Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

FAKULTÄT
FÜR MATHEMATIK, INFORMATIK
UND NATURWISSENSCHAFTEN

MASTERTHESIS

Self-Calibrating Language Models via Test-Time Discriminative Distillation

Mohamed Rissal Hedna

Field of Study: Intelligent Adaptive Systems
Matriculation No.: 7712565
1st Examiner: Prof. Dr. Chris Biemann, Universität Hamburg
2nd Examiner: Dr. Martin Semmann, Universität Hamburg

Language Technology
Department of Informatics
Faculty of Mathematics, Informatics and Natural Sciences

Universität Hamburg
Hamburg, Germany

A thesis submitted for the degree of

Master of Science (M. Sc.)

Printed on June 5, 2026

Self-Calibrating Language Models via Test-Time Discriminative Distillation

Master's Thesis submitted by: Mohamed Rissal Hedna

Date of Submission: 05.06.2026

Supervisor(s):
Jan Strich, Universität Hamburg

Committee:
1st Examiner: Prof. Dr. Chris Biemann, Universität Hamburg
2nd Examiner: Dr. Martin Semmann, Universität Hamburg

Universität Hamburg, Hamburg, Germany
Faculty of Mathematics, Informatics and Natural Sciences
Department of Informatics

Language Technology

Affidavit

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit im Masterstudiengang Informatik selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel – insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen – benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Ich versichere weiterhin, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe. Sofern im Zuge der Erstellung der vorliegenden Abschlussarbeit generative Künstliche Intelligenz (gKI) basierte elektronische Hilfsmittel verwendet wurden, versichere ich, dass meine eigene Leistung im Vordergrund stand und dass eine vollständige Dokumentation aller verwendeten Hilfsmittel gemäß der Guten Wissenschaftlichen Praxis vorliegt. Ich trage die Verantwortung für eventuell durch die gKI generierte fehlerhafte oder verzerrte Inhalte, fehlerhafte Referenzen, Verstöße gegen das Datenschutz- und Urheberrecht oder Plagiate.

I hereby declare in lieu of an oath that I have written this thesis for the Master's degree programme in Computer Science independently and have not used any aids other than those specified – in particular no Internet sources not named in the list of sources. All passages taken verbatim or in spirit from publications are labelled as such. I further certify that I have not previously submitted the thesis in another examination procedure. If, in the course of preparing this thesis, generative artificial intelligence (gAI)-based electronic tools were used, I affirm that my own contribution was the primary focus and that a complete documentation of all tools used is provided in accordance with Good Scientific Practice. I accept responsibility for any incorrect or distorted content, faulty references, violations of data protection or copyright law, or plagiarism generated by the gAI.

05.06.2026

Date

Signature

(Mohamed Rissal Hedna)

Acknowledgements

Personal

I owe my deepest thanks to my supervisors, Prof. Dr. Chris Biemann and Jan Strich. Their guidance shaped this work at every stage, from the first vague idea to the final draft, and the patience and kindness with which it was offered made the process not just productive but genuinely enjoyable. They gave me the freedom to pursue the questions that interested me most while pulling me back when the questions ran ahead of the evidence, and the balance they struck between trust and rigor is something I will carry with me well beyond this thesis.

I am grateful to my family and friends for the support that kept the rest of life functioning while this work absorbed so much of it.

Institutional

This work was carried out at the Hub of Computing and Data Science (HCDS) at the University of Hamburg. I thank the Language Technology group for providing the computational resources that made the experiments in this thesis possible.

Abstract

Large language models (LLMs) are systematically overconfident: they routinely express high certainty on questions they often answer incorrectly. Existing calibration methods either require labeled validation data, degrade under distribution shifts, or incur substantial inference costs. Recent work has shown that LLMs already contain a better-calibrated signal than the one they verbalize: the token probability of "True" when the model is asked "Is this answer correct?" (P(True)) consistently outperforms their stated confidence, a gap that is theoretically grounded as generative error is lower-bounded by roughly twice the corresponding discriminative error.

If a model can judge its own answers better than it can produce them, that judgment can be turned into a training signal. We build on this idea with **SECL** (Self-Calibrating Language Models), a test-time training (TTT) pipeline that exploits the gap as label-free self-supervision, requiring no labeled data or human supervision. SECL adapts only when the input distribution shifts, training on just 6–26% of the question stream at lower cost than the baseline it distills from. Across four small language models from three model families and four diverse domains, SECL reduces Expected Calibration Error (ECE) by 56–78%, outperforming its own supervision signal and matching or outperforming recent inference-time methods. SECL is the first method to apply TTT to calibration; seven ablations covering signal quality, gating strategy, weight accumulation, loss design, domain ordering, hyperparameter sensitivity, and layer selection confirm that each component is crucial and robust across configurations. Code: https://github.com/rissalhedna/Truthfulness

Zusammenfassung

Große Sprachmodelle (LLMs) überschätzen sich systematisch selbst: Sie äußern routinemäßig hohe Sicherheit bezüglich Fragen, die sie häufig falsch beantworten. Bestehende Kalibrierungsmethoden erfordern entweder gelabelte Validierungsdaten, verschlechtern sich bei Verteilungsverschiebungen oder verursachen erhebliche Inferenzkosten. Aktuelle Arbeiten haben gezeigt, dass LLMs bereits über ein besser kalibriertes Signal verfügen als jenes, das sie verbalisieren: Die Token-Wahrscheinlichkeit von "True", wenn das Modell gefragt wird: Ïst diese Antwort korrekt?"(P(True)), übertrifft konsistent die geäußerte Konfidenz, eine Lücke, die theoretisch dadurch begründet ist, dass der generative Fehler durch ungefähr das Doppelte des entsprechenden diskriminativen Fehlers nach unten beschränkt ist.

Wir stellen **SECL** (**SE**lf-**C**alibrating **L**anguage Models) vor, eine Test-Time-Training-Pipeline (TTT), die diese Lücke als labelfreie Selbstüberwachung nutzt und weder gelabelte Daten noch menschliche Aufsicht benötigt. SECL passt das Modell nur dann an, wenn sich die Eingabeverteilung verschiebt, und trainiert lediglich auf 6–26 % des Fragenstroms zu geringeren Kosten als der Baseline, von der es destilliert.

Über vier kleine Sprachmodelle aus drei Modellfamilien und vier diversen Domänen hinweg reduziert SECL den Expected Calibration Error (ECE) um 56–78 %, übertrifft das eigene Überwachungssignal und erreicht oder übertrifft aktuelle Inferenzzeit-Methoden. SECL ist die erste Methode, die TTT auf Kalibrierung anwendet; sieben Ablationsstudien zu Signalqualität, Gating-Strategie, Gewichtsakkumulation, Loss-Design, Domänenreihenfolge, Hyperparameter-Sensitivität und Layer-Auswahl bestätigen, dass jede Komponente entscheidend und über Konfigurationen hinweg robust ist.

Contents

List of Figures iii

List of Tables iv

1 Introduction 1
- 1.1 Research Questions . . . 2
- 1.2 Structure of the thesis . . . 3

2 Fundamentals 4
- 2.1 Neural Networks and Language Models . . . 4
  - 2.1.1 Feedforward Neural Networks . . . 4
  - 2.1.2 The Transformer Architecture . . . 5
  - 2.1.3 Large Language Models . . . 5
  - 2.1.4 Instruction Tuning and RLHF . . . 6
- 2.2 Calibration of Language Models . . . 7
  - 2.2.1 Definition of Calibration . . . 7
  - 2.2.2 Calibration Metrics . . . 7
  - 2.2.3 Sources of Miscalibration in LLMs . . . 8
  - 2.2.4 Verbalized vs. Internal Confidence . . . 8
- 2.3 Uncertainty Estimation and the Generation-Discrimination Gap . . . 9
  - 2.3.1 Entropy as an Uncertainty Signal . . . 9
  - 2.3.2 P(True) and Self-Evaluation . . . 9
  - 2.3.3 The Generation-Discrimination Gap . . . 10
  - 2.3.4 Normalizing Confidence Across Distractors . . . 10
- 2.4 Parameter-Efficient Fine-Tuning . . . 11
  - 2.4.1 Full Fine-Tuning and Its Limits . . . 11
  - 2.4.2 Low-Rank Adaptation (LoRA) . . . 11
- 2.5 Test-Time Training . . . 12
  - 2.5.1 From Domain Adaptation to Test-Time Training . . . 12
  - 2.5.2 Self-Supervision Signals for TTT . . . 12
  - 2.5.3 Continual Test-Time Adaptation . . . 12
  - 2.5.4 Test-Time Training for Language Models . . . 13

3 Related Work 14
- 3.1 Sampling-Based Uncertainty Estimation . . . 14
- 3.2 Static Probing and Lightweight Calibration . . . 14
- 3.3 Training-Based Calibration . . . 15
- 3.4 The Generation-Discrimination Gap . . . 15
- 3.5 Test-Time Adaptation for LLMs . . . 15

3.6 Positioning SECL 16

4 Methods **17**
4.1 Overview 17
4.2 Adaptive Entropy Gating 18
4.3 Normalized P(True) as Self-Supervision 19
4.3.1 Verbalized Confidence 19
4.3.2 Distractor Normalization 19
4.4 Test-Time Calibration via LoRA 20
4.4.1 Directional Training Target 20
4.4.2 Loss and Optimization 21
4.4.3 Bin-Gate Filter 21
4.5 The Full SECL Procedure 21
4.6 Design Choices and Alternatives 22

5 Experiments **24**
5.1 Datasets 24
5.1.1 GSM8K 24
5.1.2 MMLU 24
5.1.3 ARC Challenge 25
5.1.4 TruthfulQA 25
5.1.5 Continual Protocol 25
5.2 Models 25
5.3 Baselines 25
5.4 Metrics 26
5.5 Implementation Details 26

6 Evaluation **28**
6.1 Main Results 28
6.1.1 SECL Calibrates Without Labels (RQ1) 28
6.1.2 SECL Surpasses Its Signal and Approaches Supervised Calibration (RQ2) 29
6.1.3 Cost and Accuracy 31
6.1.4 Comparison with DINCO 31
6.1.5 Seed Robustness 31
6.2 Ablation Studies 32
6.2.1 Each Component Is Necessary 33
6.2.2 Signal Quality Sets the Ceiling 33
6.2.3 Robustness to Ordering and Hyperparameters 34
6.2.4 Layer Selection and Regularization 34
6.3 Additional Analyses 34
6.3.1 Per-Domain Breakdown 35
6.3.2 Negative Control: The Precondition for SECL 35

7 Conclusion **36**
7.1 Research Question 1 36
7.2 Research Question 2 37
7.3 Limitations 37

7.3.1 Signal Quality Bounds Improvement . . . 37
7.3.2 Per-Domain Calibration Is Not Uniformly Improved . . . 38
7.3.3 Calibration–Discrimination Trade-off . . . 38
7.3.4 Hyperparameter Sensitivity on Burst Size . . . 38
7.3.5 Scale . . . 38
7.4 Future Work . . . 38

A Prompt Templates **40**
A.1 Question Answering and Confidence Elicitation . . . 40
A.1.1 Main QA Prompt with Verbalized Confidence . . . 40
A.1.2 Plain QA Prompt (No Confidence) . . . 41
A.2 Discriminative Probes . . . 42
A.2.1 P(True) Verification Probe . . . 42
A.2.2 P(Know) Probes . . . 42
A.3 Distractor and Neighborhood Generation . . . 43
A.3.1 Distractor Generation . . . 43
A.3.2 Neighborhood Question Rewriting . . . 44
A.4 DINCO Baseline Prompts . . . 44
A.4.1 P(True) Probe (Yes/No Format) . . . 44
A.4.2 NLI String Templates . . . 44
A.4.3 Items Not Covered by Prompt Templates . . . 45

B Extended Evaluation Results **46**
B.1 Reliability Diagrams Across Models . . . 46
B.2 Post-Hoc Calibration: Full Results . . . 46
B.3 Full DINCO Comparison . . . 46
B.4 Per-Domain Results: Full Metrics . . . 46
B.5 Hyperparameter Sensitivity . . . 47
B.6 Domain Order Sensitivity: Full Breakdown . . . 47
B.7 LoRA Layer Position . . . 48
B.8 KL Regularization . . . 48
B.9 Why Signal Quality Determines Calibration . . . 49
B.10 Further Robustness Analyses . . . 52
B.10.1 Scaling to 8B Parameters . . . 52
B.10.2 Extended Stream Length . . . 53
B.10.3 Open-Ended Generation . . . 54
B.10.4 Negative Control: Full Results . . . 54

Appendices

References **55**

List of Figures

4.1 Overview of SECL. **(a)** *Test-Time Inference.* An entropy-based change detector (Section 4.2) monitors the input stream. If no shift is detected, the adapted model θ'_t is used directly; otherwise, a calibration burst updates it to θ'_{t+1}. **(b)** *Calibration Burst.* For each of B=50 questions: the frozen model generates an answer with confidence c_t and distractors, computes $\text{NormP}_{\text{True}}$ (Section 4.3), and applies a LoRA update when the two signals disagree by more than one bin (Section 4.4). Weights accumulate across questions without resetting. 18

6.1 Reliability diagrams: verbalized baseline vs. SECL 30
6.2 Calibration error vs. inference cost 32

B.1 Reliability diagrams across models 47
B.2 Distractor normalization across models 48
B.3 Candidate training targets compared 53
B.4 Confidence score distributions by target 53

List of Tables

5.1 LoRA trainable parameters per model 26
5.2 Full hyperparameter settings 27

6.1 Overall calibration results across all models 29
6.2 SECL vs. supervised post-hoc calibration 30
6.3 Computational cost comparison 31
6.4 Multi-seed robustness on Llama 3.2-3B 32
6.5 Gating strategy ablation 33
6.6 Effect of training target on calibration 34
6.7 Domain order robustness 34
6.8 Per-domain calibration breakdown 35

B.1 SECL combined with temperature scaling 49
B.2 SECL vs. DINCO across all models 49
B.3 Full per-domain results 50
B.4 Hyperparameter sensitivity 51
B.5 Domain order sensitivity: overall metrics 51
B.6 Per-domain ECE by ordering 51
B.7 LoRA layer ablation 52
B.8 KL regularization ablation 52
B.9 Per-domain results for Llama 3.1-8B 53
B.10 Extended 4,000-question stream 54
B.11 Open-ended TruthfulQA-Gen evaluation 54
B.12 Qwen negative control: full results 54

1 Introduction

LLMs are systematically overconfident (Jiang et al., 2021; Xiong et al., 2024), and alignment procedures such as Reinforcement Learning from Human Feedback (RLHF) (Bai et al., 2022) worsen this by rewarding agreement with human preferences over truthfulness (Sharma et al., 2024). The consequences are not abstract. In healthcare, where LLMs increasingly support triage and diagnostics (F Liu et al., 2025), a recent review of 519 studies found that only 1.2% measured calibration despite 95.4% measuring accuracy (Bedi et al., 2025). The asymmetry is telling: the field has converged on accuracy as the primary metric for clinical LLM evaluation while leaving the reliability of the model's own confidence estimates almost entirely unexamined. A model that reports 90% certainty on questions it answers correctly only 30% of the time erodes clinician trust and risks patient harm, and the same failure mode generalizes to any high-stakes setting where downstream decisions depend on knowing when to defer. Addressing this problem needs calibration methods that work without labels and can adapt to new domains at test time, since deployment distributions rarely match those seen during training.

Yet LLMs already contain a better-calibrated signal than the one they verbalize. When asked whether their own answer to a question is correct, LLMs produce probability estimates (P(True)) that are substantially better calibrated than the confidence they express during generation (Kadavath et al., 2022; Tian et al., 2023). This exposes a systematic gap between *discrimination* and *generation*, with theoretical backing: Kalai et al. (2025) shows that a model's generative error is lower-bounded by roughly twice its discriminative error. Intuitively, a model that cannot reliably produce the correct answer can still often recognize when a given answer, or its own, is wrong. The same asymmetry is familiar from human cognition: recognizing a correct answer in a multiple-choice exam is easier than producing it from scratch. For an LLM, this gap provides a continual source of self-supervision that needs no labeled data, and it is available on every question the model answers. Prior work documents this phenomenon in LLMs; our experiments test whether the same mechanism can be exploited effectively in small language models, where the gap is, if anything, more pronounced.

Prior work on calibration falls into three broad categories, each with a key limitation that SECL addresses (see Chapter 3 for more details). Sampling-based methods (Manakul et al., 2023; Kuhn et al., 2023) measure consistency across multiple generations but are

expensive, requiring many forward passes per question, and they fail on confident hallucinations where the model is consistently wrong. SECL avoids repeated sampling entirely. Static probing methods (Du et al., 2024; X Liu et al., 2024) analyze internal representations but are fit on a fixed distribution and degrade under distribution shift, exactly the regime where calibration matters most. SECL adapts continuously via test-time training. Training-based approaches (Lin et al., 2022a; Stangel et al., 2025; Damani et al., 2025) can improve calibration within a domain, but many either require supervised correctness labels or degrade out-of-distribution under standard reinforcement learning training, since the calibration objective is tied to the training distribution.

Test-time training (TTT), which adapts model weights to incoming test data instead of relying on a fixed training distribution, originated in computer vision (Sun et al., 2020; D Wang et al., 2021; Q Wang et al., 2022) and has recently been extended to LLMs for accuracy improvement (Hardt and Sun, 2024; J Hu et al., 2025; Sun et al., 2025; Zweiger et al., 2025). It has not yet been applied to calibration, due to two obstacles. First, TTT requires a self-supervision signal at test time; existing calibration signals (sampling-based consistency or discriminative probes) are expensive to compute on every question, making continuous adaptation costly enough to defeat the point of adapting in the first place. Second, naive weight updates risk catastrophic forgetting or overfitting to noisy targets, especially when the supervision signal itself is imperfect, as any unsupervised signal must be.

We show that both obstacles can be overcome. SECL uses the generation–discrimination gap as a naturally available scalar target: a normalized $P(\text{True})$ signal computed from the frozen base model serves as self-supervision to adjust verbalized confidence via lightweight Low-Rank Adaptation (LoRA) (EJ Hu et al., 2022) updates, requiring no labeled data or human supervision. Entropy-based gating triggers adaptation only on distribution shifts, so the cost of calibration is paid only when the model encounters genuinely new territory, and a conservative directional loss with bounded updates mitigates catastrophic forgetting on the long tail of in-distribution questions where no adaptation is needed.

Our contributions are as follows:

- We introduce the first **TTT method for calibration**, using the generation–discrimination gap as label-free self-supervision. Entropy-based gating limits adaptation to distribution shifts, so SECL trains only on **6–26% of the data** at a lower cost than the signal it distills.
- The adapted model **surpasses** its self-supervision signal and **matches supervised calibration without labels**, showing that SECL generalizes beyond the training signal.
- We provide **nine ablations** that isolate each design choice, showing that signal quality sets the calibration ceiling and that **each component is necessary**. SECL is robust across four architectures, forward and reversed domain orderings, and all tested hyperparameters.

1.1 Research Questions

This thesis is organized around two research questions, which together cover the feasibility and the empirical behavior of test-time calibration via the generation–discrimination gap.

