SiDyP: Simplex Diffusion with Dynamic Prior for Denoising Llama-Generated Labels

Anonymous authors
Paper under double-blind review

Abstract

The traditional process of creating labeled datasets is not only labor-intensive but also expensive. Recent breakthroughs in open-source large language models (LLMs), such as Llama-3, have opened a new avenue in generating labeled datasets automatically for various natural language processing (NLP) tasks to provide an alternative to such expensive annotation process. However, the reliability of such auto-generated labels remains a significant concern due to inherent inaccuracies. When learning from such noisy labels, the model's generalization is likely to be harmed as it is prone to overfit those label noises. In this paper, we propose the **Si**mplex Diffusion with a **Dy**namic **P**rior (**SiDyP**) model to calibrate classifier's predication, thus enhancing its robustness towards noisy labels. Our framework leverages simplex diffusion model to iteratively correct noisy labels conditioned on training dynamic trajectories obtained from classifier finetuning. The **P**rior in SiDyP refers to the potential true label candidates which was obtained according to neighborhood label distribution in text embedding space. It is **Dy**namic because we progressively distill these candidates based on the feedback of the diffusion model. Our SiDyP model can increase the performance of the BERT classifier fine-tuned on both zero-shot and few-shot Llama-3 generated noisy label datasets by an average of 5.33% and 7.69% respectively. Our extensive experiments, which explore different LLMs, diverse noise types (real-world and synthetic), ablation studies, and multiple baselines, demonstrate the effectiveness of SiDyP across a range of NLP tasks. We will make code and data publicly (under a CC BY 4.0 license) available on GitHub upon publication of the work.

1 Introduction

In the realm of machine learning, the effectiveness of Deep Neural Networks (DNNs) in a variety of applications is largely contingent on the availability of well-annotated datasets (Fisher, 1936; Deng et al., 2009; Touvron et al., 2023a). Traditionally, this annotation process has been carried out manually by subject matter experts (Ratner et al., 2017), ensuring high accuracy but at a substantial cost in terms of time and resources. In response to these constraints, the field has gradually pivoted towards alternative strategies such as active learning (Ren et al., 2021; Kartchner et al., 2020; Yu et al., 2022), transfer learning (Pan & Yang, 2009; Howard & Ruder, 2018), and weak supervision (Stephan et al., 2022; Yu et al., 2020; Lison et al., 2021). These methods help alleviate some of the burdens of manual annotation, yet they often introduce a new challenge: the incorporation of noise in the training data.

The susceptibility of DNNs, especially pre-trained language models to the noise inherent in training data is a formidable challenge, particularly for models like BERT (Devlin et al., 2019b), which can inadvertently fit to inaccuracies. This issue is compounded by weak supervision types—described by Zhou (2018) as incomplete, inexact, and inaccurate supervision—that introduce various forms of label noise. Without appropriate denoising, these models risk learning from erroneous data rather than genuine patterns. Robust denoising strategies, therefore, play a crucial role in refining training datasets. By systematically identifying and amplifying the impact of mislabeled data, these strategies ensure that models are trained on more accurate representations of the data, as demonstrated by efforts in advanced denoising techniques (Ratner et al., 2017; Yu et al., 2020; Zhang et al., 2022; Zhuang et al., 2023).

Transitioning to the era of advanced open-source language models like Llama-3 (Dubey et al., 2024), the capabilities for initial data annotation have seen remarkable improvements (Tan et al., 2024; Yu et al., 2023; Brown et al., 2020). LLMs can generate initial labels for datasets, leveraging its extensive training on diverse textual data. Although numerous methods have been proposed to enhance the capabilities of LLMs, aiming to improve the accuracy and reliability of their annotation (Yu et al., 2023; Yu & Bach, 2023; Wang et al., 2023; Oliveira et al., 2024; Li et al., 2024; Burns et al., 2023), complete immunity to inaccuracies in LLM-generated labels is unattainable, necessitating a robust mechanism to mitigate the harmful impact of their noisy labels. However, LLM-generated label noise is under exploration as previous studies mainly focus on either synthetic noise or real-world noise (Han et al., 2018b; Bae et al., 2022; Zhuang et al., 2023; Wei et al., 2020; Chen et al., 2023a). Synthetic noise is often impractical since it fails to reflect real-world scenarios, where no gold-standard dataset exists for injection. On the other hand, real-world noise is costly to obtain, as it requires subject matter experts (Ratner et al., 2017) to create labeling functions. To bridging this gap, we propose an innovative denoising approach that strengthens classifiers' resilience to LLM-generated noisy labels.

Our approach aims to purify noisy labels via transition matrix-based methods (Patrini et al., 2017; Yao et al., 2021; Zhang et al., 2021b; Xia et al., 2020; Berthon et al., 2021). Adopting the framework from Bae et al. (2022), our denoising method consists of two stages: finetuning pre-trained language classifiers (PLCs) and denoising via generative models. Finetuning a PLC on a noisy dataset yields data's embedding dynamic trajectories (Zhuang et al., 2023) and prior probability $p(\tilde{y}|x)$. By referring to the neighbor's label distribution in embedding space, we are able to collect a list of potential true label candidates and their corresponding weights. We design a simplex diffusion (Mahabadi et al., 2024) label model to reconstruct true labels from noisy labels and training dynamics. The potential true label candidates are refined progressively throughout the training of the diffusion model based on its prediction. The overall framework is presented in Figure 1.

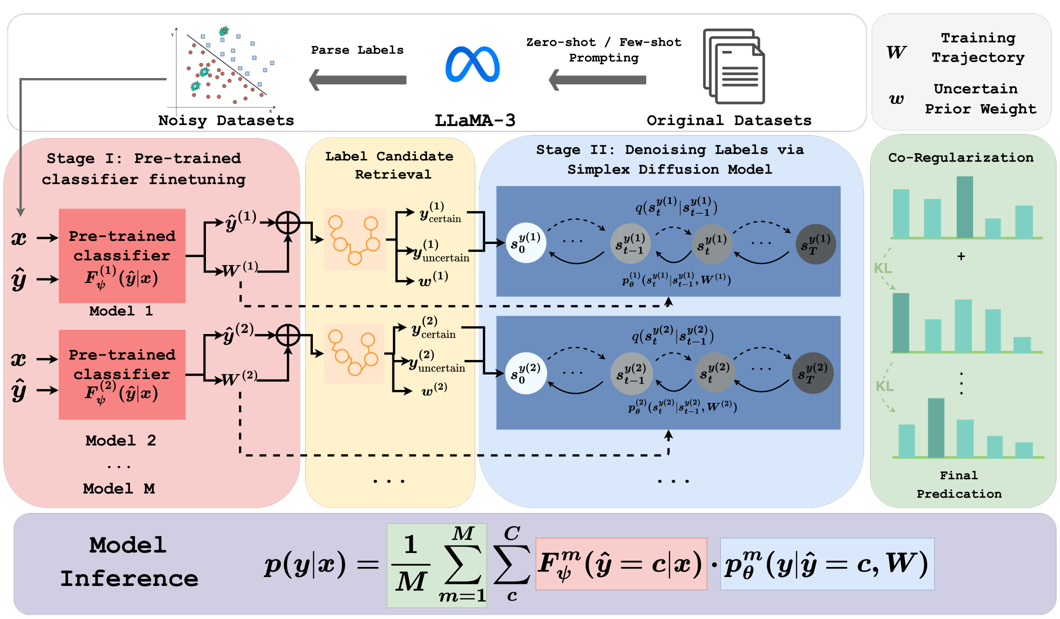

Figure 1: The SiDyP framework, containing (1) pre-trained classifier fine-tuning; (2) dynamic label candidates retrieval and distillation; (3) denoising label using simplex diffusion; (4) co-regularization between multiple model branches; (5) inference process to predict refined labels from noisy labels.

The main contribution of our work include:

- We evaluate previous state-of-the-art baselines, validated on both synthetic and real-world noise, under a novel type of noise: LLM-generated label noise. To the best of our knowledge, this is the first study aimed at enhancing learning under LLM-generated label noise.

- We propose SiDyP, a robust framework using dynamic priors to derive reliable true labels and the simplex denoising label diffusion model to calibrate classifier's predication.
- We conduct extensive experiments of our frameworks compared to 5 state-of-the-art baselines across 4 NLP tasks, 5 LLMs, and 3 different type of noises. Our approach outperforms all the baselines in all the experiments. The effectiveness of each component is also verified.

2 Background and Motivation

Problem Definition Let $\mathcal{X} \in \mathbb{R}^d$ and $\mathcal{Y} = \{0, 1, ..., c\}$ be the d-dimension input and the target label in a classification task with c classes. Following the joint probability distribution P over $\mathcal{X} \times \mathcal{Y}$, the i.i.d samples forms a gold classification dataset, $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$. Our assumption of learning from noisy labels indicates that the only accessible dataset is $\tilde{\mathcal{D}}_{\text{train}} = \{x_i, \tilde{y}_i\}_{i=1}^N$, sampling from $\tilde{P}$ over $\mathcal{X} \times \tilde{\mathcal{Y}}$ where $\tilde{\mathcal{Y}}$ are potential noisy targets. For a traditional classification problem, the training objective of a classifier f_θ is to minimize the true risk $R_L(f_\theta) := \mathbb{E}_P[L(f_\theta(x), y)]$. However, in the realm of learning from noisy labels, the only accessible risk function is the noisy empirical risk $\tilde{R}_L^{\text{emp}}(f_\theta) := \mathbb{E}_P[L(f_\theta(x), \tilde{y})]$ due to the absence of true labels y. Therefore, our goal is to find a function minimizing the true risk $R_L(f_\theta)$ during learning with noisy empirical risk $\tilde{R}_L^{\text{emp}}(f_\theta)$.

