

STGN: an Implicit Regularization Method for Learning with Noisy Labels in Natural Language Processing

Tingting Wu¹, Xiao Ding^{1*}, Minji Tang¹, Hao Zhang², Bing Qin¹, Ting Liu¹

¹Research Center for Social Computing and Information Retrieval
Harbin Institute of Technology, China

{ttwu, xding, mjtang, qinb, tliu}@ir.hit.edu.cn

²Faculty of Computing, Harbin Institute of Technology, China
zhh1000@hit.edu.cn

Abstract

Noisy labels are ubiquitous in natural language processing (NLP) tasks. Existing work, namely learning with noisy labels in NLP, is often limited to dedicated tasks or specific training procedures, making it hard to be widely used. To address this issue, SGD noise has been explored to provide a more general way to alleviate the effect of noisy labels by involving benign noise in the process of stochastic gradient descent. However, previous studies exert identical perturbation for all samples, which may cause overfitting on incorrect ones or optimizing correct ones inadequately. To facilitate this, we propose a novel stochastic tailor-made gradient noise (STGN), mitigating the effect of inherent label noise by introducing tailor-made benign noise for each sample. Specifically, we investigate multiple principles to precisely and stably discriminate correct samples from incorrect ones and thus apply different intensities of perturbation to them. A detailed theoretical analysis shows that STGN has good properties, beneficial for model generalization. Experiments on three different NLP tasks demonstrate the effectiveness and versatility of STGN. Also, STGN can boost existing robust training methods.¹

1 Introduction

Deep neural networks (DNNs) have notably succeeded on various natural language processing (NLP) tasks. However, noisy labels induced by labeling errors are prevalent in a wide range of corpora (Jia et al., 2019; Alt et al., 2020). DNNs can gradually fit correct data (i.e., noise-free data) and eventually memorize all data, including incorrect data (i.e., data with label noise) (Zhang et al., 2017), which would affect the generalization of deep models and result in the performance degradation. Hence, it is pressing and necessary to develop

*Corresponding author.

¹The code is released at <https://github.com/tangminji/STGN-sst>.

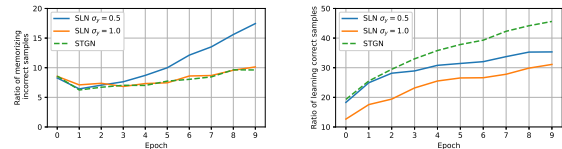


Figure 1: Ratios of memorizing incorrect samples (**Left**, i.e., training accuracy from incorrect ones (%), smaller is better) and learning correct samples (**Right**, i.e., training accuracy from correct ones (%), larger is better) during training on SST-5 with 40% label noise. Two ratios together form the training accuracy (%). Our method STGN gains more training accuracy on correct samples than existing method SLN but memorizes much less label noise.

robust methods in NLP to mitigate the impact of noisy labels.

To prevent overfitting on noisy labels, previous work mainly focuses on specific tasks (Jia et al., 2019; Meng et al., 2021) or relies on additional data annotations (Jindal et al., 2019; Hedderich et al., 2021), limiting its wide applications. For example, Le and Titov (2019) study noisy labels in entity linking, which depends on the definition of multi-instance learning. However, the framework is dedicated and hard to transfer to other tasks like text classification. As a general method, regularization can be applied to enhance generalization of the base model without excessive limitations. Specifically, implicit regularization (e.g., learning rate, dropout), arising regularization effect from optimization, can guarantee DNNs’ generalization effectively even non-use of any explicit regularization (Arora et al., 2019) (e.g., L1 norm regularization).