RQ1: Can the generation–discrimination gap serve as a label-free self-supervision signal for test-time calibration of small language models? The gap is well-documented in large models (Kadavath et al., 2022; Tian et al., 2023), but its use as a continuous training target during deployment is untested. Answering this question requires showing that a normalized $P(\text{True})$ signal is informative enough to drive weight updates, and that those updates can be applied selectively, without a labeled validation set, and without destabilizing the base model.

RQ2: Can a model adapted with this signal surpass the signal itself and match supervised calibration, and which design choices are necessary for this to hold? A self-supervised method is only useful if it does more than imitate its own training signal. This question covers both the headline result, that the adapted model exceeds the calibration of its supervisor and approaches supervised baselines, and the ablations that isolate which components, gating, the normalized $P(\text{True})$ target, the bounded LoRA update, and the directional loss, are individually required for that result. It also covers robustness across architectures, domain orderings, and hyperparameter settings.

1.2 Structure of the thesis

The remainder of this thesis is structured as follows.

Chapter 2 (Fundamentals) introduces the technical background needed to follow the rest of the thesis. It builds from neural network basics through the Transformer architecture and large language models, then covers the concepts specific to this work: calibration metrics, uncertainty estimation, parameter-efficient fine-tuning with LoRA, and test-time training.

Chapter 3 (Related Work) surveys prior work on LLM calibration, organized into the three categories above (sampling-based, static probing, and training-based methods), the literature on the generation–discrimination gap, and the recent extensions of test-time training to LLMs. It positions SECL as the first method to combine these threads.

Chapter 4 (Methods) presents SECL in full. It formalizes the entropy-based gating mechanism, the normalized $P(\text{True})$ self-supervision signal, and the test-time LoRA calibration loop, including the directional loss and the bounded update rule that together prevent catastrophic forgetting. This chapter delivers the first contribution above.

Chapter 5 (Experiments) describes the experimental setup, including the base models, the evaluation domains, and the supervised and unsupervised baselines against which SECL is compared.

Chapter 6 (Evaluation) reports the main results and the seven ablations. It shows that SECL surpasses its own self-supervision signal, matches supervised calibration without labels, and remains robust across four architectures, forward and reversed domain orderings, and all tested hyperparameters. Together with Chapter 5, it delivers the second and third contributions.

Chapter 7 (Conclusion) returns to the two research questions, summarizes the answers this thesis provides, and discusses limitations and directions for future work.

2 Fundamentals

This chapter builds the technical background needed for the rest of the thesis. It assumes familiarity with basic machine learning and probability, and climbs from neural network basics up to the four concepts that SECL combines: calibration, the generation-discrimination gap, Low-Rank Adaptation (LoRA), and test-time training (TTT). Section 2.1 introduces neural networks, the Transformer architecture, and the training pipeline that produces modern instruction-tuned LLMs. Section 2.2 defines calibration, gives the metrics we use to measure it, and explains why LLMs tend to be miscalibrated in the first place. Section 2.3 covers the uncertainty signals SECL relies on: entropy over the next-token distribution, the $P(\text{True})$ self-evaluation probe of Kadavath et al. (2022), and the formal generation-discrimination gap of Kalai et al. (2025). Section 2.4 introduces LoRA, and Section 2.5 introduces test-time training and the recent work extending it to language models.

2.1 Neural Networks and Language Models

2.1.1 Feedforward Neural Networks

A feedforward neural network, or multi-layer perceptron, is a function $f_\theta : \mathbb{R}^d \to \mathbb{R}^k$ composed of alternating linear transformations and element-wise nonlinearities. For input $x \in \mathbb{R}^d$, a network with L layers computes

$$h_0 = x, \qquad h_\ell = \sigma(W_\ell h_{\ell-1} + b_\ell), \qquad f_\theta(x) = W_L h_{L-1} + b_L, \tag{2.1}$$

where $\theta = \{W_\ell, b_\ell\}_{\ell=1}^{L}$ are the learnable weight matrices and bias vectors, and σ is a nonlinearity such as ReLU or GELU.

Training proceeds by minimizing a loss function $\mathcal{L}(\theta)$ over a dataset via gradient descent. For a batch of examples, the loss is computed, its gradient with respect to θ is obtained by backpropagation, and the parameters are updated in the negative-gradient direction. In practice, first-order adaptive optimizers such as Adam (Kingma and Ba, 2015) and its decoupled-weight-decay variant AdamW (Loshchilov and Hutter, 2019) replace vanilla stochastic gradient descent, since they track per-parameter running

statistics of the gradient and adjust the effective step size accordingly. The concepts of weights, gradients, and loss carry over unchanged to the architectures discussed below; everything that follows can be read as a more elaborate choice of parameterization for the same optimization procedure.

2.1.2 The Transformer Architecture

The Transformer (Vaswani et al., 2017) is the architecture behind every model we study. Earlier sequence models, primarily recurrent networks such as LSTMs, processed tokens one at a time and carried information forward through a fixed-size hidden state, which limited their ability to model long-range dependencies and resisted parallelization. The Transformer replaces this with self-attention, a mechanism that allows every token in a sequence to attend to every other token in a single operation.

The core operation is scaled dot-product attention. Given a sequence of input representations $X \in \mathbb{R}^{n \times d}$, three linear projections produce queries, keys, and values,

$$Q = XW_Q, \qquad K = XW_K, \qquad V = XW_V, \tag{2.2}$$

and the attention output is

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V, \tag{2.3}$$

where d_k is the dimensionality of the keys. The softmax produces a matrix of attention weights, each row of which is a probability distribution over positions, and the output at each position is a weighted sum of the value vectors. Multi-head attention runs this operation in parallel with h separate sets of projections, concatenating the outputs, so that different heads can specialize in different relational patterns.

A full Transformer block stacks multi-head attention and a position-wise feedforward network, each wrapped in a residual connection and a layer normalization. A decoder-only language model stacks L such blocks and applies a causal mask to the attention matrix so that position t only attends to positions $\leq t$. The final hidden state at each position is projected through an unembedding matrix and a softmax to produce a distribution over the vocabulary. Decoder-only Transformers, with various architectural refinements such as rotary position embeddings, grouped-query attention, and gated activation functions, form the basis of the Llama (Grattafiori et al., 2024), Qwen (Qwen Team et al., 2025), Gemma (Team et al., 2024), and Phi (Abdin et al., 2024) models that we evaluate in later chapters.

2.1.3 Large Language Models

A large language model is a decoder-only Transformer trained to predict the next token in a sequence. Given a tokenized prefix $x_{<t} = (x_1, \ldots, x_{t-1})$, the model parameterizes a conditional distribution $p_\theta(x_t \mid x_{<t})$ over the vocabulary, and the training objective is to maximize the log-likelihood of the observed tokens in a large corpus of text. At inference time, tokens are sampled or greedily selected from this distribution one at a time, with each newly generated token appended to the prefix for the next step. Tokenization, typically via byte-pair encoding (Sennrich et al., 2016) or a variant, maps raw text into a vocabulary of subword units, and is fixed once the model is trained. The vocabularies

of the models we evaluate range from roughly 32,000 (Phi-3, (Abdin et al. 2024)) to 256,000 (Gemma 2, (Team et al. 2024)) tokens.

The distribution $p_\theta(\cdot \mid x_{<t})$ is central to this thesis. Both the entropy used by our gating mechanism and the $P(\text{True})$ probe used as self-supervision are read directly off this softmax, so the shape of this distribution, how peaked or flat it is on a given input, carries most of the uncertainty information we care about. Section 2.3 returns to this point.

Two parameter scales are worth distinguishing. Models with tens or hundreds of billions of parameters, such as the larger Llama 3 variants (Grattafiori et al., 2024), are typically what is meant by "LLMs" in much of the calibration literature. Models with one to a few billion parameters, sometimes called small language models (SLMs), exhibit the same architecture and training objective but with reduced absolute capability. SECL is evaluated on SLMs specifically for two reasons. Their deployment is practically important, since on-device inference and edge applications require parameter counts in this range. In addition, as the experiments in Chapter 6 show, calibration problems on these models are often pronounced enough to give a clear signal for methods aimed at fixing them.

2.1.4 Instruction Tuning and RLHF

A base LLM trained on next-token prediction produces fluent text but does not reliably follow instructions or hold a conversation. Modern deployed models are therefore post-trained in two stages. The first is supervised fine-tuning (SFT) on instruction-response pairs (J Wei et al., 2022; Ouyang et al., 2022), which teaches the model to produce answers of the expected form when presented with a question. The second is preference-based optimization, most commonly Reinforcement Learning from Human Feedback (RLHF) (Bai et al., 2022). Human annotators compare pairs of model responses to the same prompt and mark which is preferred, a reward model is trained on these comparisons to approximate a scalar quality score, and the base model is then optimized to produce responses that the reward model rates highly. This is typically done using either reinforcement learning algorithms such as Proximal Policy Optimization (Schulman et al., 2017) or more recent direct-preference variants such as DPO (Rafailov et al., 2023) that bypass the explicit reward model.

All base models we evaluate (Llama 3.2, Qwen 2.5, Gemma 2, Phi-3) are instruction-tuned and preference-aligned variants of this kind. This matters for calibration. Preference data systematically favors confident, definitive responses over hedged ones (Sharma et al., 2024), and the policy optimized against such preferences learns to sound more certain than its underlying knowledge warrants. The mechanism is documented empirically: Tian et al. (2023) show that on RLHF-tuned models, verbalized confidence is typically better calibrated than the model's own token probabilities, an inversion of the usual ordering that points to RLHF having a particularly strong effect on internal calibration. Section 2.2.4 returns to this.

2.2 Calibration of Language Models

2.2.1 Definition of Calibration

A model is calibrated if the confidence it assigns to its predictions matches their empirical accuracy. Formally, for a model that produces predictions $\hat{y}$ with associated confidence $\hat{p} \in [0, 1]$, perfect calibration requires

$$\mathbb{P}(\hat{y} = y \mid \hat{p} = p) = p \quad \text{for all } p \in [0, 1], \tag{2.4}$$

where y is the true label (Guo et al., 2017). A model that reports 70% confidence on a set of predictions should be correct on 70% of them; if it is correct on only 40%, the model is overconfident on that bin, and if it is correct on 90%, it is underconfident.

Calibration is independent of accuracy. A model that always predicts the majority class with 50% confidence on a balanced binary task is perfectly calibrated and useless, and a model that is 95% accurate while reporting 99% confidence on every input is highly accurate and poorly calibrated. The two metrics measure different properties: accuracy measures whether predictions are right, calibration measures whether the model knows when they are. In deployment, accuracy determines how often the system is correct, and calibration determines whether a human reviewer can tell which specific outputs to trust.

2.2.2 Calibration Metrics

The standard scalar measure of calibration is the Expected Calibration Error (ECE) (Naeini et al., 2015). ECE partitions predictions into M equal-width bins by their associated confidence, then averages the absolute difference between accuracy and confidence within each bin, weighted by bin size:

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} \left|\mathrm{acc}(B_m) - \mathrm{conf}(B_m)\right|, \tag{2.5}$$

where B_m is the set of predictions in bin m, N is the total number of predictions, $\mathrm{acc}(B_m)$ is the empirical accuracy on B_m, and $\mathrm{conf}(B_m)$ is the mean confidence on B_m. ECE is reported as a single number in $[0, 1]$, with lower values indicating better calibration. We use $M = 10$ throughout this thesis, following standard practice (Guo et al., 2017).

ECE has known limitations: the choice of binning scheme affects the score, and equal-width bins can leave some bins nearly empty when confidences cluster, biasing the estimate (Nixon et al., 2019). We therefore report Adaptive ECE (equal-mass bins) alongside standard ECE in Chapter 6 as a robustness check, and supplement both with the Brier score (Brier, 1950), the mean squared error between predicted probabilities and binary correctness indicators,

$$\mathrm{BS} = \frac{1}{N} \sum_{i=1}^{N} (\hat{p}_i - \mathbb{1}[\hat{y}_i = y_i])^2, \tag{2.6}$$

which is sensitive to both calibration and refinement simultaneously.

For methods that use confidence as a signal for selective prediction, accepting answers above a threshold and abstaining otherwise, we additionally report the Area Under the Receiver Operating Characteristic (AUROC) of confidence as a binary classifier

of correctness. AUROC measures whether the model assigns higher confidence to its correct answers than to its incorrect ones, independent of any specific threshold or absolute calibration level. A model can be badly miscalibrated in the absolute sense while still ranking its correct answers above its incorrect ones, which is what selective prediction requires.

2.2.3 Sources of Miscalibration in LLMs

Modern LLMs are systematically overconfident (Jiang et al., 2021; Xiong et al., 2024). The cause is partly procedural: the post-training pipeline that turns a base language model into an instruction-following assistant introduces systematic biases that decouple confidence from correctness. Instruction-tuned models specifically have been shown to suffer from calibration problems: Heo et al. (2025) document significant overconfidence on instruction-following tasks in particular, consistent with the broader overconfidence patterns established by Xiong et al. (2024) for elicitation strategies in general.

RLHF is the most-discussed contributor, for the reasons given in Section 2.1.4. Reward models trained on human preference data systematically favor confident, definitive responses over hedged ones, and the policy optimized against these reward models inherits this bias (Sharma et al., 2024). The result is a model that sounds more reliable than it is, with confidence increasingly decoupled from correctness as the post-training procedure progresses. Tian et al. (2023) document this empirically: on RLHF-tuned models, verbalized confidence is typically better calibrated than internal token probabilities, an inversion that points to RLHF having a particularly strong effect on the calibration of internal probabilities.

These sources matter for the rest of the thesis because they are difficult to fix by retraining alone. The procedures that produce miscalibration are the same procedures that produce the instruction-following behavior that makes the models useful, and removing one would remove the other. A test-time method that corrects confidence without touching the base capabilities is therefore a natural response.

2.2.4 Verbalized vs. Internal Confidence

There are two ways to extract a confidence estimate from an LLM. The first, which we call *internal confidence*, reads the probability of the predicted token directly from the softmax over the vocabulary. For a multiple-choice question with answer tokens $\{A, B, C, D\}$, the internal confidence in answer A is the model's predicted probability $p_\theta(A \mid \text{question})$, computed in a single forward pass. Internal confidence is cheap, well-defined, and available for any model with accessible logits.

The second, which we call *verbalized confidence*, asks the model to state its confidence in natural language, for example by appending "How confident are you in your answer? Reply with a number between 0 and 100." to the prompt. The reported number is then parsed and divided by 100. Verbalized confidence requires the model to follow the elicitation instruction and to map an internal sense of certainty onto a numerical scale, both of which are non-trivial for small models.

Which of the two is better calibrated depends on the model. For RLHF-tuned models, Tian et al. (2023) report that verbalized confidence is better calibrated than the model's own token probabilities on several large RLHF models. The mechanism behind this inversion is not fully understood; one possibility is that preference optimization affects

the calibration of internal token probabilities more strongly than it affects the model's ability to introspect on its own answer through a separate elicitation, but Tian et al. (2023) document the empirical effect without committing to a specific causal account. This thesis works primarily with verbalized confidence, both because it is the signal most relevant in deployment, where users see numbers rather than logits, and because it is the signal SECL adapts directly.

2.3 Uncertainty Estimation and the Generation-Discrimination Gap

2.3.1 Entropy as an Uncertainty Signal

The most immediate uncertainty signal in an LLM is the entropy of its next-token distribution. For a distribution $p_\theta(\cdot \mid x_{<t})$ over a vocabulary V, the Shannon entropy is

$$H(p_\theta(\cdot \mid x_{<t})) = -\sum_{v \in V} p_\theta(v \mid x_{<t}) \log p_\theta(v \mid x_{<t}). \tag{2.7}$$

Entropy is maximized when the distribution is uniform and minimized when the distribution concentrates on a single token. A low-entropy distribution indicates that the model has strong prior evidence for one continuation; a high-entropy distribution indicates that many continuations are roughly equally likely. Both facts can be read off a single forward pass, which makes entropy the cheapest uncertainty signal available.

Entropy is informative but imperfect as a calibration signal. A confidently wrong model can produce low entropy on an incorrect answer, and semantic equivalence among tokens can inflate entropy on cases where the model is actually certain about the underlying content (Kuhn et al., 2023). It is therefore unsuitable as a direct self-supervision target for calibration. We instead use it as a gating signal, on the assumption that running entropy can serve as a detector of distributional change in the test stream, an assumption we validate empirically in Chapter 6. The full construction is given in the Methods chapter.

2.3.2 P(True) and Self-Evaluation

A more direct probe of the model's own belief about its answer is the P(True) construction introduced by Kadavath et al. (2022). After the model generates an answer a to a question q, the question, the answer, and a follow-up prompt asking whether the answer is correct are concatenated, and the model's probability of emitting the token "True" is read off the next-token distribution. Schematically:

$$P_\theta(\text{True} \mid q, a) = p_\theta(\text{True} \mid \text{"Q: } q\text{, A: } a\text{. Is this correct? "}), \tag{2.8}$$

with the exact prompt template varying by implementation. The key property of P(True) is that it is computed in a single forward pass, requires no labels, and provides a scalar in $[0, 1]$ that correlates with correctness.

Kadavath et al. (2022) showed that P(True) is substantially better calibrated than the probability the model assigns to the answer during generation, with calibration improving as model scale increases. The intuition is that producing a correct answer

requires generating it from a large output space, while evaluating a proposed answer requires only a comparison against the model's knowledge. The latter is generally easier, and Kalai et al. (2025) formalize this intuition as a quantitative bound (Section 2.3.3).

P(True) on a single answer is still noisy, and recent work has explored more robust variants. Wang and Stengel-Eskin (2025) propose computing P(True) against a set of plausible alternatives rather than just the model's own answer, and normalizing across the set. SECL uses a similar normalization, which we describe in Section 2.3.4.

2.3.3 The Generation-Discrimination Gap

The empirical observation that P(True) outperforms generative confidence has a theoretical counterpart. Kalai et al. (2025) analyze a family of language generation tasks and prove that the generative error of any model is lower-bounded by approximately twice its discriminative error on the associated binary task, up to additional terms that depend on the answer-space structure and the model's calibration. Formally, for a model evaluated on a population of questions with a single correct answer each, the probability of generating a wrong answer is at least roughly 2 times the probability of misclassifying a candidate answer as correct or incorrect, with corrections that vanish in the limit of well-calibrated discriminative judgments.

The intuition is the recognition-versus-recall asymmetry familiar from human cognition: selecting the correct answer from a finite set of candidates tends to be easier than producing it from scratch. This is an intuition pump rather than a formal correspondence, but it gestures at the same structure the Kalai bound makes precise: the generation task requires the model to concentrate probability on a small set of correct tokens amid many incorrect ones; the discrimination task only requires the model to put a threshold-crossing probability on a single candidate.

This gap is what SECL exploits. The discriminative probe (P(True) on the frozen base model) is better calibrated than the generative verbalized confidence on the same model. Using the former to supervise the latter therefore has a well-defined direction of potential improvement, and the gap is available on every question without any labels. Section 2.5 discusses the other half of the method: how to actually apply this signal as an update, and how to prevent it from destabilizing the base model.

2.3.4 Normalizing Confidence Across Distractors

A raw P(True) score on a single answer is scale-dependent, and the absolute number does not directly correspond to a confidence level on the same scale as verbalized confidence. Wang and Stengel-Eskin (2025) document that this affects calibration in practice through suggestibility bias: the model tends to affirm any answer presented to it, inflating P(True) regardless of correctness. A calibration target built on raw P(True) therefore inherits this bias.