With the only observable target labels being noisy, we manage to train a model that generates probability distribution of true label y given arbitrary input x, $p(y|x)$. Taking advantage of noisy labels in our training dataset, we can decompose our objective further as:

$$p(y|x) = \sum_{\tilde{y}} p(\tilde{y}|x)p(y|\tilde{y}, x)$$

In this revised objective, the prior $p(\tilde{y}|x)$ can be directly estimated by finetuning a PLC $\boldsymbol{F}_\psi$ on the accessible noisy dataset. We can approximate the posterior $p(y|\tilde{y}, x)$, expressing the probability distribution of true label y given noisy label $\tilde{y}$ and input x, by a generative model. Unlike synthetic noise, which has been extensively studied, LLM-generated label noise is more intricate, contextually influenced, and reflective of real-world class relationships (we include a more detailed discussion in Appendix G). This triggers a more challenging estimation of the posterior as the relation between $\tilde{y}$ and y becomes less predictable and more context-dependent. To tackle this, we begin by focusing on these two key aspects:

1. How can a promising and reliable true label be derived from the noisy dataset?
2. How can we estimate such probabilistic relation between true labels, corrupted labels, and input features accurately?

We define corrputed labels as one which is mislabeled thus incorrect. In the following sections, we introduce our true label candidates dynamic distillation (Section 3) and simplex denoising label diffusion model (Section 4) to address these two concerns respectively. We also adopt training dynamics during PLC fine-tuning and co-regularization mechanism (Appendix C) to make SiDyP tolerant to noises.

3 True Label Candidates Dynamic Distillation

Extracting true labels from a noisy dataset is crucial, as it directly impacts the quality of the subsequent generative posterior approximation. Our derivation of true label is based on the assumption that textual embeddings are robust enough to discriminate between clean and corrupted data samples(Ortego et al., 2021). Texts belonging to the same class typically exhibit similar semantics, making them more likely to cluster together in the embedding space. Therefore, the neighboring labels reveal information about the true labels. Different from prior works(Zhuang et al., 2023; Bae et al., 2022), we retrieve a list of true label candidates for each individual data sample (Algorithm 1). These true label candidates are distilled according to our diffusion model's feedback during training (Algorithm 2).

3.1 Label Candidate Retrieval

Our main purpose is re-assigning labels to noisy samples leveraging true label information in embedding space. First, we need to discriminate noisy samples in the dataset. During the PLC fine-tuning in Stage I, there exist training dynamics in embedding space. The noisy samples tend to exhibit larger mean and standard deviation of Euclidean distances towards their assigned labels (incorrect) compared to clean samples (Zhuang et al., 2023). We split the original dataset into $D_{\text{train}}^{\text{noisy}}$ and $D_{\text{train}}^{\text{clean}}$ by cutting off the top σ percent of training trajectories, where σ is the estimated error rate. We apply K Nearest Neighbor (KNN) algorithm on $D_{\text{train}}^{\text{noisy}}$ with $D_{\text{train}}^{\text{clean}}$ as the reference. Instead of assigning a single deterministic label, a list of label candidates and its corresponding weights (probability) are generated by KNN classifier. We manage to alleviate the uncertainty injected into training of diffusion model in Stage II by two filters: (1) we preserve the candidate if its associated probability greater than a threshold λ. These data instances are regarded as deterministic instance since their potential true label is single and certain. The remaining data instances are regarded as uncertain and linked with a list of candidates. (2) For uncertain data instances, we extract the two candidates with highest probabilities. If their summation is greater than a specified threshold γ, we then eliminate other candidates and only preserve these two dominant candidates.

Algorithm 1: Potential True Label Candidates Retrieval

Input: $\mathcal{D}_{\text{train}}^{\text{noisy}}$: $\{\mathbf{x_i}, \tilde{\mathbf{y}}_\mathbf{i}\}_\mathbf{i}^\mathbf{n}$, $\mathcal{M}_{\text{train}}$, $\mathcal{C}_{\text{knn}}$, K, λ, γ

Output: $\mathcal{D}_{\text{train}}^{\text{certain}}$: $\{\mathbf{x_i}, \mathbf{y_i}\}_\mathbf{i}^\mathbf{m}$, $\mathcal{D}_{\text{train}}^{\text{uncertain}}$: $\{\mathbf{x_i}, (\mathbf{y_i^0}, \mathbf{y_i^1}, \dots)\}_\mathbf{i}^{\mathbf{n-m}}$, $\mathcal{W}_{\text{train}}^{\text{uncertain}}$: $\{(\mathbf{w_i^0}, \mathbf{w_i^1}, \dots)\}_\mathbf{i}^{\mathbf{n-m}}$

```
1  Split D_train^noisy into {D̄_train^clean, D̄_train^noisy} according to noisy marker M_train
2  Fit D̄_train^clean into KNN classifier C_knn
3  Predict P_train : {(p_i^0, p_i^1, ...)}_i^n of entire dataset D_train^noisy using C_knn based on K neighbors
4  Initialize D_train^certain = {}, D_train^uncertain = {} and W_train^uncertain = {}
5  for i = 0 to n do
6      p_i^max = max{(p_i^0, p_i^1, ...)}
7      if p_i^max ≥ λ then
8          Insert (x_i, y_i^max) into D_train^certain
9      else
10         p_i^max1, p_i^max2 = top2{(p_i^0, p_i^1, ...)}
11         if p_i^max1 + p_i^max2 ≥ γ then
12             Insert (x_i, {y_i^max1, y_i^max2}) into D_train^uncertain
13             p_i^max1, p_i^max2 = softmax(p_i^max1, p_i^max2)
14             Insert (p_i^max1, p_i^max2) into W_train^uncertain
15         else
16             Insert (x_i, {y_i^0, y_i^1, ...}) into D_train^uncertain
17             Insert (p_i^0, p_i^1, ...) into W_train^uncertain
```

3.2 Candidate Dynamic Distillation

Our true label candidates distillation is established based on the observation that the generative model gains the capability to calibrate certain amount of noisy data instances after training on our derived deterministic (certain) dataset. Adhere to the observation, we first train our generative model only on deterministic dataset for α warm-up epochs. We rely on such capable model to evaluate our uncertain dataset over a specified iteration β. During each evaluation, if model's predicted label lies in the candidate lists, the matched label candidate will increase accordingly. The weight list will then be normalized as well to maintain a summation to 1. After candidate weight update and model evaluation for uncertain data samples, we sample a specific label candidate from the candidate list multinomially based on the candidate weights. We treat such a sample label as the true label in this training epoch. The generative model is then trained on both deterministic pair and uncertain pair. Subsequently, the loss of generative model for uncertain sample is weighted by the sampled candidate's weight.

Algorithm 2: Distill True Label from Candidates during Training

Input: $\mathcal{G}_{\text{model}}$, $\mathcal{D}^{\text{certain}}_{\text{train}}$: $\{\mathbf{x_i}, \mathbf{y_i}\}^{\mathbf{m}}_{\mathbf{i}}$, $\mathcal{D}^{\text{uncertain}}_{\text{train}}$: $\{\mathbf{x_i}, (\mathbf{y^0_i}, \mathbf{y^1_i}, \dots)\}^{\mathbf{n-m}}_{\mathbf{i}}$, $\mathcal{W}^{\text{uncertain}}_{\text{train}}$: $\{(\mathbf{w^0_i}, \mathbf{w^1_i}, \dots)\}^{\mathbf{n-m}}_{\mathbf{i}}$, α, E, β

Output: $\mathcal{G}_{\text{model}}$

for $e = 0$ *to* E **do**
- **if** $e \leq \alpha$ **then**
  - $\{\bar{\mathbf{y}}_\mathbf{i}\}^{\mathbf{m}}_{\mathbf{i}}$ = $\mathcal{G}_{\text{model}}[\{\mathbf{x_i}\}^{\mathbf{m}}_{\mathbf{i}}]$ for $\mathcal{D}^{\text{certain}}_{\text{train}}$
  - loss = $\mathcal{F}_{\text{loss}}[\{\bar{\mathbf{y}}_\mathbf{i}\}^{\mathbf{m}}_{\mathbf{i}}, \{\mathbf{y_i}\}^{\mathbf{m}}_{\mathbf{i}}]$
  - Optimize $\mathcal{G}_{\text{model}}$
- **else**
  - **for** $i = 0$ *to* β **do**
    - $\{\bar{\mathbf{y}}_\mathbf{i}\}^{\mathbf{n-m}}_{\mathbf{i}}$ = $\mathcal{G}_{\text{model}}[\{\mathbf{x_i}\}^{\mathbf{n-m}}_{\mathbf{i}}]$ for $\mathcal{D}^{\text{uncertain}}_{\text{train}}$
    - **if** $\{\bar{\mathbf{y}}_\mathbf{i}\}^{\mathbf{n-m}}_{\mathbf{i}}$ *in* $(\mathbf{y^0_i}, \mathbf{y^1_i}, \dots)$ **then**
      - Increase corresponding $\mathbf{w^*_i}$ by $\frac{1-\mathbf{w^*_i}}{\beta}$
      - $(\mathbf{w^0_i}, \mathbf{w^1_i}, \dots)$ = softmax$[(\mathbf{w^0_i}, \mathbf{w^1_i}, \dots)]$
  - $\{\mathbf{y_i}\}^{\mathbf{n-m}}_{\mathbf{i}}$= sample $(\mathbf{y^0_i}, \mathbf{y^1_i}, \dots)$ multinomially according to $\mathcal{W}^{\text{uncertain}}_{\text{train}}$
  - $\{\bar{\mathbf{y}}_\mathbf{i}\}^{\mathbf{n-m}}_{\mathbf{i}}$ = $\mathcal{G}_{\text{model}}[\{\mathbf{x_i}\}^{\mathbf{n-m}}_{\mathbf{i}}]$ for $\mathcal{D}^{\text{uncertain}}_{\text{train}}$
  - $\{\bar{\mathbf{y}}_\mathbf{i}\}^{\mathbf{m}}_{\mathbf{i}}$ = $\mathcal{G}_{\text{model}}[\{\mathbf{x_i}\}^{\mathbf{m}}_{\mathbf{i}}]$ for $\mathcal{D}^{\text{certain}}_{\text{train}}$
  - certain_loss = $\mathcal{F}_{\text{loss}}[\{\bar{\mathbf{y}}_\mathbf{i}\}^{\mathbf{m}}_{\mathbf{i}}, \{\mathbf{y_i}\}^{\mathbf{m}}_{\mathbf{i}}]$
  - uncertain_loss = $\{\bar{\mathbf{w}}_\mathbf{i}\}^{\mathbf{n-m}}_{\mathbf{i}} \times \mathcal{F}_{\text{loss}}[\{\bar{\mathbf{y}}_\mathbf{i}\}^{\mathbf{n-m}}_{\mathbf{i}}, \{\mathbf{y_i}\}^{\mathbf{n-m}}_{\mathbf{i}}]$
  - loss = certain_loss + uncertain_loss
  - Optimize $\mathcal{G}_{\text{model}}$