Among those sources of implicit regularization, the noise in stochastic gradient descent (SGD noise) is deemed as a crucial one. Previous studies (Nee-lakantan et al., 2015; HaoChen et al., 2021; Chen et al., 2021) inject gaussian noise into different positions of the gradient in the backpropagation of DNNs. However, they do not distinguish between

Method	$\tilde{\nabla}_\theta \ell(f, y)$
GNMP	$\nabla_\theta \ell(f, y) + \sigma_\theta g_\theta$ $\sigma_\theta > 0, g_\theta \in \mathbb{R}^{1 \times p},$ $g_\theta \sim \mathcal{N}(0, I_{p \times p})$
GNMO	$(\nabla_f \ell(f, y) + \sigma_f g_f) \cdot \nabla_\theta f$ $\sigma_f > 0, g_f \in \mathbb{R}^{1 \times c},$ $g_f \sim \mathcal{N}(0, I_{c \times c})$
SLN	$\nabla_\theta \ell(f, y + \sigma_y g_y)$ $\sigma_y > 0, g_y \in \mathbb{R}^{1 \times c},$ $g_y \sim \mathcal{N}(0, I_{c \times c})$

Table 1: Comparison of SGD noise variants.

correct and incorrect samples. Each sample is exerted *identical perturbation*, i.e., *the noise with the same magnitude or homovariance*. Fig. 1 shows ratios of memorizing incorrect samples and learning correct ones of a recent method SLN (Chen et al., 2021) with different noise intensities (i.e., $\sigma_y = 0.5, 1.0$). We observe an intense disturbance ($\sigma_y = 1.0$) significantly reduces incorrect samples memorization, mitigating overfitting. However, the ability to learn from correct samples also gets notably suppressed, leading to inadequate optimization.

To address this issue, we investigate a benign noise called stochastic tailor-made gradient noise (STGN) to resist inherent label noise, where *tailor-made* denotes *devising diverse disturbing magnitude for correct and incorrect samples*. Specifically, we combine multiple guidelines to separate correct and incorrect samples precisely, followed by updating the perturbation of each sample iteratively in a self-supervised manner. In this way, STGN exerts large benign noise on incorrect samples to relieve overfitting and learns correct ones without perturbation to promote sufficient optimization. In Fig. 1, STGN gains more training accuracy on correct samples than SLN ($\sigma_y = 0.5$) but memorizes much less label noise. Besides, we further provide a detailed theoretical analysis, revealing the mechanism of STGN facilitating the model generalization.

Experimental results on sentiment analysis, named entity recognition (NER), and event relation reasoning demonstrate consistent gain, which verifies the effectiveness and generality of STGN. Moreover, benign tailor-made noise can enhance the generalization of the model efficiently.

2 Preliminaries

2.1 Notations.

Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ denote the dataset with noisy labels, where \mathbf{x}_i is the i -th sample, and y_i is its corresponding label. Let $f(x; \theta)$ be the model with trainable parameters $\theta \in \mathbb{R}^p$ for any exam-

ple (x, y) . For a c -class classification task, we have the output $f(x; \theta) \in \mathbb{R}^c$ (abbr. f) and the loss of a sample $\ell(f, y)$ (abbr. ℓ). During the parameter update phase, a sample contributes $\nabla_\theta \ell(f, y)$ to the gradient descent, where $\nabla_\theta \ell(f, y)$ is calculated by the model optimizer, e.g., SGD, Adam and AdamW. *With SGD noise, the model is trained with a noisy gradient $\tilde{\nabla}_\theta \ell(f, y)$* . Following the standard notation of the Jacobian matrix, we have $\nabla_\theta \ell \in \mathbb{R}^{1 \times p}, \nabla_f \ell \in \mathbb{R}^{1 \times c}, \nabla_\theta f \in \mathbb{R}^{c \times p}$.

2.2 SGD Noise.

SGD noise is a critical one among the sources of implicit regularization (Keskar et al., 2017). Notably, inherent label noise is harmful to generalization, while SGD noise contributes to generalization due to its regularization effect. Therefore, the latter can be applied to mitigate the effects of the label noise. Moreover, as a general-purpose regularization method, it can be readily integrated into SGD or its variants and incorporates existing robust training methods against noisy labels.