The fix is to normalize across a set of plausible alternatives. Given a question q and the model's answer a, a small set of distractors $\{d_1, \dots, d_K\}$ is generated, P(True) is computed for each candidate in the set $\{a, d_1, \dots, d_K\}$, and the answer's score is normalized by the total mass on the full set. The resulting quantity lies in $[0, 1]$, integrates over baseline suggestibility, and can be read as the model's estimated probability that a is the correct answer out of the proposed alternatives. Wang and Stengel-Eskin (2025) introduced this construction for static calibration; SECL uses it as the training

target for test-time LoRA updates. The precise form, along with how distractors are generated, is given in the Methods chapter.

2.4 Parameter-Efficient Fine-Tuning

2.4.1 Full Fine-Tuning and Its Limits

The default way to adapt a pretrained model to a new objective is to update all of its parameters. For models with billions of parameters, this is expensive on two fronts. Memory scales with the parameter count: gradients and optimizer state each occupy as much memory as the model itself, so training a 7B-parameter model with Adam requires at least four times the parameter count in GPU memory for weights, gradients, and two moment buffers. Even when memory is available, full fine-tuning on small update budgets can be unstable and overfit-prone, and applying it repeatedly in a test-time setting would both risk destabilizing the model and prevent the base model from being reused across adaptations. Parameter-efficient fine-tuning (PEFT) methods address these issues by restricting updates to a small subset of new parameters inserted into the network, while keeping the pretrained weights frozen.

2.4.2 Low-Rank Adaptation (LoRA)

Low-Rank Adaptation (EJ Hu et al., 2022) is the PEFT method SECL uses. The idea is that the update ΔW to a pretrained weight matrix $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ during fine-tuning has a low effective rank, and can be parameterized as the product of two thin matrices. Formally, LoRA replaces W with

$$W' = W + \Delta W = W + \frac{\alpha}{r} BA, \tag{2.9}$$

where $B \in \mathbb{R}^{d_{\text{out}} \times r}$ and $A \in \mathbb{R}^{r \times d_{\text{in}}}$, the rank $r \ll \min(d_{\text{out}}, d_{\text{in}})$, and α is a scaling factor. At initialization, A is sampled from a Gaussian and B is zero, so $\Delta W = 0$ and the adapted model matches the base model exactly. Only A and B are updated during training; W is frozen.

The practical effect is that the number of trainable parameters drops from $d_{\text{out}} \cdot d_{\text{in}}$ to $r \cdot (d_{\text{out}} + d_{\text{in}})$, typically by two to four orders of magnitude. Gradients and optimizer state are correspondingly smaller, and the adapter can be merged back into W at inference time via $W' = W + (\alpha/r)BA$ with no additional compute. The rank r is the main capacity knob: higher r gives more expressive updates but more parameters and more instability on small training sets.

LoRA is typically applied to a subset of the linear projections in the Transformer block, most commonly the query and value projections in the attention layers (W_Q and W_V in Equation 2.3), since these were found to be the most impactful in the original study (EJ Hu et al., 2022). Our ablations vary this choice, including adapting the full set of attention projections (Q, K, V, O) and the feedforward projections, to measure how the target module set affects calibration adaptation.

2.5 Test-Time Training

2.5.1 From Domain Adaptation to Test-Time Training

Standard machine learning assumes that the test distribution matches the training distribution. This assumption fails routinely in deployment: medical models trained on one hospital's data see a different patient mix at another, and language models trained on web text encounter domain-specific jargon, dialects, or question styles not represented in training. Domain adaptation addresses this when unlabeled target-domain data is available at training time, by modifying the training procedure to produce representations that transfer. Test-time training (TTT) (Sun et al., 2020) pushes this further: the model adapts its own weights to the test stream as it encounters it, with no access to target labels and no separate adaptation phase.

The original TTT procedure of Sun et al. (2020) attaches a self-supervised auxiliary head to the model during training, for example a rotation-prediction head on image inputs. At test time, the main classification head is frozen and the auxiliary head is used to produce a self-supervised loss on the incoming test example, which is backpropagated to update shared feature layers. The prediction is then made on the adapted model. The method improved robustness to distribution shifts on image classification benchmarks, and the core idea, turning a test example into a self-supervised learning problem, has been the template for subsequent work.

2.5.2 Self-Supervision Signals for TTT

The central design choice in any TTT method is the self-supervision signal, since the quality of that signal caps the quality of the adapted model. The dominant family in computer vision is entropy minimization. TENT (D Wang et al., 2021) updates batch normalization parameters to minimize the Shannon entropy of the model's output distribution on test examples, on the principle that a confident classifier is, on average, a correct one. This is a strong assumption that can fail on miscalibrated models, which is part of the motivation for our work, and it does not transfer cleanly to autoregressive language generation, where token-level entropy is not a reliable correctness proxy.

Other approaches in the broader TTT literature use feature reconstruction, consistency across augmented views of the same input, or auxiliary tasks introduced at training time. The common pattern is that TTT methods stand or fall on whether their self-supervised signal is informative about the target task. For classification, entropy minimization and rotation prediction are reasonably informative. For calibration specifically, no previous TTT method has provided a signal at all, because the standard TTT signals are either agnostic to calibration (rotation prediction) or actively bad for it (entropy minimization, which pushes all predictions toward maximum confidence regardless of correctness). SECL's contribution on the signal side is to use the generation-discrimination gap, which is informative about calibration precisely because it is the gap calibration methods are trying to close.

2.5.3 Continual Test-Time Adaptation

Applying a TTT update once to each test example is the simple case. The more realistic setting is continual: the model sees a long stream of test examples from potentially

shifting distributions, and adaptation must accumulate across the stream without collapsing. Q Wang et al. (2022) study this setting and identify two failure modes. The first is catastrophic forgetting, in which repeated updates on recent examples erase the model's ability to perform on earlier, out-of-stream inputs. The second is error accumulation, in which noisy self-supervision signals compound over time, producing a slow drift away from useful behavior.

Their fix, CoTTA, combines stochastic restoration of weights to their pretrained values, a teacher-student update with a slowly-moving teacher, and a weighted averaging of predictions. The broader lesson is that continual TTT requires mechanisms to control when the model adapts, how much it adapts, and in what direction. SECL addresses these three concerns with entropy-based gating (Section 2.3.1), bounded LoRA updates (Section 2.4.2), and a directional loss that only moves verbalized confidence when the self-supervision signal disagrees with it by more than one bin. These design choices are discussed in detail in the Methods chapter.

2.5.4 Test-Time Training for Language Models

TTT has recently been extended from computer vision to language models, with every extension so far targeting accuracy rather than calibration. Hardt and Sun (2024) retrieve nearest-neighbor training examples at test time and fine-tune the model on them before generating the answer, showing gains on long-tail knowledge tasks. J Hu et al. (2025) formulate test-time learning for LLMs as input-perplexity minimization on the unlabeled test stream, using LoRA updates to preserve base-model knowledge. Sun et al. (2025) build TTT into the architecture itself, replacing the attention mechanism with a hidden state that is itself a model updated by self-supervised steps at test time. Zweiger et al. (2025) propose a framework in which the model adapts its own weights based on self-generated supervision signals.

These works establish that TTT for LLMs is viable and that LoRA is a natural vehicle for it, but none of them address calibration. The self-supervision signals they use (perplexity minimization, nearest-neighbor fine-tuning, reconstruction of hidden-state targets) are informative about accuracy and fluency but not about whether the model's expressed confidence matches its accuracy. SECL is, to our knowledge, the first TTT method targeting calibration, and it does so by using the generation-discrimination gap as the missing self-supervision signal.

3
Related Work

This chapter surveys prior work on LLM calibration and test-time adaptation, grouped into five threads. Sections 3.1–3.3 cover the three main families of calibration methods (sampling-based, static probing, and training-based) and identify the limitations that motivate a test-time approach. Section 3.4 covers the generation-discrimination gap, which is the theoretical and empirical basis for SECL's self-supervision signal. Section 3.5 covers test-time training for LLMs, the paradigm SECL extends. Section 3.6 then positions SECL against this landscape and identifies the specific gap in the literature that this thesis fills.

3.1 Sampling-Based Uncertainty Estimation

SelfCheckGPT (Manakul et al., 2023) detects hallucinations by comparing each sentence (S) against N=20 sampled passages at the cost of $\mathcal{O}(S \times N)$. Semantic Entropy (Kuhn et al., 2023) reduces surface sensitivity by clustering semantically equivalent responses before computing entropy, but still requires multiple generations and cannot resolve consistent falsehoods. Ma et al. (2025) improve detection in single-cluster failure cases by working on penultimate-layer logits. Xiong et al. (2024) benchmark these and other black-box elicitation methods comprehensively, finding that systematic overconfidence is inherent to all strategies; Heo et al. (2025) confirm this for instruction-tuned models specifically. All sampling methods share two limitations: high cost at inference time, and failure on consistent hallucinations where the model is confidently wrong across all sampled responses (Lin et al., 2022b).

3.2 Static Probing and Lightweight Calibration

White-box methods bypass sampling by analyzing the model's internals directly. HaloScope (Du et al., 2024) shows that hallucinations are geometrically distinct in intermediate-to-late layer embeddings, achieving strong AUROC with reasonable cross-dataset transfer, though the authors note degradation under drastic distribution shift. LitCab (X Liu et al., 2024) adds a single linear layer (<2% of parameters)

that predicts a logit bias, reducing ECE by up to 30%. Although these methods are efficient, they are static: because they are trained offline, they cannot adapt when the input distribution shifts at test time.

3.3 Training-Based Calibration

Lin et al. (2022a) showed that supervised fine-tuning with calibrated confidence labels produces well-calibrated models, though evaluation did not extend beyond math tasks. Stangel et al. (2025) use Reinforcement Learning (RL) with a logarithmic scoring rule to penalize overconfidence. TruthRL (Z Wei et al., 2025) uses a ternary reward to incentivize abstention over hallucination when models are uncertain. Damani et al. (2025) use the Brier score (Brier, 1950) as an RL reward, reducing calibration error by up to 90% in-domain, but found that standard RL degrades calibration OOD, directly motivating a test-time approach that can adapt to unseen domains. Prompting methods such as Fact-and-Reflection (Zhao et al., 2024) reduce ECE without training but remain static.

A fundamental limitation of training-based approaches is the difficulty of specifying knowledge boundaries in black-box LLMs: RLHF methods rely on human labels that introduce sycophancy (Sharma et al., 2024), and RLHF fine-tuning itself degrades calibration (Tian et al., 2023) despite models retaining well-calibrated internal judgments (Kadavath et al., 2022).

3.4 The Generation-Discrimination Gap

Kadavath et al. (2022) established that LLMs' discriminative judgments $P(\text{True})$ are well-calibrated and improve with scale. Tian et al. (2023) extended this to RLHF-tuned models, showing that verbalized confidence is typically better calibrated than conditional token probabilities, even though RLHF degrades the latter. Building on this line of work, Kalai et al. (2025) provided a theoretical basis: generative error is lower-bounded by approximately twice the misclassification rate of the corresponding binary validity problem.

Wang and Stengel-Eskin (2025) exploit this gap directly: their DINCO method normalizes verbalized confidence across natural language inference (NLI)-reweighted distractors and integrates self-consistency, outperforming prior baselines. However, DINCO is a static inference-time technique that cannot adapt when the underlying truthfulness signal is brittle under distribution shift (Haller et al., 2025). SECL addresses this by distilling the discriminative signal into the model's weights, enabling continuous adaptation. We provide a direct empirical comparison in Section 6.1.4.

3.5 Test-Time Adaptation for LLMs

Test-time training (TTT) adapts model weights using unsupervised signals from incoming test data, originating in computer vision (Sun et al., 2020; D Wang et al., 2021; Q Wang et al., 2022) and recently extended to LLMs for accuracy improvement (Hardt and Sun, 2024; J Hu et al., 2025; Sun et al., 2025; Zweiger et al., 2025). Snell et al. (2025) showed that adaptive test-time compute allocation can outperform much larger models,

and Huang et al. (2025) use calibrated confidence to allocate such compute, but their goal is accuracy, not calibration.

3.6 Positioning SECL

SECL sits at the intersection of these five threads and inherits a design constraint from each. From sampling-based methods, the recognition that repeated generation is too expensive to deploy continuously. From static probing methods, the recognition that offline-fit calibrators degrade exactly where calibration matters most, under distribution shift. From training-based methods, the recognition that supervised calibration works in-domain but does not transfer, and that RLHF as a post-training procedure is itself a source of miscalibration that retraining cannot easily undo. From the generation-discrimination gap literature, the empirical and theoretical argument that a label-free self-supervision signal exists. From test-time training for LLMs, the architectural template (LoRA updates driven by an unsupervised signal) that SECL adapts.

No prior method combines these threads. Sampling methods are expensive and cannot self-correct. Static probes are cheap but cannot adapt. Training-based methods either require labels or degrade out-of-distribution. DINCO (Wang and Stengel-Eskin, 2025) is the closest in spirit because it uses the generation-discrimination gap, but it applies the gap at inference time without updating the model, so it inherits the same distribution-shift fragility as other static methods. Test-time training methods for LLMs (Hardt and Sun, 2024; J Hu et al., 2025; Sun et al., 2025; Zweiger et al., 2025) target accuracy rather than calibration and therefore use signals (perplexity minimization, nearest-neighbor fine-tuning, reconstruction) that are uninformative about whether the model's expressed confidence matches its accuracy. SECL is the first method to apply test-time training to calibration, and it does so by identifying the generation-discrimination gap as the calibration-specific self-supervision signal that the TTT literature was missing.

4 Methods

This chapter presents SECL in full. We first state the core idea and give a pipeline overview (Section 4.1), then describe the three components in turn: the entropy-based gating mechanism that triggers adaptation on distribution shifts (Section 4.2), the normalized $P(\text{True})$ signal that serves as the self-supervision target (Section 4.3), and the test-time calibration loop that applies LoRA updates when the signal disagrees with verbalized confidence (Section 4.4). Section 4.5 gives the full procedure as pseudocode, and Section 4.6 discusses the design choices that fit these components together and the alternatives that were considered and rejected.

4.1 Overview

The core idea is simple: when the model encounters a new type of question, it checks whether its stated confidence matches its own self-assessment. If the model claims 90% confidence but its True/False self-check suggests only 30%, a small weight update corrects this mismatch. Over time, these corrections accumulate, producing better-calibrated confidence.

Concretely, SECL operates in three stages (Figure 4.1): (1) an entropy-based change detector triggers adaptation only on distribution shifts (Section 4.2); (2) a normalized discriminative signal, $\text{NormP}_{\text{True}}$, scores model answers (Section 4.3); (3) when this signal disagrees with verbalized confidence, lightweight LoRA updates reduce the gap (Section 4.4). Each component is validated individually in Chapter 6.

The three components address the two obstacles identified in Chapter 1 that have prevented previous work from applying test-time training to calibration. The gating mechanism keeps the cost of adaptation below the cost of the supervision signal it distills, which solves the expense problem. The directional loss with bounded updates and the bin-gate filter keep the method stable under imperfect self-supervision, which solves the instability problem. The next three sections describe each component in detail; Section 4.6 returns to the connections between them.

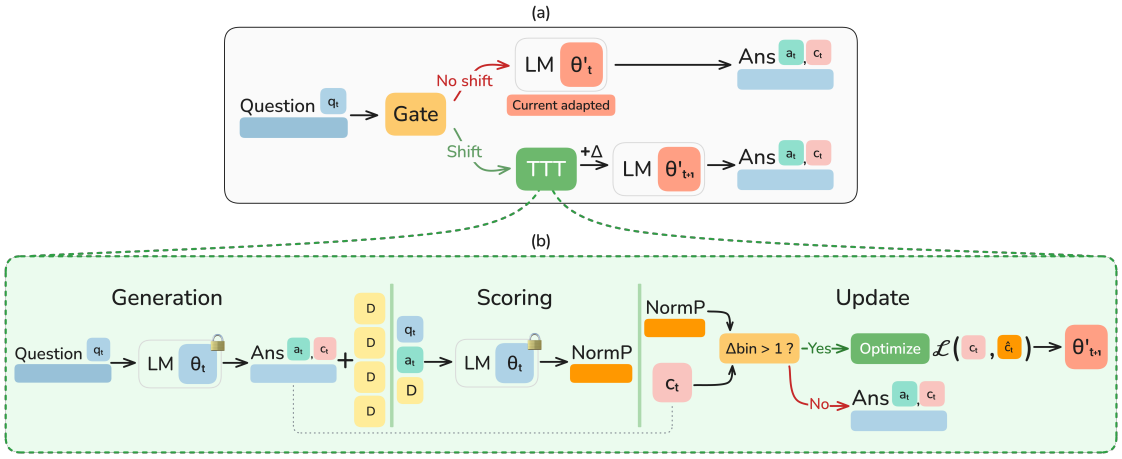

Figure 4.1: Overview of SECL. **(a)** *Test-Time Inference.* An entropy-based change detector (Section 4.2) monitors the input stream. If no shift is detected, the adapted model θ'_t is used directly; otherwise, a calibration burst updates it to θ'_{t+1}. **(b)** *Calibration Burst.* For each of B=50 questions: the frozen model generates an answer with confidence c_t and distractors, computes $\text{NormP}_{\text{True}}$ (Section 4.3), and applies a LoRA update when the two signals disagree by more than one bin (Section 4.4). Weights accumulate across questions without resetting.

4.2 Adaptive Entropy Gating

The calibration procedure (Sections 4.3–4.4) requires computing $\text{NormP}_{\text{True}}$ and running LoRA updates for each question. Once the model has adapted to a domain, these updates are redundant; the current LoRA weights already reflect the calibration characteristics of the current distribution. Applying them on every question would be wasteful in two ways: the supervision signal itself costs five forward passes to compute, and repeated gradient updates on an already-calibrated stream introduce noise without a corresponding benefit. Calibration is therefore triggered only when the input distribution shifts.

We track the entropy H_t of the model's output token distribution with an exponential moving average (EMA, smoothing factor α_{ema}) and apply the Page-Hinkley (PH) change detection test (Page, 1954). The PH test maintains a cumulative sum:

$$m_t = \sum_{s=1}^{t} \left(H_s - \bar{H}_t - \epsilon \right), \tag{4.1}$$

where $\bar{H}_t$ is the running mean entropy, H_s the entropy at step s, and ϵ is a tolerance that suppresses false alarms from minor fluctuations. An alarm fires when $m_t - \min_{s \le t} m_s > \lambda$; a higher λ requires a larger cumulative entropy deviation before triggering, controlling the trade-off between responsiveness and false alarm rate.

Upon detection, cumulative statistics are reset, and a calibration burst of B consecutive questions is initiated. Processing multiple questions per burst is crucial: a single LoRA update provides too little signal for stable adaptation, whereas a burst amortizes the cost of entering calibration mode and allows corrections to accumulate across diverse questions from the new distribution. The ablation in Section 6.2.3 shows that $B = 20$ yields ECE of 0.114 versus 0.050 for $B = 50$, confirming that sufficient distillation per trigger is critical.

LoRA weights accumulate across domains without resetting. When a new distribution shift is detected, the subsequent calibration burst adapts the existing accumulated

weights rather than starting from scratch, allowing calibration knowledge from earlier domains to compound. Section 6.2.1 shows that disabling accumulation is catastrophic: ECE degrades from 0.050 to 0.237 and AUROC drops below chance, because isolated single-question updates inject noise rather than information into the confidence token.