4 Simplex Denoising Label Diffusion Model

In terms of posterior approximation via generative models, we tackle it from the perspective of denoising diffusion models, which is designed for reconstructing high-fidelity data from pure noise iteratively. We view the true label inference as an progressively denoising process from noisy label based on input feature x. In this paper, we apply simplex diffusion model (Mahabadi et al., 2024), one of the continuous diffusion model, to approximate the true label posterior probability from noisy labels. Simplex diffusion model diffuses in simplex probability space, which aligns with our attempt to estimate the posterior distribution.

Label Simplex Representation True label y will be represented in one-hot encoded format $y \in \{0,1\}^C$. For specific category c, $y_c = 1$ and $y_i = 0$ where $i \neq c$. Given the discrete nature of one-hot data representation, we need to first map such categorical data to continuous space to fit our continuous simplex diffusion model. We map the one-hot label representation $y \in \{0,1\}^C$ to k-logit simplex to generate $s^y \in \{\pm k\}^{|C|}$, whose i-th component satisfies

$$s^c_{(i)} = \begin{cases} k, & \text{if } i = c, \\ -k & \text{otherwise.} \end{cases} \tag{1}$$

where $k \in \mathbb{R}$ is a hyperparameter.

Training Let $\boldsymbol{y} \in p_{\text{data}}$ be the one-hot representation of a label with C classes and $\boldsymbol{s^y} = \{\pm k\}^{|C|}$ be its k-logit simplex representation of $\boldsymbol{y}$. The simplex diffusion model forward process $q(\boldsymbol{s^y_t}|\boldsymbol{s^y_{t-1}})$ is defined as a Gaussian-Markov process that produces a sequence of latent variables $\boldsymbol{s^y_1}, \dots, \boldsymbol{s^y_T}$ by gradually adding Gaussian noise at each time step $t \in 1, 2, \dots, T$ with variance $\beta_t \in \mathbb{R}_{>0}$:

$$q(\boldsymbol{s^y_t}|\boldsymbol{s^y_{t-1}}) = \mathcal{N}(\boldsymbol{s^y_t}|(1-\beta_t)\boldsymbol{s^y_{t-1}}, \beta_t\mathbf{I}) \tag{2}$$

Let $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, k^2\mathbf{I})$ as we convert data into simplex space, $\alpha_t = 1 - \beta_t$, and $\bar{\alpha}_t = \prod^t_{j=1} \alpha_j$. Sampling $\boldsymbol{s^y_t}$ at an arbitrary time step t has a closed-form solution:

$$\boldsymbol{s}_t^{\boldsymbol{y}} = \sqrt{\bar{\alpha}_t}\boldsymbol{s}_0^{\boldsymbol{y}} + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t \tag{3}$$

Given a well-behaved noise schedule $\{\beta_t\}_{t=1}^{T}$, a little amount of Gaussian noise with variance β_t is injected, while a large amount $1-\beta_t$ of previous sample $\boldsymbol{s}_{t-1}^{\boldsymbol{y}}$ is preserved for each time step t. At the last time step $t = T$, our original data is expected to be no different from pure Gaussian distribution $\mathcal{N}(0, \mathbf{I})$. Therefore, in the denoising process, we can sample random noise from a standard Gaussian distribution and recover it sequentially to samples from p_{data}. Such an approximation of the reverse process $q(\boldsymbol{s}_{t-1}^{\boldsymbol{y}}|\boldsymbol{s}_t, \boldsymbol{s}_0)$ can be delivered via a neural network with parameters $\boldsymbol{\theta}$, $p_{\boldsymbol{\theta}}(\boldsymbol{s}_{t-1}^{\boldsymbol{y}}|\boldsymbol{s}_t^{\boldsymbol{y}})$. In the context of our posterior estimation, neural network is conditioned on $\boldsymbol{s}^{\tilde{\boldsymbol{y}}}$, where $\tilde{\boldsymbol{y}}$ is the noisy label, to approximate $\boldsymbol{s}_{t-1}^{\boldsymbol{y}}$ at time step t. The reverse process then is parameterized as

$$\boldsymbol{p}_{\boldsymbol{\theta}}(\boldsymbol{s}_{t-1}^{\boldsymbol{y}}|\boldsymbol{s}_t^{\boldsymbol{y}}, \boldsymbol{s}^{\tilde{\boldsymbol{y}}}, \boldsymbol{x}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{s}_t^{\boldsymbol{y}}, t|\boldsymbol{s}^{\tilde{\boldsymbol{y}}}, \boldsymbol{x}), \boldsymbol{\Sigma}_{\boldsymbol{\theta}}(\boldsymbol{s}_t^{\boldsymbol{y}}, t|\boldsymbol{s}^{\tilde{\boldsymbol{y}}}, \boldsymbol{x})) \tag{4}$$

As cross-entropy loss is typical in classification problem, we adopt it between the ground truth label and the model prediction given a noisy logit simplex $\boldsymbol{s}_t$ at time step t.

$$\mathcal{L} = \mathbb{L}_{t, q(\boldsymbol{s}_0^{\boldsymbol{y}}|\boldsymbol{s}^{\tilde{\boldsymbol{y}}}, \boldsymbol{x}_i), q(\boldsymbol{s}_t^{\boldsymbol{y}}|\boldsymbol{s}_0^{\boldsymbol{y}}, \boldsymbol{s}^{\tilde{\boldsymbol{y}}}, \boldsymbol{x}_i)}\Big[-\sum_{i=1}^{L} \log \boldsymbol{p}_{\boldsymbol{\theta}}(\boldsymbol{y}_i|\boldsymbol{s}_t^{\boldsymbol{y}_i}, t, \boldsymbol{s}^{\tilde{\boldsymbol{y}}_i}, \boldsymbol{x}_i)\Big] \tag{5}$$

Noise Schedule One important component in the diffusion forward process is the noise schedule. We follow the following cosine schedule for α_t:

$$\bar{\alpha}_t = \frac{f(t)}{f(0)}, \qquad f(t) = \cos\left(\frac{\frac{t}{T}+s}{1+s}\cdot\frac{\pi}{2}\right)^2 \tag{6}$$

Inference During the inference of the simplex diffusion model, $\boldsymbol{s}_T$ is sampled from the prior $\mathcal{N}(0, k^2\mathbf{I})$. The model predictions are iteratively denoised for $t = T, \ldots, 1$ starting from k-logit simplex Gaussian noise. This reverse process can be approximated via an adjustment of Equation (3):

$$\boldsymbol{s}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\hat{\boldsymbol{S}}_{\boldsymbol{\theta}}(\boldsymbol{s}_t, t|\boldsymbol{s}^{\tilde{\boldsymbol{y}}}, \boldsymbol{x}) + \sqrt{1-\bar{\alpha}_{t-1}}\boldsymbol{\epsilon}_t \tag{7}$$

where $\hat{\boldsymbol{S}}_{\boldsymbol{\theta}}$ is the model prediction of the ground-truth, $\boldsymbol{s}^{\tilde{\boldsymbol{y}}}$ is noisy label simplex and $\boldsymbol{x}$ is the input embedding, on which the model is conditioned. The model prediction $\hat{\boldsymbol{S}}_{\boldsymbol{\theta}}(\boldsymbol{s}_t, t|\boldsymbol{s}^{\tilde{\boldsymbol{y}}}, \boldsymbol{x})$ is regarded as the hypothetical ground-truth and corrupt it by $(t-1)$ time steps. To construct the model prediction, we project the logits produced by the underlying conditional model via argmax to match the initial k-logit representation:

$$\hat{\boldsymbol{s}}_{(i)}^{c} = \begin{cases} k, & \text{if } i = \text{argmax}(\boldsymbol{s}^{\boldsymbol{y}}), \\ -k & \text{otherwise.} \end{cases} \tag{8}$$

5 Experiments & Results

First, we introduce the tasks and datasets (20News Group, NumClaim, TREC, SemEval) that our experiments are conducted on (Section 5.1). Then, we describe our experimental setup (Section 5.2). Subsequently, we present the results of LLMs noise (Section 5.3) and synthetic noise, and real world noise (Section 5.4). Finally, we validate the effectiveness of each component in our framework (Section 5.5).

