcal{L}_G = \left\| \mathbb{E}_{x \sim p_{\text{data}}} f(x) - \mathbb{E}_{z \sim p(z)} f(G(z)) \right\|_2^2.$$

4 Implementation Details

List details needed for reproducibility:

- Dataset preparation (selection of 100 labeled samples) - CNN architecture hyperparameters - SGAN training hyperparameters - Optimizers (Adam), batch sizes - Training duration - Hardware used (GPU/CPU) - Random seed setup

5 Experiments and Results

Include classification accuracy of:

1. Supervised CNN baseline 2. SGAN (K+1 classifier only) 3. SGAN (K+1 + Feature Matching)

Insert tables and plots of accuracy curves if needed.

Example placeholder:

Method	Accuracy (%)
Baseline CNN	XX.X
SGAN (K+1)	XX.X
SGAN (K+1 + FM)	XX.X

Table 1: Placeholder results. Replace with real numbers.

6 Discussion

Interpret why SGAN improves over the baseline. Discuss stability issues, unlabeled data usage, learning dynamics, and potential limitations.

7 Conclusion

Summarize contributions: - implementation of SGAN with K+1 classifier - effective use of unlabeled data - comparison with supervised baseline - improvements observed

Suggest future extensions.

Appendix: Code Excerpts

Include short commented code excerpts: - CNN architecture - Discriminator K+1 modification - SGAN training loop Avoid long code dumps; keep it readable.