4.3 Normalized P(True) as Self-Supervision

Given a question q and the model's generated answer a, we compute $P_{\text{True}}(a \mid q)$: the token probability of "True" when the model is asked "Is the following answer to the question correct? (True/False)". Following Kadavath et al. (2022), this discriminative signal is well-calibrated and improves with scale, making it a stronger supervision target than the model's own verbalized confidence c. The theoretical grounding for this choice is the generation-discrimination gap of Kalai et al. (2025) discussed in Section 2.3.3: a model's ability to evaluate a candidate answer is systematically better than its ability to produce one, and the evaluation probability is therefore a natural supervision target for the generation-side confidence.

4.3.1 Verbalized Confidence

Following Lin et al. (2022a) and Tian et al. (2023), we elicit verbalized confidence by prompting the model to state a confidence bin (0–9) alongside its answer, where each bin corresponds to a 10-percentage-point interval (e.g., bin 7 ≈ 70–80%). We use 10 bins to ensure confidence can be expressed as a single generated token (Naeini et al., 2015; Guo et al., 2017). Rather than taking the argmax bin (hard readout), we compute a *soft* confidence as the expected value over the digit-token probability distribution:

$$c = \sum_{k=0}^{9} P(\text{bin}_k) \cdot \frac{k + 0.5}{10}, \tag{4.2}$$

where $P(\text{bin}_k)$ is the model's output probability for digit token k. This soft readout preserves information from the full distribution and provides a differentiable signal for the mean squared error (MSE) loss in Section 4.4.

4.3.2 Distractor Normalization

Raw P_{True} suffers from suggestibility bias: the model tends to affirm any answer presented to it, inflating P_{True} regardless of correctness (Wang and Stengel-Eskin, 2025), and degrades under distribution shift (Haller et al., 2025). We address both by normalizing P_{True} across distractor answers. For multiple-choice questions, we use the given answer options as distractors. For open-ended questions, we generate K=4 plausible alternatives by sampling from the model at higher temperature. The normalized signal is:

$$\text{NormP}_{\text{True}}(a) = \frac{e_a}{e_a + \sum_{k=1}^{K} e_{d_k}}, \tag{4.3}$$

where $e_x = \exp\big(P_{\text{True}}(x)/\tau\big)$, with τ as a model-specific temperature. This softmax over distractors converts the raw signal into a *relative* confidence that accounts for baseline suggestibility. When the model cannot distinguish its answer from distractors (equal

P_{True} scores), $\text{NormP}_{\text{True}} \to 1/(K+1)$, correctly indicating low confidence; when the model strongly prefers its answer, $\text{NormP}_{\text{True}} \to 1$.

Unlike DINCO (Wang and Stengel-Eskin, 2025), we use a simple softmax over distractors without NLI reweighting or self-consistency. This design choice is deliberate: the ablation in Section 6.2.2 shows that the calibration of the supervision signal is what determines the calibration of the adapted model, and the simple normalized signal is already well-calibrated enough to serve as a ceiling. Additional sophistication in the signal would add computational cost without a guaranteed improvement in calibration. The resulting $\text{NormP}_{\text{True}}$ provides a continuous training target, which is discretized into the same 10 equal-width bins used for verbalized confidence, ensuring a common scale between supervision target and model output.

4.4 Test-Time Calibration via LoRA

When the model's verbalized confidence disagrees with $\text{NormP}_{\text{True}}$, we update the model to reduce this disagreement. Updates are applied via LoRA (EJ Hu et al., 2022) on intermediate-to-late transformer layers, motivated by the finding that calibration-relevant representations concentrate in these layers (Du et al., 2024). Architecture-specific layer configurations are reported in Section 5.5, and the layer ablation is reported in Section 6.2.4.

4.4.1 Directional Training Target

Although distractor normalization suppresses systematic biases (Section 4.3.2), the discriminative signal can still be noisy on individual questions. A single P_{True} readout reflects the model's evaluation on one specific prompt with one specific answer and one specific set of distractors, and any of these can be atypical in ways that move $\text{NormP}_{\text{True}}$ away from the underlying correctness probability. Using $\text{NormP}_{\text{True}}$ directly as the training target would propagate this per-question noise into the weight update.

We therefore do not use $\text{NormP}_{\text{True}}$ directly. Instead, rather than jumping directly to the discriminative estimate, we nudge confidence toward it in small, bounded steps. Let c_i be the model's verbalized confidence and c_i^* the $\text{NormP}_{\text{True}}$ value for question i. The training target is:

$$\hat{c}_i = c_i + \alpha_{\text{step}} \cdot \text{clip}\big(c_i^* - c_i,\ -\delta,\ \delta\big), \tag{4.4}$$

where $\text{clip}(x, -\delta, \delta) = \max(-\delta, \min(x, \delta))$ clamps the correction magnitude to the interval $[-\delta, \delta]$, α_{step} controls the correction step size, and δ caps the maximum single-step adjustment. We fix $\alpha_{\text{step}} = 0.5$ and $\delta = 0.15$ across all models.

The two parameters serve different purposes. The step size α_{step} controls how aggressively the model commits to the discriminative estimate on a single question, trading responsiveness against stability: higher values converge faster on clean signals but overshoot on noisy ones. The clip bound δ places a hard ceiling on the per-question correction, which limits the damage from any single noisy $\text{NormP}_{\text{True}}$ reading no matter how far it deviates from the model's current verbalized confidence. Together, they implement a form of conservative gradient clipping in the output space rather than in the gradient space, which the ablation in Section 6.2.1 shows is more effective than plain MSE against $\text{NormP}_{\text{True}}$.

4.4.2 Loss and Optimization

The training loss is mean squared error between verbalized confidence and the directional target:

$$\mathcal{L}_i = \left(c_i - \hat{c}_i\right)^2. \tag{4.5}$$

We optimize with AdamW (Loshchilov and Hutter, 2019); learning rate and epoch count are reported in Section 5.5. Crucially, $\text{NormP}_{\text{True}}$ is computed from the base model without LoRA adapters, ensuring the supervision signal is not corrupted by ongoing adaptation. If the adapter were included in the $\text{NormP}_{\text{True}}$ computation, the adapted model would supervise itself, and any drift in verbalized confidence would be reflected back as a drifted target, producing a feedback loop with no fixed point. Decoupling the supervision signal from the adapted model breaks this loop and keeps the method stable over long streams.

4.4.3 Bin-Gate Filter

Not every question requires calibration. We skip training when the model is already approximately calibrated on a given question: specifically, when $|\text{bin}(c_i) - \text{bin}(c_i^*)| \leq 1$. This avoids gradient updates on questions where verbalized and discriminative confidence already agree, reducing computation and limiting noise from marginal disagreements. We use threshold 1 as default; the ablation in Section 6.2.1 shows that combining entropy gating with the bin-gate filter processes only 25.6% of the stream while matching the calibration of always-on adaptation.

4.5 The Full SECL Procedure

Algorithm 1 gives the full SECL procedure as pseudocode. The outer loop processes the test stream one question at a time. For each question, the frozen base model produces an answer and its verbalized confidence (line 4). The entropy of the answer distribution updates the Page-Hinkley change detector (line 5). When a distribution shift is detected, a calibration burst of B questions begins (line 7); during the burst, $\text{NormP}_{\text{True}}$ is computed from the base model (line 10), the bin-gate filter checks whether the question is worth training on (line 11), and if so, the directional target and MSE loss drive a LoRA update (lines 12–14). The adapted model is then used for any non-burst questions between shifts (line 17).

Two properties of the procedure are worth noting. First, the LoRA adapter ϕ is never reset: the update on line 14 modifies the adapter in place, and subsequent bursts continue from the current state. This is the weight accumulation that Section 6.2.1 shows is essential. Second, the $\text{NormP}_{\text{True}}$ computation on line 10 uses θ, not $\theta + \phi$. The supervision signal comes from the base model throughout, so no feedback loop forms between the adapter and its target.

Algorithm 1 SECL: Self-Calibrating Language Models via Test-Time Discriminative Distillation

Require: Base model θ, stream $\{q_t\}_{t=1}^{T}$, LoRA adapter ϕ initialized to zero, Page-Hinkley detector PH with threshold λ, burst length B, step size α_{step}, clip δ, bin-gate threshold β

$\phi \leftarrow 0$; PH.reset()
for $t = 1, \ldots, T$ **do**
  $a_t, c_t, H_t \leftarrow \text{generate}(\theta + \phi, q_t)$ ▷ answer, verbalized confidence, entropy
  PH.update(H_t)
  if PH.alarm() **or** in_burst **then**
    if not in_burst **then**
      in_burst $\leftarrow$ **true**; burst_remaining $\leftarrow B$; PH.reset()
    end if
    $d_1, \ldots, d_K \leftarrow \text{distractors}(q_t, a_t)$
    $c_t^* \leftarrow \text{NormP}_{\text{True}}(\theta, q_t, a_t, d_{1:K})$ ▷ computed on base model, no LoRA
    if $|\text{bin}(c_t) - \text{bin}(c_t^*)| > \beta$ **then**
      $\hat{c}_t \leftarrow c_t + \alpha_{\text{step}} \cdot \text{clip}(c_t^* - c_t, -\delta, \delta)$
      $\mathcal{L} \leftarrow (c_t - \hat{c}_t)^2$
      $\phi \leftarrow \phi - \eta \nabla_\phi \mathcal{L}$ ▷ AdamW step on LoRA parameters
    end if
    burst_remaining $\leftarrow$ burst_remaining -1
    if burst_remaining $= 0$ **then** in_burst $\leftarrow$ **false**
    end if
  end if
end for

4.6 Design Choices and Alternatives

SECL combines three components that each have obvious alternatives. This section briefly justifies each choice against its main alternative, with reference to the ablations in Chapter 6.

Why entropy gating rather than a fixed schedule? A fixed-schedule approach, for example running a calibration burst every N questions, has the advantage of simplicity but the disadvantage of wasting compute on stable streams and missing shifts that fall between scheduled bursts. Entropy gating ties adaptation to the underlying property we care about (distributional change) rather than a clock. The ablation in Section 6.2.1 shows that entropy-gated bursts match always-on adaptation at one quarter of the compute.

Why $\text{NormP}_{\text{True}}$ rather than raw P_{True} or self-consistency? Raw P_{True} is known to suffer from suggestibility bias (Wang and Stengel-Eskin, 2025) and performs worse as a supervision signal than its normalized counterpart (Section 6.2.1). Self-consistency is a plausible alternative since it also produces a scalar confidence without labels, but the ablation in Section 6.2.2 shows that self-consistency is a systematically biased proxy for correctness at the scale of the generation distribution, and using it as the training target degrades calibration to 2.5 times worse than the untrained baseline. $\text{NormP}_{\text{True}}$ is the best-calibrated signal available at reasonable cost.

Why LoRA rather than full fine-tuning or prompt tuning? Full fine-tuning would require gradients and optimizer state for the full model at test time, which is prohibitive for 3–8B parameter models. It would also make catastrophic forgetting considerably harder to control, because every parameter is in play on every update. Prompt tuning, at the other extreme, modifies only the input and therefore cannot change the model's confidence representation directly. LoRA sits at the right point on this spectrum: it modifies a small subset of parameters (0.01–0.02% of total) concentrated in the layers most relevant to calibration, keeping memory and stability manageable while still being expressive enough to close the generation-discrimination gap.

Why the directional loss rather than plain MSE? Plain MSE against $\text{NormP}_{\text{True}}$ treats every per-question supervision signal as equally trustworthy, which overweights noisy readings and can drive the adapted confidence past the underlying correctness probability. The directional formulation limits both the magnitude and the direction of each update, which is effectively a form of output-space clipping. The ablation in Section 6.2.1 shows this raises ECE from 0.085 (plain MSE) to 0.052 (directional) on the same model and signal.

5 Experiments

This chapter describes the experimental setup used to evaluate SECL: the datasets, models, baselines, metrics, and implementation details. Results from these experiments are reported in Chapter 6.

5.1 Datasets

We evaluate on four datasets that differ in reasoning type, difficulty, and degree of model overconfidence. We sample 500 questions per domain for a total of 2,000 questions per run.

5.1.1 GSM8K

GSM8K (Cobbe et al., 2021) is a dataset of 8,500 grade-school math word problems created by OpenAI to evaluate multi-step arithmetic reasoning. Each problem requires two to eight steps of elementary arithmetic to reach a single numeric answer, making it a test of sequential reasoning rather than factual recall. We use the first 500 problems of the train split. GSM8K is the most reasoning-intensive of our four domains, and models are often well-calibrated here because incorrect reasoning chains tend to produce visibly uncertain answers.

5.1.2 MMLU

MMLU (Hendrycks et al., 2021) (Massive Multitask Language Understanding) is a multiple-choice benchmark covering 57 subjects ranging from elementary mathematics to professional law and medicine, designed to measure breadth of world knowledge acquired during pretraining. Each question has four options and a single correct answer. We sample 500 questions round-robin across subjects from the test split to ensure balanced subject coverage. MMLU tests factual knowledge rather than reasoning, and models are frequently overconfident here, asserting high confidence on questions outside their knowledge.

5.1.3 ARC Challenge

ARC Challenge (Clark et al., 2018) (AI2 Reasoning Challenge) consists of grade-school science questions specifically filtered to contain only questions that simple retrieval and word-association methods answer incorrectly, making it a harder subset than standard science QA. Each question is multiple-choice with three to five options. We use the first 500 questions of the test split. ARC sits between GSM8K and MMLU in character, requiring both factual science knowledge and light reasoning.

5.1.4 TruthfulQA

TruthfulQA (Lin et al., 2022b) is a benchmark of 817 questions adversarially constructed to elicit common human misconceptions, such as false beliefs and conspiracy theories, on which models tend to reproduce the popular but incorrect answer. We use the MC1 (single-correct multiple-choice) variant of the validation split, first 500 questions. TruthfulQA is the domain where models are most severely miscalibrated, since the adversarial construction specifically targets confident wrong answers, which makes it the most informative domain for evaluating a calibration method.

5.1.5 Continual Protocol

The four domains are presented sequentially (GSM8K → MMLU → ARC → TruthfulQA), forming a single stream of 2,000 questions with distribution shifts between domains. Per-domain and ordering results are reported in Section 6.3.1 and Appendix B.6; an additional open-ended variant using TruthfulQA generation answers is reported in Appendix B.10.3.

5.2 Models

We evaluate four instruction-tuned small language models spanning 2–8B parameters and three model families: Llama 3.2-3B-Instruct and Llama 3.1-8B-Instruct (Grattafiori et al., 2024), Gemma 2-2B-IT (Team et al., 2024), and Phi 3.5-Mini-Instruct (3.8B) (Abdin et al., 2024). These were selected because all four exhibit a measurable generation-discrimination gap: their $\text{NormP}_{\text{True}}$ signal is better calibrated than their verbalized confidence, which is the prerequisite for SECL. A negative control on Qwen 2.5-3B (Qwen Team et al., 2025), where this gap is absent, is reported in Section 6.3.2.

5.3 Baselines

We compare against two baselines representing the cost-quality extremes for label-free calibration. The *Verbalized* baseline uses the model's verbalized confidence c (soft readout, Eq. 4.2) directly with no adaptation, representing a zero-cost lower reference. The *P(True) Norm* baseline reports the distractor-normalized discriminative signal (Eq. 4.3) directly as the confidence estimate; it serves as an upper-bound reference for signal quality, since it requires five discriminative forward passes per question, does not modify model weights, and represents the quality of the supervision signal that SECL distills.

5.4 Metrics

We report Expected Calibration Error (ECE; (Naeini et al. 2015; Guo et al. 2017)) as our primary calibration metric, Brier score (Brier, 1950) as a composite measure of calibration and discrimination, AUROC for discrimination quality, and task accuracy to verify that calibration updates do not degrade correctness. Formal definitions are provided in Section 2.2.2 of Chapter 2.

We additionally report Adaptive ECE (AdaECE), which uses 10 equal-mass bins rather than equal-width bins, to verify that calibration improvements are robust to the choice of binning strategy. Equal-width binning can leave some bins nearly empty when confidences cluster; equal-mass binning guarantees the same number of predictions per bin, producing a complementary estimate of calibration error that does not depend on the distribution of confidences over the $[0, 1]$ range.

5.5 Implementation Details

All values below are the best configuration determined through the ablation studies in Chapter 6. Table 5.2 consolidates the full hyperparameter set in a single reference.

LoRA configuration. LoRA (rank r=8, α=16) is applied to the query and value projection matrices of the last 4–8 transformer layers, targeting intermediate-to-late layers where calibration-relevant representations concentrate (Du et al., 2024). The exact layer range is model-dependent. Phi uses a fused `qkv_proj` module rather than separate `q_proj` and `v_proj`, so the adapter is placed on the fused projection in that case. This setup modifies 328K–1,049K parameters per model, or roughly 0.01–0.02% of total parameters; the exact counts are reported in Table 5.1.

Model	Total params	LoRA layers	LoRA params	% of total
Llama 3.2-3B	3.21B	4	327,680	0.010%
Llama 3.1-8B	8.03B	8	1,048,576	0.013%
Gemma 2-2B	2.61B	8	491,520	0.019%
Phi 3.5-Mini	3.82B	8	786,432	0.021%

Table 5.1: LoRA trainable parameters per model (rank r=8). Llama 3.2-3B: last 4 layers (24–27 of 28); Llama 3.1-8B: last 8 layers; Gemma and Phi: last 8 layers (Gemma: 18–25 of 26; Phi: 24–31 of 32).

Training. We optimize with AdamW (Loshchilov and Hutter, 2019) at a learning rate of 5×10^{-5} for 3 epochs per question. The directional target uses $\alpha_{\text{step}} = 0.5$ and clip bound $\delta = 0.15$. The bin-gate threshold is set to 1 bin, so questions where verbalized and discriminative confidence already agree within one bin receive no gradient update. The entropy gate uses EMA smoothing $\alpha_{\text{ema}} = 0.05$, Page-Hinkley tolerance $\epsilon = 0.05$, detection threshold $\lambda = 3.0$, burst length $B = 50$, and a warmup of 30 questions before first detection. The normalization temperature τ is fixed per model family from preliminary exploratory runs of the P(True) baseline and held constant across datasets: τ=0.7 for Llama 3.2-3B, τ=1.5 for Gemma and Phi, and τ=3.0 for Llama 3.1-8B.

The full sweep over temperatures and the reasoning for these selections is reported in Section 6.2.1.

Hardware, software, and reproducibility. All experiments were conducted on a shared workstation equipped with NVIDIA A100 (80 GB) and RTX A6000 (48 GB) GPUs. Each 2,000-question SECL run takes approximately 2–4 GPU-hours on a single GPU, depending on model size. We use PyTorch 2.9 (Paszke et al., 2019), HuggingFace Transformers 4.57 (Wolf et al., 2020), and the PEFT library (v0.17) with CUDA 12.8. The main table reports fixed-seed runs; multi-seed robustness on Llama 3.2-3B is reported in Section 6.1.5.