5.1 Tasks and Datasets

For our experiments, we include financial numerical claim detection from Shah et al. (2024), question classification from Li & Roth (2002), semantic relation classification task from Hendrickx et al. (2019), and news topic modeling task from Lang (1995). A summary of datasets used with the train-validation-test split is provided in table 1. We provide brief details about each task and dataset in Appendix A.

Dataset	# Labels	Dataset Size		
		Train	Valid	Test
NumClaim	2	1715	429	537
TREC	6	5033	500	500
SemEval	9	1749	178	600
20News	20	9051	2263	7532

Table 1: Summary of datasets used. Dataset size denotes the number of samples in the benchmark.

5.2 Experimental Setup

Baselines We compare SiDyP with the most relevant state-of-the-art baselines from three different categories in the realm of learning from noisy labels: (1) *Basic Performances* without specific design tackling noisy labels (Devlin et al., 2019a); (2) *Multi-Model Training Strategies*: **Co-Teaching** (Han et al., 2018a) and **JoCoR** (Wei et al., 2020). **Co-Teaching** trains two networks simultaneously and selects small-loss instances as clean samples for subsquent training. **JoCoR** also trains two networks simultaneously and use co-regularization to achieve agreement to filter out noisy samples by selecting instances with small losses; (3) *Generative Models for Noisy Maxtrix Estimation*: **NPC** (Bae et al., 2022) and **DyGen** (Zhuang et al., 2023). **NPC** utilize a generative model to calibrate the prediction of classifiers trained on noisy labels via a transition matrix. **DyGen** leverages the training dynamics to detect noisy samples and use a generative model to calibrate.

Evaluation We evaluate all the experiments using accuracy on clean test datasets. We only run the model on the test dataset at the point when the validation accuracy achieves the highest during training. The reported test performances of all baselines and our SiDyP is selected by this procedure. Given that the success of existing weakly-supervised learning methods relies heavily on clean validation samples (Zhu et al., 2023), we use noisy validation sets for model selections in all experiments. All experiments are run under 5 random seeds. We report the mean of the performances and the standard deviation.

Implementation Details We implement SiDyP using PyTorch (Paszke et al., 2019) and HuggingFace (Wolf et al., 2020). We use BERT (Devlin et al., 2019a) as our PLC in Stage I. For our baselines which contains PLC fine-tuning on noisy label datasets (**NPC**, **DyGen**, **GaDyP**), we use only one coherent PLC results for their individual post process to ensure a fair comparison as random seeds affect network initialization, synthetic noise generation, etc. More training details are revealed in Appendix D.

5.3 LLMs Noise Experiments

We run extensive experiments on various tasks and diversified LLM noises. First, we examine our framework in NumClaim, TREC, and SemEval labelled by `Llama-3-70b-chat-hf` (Dubey et al., 2024) in both zero-shot and few-shot manner. We only prompt 20News Group in zero-shot manner as it is a document level task, and Llama-3-70b has a context length limitation of 8192, which is not sufficient for few-shot learning. Then, to test SiDyP under diversified LLM noises, we prompt `Meta-Llama-3.1-70B-Instruct-Turbo` (Dubey et al., 2024), `Meta-Llama-3.1-405B-Instruct-Turbo` (Dubey et al., 2024), `gpt-4o` (OpenAI et al., 2024), and `Mixtral-8x22B-Instruct-v0.1` (Jiang et al., 2024) in both zero-shot and few-shot prompting manners on SemEval task. We address the experiment details and results in the following.

LLM Prompting For both zero-shot and few-shot manners, we use same prompts of same tasks for different LLMs (See prompting details in Appendix B.2). Notably, when prompting the LLM to label data, it is not guaranteed that it would follow the instructions and output in the specified format. It leads to missing labels for some data samples in our annotated datasets. Although we observe that the portion of missing labels is trivial (i.e. highest missing label ratio (only 0.014%) happens in 20News Group dataset. See full statistics in Appendix E), we still want to preserve those data samples to maintain data's integrity for training. Therefore, we randomly assign a label to those missing-label samples according to a uniform distribution over all labels. We use the dataset after random assignment for both training and validation. We do not apply random assignment for test dataset and report LLMs' raw accuracy in Table 2 and 3.

Results Table 2 shows the results of Llama-3-70b on all four tasks. Our method (SiDyP) outperforms all baselines by a notable margin 2.05& across all tasks in both prompting manners. There are averagely 6.34% samples of a fine-tuned PLC, and 5.77% of raw Llama-3-70b labelled samples successfully corrected by SiDyP. The performance gain on SemEval task is the most significant, achieving an average increase of 3.7%. This indicates that SiDyP is robust to high noise ratio dataset. Although the base performance of NumClaim is competitive, SiDyP is able to bring an average of 20.19% marginal increase. For NumClaim in few-shot manner, our method is the only one to outperform Llama-3-70b raw labelling accuracy and fine-tuned PLC. We also observe that both methods of multi-model training strategies struggle in these tasks. We think it's because of its training from scratch as PLC possesses prior knowledge that would be helpful despite that they are prone to noisy labels. Transition matrix-based methods performs generally better as it leverages pre-trained models and calibrate it via a post-process.

Datasets (→)	**NumClaim**		**TREC**		**SemEval**		**20News**
Method (↓)	**Zero-shot**	**Few-shot**	**Zero-shot**	**Few-shot**	**Zero-shot**	**Few-shot**	**Zero-shot**
Llama-3-70b	89.94	95.53	81.80	84.00	47.50	48.50	74.04
PLC	90.54±0.72	95.11±0.30	80.64±0.94	77.72±1.34	51.59±0.44	50.46±0.72	71.2±0.52
Co-teaching	82.31±1.11	83.77±4.05	69.20±2.09	67.20±2.21	46.53±4.16	44.29±6.18	35.28±12.18
JoCoR	83.35±1.97	85.82±2.05	70.80±3.00	65.82±2.17	45.66±3.25	44.11±2.23	42.39±11.98
NPC	90.83±0.62	95.04±0.61	79.48±1.97	78.88±1.47	50.73±1.70	47.53±1.26	70.60±0.51
DyGen	<u>91.13±0.30</u>	<u>95.41±0.28</u>	<u>82.88±0.71</u>	<u>84.80±0.86</u>	<u>60.86±0.81</u>	<u>60.79±2.23</u>	<u>71.42±0.31</u>
SiDyP	**93.63±0.84**	**95.97±0.15**	**84.76±0.79**	**85.60±0.44**	**64.26±0.27**	**64.79±0.96**	**72.66±0.58**

Table 2: Performance comparison of Llama-3-70b on zero-shot and few-shot learning tasks across multiple datasets, including NumClaim, TREC, SemEval, and 20News. Results are reported as classification accuracy with mean and standard deviations of 5 runs under different seed. **Bold** represents the best performance, while <u>underline</u> presents the second-best performance. Same seed setting and presentation apply in the following tables.

Robustness Check for Diversified LLMs Instead of limiting to Llama-3-70b, we extend our experiments to a variety of LLMs of different families with different sizes. We follow the same prompting and assignment procedure as describe above (See details in Appendix B.1). We aim to check the robustness of our SiDyP framework under multiple LLM-generated label noise. Table 3 shows the results of various types of LLM label noise on SemEval. Our method (SiDyP) achieves a significantly better performance compared to all baselines across all LLMs and both prompting manners. Specifically, SiDyP obtain an average of 4.47% performance gain than the second best baseline. Comparing to a fine-tuned PLC on noisy dataset, our method is able to boost the performance by an average of 8.02%. Notably, a significant average increase of 11.73% than LLMs raw accuracy is brought by our method. Combining all, we validate that our method is robust and resilient to different types of LLM noise and different prompting methods.

5.4 Synthetic and Real-world Noise Experiments

Observing significant performance improvement in LLM-generated label noises, we further test our method under different families of noises, synthetic and real-world, on SemEval task. We reveal the experiment details and results below.