2.3 Previous SGD Noise Variants

GNMP. Neelakantan et al. (2015) propose an SGD noise, adding Gaussian noise g_θ with mean-zero and constant standard deviation σ_θ on the gradient of loss w.r.t model parameters $\nabla_\theta \ell(f, y)$, obtaining noisy gradient $\tilde{\nabla}_\theta \ell(f, y)$.

GNMO. HaoChen et al. (2021) study an SGD noise induced by Gaussian noise g_f with mean-zero and constant standard deviation σ_f on the gradient of loss w.r.t model output $\nabla_f \ell(f, y)$, getting noisy gradient $\tilde{\nabla}_\theta \ell(f, y)$.

SLN. Chen et al. (2021) devise an SGD noise resulting from stochastic label noise, injecting Gaussian noise g_y with mean-zero and constant standard deviation σ_y to the one-hot labels y , acquiring noisy gradient $\tilde{\nabla}_\theta \ell(f, y)$.

The concrete forms of $\tilde{\nabla}_\theta \ell(f, y)$ in these methods are shown in Table 1, respectively. HaoChen et al. (2021) prove that in an over-parameterized setting, SLN or GMNO recovers the sparse groundtruth, whereas GMNP overfits to dense solutions. With the same and small training error, sparse solutions generalize better than dense solutions, whose intuitive explanation is called Occam’s razor, and No Free Lunch theorem (Shalev-Shwartz and Ben-David, 2014). Therefore, GNMO and SLN are preferable to GNMP at improving generalization when learning with noisy labels.

3 Method

3.1 Details of STGN

Previous SGD noise variants fail to consider the specialty of learning with noisy labels. In their studies, every sample is disturbed with identical magnitude. With no explicit discrimination between correct and incorrect samples, the intense perturbation may lead to inadequate model optimization, while minor disturbance can result in overfitting incorrect ones. To address this issue, we propose stochastic tailor-made gradient noise (STGN), i.e., *in a mini-batch, we reduce the disturbance on correct samples and increase the perturbation on incorrect ones*. STGN can be regarded as a kind of SGD noise, which designs tailor-made stochastic gradient noise for different samples. The exact format of our approach is as follows:

$$\left(\nabla_f \ell(f, y) + \frac{\sigma_f^t}{S} u_f\right) \cdot \nabla_\theta f, \quad (1)$$

where $\sigma_f^t > 0$ denotes disturbance intensity, and t is the current number of epoch. We add gradient noise sampled from a uniform distribution, where $u_f \in \mathbb{R}^{1 \times c}$ and $u_f \sim \mathcal{U}\left(-\frac{I_{c \times c}}{2}, \frac{I_{c \times c}}{2}\right)$. $I_{c \times c}$ denotes the $c \times c$ identity matrix. Notably, u_f follows a uniform distribution rather than the Gaussian distribution in previous work. This is mainly because compared with uniform distribution, sampling from Gaussian distribution may generate extremely large values, bringing additional large random disturbances, which is detrimental to reducing the perturbation on correct samples. We use a softmax function $S(f(x; \theta)) \in [0, 1]^c$ to obtain the probability of each class, where S indicates the softmax output of a sample.

We investigate an SGD noise by perturbing the gradient of loss w.r.t model output. But unlike existing studies, the standard deviation σ_f^t varies with every epoch’s judgment (i.e., whether the sample is correct or not) and is updated iteratively. Specifically, for each sample at the t -th epoch, we enlarge its σ_f^t if its label is noisy and diminish its σ_f^t otherwise. The problem is how to distinguish correct samples from incorrect ones.

Correct & Incorrect Samples Discrimination.