Component	Parameter	Value
LoRA	Rank r	8
	Scaling factor α	16
	Target layers (Llama 3.2-3B)	Last 4
	Target layers (Llama 3.1-8B, Gemma, Phi)	Last 8
	Target modules (Llama, Gemma)	`q_proj`, `v_proj`
	Target modules (Phi)	`qkv_proj`
Optimization	Optimizer	AdamW
	Learning rate	5×10^{-5}
	Epochs per question	3
Directional loss	Step size α_{step}	0.5
	Clip bound δ	0.15
Bin-gate	Threshold	1 bin
Page-Hinkley	Tolerance ϵ	0.05
	Detection threshold λ	3.0
	EMA smoothing α_{ema}	0.05
	Warmup period	30 questions
Burst	Burst size B	50
Normalization τ	Llama 3.2-3B	**0.7**
	Gemma 2-2B	**1.5**
	Phi 3.5-Mini (3.8B)	**1.5**
	Llama 3.1-8B	**3.0**
Other	Weight accumulation	On
	Distractors k	4

Table 5.2: Full hyperparameter settings used across all experiments. Model-specific values are noted where applicable. LoRA modifies 0.01–0.02% of model parameters; the gating parameters control when and how often calibration bursts are triggered; τ values were selected per model family on preliminary P(True) Norm runs (see Section 6.2.1) and held constant across datasets.

6 Evaluation

This chapter evaluates SECL against the two research questions posed in Chapter 1. Section 6.1 presents the main results and answers both questions directly: that the generation-discrimination gap is a usable label-free signal for test-time calibration (RQ1), and that a model adapted with this signal surpasses the signal itself and approaches supervised calibration (RQ2, first part). Section 6.2 reports ablation studies that isolate which design choices are necessary for this result and confirm its robustness (RQ2, second part). Section 6.3 reports additional analyses that probe the scope and limits of the method.

6.1 Main Results

This section answers the two research questions. Section 6.1.1 establishes that SECL reduces calibration error substantially across four models using only the label-free generation-discrimination signal, answering RQ1. Section 6.1.2 establishes that the adapted model surpasses its own supervision signal and approaches supervised calibration without labels, answering the first part of RQ2. Sections 6.1.3 and 6.1.4 situate these results against the cost of the method and against the closest prior approach.

6.1.1 SECL Calibrates Without Labels (RQ1)

RQ1 asks whether the generation-discrimination gap can serve as a label-free self-supervision signal for test-time calibration. The answer is yes. Table 6.1 reports calibration metrics across the full 2,000-question stream for all four models. Compared with the verbalized baseline, SECL reduces Expected Calibration Error by 56% on Phi to 78% on Gemma, using no labeled data at any point. Adaptive ECE, which uses equal-mass rather than equal-width bins, tracks ECE closely, confirming that the improvement is not an artifact of the binning scheme.

The largest gains appear where miscalibration is most severe. On a per-domain basis (Section 6.3.1), MMLU and TruthfulQA, where the verbalized baseline is most overconfident, show the biggest reductions, while the already-better-calibrated GSM8K

Model	Method	ECE↓	AdaECE↓	Brier↓	AUROC↑	Acc
Llama 3.2-3B	Verbalized	.170	.167	.292	.510	.576
	Self-Consistency†	.093	.096	**.211**	**.728**	.624
	+ Temp Scaling*	.047	.075	.250	.504	.576
	P(True) Norm	.065	.067	.223	.694	.576
	+ Temp Scaling*	**.029**	**.030**	.218	.692	.576
	SECL (Ours)	.050	.060	.241	.587	.577
Gemma 2-2B	Verbalized	.256	.252	.314	.558	.516
	+ Temp Scaling*	.047	.037	.249	.550	.516
	P(True) Norm	.141	.142	.259	**.650**	.516
	+ Temp Scaling*	**.037**	**.033**	**.233**	.647	.516
	SECL (Ours)	.056	.060	.254	.548	.515
Phi 3.5-Mini	Verbalized	.251	.240	.275	.600	.667
	+ Temp Scaling*	**.047**	**.049**	.215	.583	.667
	P(True) Norm	.154	.151	.227	**.675**	.667
	+ Temp Scaling*	.059	.060	**.205**	.664	.667
	SECL (Ours)	.110	.119	.251	.521	.665
Llama 3.1-8B	Verbalized	.225	.225	.258	.684	.644
	+ Temp Scaling*	**.080**	.089	.213	.681	.644
	P(True) Norm	.120	.117	.211	**.718**	.646
	+ Temp Scaling*	.108	.103	**.208**	.716	.646
	SECL (Ours)	.083	**.080**	.222	.643	.646

Table 6.1: Overall results across the full 2,000-question stream. **Bold**: best overall per model. Underline: best among label-free methods. *Requires ground-truth labels (5-fold CV). †Self-Consistency requires N=10 sampling passes; reported for Llama only to contextualize SECL's single-pass efficiency.

and ARC show smaller gains. This is the expected behavior for a method that closes the gap between a noisy generative signal and a cleaner discriminative one, and it is the first piece of evidence that the gap is doing the work: the improvement is concentrated exactly where the gap is widest.

Figure 6.1 visualizes the effect for Llama. The verbalized baseline is systematically overconfident, with predictions sitting well below the diagonal in the high-confidence bins. After SECL, predictions shift onto the diagonal, and the false near-100% certainty visible in the baseline's top bin is eliminated. Reliability diagrams for the remaining models, which show the same pattern, are provided in Appendix B.1.

6.1.2 SECL Surpasses Its Signal and Approaches Supervised Calibration (RQ2)

The first part of RQ2 asks whether a model adapted with the generation-discrimination signal can surpass the signal itself and match supervised calibration. Both hold.

SECL surpasses its own supervision signal on all four models. In Table 6.1, SECL's ECE is lower than that of P(True) Norm, the very signal it distills, despite SECL training on only 6–26% of the stream (Section 6.2.1). On Llama, SECL reaches ECE 0.050 against the signal's 0.065; on Gemma, 0.056 against 0.141. The model internalizes the discriminative signal well enough to generalize beyond the specific questions it trained on, rather than merely reproducing the signal's per-question outputs.

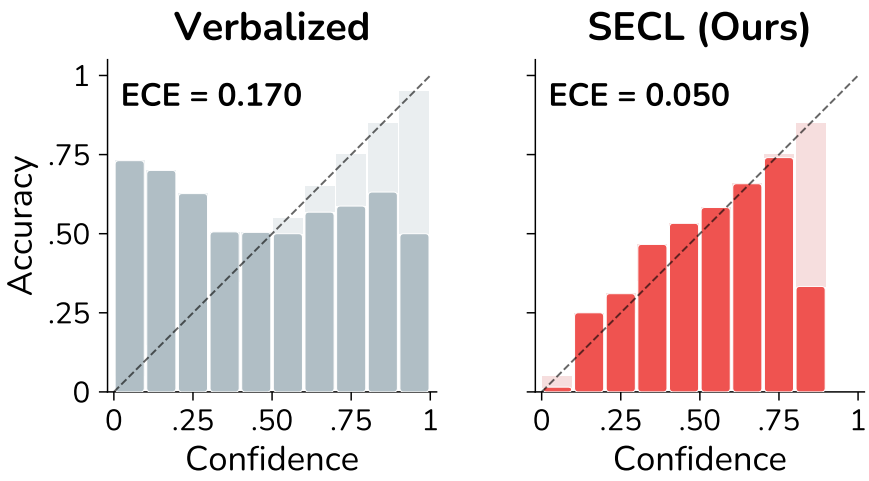

Figure 6.1: Reliability diagrams for Llama 3.2-3B. **Left**: verbalized baseline (ECE = 0.170). **Right**: after SECL (ECE = 0.050, a 71% reduction). SECL shifts predictions onto the diagonal and eliminates the baseline's false near-100% certainty.

Model	Method	ECE↓	AdaECE↓	Brier↓	AUROC↑	Conf. range
Llama 3.2-3B	Verb. + Temp	.047	.075	.250	.504	[.44, .56]
	Verb. + Platt	.021	.057	.244	.481	[.56, .61]
	P(True) + Temp	.029	.030	.218	.692	[.14, .86]
	SECL (no labels)	.050	.060	.241	.587	[.05, .85]
Gemma 2-2B	Verb. + Temp	.047	.037	.249	.550	[.41, .59]
	Verb. + Platt	.035	.050	.250	.550	[.42, .55]
	P(True) + Temp	.037	.033	.233	.647	[.26, .74]
	SECL (no labels)	.056	.060	.254	.548	[.05, .95]
Phi 3.5-Mini	Verb. + Temp	.047	.049	.215	.583	[.29, .71]
	Verb. + Platt	.052	.050	.216	.585	[.16, .69]
	P(True) + Temp	.059	.060	.205	.664	[.21, .79]
	SECL (no labels)	.110	.119	.251	.521	[.05, .95]
Llama 3.1-8B	Verb. + Temp	.080	.089	.213	.681	[.35, .74]
	Verb. + Platt	.063	.052	.206	.677	[.02, .75]
	P(True) + Temp	.108	.103	.208	.716	[.02, .99]
	SECL (no labels)	.083	.080	.222	.643	[.25, .95]

Table 6.2: Post-hoc calibration baselines (5-fold CV). Supervised methods achieve low ECE by compressing the confidence range toward the base rate; SECL preserves a wide range without labels. Full fitted temperatures and the SECL+Temp combination are reported in Appendix B.2.

SECL also approaches supervised calibration without using labels. The supervised post-hoc baselines in Table 6.2, temperature and Platt scaling fitted with ground-truth labels, achieve lower absolute ECE on some models, but they do so by compressing the confidence range. For Llama, temperature scaling requires $T = 17.6$, collapsing all predictions into $[0.44, 0.56]$, essentially predicting the base rate; Platt scaling collapses further into $[0.56, 0.61]$ and drives AUROC below chance to 0.481. SECL preserves a wide confidence range ($[0.05, 0.85]$ on Llama) while improving AUROC from 0.510 to 0.587. The supervised methods buy low ECE by destroying discriminative information; SECL achieves comparable calibration while keeping it, and without any labels.

Model	Trained	Skipped	SECL	P(True) Norm
Llama 3.2-3B	512 (25.6%)	1,488	9,168	12,000
Llama 3.1-8B	251 (12.6%)	1,749	5,514	12,000
Gemma 2-2B	119 (5.9%)	1,881	3,666	12,000
Phi 3.5-Mini	160 (8.0%)	1,840	4,240	12,000

Table 6.3: Computational cost in forward-pass equivalents over the full 2,000-question stream. *Trained* questions receive TTT adaptation (≈15 FWD-eq each); *skipped* questions are generation-only (1 FWD-eq). The P(True) Norm baseline costs 6 FWD-eq per question. SECL is cheaper than the baseline on all four models.

Taken together, these two results answer the first part of RQ2: the adapted model is not a passive imitator of its signal but exceeds it, and it reaches the territory of supervised methods while retaining the discriminative information those methods discard.

6.1.3 Cost and Accuracy

For the result to support deployment, the calibration gain must not come at the price of prohibitive cost or degraded task performance. Neither occurs.

Accuracy is preserved. Across all models, accuracy differs by less than one percentage point from the verbalized baseline (Table 6.1), and per-domain shifts are at most three points (Section 6.3.1). Unlike reinforcement-learning approaches that can degrade task performance (Damani et al., 2025; Stangel et al., 2025), SECL updates only the confidence representation, leaving the model's answers essentially unchanged.

Cost stays below the signal SECL distills. Table 6.3 reports computational cost in forward-pass equivalents over the full stream. Because entropy gating restricts adaptation to detected distribution shifts, SECL trains on only 6–26% of questions and is cheaper than running the P(True) Norm signal on every question. Gemma and Phi trigger infrequently (6–8% of the stream), reducing cost to roughly a third of the signal baseline, while even Llama, which triggers most often, stays below it.

6.1.4 Comparison with DINCO

The closest prior method is DINCO (Wang and Stengel-Eskin, 2025), which also exploits the generation-discrimination gap but applies it at inference time without adapting the model, at roughly ten forward passes per question. SECL achieves lower or equal ECE on all four models at a fraction of DINCO's cost (Figure 6.2). DINCO achieves stronger AUROC through its sampling passes and NLI reweighting, but at substantially higher cost and lower task accuracy, since its beam-search answer selection underperforms greedy decoding. On Gemma, DINCO fails outright, with ECE 0.408 worse than the unadapted baseline, while SECL's distillation still succeeds (ECE 0.056); this is direct evidence that adapting the model is more robust across architectures than applying the gap statically. The full cross-model comparison is reported in Appendix B.3.

6.1.5 Seed Robustness

To verify that the main results are not seed-dependent, we report Llama 3.2-3B results across three random seeds (42, 43, 44) on the same 2,000-question sequential stream.

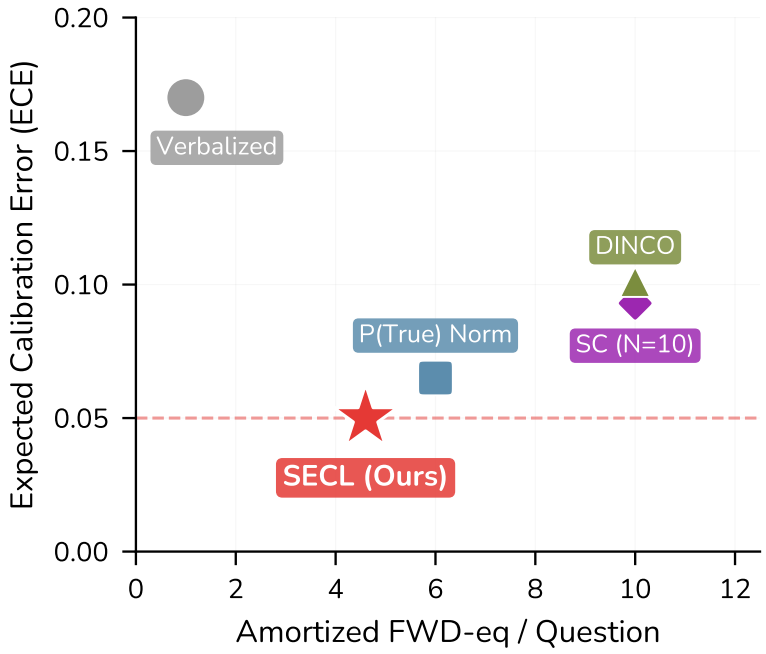

Figure 6.2: Calibration error vs. inference cost for Llama 3.2-3B (lower-left is better). SECL achieves the lowest calibration error at a fraction of the cost of P(True) Norm, DINCO, and Self-Consistency.

Metric	Seed 42	Seed 43	Seed 44	Mean ± Std
Verbalized ECE	.170	.176	.176	.174 ± .003
SECL ECE	.050	.050	.039	**.046 ± .005**
Verbalized AUROC	.510	.512	.512	.511 ± .001
SECL AUROC	.587	.601	.592	**.593 ± .006**

Table 6.4: Llama 3.2-3B multi-seed robustness (seeds 42, 43, 44). SECL remains strongly better calibrated across all seeds; ECE standard deviation of 0.005 is small relative to the 0.12 absolute improvement over the baseline.

The ECE standard deviation of 0.005 is small relative to the improvement over the verbalized baseline (roughly 0.12 absolute). The headline ECE of 0.050 reported in Table 6.1 is within one standard deviation of the three-seed mean (0.046), confirming that the main-table result is not a cherry-picked seed. AUROC improvements are similarly stable across seeds.

6.2 Ablation Studies

The main results establish that SECL works. The ablations in this section serve a different and secondary purpose: they isolate which of SECL's design choices are necessary for that result, addressing the second part of RQ2, and they confirm that the result is robust to domain ordering and hyperparameter settings. Each ablation varies one component while holding all others fixed, on Llama 3.2-3B unless noted. Three

Strategy	ECE↓	Brier↓	AUROC↑	Trained
Always-on MSE	0.047	0.240	0.593	100%
Bin-gate (≤ 1)	0.052	0.242	0.585	55.8%
Bin-gate (≤ 2)	0.044	0.242	0.578	32.6%
Entropy-gated ($B = 50$)	0.050	0.241	0.587	25.6%

Table 6.5: Gating strategy ablation (Llama 3.2-3B). Entropy-gated bursts match always-on ECE while training on a quarter of the stream.

findings emerge: the adaptation mechanism is cheap, signal quality sets the ceiling, and each component is individually necessary.

6.2.1 Each Component Is Necessary

Entropy gating. Gating determines when adaptation is triggered. Training on every question (no gating) achieves ECE 0.047 against SECL's 0.050, a negligible difference at four times the compute. Combined with the bin-gate filter, entropy gating processes only 25.6% of the stream while matching always-on calibration (Table 6.5). Gating is therefore necessary for cost, not for quality.

Weight accumulation. Calibration knowledge must compound across questions. Resetting the LoRA weights after each question yields ECE 0.237 and drives AUROC to 0.484, below chance: isolated single-question updates inject noise into the confidence token rather than information. Accumulation is necessary and load-bearing.

Directional loss. Replacing the bounded directional target with plain MSE toward $\text{NormP}_{\text{True}}$ degrades ECE from 0.052 to 0.085. Conservative clipping prevents overshooting on noisy per-question signals, confirming that the update direction matters more than its magnitude.

Distractor normalization. Raw $P(\text{True})$ already beats the verbalized baseline (ECE 0.161 vs. 0.170), confirming the gap exists. Normalization across distractors reduces this further to 0.065 by converting absolute affirmation into relative preference, suppressing the suggestibility bias of Wang and Stengel-Eskin (2025).

6.2.2 Signal Quality Sets the Ceiling

The most consequential design choice is the supervision signal itself. Replacing $\text{NormP}_{\text{True}}$ with Self-Consistency (X Wang et al., 2023) as the training target, holding everything else fixed, degrades ECE from 0.050 to 0.432, which is 2.5 times worse than the untrained baseline (Table 6.6). The adaptation mechanism faithfully distills whatever signal it receives: Self-Consistency's systematic overconfidence propagates directly into verbalized confidence, while $\text{NormP}_{\text{True}}$'s tighter correspondence to correctness yields well-calibrated outputs. The practical consequence is that SECL's calibration can never exceed that of its signal, which is why the choice of $\text{NormP}_{\text{True}}$ over alternatives is the central design decision. A detailed analysis of why Self-Consistency is a biased proxy for correctness is provided in Appendix B.9.

Training Target	ECE↓	Brier↓	AUROC↑	Acc
None (Verbalized)	.170	.292	.510	.576
Self-Consistency (N=10)	.432	.470	.443	.564
P(True) Norm (ours)	**.050**	**.241**	**.587**	.577

Table 6.6: Effect of the training target on Llama 3.2-3B. All rows use the identical SECL pipeline; only the pseudo-label differs. A poor signal makes calibration worse than no adaptation at all.

Model	Verbalized	SECL $\mathcal{O}$	SECL $\mathcal{R}$
Llama 3.2-3B	.170	**.050** (−71%)	.068 (−60%)
Llama 3.1-8B	.225	**.083** (−63%)	.189 (−16%)
Gemma 2-2B	.256	**.056** (−78%)	.149 (−42%)
Phi 3.5-Mini	.251	**.110** (−56%)	.173 (−31%)

Table 6.7: Domain order robustness (ECE). $\mathcal{O}$: default order. $\mathcal{R}$: reversed. SECL improves over the baseline under both orderings for all models.