Noise Generation We inject three types of synthetic noises, including **Symmetric Noise (SN)**, **Asymmetric Noise (ASN)**, and **Instance-Dependent Noise (IDN)**. Symmetric Noise flips labels

Dataset (→)	**SemEval**							
Method (↓)	**Llama-3.1-70b**		**Llama-3.1-405b**		**GPT4o**		**Mixtral-8x22b**	
	Zero-shot	**Few-shot**	**Zero-shot**	**Few-shot**	**Zero-shot**	**Few-shot**	**Zero-shot**	**Few-shot**
Base	52.66	55.16	55.16	52.16	56.50	57.66	42.66	40.83
PLC	60.26±0.89	57.70±1.10	54.76±1.24	53.96±0.12	58.63±0.86	61.56±0.93	49.29±1.31	46.33±1.32
Co-teaching	52.50±5.35	54.09±3.56	45.51±1.96	51.36±0.89	52.13±5.36	60.91±5.58	39.3±6.79	27.35±2.55
JoCoR	45.06±0.97	44.26±9.55	45.39±4.29	50.28±3.07	53.31±5.43	53.05±4.78	32.94±8.73	27.26±1.46
NPC	60.13±0.77	57.49±3.00	55.06±2.99	54.53±1.24	59.56±0.90	61.40±1.53	47.56±1.26	41.96±0.70
DyGen	68.53±0.88	64.53±2.85	59.69±1.31	51.69±2.02	62.63±0.91	64.03±0.82	50.63±6.43	40.23±1.41
SiDyP	**71.66±0.91**	**67.43±1.36**	**62.76±0.99**	**60.46±2.06**	**66.86±0.48**	**68.83±1.07**	**57.96±1.94**	**50.66±2.02**

Table 3: Performance comparison of Llama-3.1-70b, Llama-3.1-405b, GPT4o, and Mixtral-8×22b on zero-shot and few-shot learning tasks on SemEval. "Base" represents LLM's raw accuracy on test sets.

uniformly to other classes (Zhuang et al., 2023; Bae et al., 2022; Han et al., 2018a). Asymmetric Noise flips labels with similar classes (Zhuang et al., 2023; Bae et al., 2022). Instance-Dependent Noise flips label with a probability proportional to the features of the sample (Zhuang et al., 2023; Bae et al., 2022). As synthetic noise is controlled, we use the noise ratio of 50% to make a comparison with LLM noise. We choose 50% because LLM noises ratio on SemEval are around 50%. For real-world noise, we take majority vote on the 164 labeling functions' output provided in WRENCH (Zhang et al., 2021a) for the SemEval dataset.

Results In Table 4, we present the results of various synthetic noises and real-world noises on SemEval. SiDyP achieves an average of 2.80% increase compared to the second-best baseline. We observe that the performance increase between SiDyP and a strong baseline DyGen on LLM noises (5.21%) is higher than it on synthetic noises (3.26%). This is because DyGen performs better on synthetic datasets as such noises are less intricate (Zhuang et al., 2023). It further validates that LLM-generated label noises align more with real-world noise, making it more challenging for other baselines to arrive at accurate estimates. SiDyP, on the other hand, is resilient to all types of label noise, and brings improvement consistently. Moreover, all baselines are prone to the real-world noise as they struggle to be comparable with Base and PLC performances. SiDyP is the only one outperforming them by 3.36% and 1.73% increase respectively.

Datasets (→)	**SemEval**			
Method (↓)	**SN**	**ASN**	**IDN**	**Real World**
Base	50.00	50.00	50.00	82.50
PLC	65.06±2.13	40.96±2.60	59.83±2.65	84.13±0.68
Co-teaching	49.78±7.82	38.79±9.04	37.00 ±3.88	70.2±0.7
JoCoR	51.66±7.88	44.84±4.75	41.91±6.64	69.71±1.17
NPC	57.73±3.61	42.60±5.46	54.16±4.91	81.23±1.88
DyGen	73.06±2.07	53.16±5.46	71.40±1.80	82.3±0.13
SiDyP	**74.26±1.99**	**59.63±3.06**	**73.19±2.22**	**85.86±0.52**

Table 4: Performance comparison on SemEval with synthetic noise (SN, ASN, IDN) and real-world noise.

5.5 Effectiveness of Different Components

We investigate the effectiveness of each component in our SiDyP framework on Llama-3-70b labelled SemEval dataset in both zero-shot and few-shot manners. We eliminate them individually to validate their impact on performances: (1) Replacing our dynamic distillation priors with fix certain priors (for each sample, it's only associated with one fix certain label) in Stage II; (2) Substituting Stage II's generative model, simplex diffusion model with Dirichlet variational auto-encoder (VAE) (Joo et al., 2019) and Gaussian diffusion model (Sohl-Dickstein et al., 2015; Han et al., 2022; Chen et al., 2023b). Table 5 indicates the result. All experiments are conduct using same PLC fine-tuned results, and share the same value of hyper-parameters. Our simplex denoising label diffusion model surpasses Dirchlete VAE by an average of 2.17%. We believe such an enhancement comes from the de-noising capability of diffusion model. Moreover, it outperforms the Gaussin diffusion model by 8.58%. Our simplex denoising label diffusion model, which diffuses in probability simplex space, constructs a more reliable and accurate label probability from noisy labels. Besides, our dynamic prior distillation brings 1.53% increase. We further validate the improvement source of our dynamic prior by comparing the portion of correct labels we collect with fix prior method (See Appendix F for more details). Combining all, it confirms that our candidate retrieval algorithm could derive more true labels, and our prior distillation could find the correct labels among the candidates.

6 Related Work

Weak-supervision in machine learning includes incomplete, inexact, and inaccurate categories, each tailored to specific imperfections in data (Zhou, 2018). Inexact supervision deals with broad labels, while inaccurate supervision, where labels are erroneous, employ techniques like data programming (Ratner et al., 2017), human-in-the-loop strategies (Zhang et al., 2022), and contrastive loss for enhanced learning from data similarities and differences (Yu et al., 2020). Zhang et al. (2021a) apply a two-stage model to manage inaccurate supervision, initially denoising data before training on refined labels. In the landscape of learning from noisy labels, Iscen et al. (2022) proposed that there supposed to be similarities among training instances in the feature/embedding space, leading to the consistency of labels between data instances and their neighbors. NPC proposed by Bae et al. (2022), lies in the class of transition matrix base method. The true label is inferred by a prior, estimated by a pretrianed classifer, and a posterior, approximated by a generative model. DyGen (Zhuang et al. (2023)) infers true label based on the training dynamics during finetuning the pretrained language model. The feasibility of Diffusion Models in classification problems are explored and validated by Han et al. (2022). Chen et al. (2023a) is the very first to exploit the Gaussian diffusion model in the context of noisy label learning. LLMs have also been leveraged to iteratively expand label space under extremely weak supervision. X-MLClass (Li et al., 2024) demonstrated significant improvements in label discovery and multi-label classification accuracy in open-world settings. Additionally, explanation-aware ensembling methods like EASE (Yu et al., 2023) further illustrate how LLMs can be used to improve in-context learning by effectively guiding predictions and mitigating label noise.

Datasets (→)	**SemEval**	
Method (↓)	**Zero-shot**	**Few-shot**
FP + Dir-VAE	60.86$\pm$0.81	60.79$\pm$2.23
FP + Sim-Diff	<u>62.73$\pm$1.06</u>	<u>63.26$\pm$1.06</u>
DP + Gau-Diff	54.53$\pm$3.48	57.36$\pm$3.64
DP + Sim-Diff (**SiDyP**)	**64.26$\pm$0.27**	**64.79$\pm$0.96**

Table 5: Different components efficacy on zero-shot and few-shot labelled SemEval by Llama-3-70b. "FP"=fix prior. "DP"=our dynamic prior. "Dir-VAE"=Dirchlete VAE. "Gau-Diff"=Gaussian diffusion model. "Sim-Diff"=simplex diffusion model.

7 Discussion

In this paper, we propose a denoising framework, SiDyP, to enhance the learning from Llama-3 generated labels noise. Leveraging the principle of partial label learning and neighbor consistency, our label candidate retrieval and prior dynamic refinement algorithm alleviate the harm of incorrect labels during the training of a classifier. We introduce a simplex diffusion model to reconstruct categorical label data and utilize it as a posterior probability distribution estimator to calibrate the inaccurate prior distribution. Our framework boosts few-shot Llama-3 classification accuracy by a 7.69% average increase across all datasets of diverse noise ratios. We believe that our work sheds light on the realm of employing the diffusion model in the context of learning from noisy labels as well as the topics of calibrating incorrect llm-generated datasets.

References

HeeSun Bae, Seungjae Shin, Byeonghu Na, JoonHo Jang, Kyungwoo Song, and Il-Chul Moon. From noisy prediction to true label: Noisy prediction calibration via generative model. In *International Conference on Machine Learning*, pp. 1277–1297. PMLR, 2022.

Antonin Berthon, Bo Han, Gang Niu, Tongliang Liu, and Masashi Sugiyama. Confidence scores make instance-dependent label-noise learning possible, 2021.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, Ilya Sutskever, and Jeff Wu. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision, 2023. URL https://arxiv.org/abs/2312.09390.

Jian Chen, Ruiyi Zhang, Tong Yu, Rohan Sharma, Zhiqiang Xu, Tong Sun, and Changyou Chen. Label-retrieval-augmented diffusion models for learning from noisy labels, 2023a.

Jian Chen, Ruiyi Zhang, Tong Yu, Rohan Sharma, Zhiqiang Xu, Tong Sun, and Changyou Chen. Label-retrieval-augmented diffusion models for learning from noisy labels, 2023b. URL https://arxiv.org/abs/2305.19518.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019a. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019b. URL https://arxiv.org/abs/1810.04805.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie

Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2): 179–188, 1936.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems*, 31, 2018a.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels, 2018b.

Xizewen Han, Huangjie Zheng, and Mingyuan Zhou. Card: Classification and regression diffusion models, 2022.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid O Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. *arXiv preprint arXiv:1911.10422*, 2019.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

Ahmet Iscen, Jack Valmadre, Anurag Arnab, and Cordelia Schmid. Learning with neighbor consistency for noisy labels, 2022.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024. URL https://arxiv.org/abs/2401.04088.

Weonyoung Joo, Wonsung Lee, Sungrae Park, and Il-Chul Moon. Dirichlet variational autoencoder, 2019. URL https://arxiv.org/abs/1901.02739.

David Kartchner, Wendi Ren, David Nakajima An, Chao Zhang, and Cassie S Mitchell. Regal: Rule-generative active learning for model-in-the-loop weak supervision. *Advances in neural information processing systems*, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

Ken Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 331–339, 1995.

Xin Li and Dan Roth. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002.