Deep networks can effectively filter out noise-free samples from the noisy training data by empirically treating small-loss examples as correct ones (Jiang et al., 2018; Han et al., 2018). Inspired by this loss-based separation, we exploit model loss to separate

noisy data. Let ϵ denote the *noise level*, measuring how many noisy labels are corrupted from ground-truth labels. Here, ϵ is a scalar and can easily be estimated in practice (Patrini et al., 2017). For any sample in a mini-batch, we contrive a simple but effective guideline g_1 to determine whether the data is noise-free or not. g_1 is defined as

$$g_1 = \text{sign}(\ell - \tau), \quad (2)$$

where $\text{sign}(\cdot)$ is the sign function, and ℓ indicates the loss of a sample. τ is the ϵ' -th percentile loss in the mini-batch, where ϵ' is uniquely determined by ϵ (e.g., $\epsilon = 0.4$, $\epsilon' = (1 - \epsilon) \cdot 100\% = 60(\%)$). $g_1 = -1$ if $\ell < \tau$, which implies the sample is noise-free and conversely $g_1 = 1$. $g_1 = 0$ if $\ell = \tau$.

However, it is hard to avoid memorizing noisy labels from the beginning of training since noisy samples cannot be distinguished without sufficient training. To alleviate this issue, we initialize σ_f^t as constant σ_f for each example, where $\sigma_f > 0$. Then we employ more guidelines to facilitate automatic and precise judgment regarding correct and incorrect samples. Specifically, Toneva et al. (2019) find that samples with label noise are among the most forgotten ones by DNNs and characterize the phenomenon by defining *forgetting events*. Namely, *if a training example transitions from being classified correctly to incorrectly at the t -th epoch, its number of forgetting events increase by one*. Inspired by the learning dynamics of DNNs, we propose guideline g_2 to separate correct data from incorrect data as follows:

$$g_2 = \text{sign}(\mathfrak{f}^t - \lambda), \quad (3)$$

where \mathfrak{f}^t is a sample’s number of forgetting events at the t -th epoch. λ denotes a threshold to separate incorrect data from correct ones. $g_2 = -1$ if $\mathfrak{f}^t < \lambda$, which indicates the sample is noise-free and conversely $g_2 = 1$. $g_2 = 0$ if $\mathfrak{f}^t = \lambda$.

By integrating the above guidelines into a general framework, we harvest precise and stable ways for identification. Predictions depending on more guidelines are analogous to the diversity advantage in ensemble learning, thus ensuring accurate discrimination results. Sign function imposes impactful constraints on calculating guidelines, guaranteeing stable outputs, and avoids drastically changed outputs among different epochs. Suppose we have m guidelines in total, which output a total contribution no more than \mathcal{C} , we have

Algorithm 1 STGN training.

Require: Number of epochs T , optimizer \mathcal{O} , mini-batch $\mathcal{B} = (x_i, y_i)_{i=1}^{n_b}$ of batch size n_b , constant $\sigma_f > 0$, $u_f \sim \mathcal{U}\left(-\frac{I_{c \times c}}{2}, \frac{I_{c \times c}}{2}\right)$.

- 1: $\sigma_f^0 \leftarrow \sigma_f$ $\triangleright \sigma_f^0 \in \mathbb{R}^N$
- 2: **for** $t = 1$ to T **do**
- 3: Sample \mathcal{B} randomly
- 4: $f \leftarrow f(\{x_i\}_{i=1}^{n_b}; \theta)$ \triangleright line 4-5: forward pass
- 5: $\ell \leftarrow \ell(f, \{y_i\}_{i=1}^{n_b})$
- 6: $S \leftarrow \text{Softmax}(f)$ \triangleright obtain each class's probability
- 7: $\nabla_f \ell \leftarrow \text{Backward}(\ell, f)$ \triangleright line 7-13: backward pass
- 8: **for** $i = 1, \dots, n_b$ **do** \triangleright each sample in \mathcal{B}
- 9: Σ_i^t is calculated by Eq. (4)
- 10: $\sigma_{f_i}^t \leftarrow \sigma_{f_i}^{t-1} + \gamma \Sigma_i^t$
- 11: $\tilde{\nabla}_f \ell_i \leftarrow \nabla_f \ell_i + \frac{\sigma_{f_i}^t}{S_i} u_f$
- 12: **end for**
- 13: $\tilde{\nabla}_\theta \ell \leftarrow \text{Backward}(f, \theta)$
- 14: $\mathcal{O}.\text{step}(\tilde{\nabla}_\theta \ell)$ \triangleright update model
- 15: **end for**