6.2.3 Robustness to Ordering and Hyperparameters

The RQ2 result holds across the axes we tested. SECL improves over the baseline under both forward and reversed domain orderings on all four models, with reductions of 16–71% (Table 6.7). It is also insensitive to most hyperparameters: the step size α_{step} and clip bound δ move ECE by at most 0.015 across tested values, leaving burst length B as the single consequential hyperparameter (B=20 yields 0.114 versus 0.050 for B=50). The full hyperparameter sweep and the per-domain ordering breakdown are reported in Appendix B.5 and Appendix B.6.

6.2.4 Layer Selection and Regularization

Two further ablations confirm secondary design choices. On LoRA layer placement, mid and late layers achieve the same ECE (0.039) on Llama, while adapting all layers worsens ECE despite improving AUROC, indicating that adapting too many layers introduces calibration noise; we use late layers following evidence that calibration-relevant representations concentrate there (Du et al., 2024). On Gemma and Phi, halving the adapter count from eight to four layers roughly doubles ECE, so eight late layers is the default. We also tested adding a KL-divergence term to preserve the base distribution; a small weight helps marginally on Llama but degrades Gemma and Phi, so we use no KL term. Full results for both ablations are in Appendix B.7 and Appendix B.8.

6.3 Additional Analyses

The preceding sections answered the research questions. This section reports two analyses that bear directly on the scope of those answers: a per-domain breakdown that shows where the calibration gains concentrate, and a negative control that establishes the precondition under which SECL works at all. Further analyses that confirm the results extend to longer streams, open-ended generation, and an 8B model are reported in Appendix B.10.

Model	Method	GSM8K	MMLU	ARC	TQA
Llama 3.2-3B	Verbalized	.218	.106	**.095**	.372
	P(True) Norm	.133	.091	.117	.089
	SECL	**.070**	**.067**	.112	**.068**
Gemma 2-2B	Verbalized	.549	.194	**.082**	.267
	P(True) Norm	.395	.163	.185	.104
	SECL	**.356**	**.054**	.144	.210
Phi 3.5-Mini	Verbalized	.290	.264	**.109**	.343
	P(True) Norm	.261	.171	.129	.146
	SECL	.283	**.054**	.129	.229

Table 6.8: Per-domain ECE for the three smaller models. Gains concentrate on the most-miscalibrated domains (MMLU, TruthfulQA). Best per domain in **bold**. Full metrics including AdaECE, Brier, AUROC, and accuracy are in Appendix B.4.

6.3.1 Per-Domain Breakdown

The aggregate ECE reductions hide substantial per-domain variation, and the pattern of that variation is itself evidence for RQ1. Table 6.8 reports per-domain ECE for the three smaller models. SECL delivers its largest reductions on MMLU and TruthfulQA, the domains where the verbalized baseline is most miscalibrated, and smaller reductions on the better-calibrated GSM8K and ARC. ARC is the one domain where SECL slightly increases ECE under the forward ordering across all models; Section 6.2.3 and Appendix B.6 show this is a domain-sequencing artifact that reverses under the opposite ordering, not a systematic limitation. Accuracy is preserved in every domain, with the largest shift being three percentage points.

6.3.2 Negative Control: The Precondition for SECL

RQ1's answer carries an implicit scope condition: SECL works only when a generation-discrimination gap exists to exploit. We test this directly on Qwen 2.5-3B (Qwen Team et al., 2025), a model where the gap is absent. Its best P(True) Norm result (ECE 0.257) is worse than its verbalized baseline (ECE 0.247) at every normalization temperature, meaning the discriminative signal carries no calibration advantage over generation. Because SECL distills the discriminative signal into generative confidence, it cannot improve calibration when that signal is itself uninformative, and on Qwen it correctly produces no improvement. This is not a failure but a confirmation of the precondition: the cheap $\text{NormP}_{\text{True}}$ versus verbalized comparison can be run on any candidate model in advance to determine whether SECL will help. Full per-temperature results are in Appendix B.10.4.

7 Conclusion

This thesis introduced SECL, a test-time training pipeline that exploits the generation–discrimination gap as label-free self-supervision to continuously improve calibration without labeled data or human supervision. Across four small language models from three model families and four diverse domains, SECL reduces Expected Calibration Error by 56–78% while preserving task accuracy, training on only 6–26% of the question stream via entropy-gated adaptation. This chapter returns to the two research questions posed in Chapter 1, summarizes the answers this thesis provides, and discusses limitations and directions for future work.

7.1 Research Question 1

RQ1: Can the generation–discrimination gap serve as a label-free self-supervision signal for test-time calibration of small language models?

The answer is yes. The key evidence comes from three results in Chapter 6. First, Section 6.1.1 shows that a normalized $P(\text{True})$ signal drives effective weight updates on all four evaluated models, reducing ECE by 56–78% relative to the verbalized baseline. Second, Section 6.2.1 confirms that the gap itself is what produces these gains: raw $P(\text{True})$ already outperforms verbalized confidence, and distractor normalization tightens this further by converting absolute affirmation into relative preference among candidates. Third, Section 6.2.1 shows that the signal can be applied selectively through entropy-based gating, training on only 25.6% of the stream without loss of calibration quality, which addresses the cost obstacle that had prevented earlier applications of TTT to calibration.

SECL does not destabilize the base model. Accuracy differs by less than one percentage point overall (Section 6.1.3), and the directional loss with bounded updates (Section 6.2.1) prevents overshooting on noisy targets. Together, these results establish that the generation–discrimination gap is not only theoretically available as self-supervision but practically usable as a training signal at test time.

7.2 Research Question 2

RQ2: Can a model adapted with this signal surpass the signal itself and match supervised calibration, and which design choices are necessary for this to hold?

The answer is again yes, on both fronts. On the surpassing-signal claim, Section 6.1.1 shows that SECL's ECE is lower than the P(True) Norm signal on all four models, despite using that same signal as its sole supervision target. The adapted model internalizes the signal well enough to generalize beyond the specific questions on which it was trained. On the supervised-calibration claim, Section 6.1.2 shows that temperature and Platt scaling fitted with ground-truth labels achieve lower absolute ECE but collapse predictions into narrow confidence ranges, sacrificing the confidence discrimination that makes calibration practically useful. SECL preserves a wide confidence range without any labels, which is the more meaningful match.

The ablation studies in Section 6.2 isolate which components are individually necessary. Entropy gating (Section 6.2.1) is necessary for cost but not for quality, as always-on adaptation achieves similar ECE at four times the compute. The directional loss (Section 6.2.1) is necessary for quality: plain MSE against $\text{NormP}_{\text{True}}$ degrades ECE by 64%. Weight accumulation across questions (Section 6.2.1) is necessary and load-bearing: resetting LoRA weights after each question drives AUROC below chance, showing that calibration knowledge must compound across the stream to produce any benefit at all. Distractor normalization (Section 6.2.1) is necessary for signal quality: raw $P(\text{True})$ produces substantially worse calibration gains than the normalized variant. The signal itself is the ceiling: Section 6.2.2 shows that substituting Self-Consistency for $\text{NormP}_{\text{True}}$ while keeping every other component identical degrades ECE to 2.5× worse than the untrained baseline, confirming that the adaptation mechanism faithfully distills whatever signal it receives.

Robustness holds across the axes tested. Section 6.2.3 shows that forward and reversed domain orderings both produce ECE reductions of 16–71% on all four models. Section 6.2.3 shows that α_{step} and δ have minimal impact on ECE across tested values, leaving burst length B as the single consequential hyperparameter. This combination, each component individually necessary and the overall configuration robust to most hyperparameter choices, is what answers the second half of RQ2: SECL's effectiveness is not an artifact of a particular tuning choice or domain sequence.

7.3 Limitations

7.3.1 Signal Quality Bounds Improvement

SECL's calibration gains are bounded by the quality of the $\text{NormP}_{\text{True}}$ supervision signal. When the discriminative signal is only marginally better than verbalized confidence, distillation yields smaller improvements. On Gemma TruthfulQA, for example, SECL reduces ECE from 0.267 (Verbalized) to 0.210, but P(True) Norm achieves 0.104, indicating that the distillation captures only part of the available signal on adversarially constructed questions. Integrating richer discriminative signals beyond binary $P(\text{True})$, such as multi-step verification or ensemble-based judgments, could close this gap. SECL requires a measurable generation–discrimination gap, which we verified for all four evaluated model families. For models where this gap is absent, such as Qwen 2.5-3B (Qwen

Team et al., 2025) where $\text{NormP}_{\text{True}}$ underperforms the verbalized baseline at all tested temperatures (Appendix B.10.4), SECL correctly produces no improvement, since no useful signal exists to distill. Characterizing when and why this gap closes is an important direction for future work.

7.3.2 Per-Domain Calibration Is Not Uniformly Improved

ARC ECE increases slightly in the forward ordering across all three smaller models (e.g., Llama: 0.095 → 0.112), but this effect reverses under alternative orderings (Appendix B.6), indicating a domain-sequencing artifact from cumulative LoRA weight transfer rather than a systematic limitation. Aggregate ECE remains improved under every ordering tested.

7.3.3 Calibration–Discrimination Trade-off

SECL improves AUROC on Llama 3.2-3B (+0.077) but degrades it on Phi (−0.079) and on Llama 3.1-8B (−0.041; Appendix B.10.1). Since the LoRA updates target the confidence token, they can redistribute probability mass in ways that improve bin-level calibration at the cost of per-question ranking quality. Brier score, which captures both components, improves across all models, but applications requiring fine-grained discrimination should weigh this trade-off.

7.3.4 Hyperparameter Sensitivity on Burst Size

The burst size B is the most important hyperparameter: B=20 yields ECE 0.114 versus 0.050 for B=50 on Llama (Section 6.2.3), indicating that sufficient distillation per trigger is necessary. In deployment settings with very rapid distribution shifts, fewer than 50 questions per domain, the method may not accumulate enough training signal.

7.3.5 Scale

We evaluate models up to 8B parameters. While the generation–discrimination gap is theoretically expected to widen with scale (Kalai et al., 2025), we have not verified SECL's effectiveness beyond 8B, where the computational cost of LoRA updates would also increase.

7.4 Future Work

The results of this thesis suggest a broader principle: when a model's ability to evaluate exceeds its ability to generate, the gap can be distilled back into the model's outputs via self-supervised test-time adaptation. Calibration is a natural first target because the discriminative signal is scalar and cheap to compute, but the same approach applies to any task where an analogous evaluation–generation gap exists. Factual accuracy and reasoning consistency are the two most direct extensions: in both cases, the model's ability to verify a candidate output is often better than its ability to produce one, and the same gating-plus-distillation architecture should transfer with only the signal changed.

Three specific directions follow from the limitations above. First, richer discriminative signals beyond binary P(True), such as multi-step verification, chain-of-thought self-evaluation, or ensemble-based judgments across multiple probe prompts, could raise the calibration ceiling identified in Section 7.3.1. Second, scaling SECL to models beyond 8B parameters would test whether the theoretical widening of the gap with scale (Kalai et al., 2025) translates into larger practical gains, and would establish the computational regime in which test-time LoRA updates remain viable. Third, characterizing when the gap is absent, as in the Qwen 2.5-3B case, would clarify the preconditions for SECL and potentially yield a diagnostic that indicates in advance whether a given model is a viable candidate for test-time calibration.

The broader claim the thesis supports is that the generation–discrimination gap (Kalai et al., 2025) is not a deficiency to be tolerated but a resource to be exploited. SECL uses the gap for calibration specifically; the methodology is general, and the same self-supervision principle should extend to the full set of tasks where recognition precedes recall.

A Prompt Templates

This appendix documents every prompt template used in the experiments reported in Chapter 5 and Chapter 6. Prompts are reproduced verbatim from the implementation. Placeholder fields are shown in blue angle brackets. Where a model-specific chat template was applied (via HuggingFace `AutoTokenizer`), the user-facing content shown here was wrapped using each model's official chat template before tokenization.

Eight distinct LLM prompt templates are used across the codebase, plus two NLI string templates internal to the DINCO baseline. They fall into four groups. Section A.1 contains the prompts used to elicit answers and verbalized confidence. Section A.2 contains the discriminative probes used to compute the $P(\text{True})$ and $P(\text{Know})$ supervision signals. Section A.3 contains the prompts used to generate distractors and neighborhood questions. Section A.4 documents the prompts and NLI templates used in the DINCO baseline.

A.1 Question Answering and Confidence Elicitation

A.1.1 Main QA Prompt with Verbalized Confidence

This prompt (`create_qa_prompt` in `utils.py`, lines 340–362) is used to elicit both the model's answer and its verbalized confidence in a single forward pass. It is used by the verbalized baseline (Section 5.3) and by SECL during both inference and training. The confidence bin token is read from the position immediately following `Confidence: bin`, and the soft readout in Equation 4.2 is computed over the digit-token probability distribution at that position.

Prompt 1: Main QA + Confidence (`create_qa_prompt`)

```
Your task is to answer the question based on factual information in your
own knowledge.
Please adhere to the following guidelines when formulating the answer:
You must rate how confident you are that your final answer is correct
using one of these categories:
```

```
<confidence_bullets>
IMPORTANT: Keep your reasoning BRIEF (1-2 sentences maximum). Then
provide the final answer with your confidence level. The reasoning
process must be enclosed within <think> </think> tags.
Do not add anything after the final answer.

Format:
<think>Your brief reasoning here (1-2 sentences max)</think>
Answer: <your answer>
Confidence: <confidence level>

For example:
Question: What is the capital of France?
<think>Paris is the capital of France.</think>
Answer: Paris.
Confidence: bin<num_bins-1>.

Now answer the following question:
Question: <question>
```

The <confidence_bullets> field is generated dynamically by _confidence_bullets(num_bins). For num_bins = 10 (the default used throughout all experiments), the rendered output is:

Confidence Bullets (num_bins = 10, rendered)

```
  - bin0. - you are ~0-10% confident your final answer is correct
- bin1. - you are ~10-20% confident your final answer is correct
- bin2. - you are ~20-30% confident your final answer is correct
- bin3. - you are ~30-40% confident your final answer is correct
- bin4. - you are ~40-50% confident your final answer is correct
- bin5. - you are ~50-60% confident your final answer is correct
- bin6. - you are ~60-70% confident your final answer is correct
- bin7. - you are ~70-80% confident your final answer is correct
- bin8. - you are ~80-90% confident your final answer is correct
- bin9. - you are ~90-100% confident your final answer is correct
```

A.1.2 Plain QA Prompt (No Confidence)

This prompt (create_plain_qa_prompt in run_dinco.py, lines 67–69) is used by the DINCO baseline during beam-search answer generation, since DINCO computes confidence post-hoc from $P(\text{True})$ scores rather than eliciting a verbalized confidence bin.

Prompt 2: Plain QA (create_plain_qa_prompt)

```
Answer the following question concisely.

Question: <question>
Answer:
```

A.2 Discriminative Probes

A.2.1 P(True) Verification Probe

This prompt (`get_discriminative_confidence` in `run_continual_ttt.py` lines 572–580, and `get_p_true` in `run_baselines.py` lines 103–110) is the primary discriminative signal used in SECL. After the model generates an answer, the question, the model's reasoning, and the final answer are presented together and the model is asked to evaluate correctness. The probability of the `True` token is read from the next-token distribution. When a candidate list is provided, the optional `candidates_section` prefix is prepended.

Prompt 6: P(True) Verification (`get_discriminative_confidence` / `get_p_true`)

```
<candidates_section>Question: <question>
Proposed Solution:
<reasoning>
Final Answer: <answer>

Is this answer correct? Please respond with only "True" or "False".
```

When a candidate list is provided (the distractor-normalization setting used in SECL), the `<candidates_section>` field is prepended as:

Candidates Section (prepended when `candidate_list` is set)

```
Here are some brainstormed answers:
1. <candidate_1>
2. <candidate_2>
⋮
<k>. <candidate_k>
```

When no candidate list is provided (the raw P(True) baseline), `<candidates_section>` is empty and the prompt begins directly with `Question:`.

A.2.2 P(Know) Probes

Two P(Know) variants were explored (`get_discriminative_confidence` in `run_continual_ttt.py`, lines 560–569, `use_p_know=True`). These frame the discriminative judgment in terms of whether the model knows the answer, rather than whether the proposed answer is correct. Neither outperformed P(True) in preliminary experiments; they are documented here for completeness.

Thresholded variant (`claimed_bin` set). When the model has expressed a verbalized confidence bin, that percentage is included in the query:

Prompt 7: P(Know) with Claimed Confidence

```
Question: <question>
Are you at least <pct_display>% confident that you know the correct answer
to this question?
Please respond with only "True" or "False".
```

100% variant (`claimed_bin=None`). When no claimed confidence is available, the probe defaults to 100%:

Prompt 8: P(Know) at 100%

```
Question: <question>
Are you 100% confident that you know the correct answer to this question?
Please respond with only "True" or "False".
```

A.3 Distractor and Neighborhood Generation

A.3.1 Distractor Generation

This prompt (`generate_distractors` in `run_continual_ttt.py`, lines 489–498) generates $K = 4$ plausible alternative answers at sampling temperature 0.8. For multiple-choice datasets (MMLU, ARC, TruthfulQA-MC1), the provided answer options are used directly as distractors and this prompt is not invoked.

Prompt 5: Distractor Generation

```
Question: <question>
Original answer: <base_answer>
Generate <k> plausible alternative answers to this question that someone
might reasonably give.
Requirements:
- Output exactly <k> answers, numbered 1-<k>, one per line.
- Do NOT repeat or paraphrase the original answer.
- Each alternative should be plausible and different from the original.
- <format_hint>
```

The `<format_hint>` field takes one of two values:

Format Hint Variants

Numeric:
Use a single short numeric answer (one number) per line. Each answer must be a DIFFERENT number.

Non-numeric:
Keep each answer under 6 words. No explanations.

A.3.2 Neighborhood Question Rewriting

This prompt (`generate_neighborhood_questions` in `run_continual_ttt.py`, lines 393–396) was explored as an additional source of self-supervision by generating semantically neighboring questions. It was not adopted in the final SECL configuration and is documented here for completeness.

Prompt 4: Neighborhood Question Rewriting

```
Rewrite this question <n_neighbors> different ways:
"<question>"
1.
```

The model is expected to continue the numbered list. The leading `1.` is part of the prompt to encourage the list format.

A.4 DINCO Baseline Prompts

A.4.1 P(True) Probe (Yes/No Format)

The DINCO baseline (Section 6.1.4) uses a Yes/No formulation of the discriminative probe (`PTRUE_PROMPT` in `run_dinco.py`, lines 78–84), following the format of the original implementation (Wang and Stengel-Eskin, 2025). This is distinct from the True/False probe used by SECL (Section A.2.1).

Prompt 3: DINCO P(True) Probe (`PTRUE_PROMPT`)

```
Below is a question and a candidate answer. Your task is to determine
whether the answer is correct or not. Only output "Yes" (correct) or "No"
(incorrect).

Question: <question>
Candidate answer: <candidate_answer>
```

A.4.2 NLI String Templates

DINCO additionally uses an NLI classifier (`compute_pairwise_nli` and `compute_sc` in `run_dinco.py`) to reweight distractor scores. These are not LLM prompts but structured string templates passed directly to the NLI model's tokenizer:

Prompt 9: DINCO NLI Templates (not LLM prompts)

```
Premise:
Question: <question>\nAnswer: <text>

Hypothesis:
Answer: <text>
```

The NLI model outputs entailment/neutral/contradiction labels over these premise-hypothesis pairs; the scores are used to reweight candidate answers before computing DINCO's normalized confidence estimate.