Xintong Li, Jinya Jiang, Ria Dharmani, Jayanth Srinivasa, Gaowen Liu, and Jingbo Shang. Open-world multi-label text classification with extremely weak supervision, 2024. URL https://arxiv.org/abs/2407.05609.

Pierre Lison, Jeremy Barnes, and Aliaksandr Hubin. skweak: Weak supervision made easy for nlp. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-demo.40. URL http://dx.doi.org/10.18653/v1/2021.acl-demo.40.

Rabeeh Karimi Mahabadi, Hamish Ivison, Jaesung Tae, James Henderson, Iz Beltagy, Matthew E. Peters, and Arman Cohan. Tess: Text-to-text self-conditioned simplex diffusion, 2024.

Vitor Oliveira, Gabriel Nogueira, Thiago Faleiros, and Ricardo Marcacini. Combining prompt-based language models and weak supervision for labeling named entity recognition on legal documents. *Artificial Intelligence and Law*, pp. 1–21, 2024.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

Diego Ortego, Eric Arazo, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Multi-objective interpolation training for robustness to label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6606–6615, 2021.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019. URL https://arxiv.org/abs/1912.01703.

Giorgio Patrini, Alessandro Rozza, Aditya Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: a loss correction approach, 2017.

Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, volume 11 (3), pp. 269. NIH Public Access, 2017.

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B. Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning, 2021. URL https://arxiv.org/abs/2009.00236.

Agam Shah, Arnav Hiray, Pratvi Shah, Arkaprabha Banerjee, Anushka Singh, Dheeraj Eidnani, Bhaskar Chaudhury, and Sudheer Chava. Numerical claim detection in finance: A new financial dataset, weak-supervision model, and market analysis. *arXiv preprint arXiv:2402.11728*, 2024.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015.

Andreas Stephan, Vasiliki Kougia, and Benjamin Roth. Sepll: Separating latent class labels from weak supervision noise, 2022. URL https://arxiv.org/abs/2210.13898.

Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. Large language models for data annotation: A survey, 2024. URL https://arxiv.org/abs/2402.13446.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023a. URL https://arxiv.org/abs/2307.09288.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023b.

Song Wang, Zhen Tan, Ruocheng Guo, and Jundong Li. Noise-robust fine-tuning of pretrained language models via external guidance, 2023. URL https://arxiv.org/abs/2311.01108.

Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with co-regularization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13726–13735, 2020.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. Rush. Huggingface's transformers: State-of-the-art natural language processing, 2020. URL https://arxiv.org/abs/1910.03771.

Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. Part-dependent label noise: Towards instance-dependent label noise, 2020.

Yu Yao, Tongliang Liu, Bo Han, Mingming Gong, Jiankang Deng, Gang Niu, and Masashi Sugiyama. Dual t: Reducing estimation error for transition matrix in label-noise learning, 2021.

Peilin Yu and Stephen Bach. Alfred: A system for prompted weak supervision. In Danushka Bollegala, Ruihong Huang, and Alan Ritter (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pp. 479–488, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-demo.46. URL https://aclanthology.org/2023.acl-demo.46.

Yue Yu, Simiao Zuo, Haoming Jiang, Wendi Ren, Tuo Zhao, and Chao Zhang. Fine-tuning pre-trained language model with weak supervision: A contrastive-regularized self-training approach. *arXiv preprint arXiv:2010.07835*, 2020.

Yue Yu, Lingkai Kong, Jieyu Zhang, Rongzhi Zhang, and Chao Zhang. Actune: Uncertainty-aware active self-training for semi-supervised active learning with pretrained language models, 2022. URL https://arxiv.org/abs/2112.08787.

Yue Yu, Jiaming Shen, Tianqi Liu, Zhen Qin, Jing Nathan Yan, Jialu Liu, Chao Zhang, and Michael Bendersky. Explanation-aware soft ensemble empowers large language model in-context learning. *arXiv preprint arXiv:2311.07099*, 2023.

Jieyu Zhang, Yue Yu, Yinghao Li, Yujing Wang, Yaming Yang, Mao Yang, and Alexander Ratner. Wrench: A comprehensive benchmark for weak supervision. *arXiv preprint arXiv:2109.11377*, 2021a.

Rongzhi Zhang, Yue Yu, Pranav Shetty, Le Song, and Chao Zhang. Prboost: Prompt-based rule discovery and boosting for interactive weakly-supervised learning. *arXiv preprint arXiv:2203.09735*, 2022.

Yivan Zhang, Gang Niu, and Masashi Sugiyama. Learning noise transition matrix from only noisy labels via total variation regularization, 2021b.

Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National science review*, 5(1): 44–53, 2018.

Dawei Zhu, Xiaoyu Shen, Marius Mosbach, Andreas Stephan, and Dietrich Klakow. Weaker than you think: A critical look at weakly supervised learning, 2023. URL https://arxiv.org/abs/2305.17442.

Yuchen Zhuang, Yue Yu, Lingkai Kong, Xiang Chen, and Chao Zhang. Dygen: Learning from noisy labels via dynamics-enhanced generative modeling. *arXiv preprint arXiv:2305.19395*, 2023.

A Dataset and Task Detail

- **Numerical Claim Detection (NumClaim)**: This involves extracting numerical claims from financial texts like analysts' reports to forecast stock price volatility. Using a dataset with binary labels for sentences, this task distinguishes between "in-claim" sentences that predict financial outcomes and "out-of-claim" sentences that state factual information.
- **Question Classification (TREC)**: This task involves classifying questions into predefined categories based on their intent and content, as outlined in the TREC dataset from Li & Roth (2002) study. Using a dataset of labeled questions, this task assigns each question to one of six categories: location, entity, description, human, numeric value, and abbreviation. The goal is to determine the type of answer each question seeks, thereby facilitating targeted information retrieval and enhancing the efficiency of question-answering systems.
- **Semantic Relation Extraction (SemEval)**: This task focuses on the multi-way classification of semantic relations between pairs of nominals, as defined in SemEval-2010 Task 8 (Hendrickx et al., 2019). Utilizing a dataset where each pair of nominals is annotated with one of nine (Cause-Effect, Instrument-Agency, etc.) possible semantic relations, this task involves determining the specific type of relationship that exists between the two terms. The nine categories include Cause-Effect, Instrument-Agency, Product-Producer, Content-Container, Entity-Origin, Entity-Destination, Component-Whole, Member-Collection, and Message-Topic. The objective is to enhance the understanding of linguistic patterns and to improve the semantic analysis capabilities of natural language processing systems.
- **News Topic Modeling (20News)**: This task involves classifying news articles into different topics using the well-known 20 Newsgroups dataset (Lang, 1995). The dataset contains around 20,000 documents collected from newsgroups, organized into 20 different categories such as 'rec.sport.baseball', 'comp.graphics', and 'sci.med'. Each document is assigned to one of these categories. The task's objective is to train models to effectively capture the topical structure of news articles, which helps improve text categorization and topic detection capabilities in natural language processing applications.

B LLM Prompting Details

B.1 Model Implementation Details

We use the `Llama-3-70b-chat-hf` (Touvron et al., 2023b) model for all of our inferences. We take advantage of API from together.ai. We are grateful to them for providing free credits and making it possible. We use the model with a $temperature$ value of 0.00 (for reproducibility) and max_token of 100. The same hyper-parameters are used for `Meta-Llama-3.1-70B-Instruct-Turbo`, `Meta-Llama-3.1-405B-Instruct-Turbo`, `Mixtral-8x22B-Instruct-v0.1`, and `gpt-4o`.

B.2 Prompt Templates

Numerical Claim Detection

We use the following zero-shot prompt for numerical claim detection:

prompt_json = [

"role": "user", "content": f"Classify the following sentence into 'INCLAIM', or 'OUTOFCLAIM' class. 'INCLAIM' refers to predictions or expectations about financial outcomes, it can be thought of as 'financial forecasts'. 'OUTOFCLAIM' refers to sentences that provide numerical information or established facts about past financial events. Now, for the following sentence provide the label in the first line and provide a short explanation in the second line. The sentence: sentence",

]

We use the following few-shot prompt for numerical claim detection:

prompt_json = [

"role": "user", "content": f"Classify the following sentence into 'INCLAIM', or 'OUTOFCLAIM' class. 'INCLAIM' refers to predictions or expectations about financial outcomes, it can be thought of as 'financial forecasts'. 'OUTOFCLAIM' refers to sentences that provide numerical information or established facts about past financial events. Here are two examples: \nExample 1: consolidated total capital was $2.9 billion for the quarter. // OUTOFCLAIM\nExample 2: we expect revenue growth to be in the range of 5.5% to 6.5% year on year. // INCLAIM \nNow, for the following sentence provide the label in the first line and provide a short explanation in the second line. The sentence: {sentence}",

]

TREC

We use the following zero-shot prompt for the TREC dataset:

prompt_json = [

"role": "user", "content": f"For the following question, which belongs to a specific category, categorize it into one of the following classes based on the type of answer it requires: Abbreviation (ABBR), Entity (ENTY), Description (DESC), Human (HUM), Location (LOC), Numeric (NUM). Provide the label in the first line and provide a short explanation in the second line. The question: {question},

]