$$\Sigma^t = \frac{C}{m} \sum_{j=1}^m \text{sign}(M_j^t), \quad (4)$$

where M_j^t is the j -th method to build the guideline at the t -th epoch. Let $m = 2$, $C = 1$, and Σ^t denote a sample's discriminant output, and we iteratively update σ_f^t via

$$\sigma_f^t \leftarrow \sigma_f^{t-1} + \gamma \Sigma^t, \quad (5)$$

where $\Sigma^t \sim \{-1, -0.5, 0, 0.5, 1\}$, and $\gamma > 0$ is the coefficient. The update of σ_f^t can be regarded as a *random walk* in one-dimensional space driven by mean-zero updates. Let $\sigma_f^t \in [0, \sigma_{max}]$. Then a correct (or incorrect) sample will eventually converge to $\sigma_f^t = 0$ (or $\sigma_f^t = \sigma_{max}$) with high probability, which meets the goal of STGN. Algorithm 1 depicts the complete implementation process. We divide the backpropagation process into two sections (Lines 7, 13). The model first backpropagates from loss ℓ to model output f , obtaining the gradient $\nabla_f \ell$ (Line 7). From Lines 8-12, we inject benign noise into $\nabla_f \ell$, leading to the gradient $\nabla_f \ell$ changing to $\tilde{\nabla}_f \ell$. After that, it continues the backpropagation from noisy gradient $\tilde{\nabla}_f \ell$ to the learnable parameters θ (Line 13).

3.2 Theoretical Analysis

3.2.1 Properties of STGN

We analyze STGN's properties from three different perspectives based on Theorem 1.

Theorem 1. *STGN induces noise $e \in \mathbb{R}^{1 \times p}$ of multivariate uniform distribution with mean-zero and covariance matrix $V \in \mathbb{R}^{p \times p}$, where $V_{i,j} = \frac{(\sigma_f^t)^2}{12} \left(\frac{\nabla_{\theta_i} f}{S} \right)^T \frac{\nabla_{\theta_j} f}{S}$, $\forall i, j \in \{1, \dots, p\}$. $\dot{\cdot}$ denotes the element-wise division. Note that the standard deviation of noise on the i -th parameter θ_i is $\sqrt{V_{i,i}} = \frac{\sigma_f^t}{2\sqrt{3}} \left\| \frac{\nabla_{\theta_i} f}{S} \right\|_2$, where $\|\cdot\|$ denotes the L_2 norm.*

Proof. Suppose $u_f \in \mathbb{R}^{1 \times c}$, $\nabla_\theta \ell \in \mathbb{R}^{1 \times p}$, $\nabla_f \ell \in \mathbb{R}^{1 \times c}$, $\nabla_\theta f \in \mathbb{R}^{c \times p}$, $\nabla_{\theta_i} f \in \mathbb{R}^{c \times 1}$. The noisy gradient is

$$\begin{aligned} \tilde{\nabla}_\theta \ell(f, y) &= \left(\nabla_f \ell(f, y) + \frac{\sigma_f^t}{S} u_f \right) \cdot \nabla_\theta f \\ &= \nabla_\theta \ell(f, y) + \frac{\sigma_f^t}{S} u_f \cdot \nabla_\theta f. \end{aligned} \quad (6)$$