A.4.3 Items Not Covered by Prompt Templates

Two processing steps operate on raw text without LLM prompt wrappers and are noted here for completeness. The entropy gating mechanism (`compute_question_entropy` in `run_continual_ttt.py`, implemented in `page_hinkley_entropy.py`) tokenizes the raw question text directly to compute entropy over the model's next-token distribution, with no surrounding prompt. Dataset loading (`load_qa_dataset` in `utils.py`) appends multiple-choice options to the question string before any prompt is constructed; this string concatenation is not itself a prompt template.

B
Extended Evaluation Results

This appendix contains supporting tables and figures referenced from Chapter 6. They provide the full detail behind the summarized results in the main text and are organized to follow the order in which they are cited.

B.1 Reliability Diagrams Across Models

Section 6.1.1 shows the verbalized-vs-SECL reliability diagram for Llama as the representative case. Figure B.1 extends this to Gemma and Phi, and Figure B.2 shows the effect of distractor normalization on the raw $P(\text{True})$ signal across the three models. The pattern is consistent: SECL shifts predictions toward the diagonal and eliminates near-100% false certainty across all architectures.

B.2 Post-Hoc Calibration: Full Results

Table B.1 reports the combination of SECL with supervised temperature scaling, referenced in Section 6.1.2. SECL and post-hoc recalibration are complementary: SECL improves the underlying confidence signal, and temperature scaling then recalibrates it. The combination achieves the best or near-best ECE on three of four models.

B.3 Full DINCO Comparison

Table B.2 gives the complete cross-model comparison with DINCO summarized in Section 6.1.4.

B.4 Per-Domain Results: Full Metrics

Table B.3 reports the complete per-domain metrics for the three smaller models, expanding the ECE-only summary in Section 6.3.1.

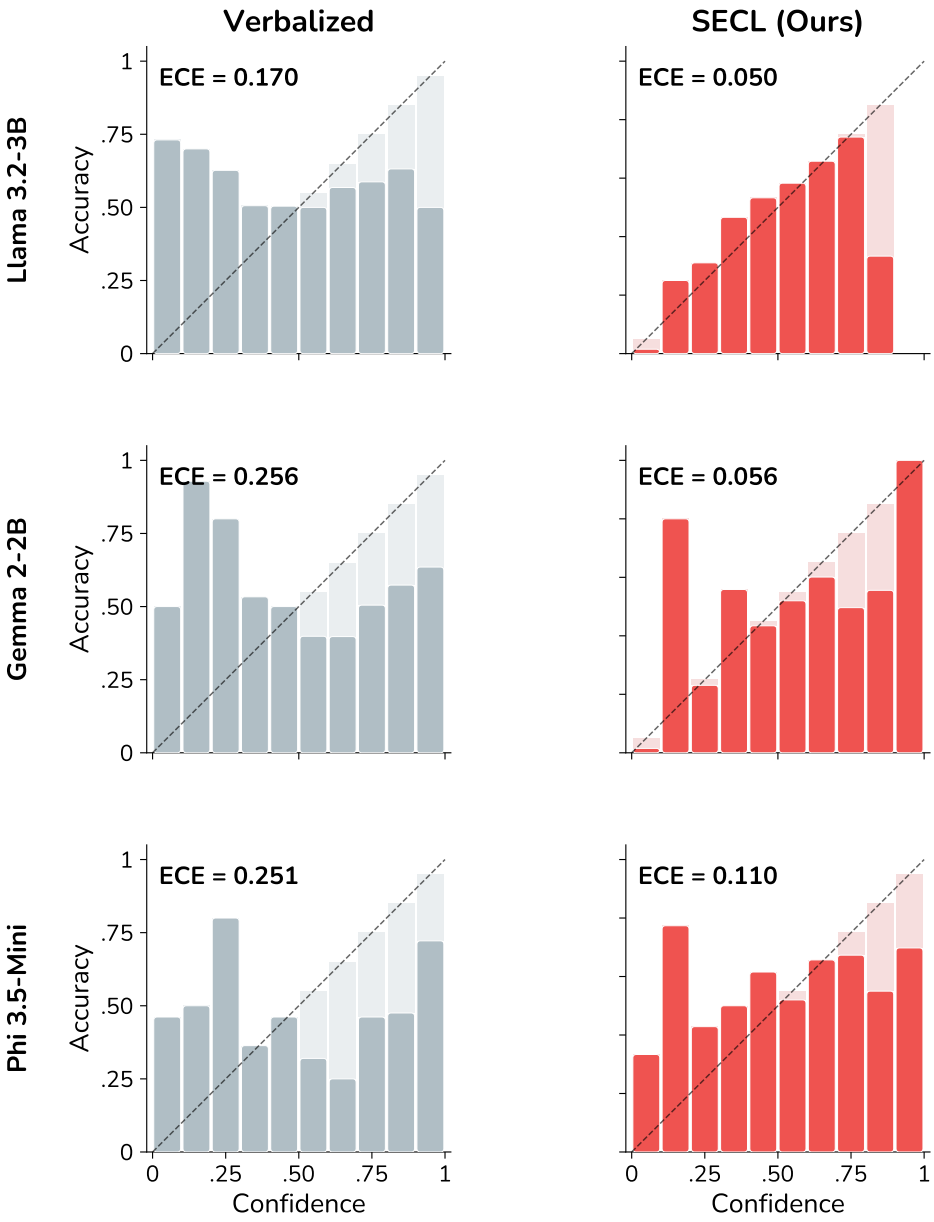

Figure B.1: Verbalized baseline (**left**) vs. SECL (**right**) for Llama 3.2-3B (ECE: 0.170 → 0.050), Gemma 2-2B (0.256 → 0.056), and Phi 3.5-Mini (0.251 → 0.110).

B.5 Hyperparameter Sensitivity

Table B.4 reports the full hyperparameter sweep on Llama 3.2-3B referenced in Section 6.2.3.

B.6 Domain Order Sensitivity: Full Breakdown

Tables B.5 and B.6 give the complete forward-vs-reversed comparison summarized in Section 6.2.3, including the per-domain ARC effect discussed there.

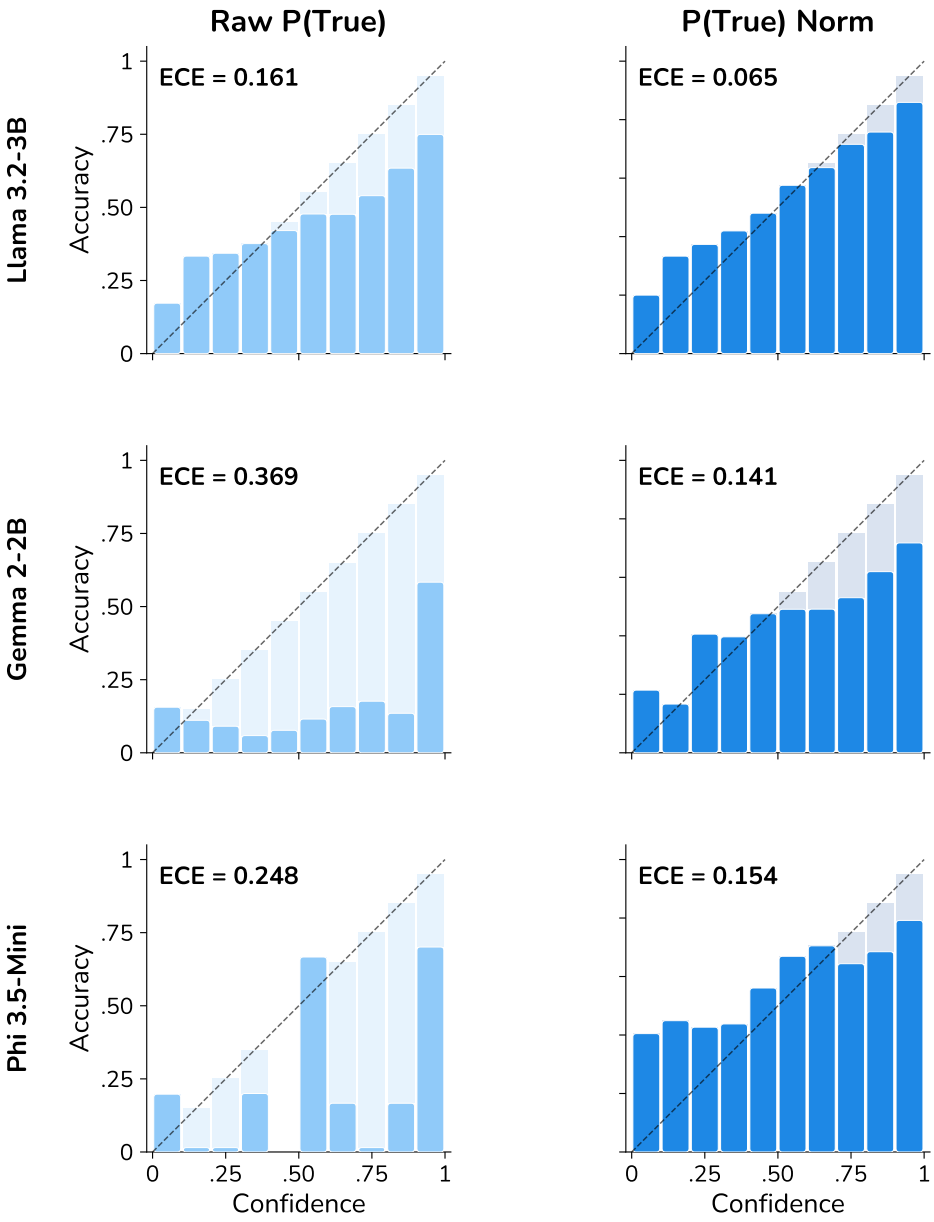

Figure B.2: Raw $P(\text{True})$ (**left**) vs. $\text{NormP}_{\text{True}}$ (**right**) for the three non-8B models. Distractor normalization consistently reduces calibration error.

B.7 LoRA Layer Position

Table B.7 reports the layer ablation summarized in Section 6.2.4.

B.8 KL Regularization

Table B.8 reports the KL-divergence regularization ablation summarized in Section 6.2.4. We tested adding a term $\beta \cdot D_{\text{KL}}(p_{\text{base}} \,\|\, p_{\text{adapted}})$ to preserve the base distribution. A small β helps marginally on Llama but degrades Gemma and Phi, so we use $\beta = 0$ throughout.

Model	Method	ECE↓	AdaECE↓	Brier↓	AUROC↑
Llama 3.2-3B	P(True) Norm + Temp*	.029	.030	.218	.692
	SECL	.050	.060	.241	.587
	SECL + Temp*	.049	.053	.241	.584
Gemma 2-2B	P(True) Norm + Temp*	.037	.033	.233	.647
	SECL	.056	.060	.254	.548
	SECL + Temp*	.011	.037	.249	.542
Phi 3.5-Mini	P(True) Norm + Temp*	.059	.060	.205	.664
	SECL	.110	.119	.251	.521
	SECL + Temp*	.097	.082	.232	.516
Llama 3.1-8B	P(True) Norm + Temp*	.108	.103	.208	.716
	SECL	.083	.080	.222	.643
	SECL + Temp*	.062	.045	.213	.638

Table B.1: SECL combined with temperature scaling (5-fold CV). *Requires ground-truth labels.

Model	Method	FWD-eq	ECE↓	Brier↓	AUROC↑	Acc
Llama 3.2-3B	Verbalized	1	.170	.292	.510	.576
	DINCO	~10	.101	**.207**	**.762**	.466
	SECL	4.6	**.050**	.241	.587	**.577**
Phi 3.5-Mini	Verbalized	1	.251	.275	.600	**.667**
	DINCO	~10	**.110**	**.212**	**.749**	.560
	SECL	2.1	**.110**	.251	.521	.665
Gemma 2-2B	Verbalized	1	.256	.314	.558	**.516**
	DINCO	~10	.408	.410	.566	.327
	SECL	1.8	**.056**	**.254**	.548	.515
Llama 3.1-8B	Verbalized	1	.225	.258	.684	.644
	DINCO	~10	.117	**.210**	**.756**	.522
	SECL	2.8	**.083**	.222	.643	**.646**

Table B.2: SECL vs. DINCO. SECL achieves better calibration at lower cost; DINCO achieves better discrimination at roughly ten forward passes per question.

B.9 Why Signal Quality Determines Calibration

Section 6.2.2 shows that substituting Self-Consistency for $\text{NormP}_{\text{True}}$ degrades calibration below the untrained baseline. This appendix explains why.

SECL minimizes $\mathcal{L} = \mathbb{E}[(c(q) - t(q))^2]$, where $c(q)$ is verbalized confidence and $t(q)$ is the training target. With sufficient capacity and convergence, $c^*(q) \to t(q)$, so the trained model inherits the calibration of its target: $\text{ECE}(c^*) \to \text{ECE}(t)$. The target bounds the achievable calibration.

Self-Consistency is a biased proxy for correctness. As $N \to \infty$, the Self-Consistency score approaches $p_{\text{gen}}(\text{mode} \mid q)$, the mass the generation distribution places on its modal answer. For modern LLMs this is typically high (above 0.7) regardless of correctness, so among questions where Self-Consistency ≈ 1, the actual fraction correct can be well

Model	Method	Domain	ECE↓	AdaECE↓	Brier↓	AUROC↑	Acc
Llama 3.2-3B	Verbalized	GSM8K	.218	.215	.294	.565	.472
		MMLU	.106	.109	.253	.601	.562
		ARC	.095	.112	.207	.568	.726
		TQA	.372	.371	.414	.301	.544
	P(True) Norm	GSM8K	.133	.138	.264	.594	.476
		MMLU	.091	.090	.239	.652	.566
		ARC	.117	.132	.210	.624	.728
		TQA	.089	.092	.180	.818	.532
	SECL	GSM8K	.070	.079	.249	.573	.488
		MMLU	.067	.060	.249	.549	.558
		ARC	.112	.106	.207	.616	.714
		TQA	.068	.114	.260	.454	.546
Gemma 2-2B	Verbalized	GSM8K	.549	.549	.446	.658	.186
		MMLU	.194	.178	.271	.608	.576
		ARC	.082	.085	.201	.534	.738
		TQA	.267	.263	.338	.417	.566
	P(True) Norm	GSM8K	.395	.394	.326	.685	.186
		MMLU	.163	.175	.263	.616	.576
		ARC	.185	.182	.229	.597	.738
		TQA	.104	.104	.213	.729	.566
	SECL	GSM8K	.356	.356	.271	.609	.180
		MMLU	.054	.064	.238	.594	.578
		ARC	.144	.136	.213	.591	.728
		TQA	.210	.198	.293	.343	.574
Phi 3.5-Mini	Verbalized	GSM8K	.290	.290	.304	.563	.650
		MMLU	.264	.257	.286	.602	.648
		ARC	.109	.105	.150	.588	.826
		TQA	.343	.336	.362	.592	.546
	P(True) Norm	GSM8K	.261	.257	.303	.559	.650
		MMLU	.171	.158	.243	.648	.648
		ARC	.129	.119	.157	.680	.826
		TQA	.146	.141	.205	.771	.546
	SECL	GSM8K	.283	.280	.297	.575	.658
		MMLU	.054	.058	.223	.604	.644
		ARC	.129	.129	.167	.499	.826
		TQA	.229	.224	.319	.407	.532

Table B.3: Complete per-domain metrics. TQA denotes TruthfulQA.

Parameter	Value	ECE↓	AdaECE↓
$\alpha_{\text{step}}{}^{\dagger}$	0.2	.067	.083
	0.3	.066	.069
	0.5	**.052**	**.073**
$\delta^{\dagger}$	**0.15**	**.052**	**.073**
	0.20	.064	.067
Loss†	Plain MSE	.085	.088
Burst $B^{\ddagger}$	20	.114	–
	50	**.050**	**.060**

Table B.4: Hyperparameter sensitivity (Llama 3.2-3B). †Without entropy gating. ‡With entropy gating.

Model	Order	ECE↓	AdaECE↓	Brier↓	AUROC↑	Acc
Llama 3.2-3B	Verbalized	.170	.167	.292	.510	.576
	SECL $\mathcal{O}$	.050	.060	.241	.587	.577
	SECL $\mathcal{R}$	.068	.082	.248	.578	.575
Gemma 2-2B	Verbalized	.256	.252	.314	.558	.516
	SECL $\mathcal{O}$	.056	.060	.254	.548	.515
	SECL $\mathcal{R}$	.149	.133	.258	.625	.520
Phi 3.5-Mini	Verbalized	.251	.240	.275	.600	.667
	SECL $\mathcal{O}$	.110	.119	.251	.521	.665
	SECL $\mathcal{R}$	.173	.170	.250	.575	.674
Llama 3.1-8B	Verbalized	.225	.225	.258	.684	.644
	SECL $\mathcal{O}$	.083	.080	.222	.643	.646
	SECL $\mathcal{R}$	.189	.191	.272	.580	.629

Table B.5: Domain order sensitivity. $\mathcal{O}$: GSM8K→MMLU→ARC→TruthfulQA. $\mathcal{R}$: reversed.

Model	Order	GSM8K	MMLU	ARC	TQA
Llama 3.2-3B	$\mathcal{O}$	.070	.067	.112	.068
	$\mathcal{R}$	.117	.032	.081	.134
Gemma 2-2B	$\mathcal{O}$	.356	.054	.144	.210
	$\mathcal{R}$	.396	.096	.046	.189
Phi 3.5-Mini	$\mathcal{O}$	.283	.054	.129	.229
	$\mathcal{R}$	.068	.213	.099	.343
Llama 3.1-8B	$\mathcal{O}$	.066	.120	.198	.038
	$\mathcal{R}$	.206	.217	.234	.223

Table B.6: Per-domain ECE for original ($\mathcal{O}$) vs. reversed ($\mathcal{R}$) order. ARC improves under reversal for the three smaller models, confirming the forward-order ARC increase is a sequencing artifact.

Model	Layers	Range	ECE↓	Brier↓	AUROC↑	Acc
Llama 3.2-3B	Early	0–3	.046	.235	.619	.591
	Mid	12–15	**.039**	.233	.628	.573
	Late (default)	24–27	**.039**	.241	.592	.570
	Last 8	20–27	.044	.240	.609	.558
	All	0–27	.058	**.232**	**.640**	**.588**
Gemma 2-2B	Last 4	22–25	.116	.277	.464	.513
	Last 8 (default)	18–25	**.056**	**.254**	**.548**	**.515**
Phi 3.5-Mini	Last 4	28–31	.222	.277	**.524**	**.669**
	Last 8 (default)	24–31	**.110**	**.251**	.521	.665

Table B.7: LoRA layer ablation. **Top**: position on Llama (4 layers each except All). **Bottom**: count on Gemma and Phi.

Model	β	ECE↓	Brier↓	AUROC↑	Acc
Llama 3.2-3B	**0**	.050	**.241**	.587	**.577**
	0.01	**.044**	.242	**.591**	.573
	0.1	.149	.279	.527	.569
Gemma 2-2B	**0**	**.056**	**.254**	.548	.515
	0.01	.127	.271	.486	.514
	0.1	.238	.305	**.551**	**.520**
Phi 3.5-Mini	**0**	**.110**	**.251**	.521	.665
	0.01	.144	.252	.506	.669
	0.1	.248	.272	**.598**	**.673**

Table B.8: KL regularization ablation. We use $\beta = 0$ in all reported experiments.

below one, violating the calibration condition. Training on this signal teaches the model to report generation concentration as confidence, producing overconfidence.