We use the following few-shot prompt for the TREC dataset:

prompt_json = [

"role": "user", "content": f"For the following question, which belongs to a specific category, categorize it into one of the following classes based on the type of answer it requires: Abbreviation (ABBR), Entity (ENTY), Description (DESC), Human (HUM), Location (LOC), Numeric (NUM). Here are six examples:\nExample 1: how did serfdom develop in and then leave russia ? // DESC\nExample 2: what films featured the character popeye doyle ? // ENTY\nExample 3: what contemptible scoundrel stole the cork from my lunch ? // HUM\nExample 4: what is the full form of .com ? // ABBR\nExample 5: what sprawling u.s. state boasts the most airports ? // LOC\nExample 6: when was ozzy osbourne born ? // NUM \nNow for the following question provide the label in the first line and provide a short explanation in the second line. The question: {question},

]

SemEval

We use the following zero-shot prompt for the SemEval dataset:

prompt_json = [

"role": "user", "content": f"The task is to identify the type of semantic relationship between two nominals in a given sentence. Below are the definitions of the nine relationship categories you must choose from:\nCause-Effect (CE): An event or object leads to an effect.\nInstrument-Agency (IA): An agent uses an instrument.\nProduct-Producer (PP): A producer causes a product to exist.\nContent-Container (CC): An object is physically stored in a delineated area of space.\nEntity-Origin (EO): An entity is coming or is derived from an origin (e.g., position or material).\nEntity-Destination (ED): An entity is moving towards a destination.\nComponent-Whole (CW): An object is a component of a larger whole.\nMember-Collection (MC): A member forms a nonfunctional part of a collection.\nMessage-Topic (MT): A message, written or spoken, is about a topic.\nFor the provided sentence below, determine the most accurate relationship category based on the descriptions provided. Respond by selecting the label (e.g., CE, IA, PP, etc.) that best matches the relationship expressed in the sentence. Provide the label in the first line and provide a short explanation in the second line. The sentence: {sentence},

]

We use the following few-shot prompt for the SemEval dataset:

prompt_json = [

"role": "user", "content": f"The task is to identify the type of semantic relationship between two nominals in a given sentence. Below are the definitions of the nine relationship categories you must choose from:\nCause-Effect (CE): An event or object leads to an effect. (Example: As the right front wheel of Senna 's car hit the wall , the violent impact caused a torsion on the steering column , causing it to break .)\nInstrument-Agency (IA): An agent uses an instrument. (Example: The necromancer wields the power of death itself , a power no enemy can stand against .)\nProduct-Producer (PP): A producer causes a product to exist. (Example: This website , www.fertilityuk.org , shows how to interpret the changes that take place in the mucus secretions produced by the cells lining the cervix .)\nContent-Container (CC): An object is physically stored in a delineated area of space. (Example: I sent you a suitcase with cash in it so you can fill it up with wine gummies .)\nEntity-Origin (EO): An entity is coming or is derived from an origin (e.g., position or material) (Example: I have always felt so relieved that Roy and the boys had left the creek .).\nEntity-Destination (ED): An entity is moving towards a destination. (Example: The machine blows water into the connecting conduit .)\nComponent-Whole (CW): An object is a component of a larger whole. (Example: He noticed a speck of blood on the man 's thumb and what he thought were several corresponding drops on the driver 's door of the truck .)\nMember-Collection (MC): A member forms a nonfunctional part of a collection. (Example: With the conquest of Jerusalem in 1099 , Geoffrey de Bouillon established a chapter of secular canons in the basilica of the Holy Sepulcher to offer the sacred liturgy according to the Latin rite .)\nMessage-Topic (MT): A message, written or spoken, is about a topic. (Example: A number of scientific criticisms of Duesberg 's hypothesis were summarised in a review article in the journal Science in 1994 .)\nFor the provided sentence below, determine the most accurate relationship category based on the descriptions provided. Respond by selecting the label (e.g., CE, IA, PP, etc.) that best matches the relationship expressed in the sentence. Provide the label in the first line and provide a short explanation in the second line. The sentence: {sentence},

]

20News

We use the following zero-shot prompt for the 20News dataset:

prompt_json = [

"role": "user", "content": f"The task is to classify the given text into one of the 20 news group categories. Below are the 20 categories you must choose from:\n1. 'alt.atheism': Discussions related to atheism.\n2. 'comp.graphics': Topics about computer graphics, including software and hardware.\n3. 'comp.os.ms-windows.misc': Discussions about the Microsoft Windows operating system.\n4. 'comp.sys.ibm.pc.hardware': Topics related to IBM PC hardware.\n5. 'comp.sys.mac.hardware': Discussions about Mac hardware.\n6. 'comp.windows.x': Topics about the X Window System.\n7. 'misc.forsale': Posts related to buying and selling items.\n8. 'rec.autos': Discussions about automobiles.\n9. 'rec.motorcycles': Topics related to motorcycles.\n10. 'rec.sport.baseball': Discussions about baseball.\n11. 'rec.sport.hockey': Discussions about hockey.\n12. 'sci.crypt': Topics about cryptography and encryption.\n13. 'sci.electronics': Discussions about electronic systems and devices.\n14. 'sci.med': Topics related to medical science and healthcare.\n15. 'sci.space': Discussions about space and astronomy.\n16. 'soc.religion.christian': Topics about Christianity and related discussions.\n17. 'talk.politics.guns': Discussions about gun politics and related debates.\n18. 'talk.politics.mideast': Topics about politics in the Middle East.\n19. 'talk.politics.misc': General political discussions not covered by other categories.\n20. 'talk.religion.misc': Discussions about miscellaneous religious topics.\nFor the provided text below, determine the most appropriate category based on the descriptions above. Respond by selecting the label (e.g., alt.atheism, comp.graphics, etc.) that best matches the topic of the text. Provide the label in the first line and a brief explanation in the second line. The sentence: {sentence},

]

C Training Dynamics and Co-Regularization

Training Dynamics The training dynamics during PLC fine-tuning (Stage I in Figure 1) is not only beneficial for clean and noisy sample separation (as we discuss in Section 3), but also contains rich information attributing to generative model learning (Stage II in Figure 1) (Zhuang et al., 2023). Leveraging such dynamics, our empirical objective becomes:

$$p(y|x) \propto \sum_{\hat{y}} p(\hat{y}|x) p(y|\hat{y}, W)$$

where W denotes the training dynamics for each sample.

Co-Regularization Although we manage to mitigate the negative impact of label noises (Section 3,4), it is inevitable that small deviations in $p(\hat{y}|x)$ and $p(y|\hat{y}, x)$ could propagate to later stages, thus affecting the objective $p(y|x)$. We leverage multiple branches with identical architecture but different initializations (Zhuang et al., 2023). A co-regularization loss across branches is introduced to achieve consensus. Such a loss is calculated as the KL Divergence between the consensus probability (the average probability of models' predicted probability in different model branches) and each individual model's predicted probability. We apply co-regularization mechanism to both Stage I PLC $\mathbf{F}_{\varphi}(\hat{y}|x)$ and Stage II generative model $p_\theta(y|\hat{y}, x)$. To begin, we initialize M copies of $\mathbf{F}_{\varphi}^{(m)}(\hat{y}|x)$ and $p_\theta^{(m)}(y|\hat{y}, x)$. Passing instances x_i to different model branches, we can obtain corresponding model predicted probabilities $p_i^{(m)}$. Then, a aggregated probability q_i can be calculated by averaging all predicted probabilities:

$$q_i = \frac{1}{M} \sum_{m=1}^{M} p_i^{(m)}$$

Given these, a co-regularization loss can be calculated as follows:

$$\begin{aligned}\ell_{\text{CR}} &= \frac{1}{MN} \sum_{i=1}^{N} \sum_{m=1}^{M} \text{KLK}(q_i || p_i^{(m)}) \\ &= \frac{1}{MN} \sum_{i=1}^{N} \sum_{m=1}^{M} \sum_{c=1}^{C} q_{ic} \log \left(\frac{q_{ic} + \epsilon}{p_{ic}^{(m)} + \epsilon} \right)\end{aligned}$$

where ϵ indicates a small positive number to avoid division by zero.

D SiDyP Training Details

All experiments are conducted on CPU: Intel(R) Xeon(R) W-2295 CPU @ 3.00GHz and GPU: NVIDIA GeForce RTX A6000 GPUs using Python 3.11.5 and PyTorch 2.0.1. Table 6 indicates all specific hyper-parameters we use in different datasets. We use Adam (Kingma & Ba, 2017) as optimizer. E_{BERT} is the training epochs for the BERT classifier. E_{SD} is the training epochs for the simplex diffusion model. σ is the estimated error rate in Algorithm 1. λ is the threshold that we separate certain and uncertain prior in Algorithm 1. γ is the threshold that we preserve the dominance candidates in uncertain prior in Algorithm 1. In Algorithm 2, α is the warmup epochs for Stage II generative model training. m is the number of model branch. β is the number of sample times that we use to refine our uncertain prior based on model's predictions.

Time Complexity We perform Big-O analysis for SiDyP. The time complexity for SiDyP is $O(W^2 \times T)$ where W denotes the embedding size of training dynamics and T is either training timesteps or inference timesteps of our simplex diffusion model. We choose γ based on our empirical estimation. To make a fair comparison, we use the same estimate error rate in all other baselines which requires one. We grid search these hyper-parameters: λ in [0.7, 0.8, 0.9, 1.0], γ in [0.4, 0.6, 0.8], α in [1, 2, 3, 4, 5, 6], β in [2, 4, 6, 8], K in [10, 20, 30], train timesteps in [400, 500, 600, 700, 800], inference timesteps in [10, 20, 50, 100], learning rate in [1e-3, 6e-4, 3e-4, 1e-5].