The noise on $\nabla_\theta \ell(f, y)$ is $e = \frac{\sigma_f^t}{S} u_f \cdot \nabla_\theta f \in \mathbb{R}^{1 \times p}$. Note that $u_f \sim \mathcal{U}\left(-\frac{I_{c \times c}}{2}, \frac{I_{c \times c}}{2}\right)$, let e_i be the i -th entry of e , and we have

$$e_i = \sigma_f^t \sum_{k=1}^c \frac{\partial f_k}{\partial \theta_i} \frac{u_{f_k}}{S_k} = \sigma_f^t \sum_{k=1}^c \frac{\nabla_{\theta_i} f_k}{S_k} u_{f_k}. \quad (7)$$

Hence,

$$\mathbb{E}[e_i^2] = \frac{(\sigma_f^t)^2}{12} \left\| \frac{\nabla_{\theta_i} f}{S} \right\|_2^2, \mathbb{E}[e_i e_j] = \frac{(\sigma_f^t)^2}{12} \left(\frac{\nabla_{\theta_i} f}{S} \right)^T \frac{\nabla_{\theta_j} f}{S} \quad (8)$$

□

Perspective 1: Menon et al. (2019) propose not overly trusting any single sample to help mitigate the label noise effect. Besides, Lukasik et al. (2020) evade overconfidence to improve generalization. In Theorem 1, $S \in \mathbb{R}^{c \times 1}$ is in the denominator (element-wise division) of $\sqrt{V_{i,i}}$, bringing the SGD noise introduced by STGN dependent on the confidence of S . When S is confident (i.e., a term in S closes to 1 and others to 0), it will lead to high variance and convergence difficulty since there are small numbers in the denominator. Therefore, we derive Property 1 as follows.

Property 1. *STGN mitigates overfitting on noisy labels by preventing overconfident prediction.*

Perspective 2: Achille and Soatto (2018) prove that compared with sharp minima, flat minima in landscape generalize well. In Theorem 1, $\|\nabla_{\theta_i} f\|_2$

is in the numerator of $\sqrt{V_{i,i}}$. It can be very large around the sharp landscape, which means the SGD noise has high variance. The high variance makes the training hard to converge, which helps escape from sharp minima. Therefore, we derive Property 2 as follows.

Property 2. *The standard deviation of the noise $\sqrt{V_{i,i}} = \frac{\sigma_f^t}{2\sqrt{3}} \left\| \frac{\nabla_{\theta_i} f}{S} \right\|_2$ derived from STGN correlates with the landscape and contributes to generalization.*

Perspective 3: In machine learning, Han et al. (2020) discover optimization shares the goal with generalization at the beginning of training. Sufficient optimization can evade underfitting. However, the objective of optimization will diverge from that of generalization once optimization is well done, as too much optimization can trigger overfitting. In Theorem 1, with the training, zero perturbation (i.e., $\sigma_f^t \rightarrow 0$) on correct data prompts more thorough optimization, while massive disturbance (i.e., $\sigma_f^t \rightarrow \sigma_{max}$) on incorrect data alleviates memorizing noisy labels. The auto-adaptive operations for correct and incorrect samples ensure no divergence between the optimization and generalization goals. Therefore, we derive Property 3 as follows.

Property 3. *STGN exerts large perturbation on incorrect samples while learns correct samples without disturbance, ensuring the goals of optimization and generalization are always the same.*

3.2.2 STGN vs. Existing SGD Noise Variants

To explore the relation between STGN and existing methods, we summarize the following Theorem 2.

Theorem 2. *The existing SGD noise variants are special cases of STGN. Specifically, STGN would degenerate to GNMO if $\sigma_f^t = \sigma_f$ and removing $\frac{\dot{S}}{S}$ is satisfied, and degenerate to SLN when $\sigma_f^t = \sigma_f$ condition is met, where $\sigma_f > 0$ is constant.*