$\text{NormP}_{\text{True}}$ is less biased. When the model cannot distinguish its answer from distractors, $\text{NormP}_{\text{True}} \to 1/(K+1)$, correctly signaling low confidence; when it strongly prefers its answer, $\text{NormP}_{\text{True}} \to 1$. This maps the discriminative signal onto $[0, 1]$ in closer correspondence with correctness probability. Figures B.3 and B.4 provide visual evidence.

B.10 Further Robustness Analyses

This section reports three analyses confirming that the main results extend beyond the default experimental setting: scaling to 8B parameters, longer streams, and open-ended generation.

B.10.1 Scaling to 8B Parameters

Table B.9 reports per-domain results for Llama 3.1-8B. The aggregate result appears in the main table (Table 6.1): SECL reduces ECE by 63% while adapting only 12.6%

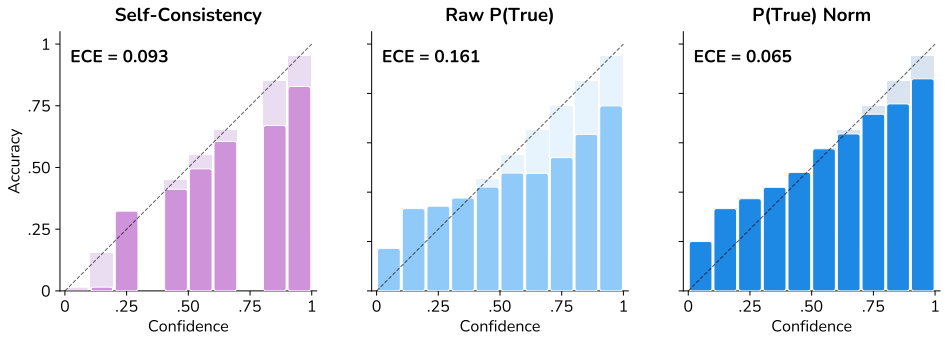

Figure B.3: Reliability diagrams for candidate training targets on Llama 3.2-3B. Self-Consistency (**left**, ECE 0.093) is biased; raw P(True) (**center**, ECE 0.161) shows suggestibility bias; P(True) Norm (**right**, ECE 0.065) tracks the diagonal most closely.

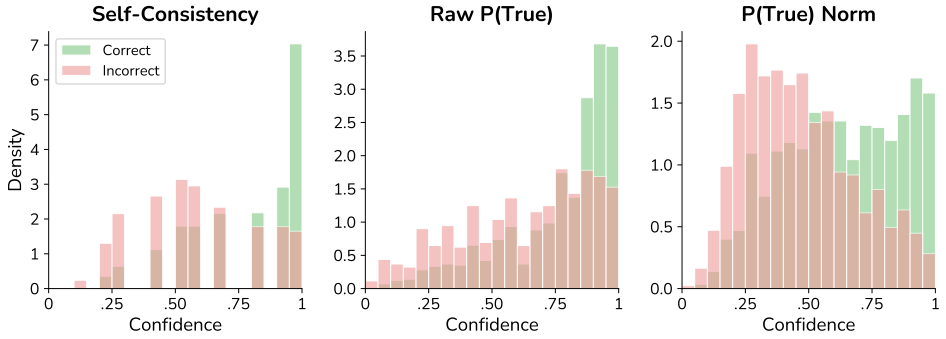

Figure B.4: Confidence distributions for correct (green) and incorrect (red) predictions. P(True) Norm separates the two classes best.

of questions. The per-domain pattern mirrors the smaller models, with the largest gains on GSM8K, MMLU, and TruthfulQA.

Method	Domain	ECE↓	Brier↓	AUROC↑	Acc
Verbalized	GSM8K	.298	.305	.714	.602
	MMLU	.208	.251	.644	.670
	ARC	.108	.164	.633	.798
	TQA	.287	.310	.669	.506
SECL	GSM8K	**.066**	.238	.602	.572
	MMLU	**.120**	.223	.591	.686
	ARC	.198	.178	.632	.824
	TQA	**.038**	.249	.552	.504

Table B.9: Per-domain results for Llama 3.1-8B.

B.10.2 Extended Stream Length

Table B.10 doubles the stream to 4,000 questions. SECL's improvements hold on Llama and Gemma. On Phi, calibration degrades because the infrequent gating provides too few training bursts over the longer stream, the same under-distillation effect seen with small burst length in Section 6.2.3.

Model	Method	ECE↓	Brier↓	AUROC↑	Acc
Llama 3.2-3B	Verbalized	.190	.313	.508	.577
	SECL	**.051**	**.239**	**.607**	**.581**
Gemma 2-2B	Verbalized	.276	.328	**.548**	.508
	SECL	**.086**	**.263**	.500	**.512**
Phi 3.5-Mini	Verbalized	**.192**	**.245**	.593	.687
	SECL	.259	.260	**.712**	**.696**

Table B.10: 4,000-question comparison. SECL improves Llama and Gemma; Phi under-distills due to infrequent gating.

B.10.3 Open-Ended Generation

Table B.11 replaces the final domain with the TruthfulQA generation split to test open-ended answers. SECL reduces ECE by 76% on the 2,000-question stream and 66% on the 4,000-question stream, confirming the method transfers beyond multiple-choice without modification.

Method	Order/Stream	ECE↓	Brier↓	AUROC↑	Acc
Verbalized	Fwd / 2k	.195	.287	.586	**.506**
P(True) Norm	Fwd / 2k	.173	.273	**.625**	**.506**
SECL	Fwd / 2k	**.047**	**.239**	.618	.500
SECL	Rev / 2k	**.047**	.241	**.625**	.497
Verbalized	Fwd / 4k	.179	.278	.598	**.525**
SECL	Fwd / 4k	**.061**	**.237**	**.625**	.522

Table B.11: TruthfulQA generation-domain evaluation (Llama 3.2-3B). SECL transfers to open-ended answers without modification.

B.10.4 Negative Control: Full Results

Table B.12 gives the full per-temperature results for the Qwen negative control summarized in Section 6.3.2.

Method	ECE↓
Verbalized	**.247**
P(True) Norm τ=0.3	.290
P(True) Norm τ=0.7	.263
P(True) Norm τ=1.0	.257
P(True) Norm τ=1.5	.265
P(True) Norm τ=2.0	.267
P(True) Norm τ=3.0	.291

Table B.12: Qwen 2.5-3B. All P(True) Norm temperatures yield worse ECE than the verbalized baseline, confirming the absence of a usable generation-discrimination gap.

References

Marah Abdin et al. 2024. Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone. arXiv: 2404.14219. (Cited on pages 5 sq., 25).

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova Dassarma, Dawn Drain, Stanislav Fort, Deep Ganguli, T. J. Henighan, Nicholas Joseph, Saurav Kadavath, John Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Chris Olah, Benjamin Mann, and Jared Kaplan. 2022. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. arXiv: 2204.05862. (Cited on pages 1, 6).

Suhana Bedi, Yutong Liu, Lucy Orr-Ewing, Dev Dash, Sanmi Koyejo, Alison Callahan, Jason A. Fries, Michael Wornow, Akshay Swaminathan, Lisa Soleymani Lehmann, Hyo Jung Hong, Mehr Kashyap, Akash R. Chaurasia, Nirav R. Shah, Karandeep Singh, Troy Tazbaz, Arnold Milstein, Michael A. Pfeffer, and Nigam H. Shah. 2025. Testing and Evaluation of Health Care Applications of Large Language Models: A Systematic Review. *JAMA* 333, no. 4 (January): 319–328. (Cited on page 1).

Glenn W. Brier. 1950. Verification of Forecasts Expressed in Terms of Probability. *Monthly Weather Review* (Boston MA, USA) 78 (1): 1–3. (Cited on pages 7, 15, 26).

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think You Have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge. arXiv: 1803.05457. (Cited on page 25).

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training Verifiers to Solve Math Word Problems. arXiv: 2110.14168. (Cited on page 24).

Mehul Damani, Isha Puri, Stewart Slocum, Idan Shenfeld, Leshem Choshen, Yoon Kim, and Jacob Andreas. 2025. Beyond Binary Rewards: Training LMs To Reason About Their Uncertainty. arXiv: 2507.16806. (Cited on pages 2, 15, 31).

Xuefeng Du, Chaowei Xiao, and Yixuan Li. 2024. HaloScope: Harnessing Unlabeled Llm Generations For Hallucination Detection. In *Advances In Neural Information Processing Systems,* 37:102948–102972. Vancouver, Canada. (Cited on pages 2, 14, 20, 26, 34).

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The Llama 3 Herd of Models. arXiv: 2407.21783. (Cited on pages 5 sq., 25).

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On Calibration of Modern Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70,* 1321–1330. ICML'17. Sydney, Australia. (Cited on pages 7, 19, 26).

Patrick Haller, Mark Ibrahim, Polina Kirichenko, Levent Sagun, and Samuel J. Bell. 2025. LLM Knowledge Is Brittle: Truthfulness Representations Rely On Superficial Resemblance. arXiv: 2510.11905. (Cited on pages 15, 19).

Moritz Hardt and Yu Sun. 2024. Test-Time Training on Nearest Neighbors for Large Language Models. In *The Twelfth International Conference on Learning Representations, ICLR 2024.* Vienna, Austria. (Cited on pages 2, 13, 15 sq.).

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring Massive Multitask Language Understanding. In *International Conference on Learning Representations, ICLR 2021.* Vienna, Austria. (Cited on page 24).

Juyeon Heo, Miao Xiong, Christina Heinze-Deml, and Jaya Narain. 2025. Do LLMs Estimate Uncertainty Well In Instruction-Following? In *The Thirteenth International Conference On Learning Representations, ICLR 2025.* Singapore, Singapore. (Cited on pages 8, 14).

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation Of Large Language Models. In *Proceedings Of The International Conference On Learning Representations, ICLR 2022.* Virtual Event. (Cited on pages 2, 11, 20).

Jinwu Hu, Zitian Zhang, Guohao Chen, Xutao Wen, Chao Shuai, Wei Luo, Bin Xiao, Yuanqing Li, and Mingkui Tan. 2025. Test-Time Learning for Large Language Models. In *Proceedings of the 42nd International Conference on Machine Learning,* 267:24823–24849. Proceedings of Machine Learning Research. Vancouver, Canada: PMLR, July. (Cited on pages 2, 13, 15 sq.).

Chengsong Huang, Langlin Huang, Jixuan Leng, Jiacheng Liu, and Jiaxin Huang. 2025. Efficient Test-Time Scaling Via Self-Calibration. In *Proceedings of the NeurIPS 2025 Workshop on Efficient Reasoning (ER).* San Diego, USA. (Cited on page 16).

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How Can We Know When Language Models Know? On The Calibration Of Language Models For Question Answering. *Transactions of the Association for Computational Linguistics* (Cambridge, MA) 9:962–977. (Cited on pages 1, 8).

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language Models (Mostly) Know What They Know. arXiv: 2207.05221. (Cited on pages 1, 3 sq., 9, 15, 19).

Adam Tauman Kalai, Ofir Nachum, Santosh S. Vempala, and Edwin Zhang. 2025. Why Language Models Hallucinate. arXiv: 2509.04664. (Cited on pages 1, 4, 10, 15, 19, 38 sq.).

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.* (Cited on page 4).

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic Uncertainty: Linguistic Invariances for Uncertainty Estimation in Natural Language Generation. In *Proceedings of the International Conference on Learning Representations, ICLR 2023.* Kigali, Rwanda. (Cited on pages 1, 9, 14).

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022a. Teaching Models to Express Their Uncertainty in Words. *Transactions on Machine Learning Research,* (cited on pages 2, 15, 19).

———. 2022b. TruthfulQA: Measuring How Models Mimic Human Falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022,* 3214–3252. Dublin, Ireland: Association for Computational Linguistics, May. (Cited on pages 14, 25).

Fenglin Liu, Hongjian Zhou, Boyang Gu, Xinyu Zou, Jinfa Huang, Jinge Wu, Yiru Li, Sam S. Chen, Yining Hua, Peilin Zhou, Junling Liu, Chengfeng Mao, Chenyu You, Xian Wu, Yefeng Zheng, Lei Clifton, Zheng Li, Jiebo Luo, and David A. Clifton. 2025. Application Of Large Language Models In Medicine. *Nature Reviews Bioengineering* 3, no. 6 (June): 445–464. (Cited on page 1).

Xin Liu, Muhammad Khalifa, and Lu Wang. 2024. LitCab: Lightweight Language Model Calibration Over Short- And Long-Form Responses. In *The Twelfth International Conference On Learning Representations, ICLR 2024,* 13033–13047. Vienna, Austria. (Cited on pages 2, 14).

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *Proceedings of the International Conference on Learning Representations, ICLR 2019.* New Orleans, USA. (Cited on pages 4, 21, 26).

Huan Ma, Jiadong Pan, Jing Liu, Yan Chen, Joey Tianyi Zhou, Guangyu Wang, Qinghua Hu, Hua Wu, Changqing Zhang, and Haifeng Wang. 2025. Semantic Energy: Detecting LLM Hallucination Beyond Entropy. arXiv: 2508.14496. (Cited on page 14).

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing,* 9004–9017. Singapore, Singapore: Association for Computational Linguistics, December. (Cited on pages 1, 14).

Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. 2015. Obtaining Well Calibrated Probabilities Using Bayesian Binning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence,* 2901–2907. AAAI'15. Austin, Texas: AAAI Press. (Cited on pages 7, 19, 26).

Jeremy Nixon, Michael W. Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. 2019. Measuring Calibration in Deep Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.* (Cited on page 7).

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems.* (Cited on page 6).

E. S. Page. 1954. Continuous Inspection Schemes. *Biometrika* 41 (1-2): 100–115. (Cited on page 18).

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems,* vol. 32. Curran Associates, Inc. (Cited on page 27).

Qwen Team, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 Technical Report. arXiv: 2412.15115. (Cited on pages 5, 25, 35, 37).

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: your language model is secretly a reward model. In *Proceedings of the 37th International Conference on Neural Information Processing Systems.* NIPS '23. New Orleans, LA, USA: Curran Associates Inc. (Cited on page 6).

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. arXiv: 1707.06347 [cs.LG]. (Cited on page 6).

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* 1715–1725. Berlin, Germany: Association for Computational Linguistics, August. (Cited on page 5).

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R. Bowman, Esin DURMUS, Zac Hatfield-Dodds, Scott R Johnston, Shauna M Kravec, Timothy Maxwell, Sam McCandlish, Kamal Ndousse, Oliver Rausch, Nicholas Schiefer, Da Yan, Miranda Zhang, and Ethan Perez. 2024. Towards Understanding Sycophancy in Language Models. In *The Twelfth International Conference on Learning Representations, ICLR 2024.* Vienna, Austria. (Cited on pages 1, 6, 8, 15).

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2025. Scaling LLM Test-Time Compute Optimally Can Be More Effective Than Scaling Model Parameters. In *Proceedings of the International Conference on Learning Representations, ICLR 2025.* Singapore, Singapore. (Cited on page 15).

Paul Stangel, David Bani-Harouni, Chantal Pellegrini, Ege Özsoy, Kamilia Zaripova, Matthias Keicher, and Nassir Navab. 2025. Rewarding Doubt: A Reinforcement Learning Approach to Calibrated Confidence Expression of Large Language Models. arXiv: 2503.02623. (Cited on pages 2, 15, 31).

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, Tatsunori Hashimoto, and Carlos Guestrin. 2025. Learning To (Learn At Test Time): RNNs With Expressive Hidden States. In *Proceedings of the 42nd International Conference on Machine Learning,* 267:57503–57522. Proceedings of Machine Learning Research. Vancouver, Canada: PMLR, July. (Cited on pages 2, 13, 15 sq.).

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. 2020. Test-Time Training with Self-Supervision for Generalization Under Distribution Shifts. In *Proceedings of the 37th International Conference on Machine Learning,* 119:9229–9248. Proceedings of Machine Learning Research. Online: PMLR, July. (Cited on pages 2, 12, 15).

Gemma Team et al. 2024. Gemma 2: Improving Open Language Models At A Practical Size. arXiv: 2408.00118. (Cited on pages 5 sq., 25).

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. 2023. Just Ask For Calibration: Strategies For Eliciting Calibrated Confidence Scores From Language Models Fine-Tuned With Human Feedback. In *Proceedings Of The 2023 Conference On Empirical Methods In Natural Language Processing,* 5433–5442. Singapore, Singapore: Association for Computational Linguistics, December. (Cited on pages 1, 3, 6, 8 sq., 15, 19).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems,* 6000–6010. NIPS'17. Long Beach, California, USA: Curran Associates Inc. (Cited on page 5).

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. 2021. Tent: Fully Test-Time Adaptation by Entropy Minimization. In *International Conference on Learning Representations, ICLR 2021.* Vienna, Austria. (Cited on pages 2, 12, 15).

Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. 2022. Continual Test-Time Domain Adaptation. In *2022 IEEE/CVF Conference On Computer Vision and Pattern Recognition (CVPR),* 7191–7201. New Orleans, USA. (Cited on pages 2, 13, 15).

Victor Wang and Elias Stengel-Eskin. 2025. Calibrating Verbalized Confidence With Self-Generated Distractors. arXiv: 2509.25532. (Cited on pages 10, 15 sq., 19 sq., 22, 31, 33, 44).

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *Proceedings of the International Conference on Learning Representations, ICLR 2023.* Kigali, Rwanda. (Cited on page 33).

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned Language Models are Zero-Shot Learners. In *International Conference on Learning Representations.* (Cited on page 6).

Zhepei Wei, Xiao Yang, Kai Sun, Jiaqi Wang, Rulin Shao, Sean Chen, Mohammad Kachuee, Teja Gollapudi, Tony Liao, Nicolas Scheffer, Rakesh Wanga, Anuj Kumar, Yu Meng, Wen-tau Yih, and Xin Luna Dong. 2025. TruthRL: Incentivizing Truthful Llms Via Reinforcement Learning. arXiv: 2509.25760. (Cited on page 15).

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations,* edited by Qun Liu and David Schlangen, 38–45. Online: Association for Computational Linguistics, October. (Cited on page 27).

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2024. Can Llms Express Their Uncertainty? An Empirical Evaluation Of Confidence Elicitation In LLMs. In *The Twelfth International Conference On Learning Representations, ICLR 2024.* Vienna, Austria. (Cited on pages 1, 8, 14).

Xinran Zhao, Hongming Zhang, Xiaoman Pan, Wenlin Yao, Dong Yu, Tongshuang Wu, and Jianshu Chen. 2024. Fact-And-Reflection (FaR) Improves Confidence Calibration Of Large Language Models. In *Findings of the Association for Computational Linguistics: ACL 2024,* 8702–8718. Bangkok, Thailand: Association for Computational Linguistics, August. (Cited on page 15).

Adam Zweiger, Jyo Pari, Han Guo, Yoon Kim, and Pulkit Agrawal. 2025. Self-Adapting Language Models. In *Proceedings of the Neural Information Processing Systems (NeurIPS).* San Diego, USA. (Cited on pages 2, 13, 15 sq.).