LLM (→)	Llama-3-70b						
Datasets (→)	**NumClaim**		**TREC**		**SemEval**		**20News**
Method (↓)	**Zero-shot**	**Few-shot**	**Zero-shot**	**Few-shot**	**Zero-shot**	**Few-shot**	**Zero-shot**
E_{BERT}	20	20	20	20	20	20	20
batch size	128	128	128	128	128	128	128
learning rate (BERT)	5e-5	5e-5	5e-5	5e-5	5e-5	5e-5	5e-5
max length	128	128	64	64	128	128	128
σ	0.1	0.05	0.3	0.3	0.5	0.5	0.5
λ	0.9	0.9	0.9	0.9	0.9	0.9	0.9
γ	0.8	0.8	0.8	0.8	0.8	0.8	0.8
α	2	1	1	1	2	3	4
m	3	3	3	3	3	3	3
β	4	4	4	4	4	4	4
E_{SD}	10	10	10	10	10	10	10
batch size (SD)	128	128	128	128	128	128	128
learning rate (SD)	6e-4	6e-4	6e-4	6e-4	6e-4	6e-4	6e-4
train timesteps	800	500	800	600	800	500	500
inference timesteps	10	10	50	80	10	10	10
K	20	20	20	10	10	10	10

Table 6: Training hyper-parameters details for SiDyP on all six Llama-3 generated datasets.

E LLM Noise Ratio

See Table 8

LLM (→)	Llama-3-70b						
Datasets (→)	**NumClaim**		**TREC**		**SemEval**		**20News**
Method (↓)	**Zero-shot**	**Few-shot**	**Zero-shot**	**Few-shot**	**Zero-shot**	**Few-shot**	**Zero-shot**
Noise Ratio (Original)	91.69	95.85	70.35	69.72	50.96	50.64	76.13
No Answer Ratio	0.00	0.00	$3.6e^{-4}$	$1.8e^{-4}$	$2.5e^{-3}$	$4.1e^{-3}$	$1.4e^{-2}$
Noise Ratio (After RA)	91.69	95.85	70.35	69.72	50.96	50.64	76.23

Table 7: Llama-3-70b label noise ratio on training sets of NumClaim, TREC, and SemEval in zero-shot and few-shot manners, and 20News Group in zero-shot manner. "RA" represents random assignment.

Dataset (→)	SemEval							
Method (↓)	**Llama-3.1-70b**		**Llama-3.1-405b**		**GPT4o**		**Mixtral-8x22b**	
	Zero-shot	**Few-shot**	**Zero-shot**	**Few-shot**	**Zero-shot**	**Few-shot**	**Zero-shot**	**Few-shot**
Noise Ratio (Original)	57.39	56.66	57.70	55.78	60.61	61.49	44.94	44.42
No Answer Ratio	0.00	0.00	0.001	0.0005	0.00	0.00	0.009	0.001
Noise Ratio (After RA)	57.39	56.66	57.75	55.78	60.61	61.49	44.94	44.42

Table 8: Label noise ratio of SemEval training set by Llama-3.1-70b, Llama-3.1-405b, GPT4o, and Mixtral-8×22b in both zero-shot and few-shot manners. "RA" represents random assignment.

F Label Candidate Efficacy

We calculate the accuracy of our label candidate compared to true labels for Llama-3-70b zero-shot labeled 20News Group, NumClaim, Trec, and SemEval across a wide-range of certain threshold λ and dominant threshold γ. For certain candidate, the accuracy is easy to calculate as we can directly compare to its corresponding true label. For uncertain candidate, we either compare the specific candidate with maximum probability with true label, or we check if true label lies in our uncertain candidate. Notably, when $\lambda = \gamma = 0$, our dynamic prior turns into fix prior. Our label candidate achieves an average of 9.5% improvement compared to fix prior. Figure 2 presents the entire distribution of our dynamic prior accuracy.

G LLM-generated Label Noise Characteristics

We plot SemEval's noise distribution of three different types of noise: LLM, synthetic, real-world in Figure 3. Except for real-world noise which has lower noise ratio (16%), both LLM-generated noise and synthetic noise's ratio are around 50%. Our observations are listed in the follow:

- Although the noise ratio of LLM-generated labels is comparable to that of synthetic noise, the correct ratio (the diagonal) is more diverse. In contrast, the correct ratios for all three types of synthetic noise are approximately 50%, reflecting an equal distribution of noise injection across classes.
- In synthetic noise, incorrect labels often show clear patterns (e.g., being consistently off by one class in ASN, noise distributed relatively equally in SN). The label noise introduced by IDN changes significantly depending on the seed used. Such a sensitivity to initial random state impacts model's robustness.
- While the distribution of synthetic noise indicates that this type of mislabeling often lacks contextual correlation, LLM-generated label noise reflects underlying relationships between classes (as evidenced by the similarity among the three LLMs), making it more aligned with real-world noise.

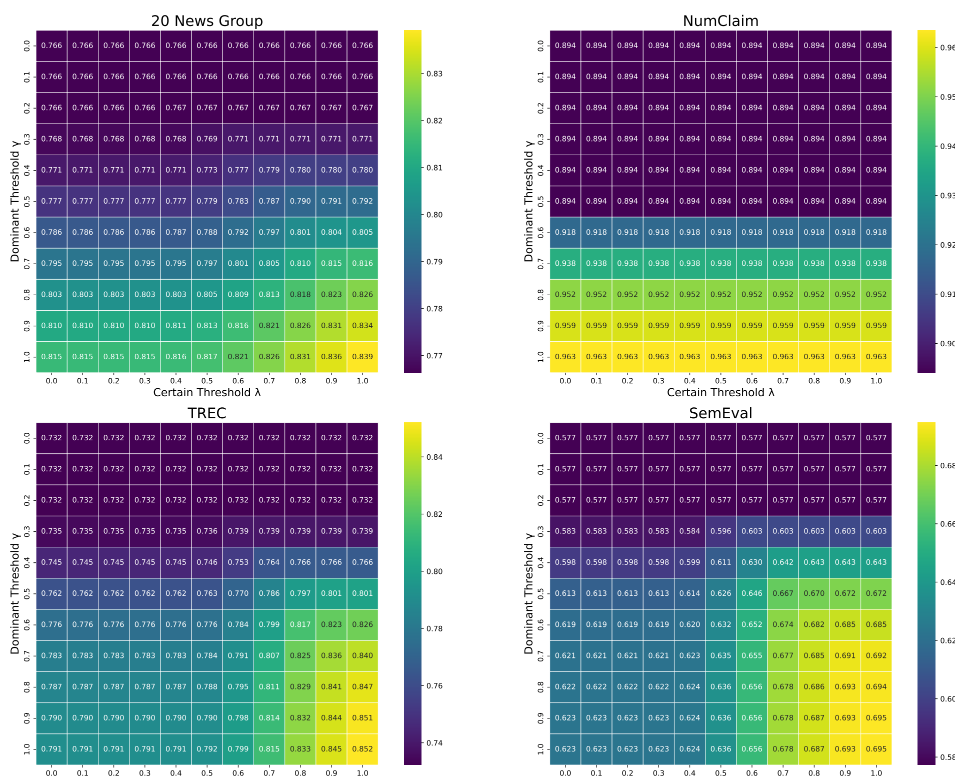

Figure 2: Label candidate accuracy distribution across different combinations of certain threshold λ and dominant threshold γ on 20 News Group, NumClaim, TREC, and SemEval labelled by Llama-3-70b in the zero-shot manner.

Figure 3: Noise distribution of differet types of noise: IDN under three seeds, Llama-3-70b zeroshot, Llama-3-70b fewshot, gpt4o, SN, ASN, and real-world

H Candidate Distillation Efficacy

Figure 4 presents the performance increase brought by our candidate dynamic distillation algorithm. We use all four datasets labelled by Llama-3-70B. We obtain the amount of data instances in our training set of each dataset being corrected. The corrected uncertain ratio is calculated by such an amount dividing the total number of uncertain data instances which contains true labels in their candidates. We observe that more noise the datasete has, more significant improvement our distillation can bring. Notably, it is able to correct 8.6% label in SemEval few-shot prompting.

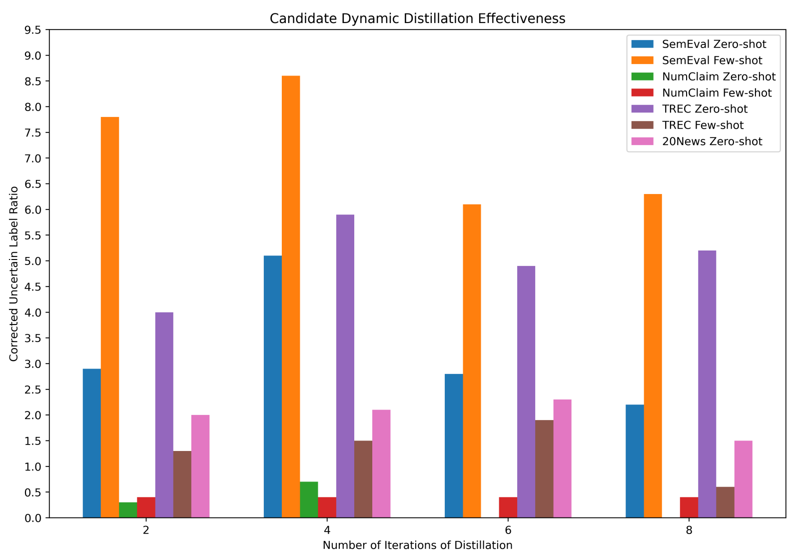

Figure 4: The ratio of uncertain labels being corrected by our candidate dynamic distillation.