Proof. Suppose $\nabla_{\theta} \ell \in \mathbb{R}^{1 \times p}$, $\nabla_f \ell \in \mathbb{R}^{1 \times c}$, $\nabla_{\theta} f \in \mathbb{R}^{c \times p}$, $\nabla_S \ell \in \mathbb{R}^{1 \times c}$, $\nabla_{\theta} S \in \mathbb{R}^{c \times p}$, $g_f \in \mathbb{R}^{1 \times c}$ and $g_f \sim \mathcal{N}(0, I_{c \times c})$, $g_y \in \mathbb{R}^{1 \times c}$ and $g_y \sim \mathcal{N}(0, I_{c \times c})$, $u_f \in \mathbb{R}^{1 \times c}$ and $u_f \sim \mathcal{U}\left(-\frac{I_{c \times c}}{2}, \frac{I_{c \times c}}{2}\right)$. Due to the mean-zero of the perturbations (i.e., g_f , g_y , and u_f), we focus on analyzing their standard deviations or variances of the perturbation, which are irrelevant to specific distributions. Therefore, suppose we ignore the form of noise distribution in SGD noise.

For GNMO, the noisy gradient is

$$\begin{aligned} \tilde{\nabla}_{\theta} \ell(f, y) &= (\nabla_f \ell(f, y) + \sigma_f g_f) \cdot \nabla_{\theta} f \\ &= \nabla_{\theta} \ell(f, y) + \sigma_f g_f \cdot \nabla_{\theta} f. \end{aligned} \quad (9)$$

The noise on $\nabla_{\theta} \ell(f, y)$ is $e = \sigma_f g_f \cdot \nabla_{\theta} f \in \mathbb{R}^{1 \times p}$. Note that $g_f \sim \mathcal{N}(0, I_{c \times c})$. Let e_i be the i -th entry of e , and we have

$$e_i = \sigma_f \sum_{k=1}^c \frac{\partial f_k}{\partial \theta_i} g_{fk}. \quad (10)$$

Hence,

$$\mathbb{E}[e_i^2] = \sigma_f^2 \|\nabla_{\theta_i} f\|_2^2, \mathbb{E}[e_i e_j] = \sigma_f^2 (\nabla_{\theta_i} f)^T \nabla_{\theta_j} f. \quad (11)$$

For STGN in Eq. (6), where we set $\sigma_f^t = \sigma_f$ and remove $\frac{\dot{S}}{S}$, Eq. (6) is reduced to Eq. (9).

For SLN, the loss function is restricted to cross-entropy loss. $S = S(f(x))$ denotes the softmax function, and the noisy gradient is as follows.

$$\begin{aligned} \tilde{\nabla}_{\theta} \ell(f, y) &= \nabla_{\theta} \ell(f, y + \sigma_y g_y) \\ &= \nabla_S \ell(f, y + \sigma_y g_y) \cdot \nabla_{\theta} S \\ &= -\left(\frac{y + \sigma_y g_y}{S}\right) \cdot \nabla_{\theta} S \\ &= -\left(\frac{y}{S}\right) \cdot \nabla_{\theta} S - \left(\frac{\sigma_y g_y}{S}\right) \cdot \nabla_{\theta} S \\ &= \nabla_{\theta} \ell(f, y) - \frac{\sigma_y}{S} g_y \cdot \nabla_{\theta} S. \end{aligned} \quad (12)$$

The noise on $\nabla_{\theta} \ell(f, y)$ is $e = -\frac{\sigma_y}{S} g_y \cdot \nabla_{\theta} S \in \mathbb{R}^{1 \times p}$. Note that $g_y \sim \mathcal{N}(0, I_{c \times c})$, let e_i be the i -th entry of e , and we have

$$e_i = -\sigma_y \sum_{k=1}^c \frac{\partial S_k}{\partial \theta_i} \frac{g_{yk}}{S_k}. \quad (13)$$

Hence,

$$\mathbb{E}[e_i^2] = \sigma_y^2 \left\| \frac{\nabla_{\theta_i} S}{S} \right\|_2^2, \mathbb{E}[e_i e_j] = \sigma_y^2 \left(\frac{\nabla_{\theta_i} S}{S}\right)^T \frac{\nabla_{\theta_j} S}{S}. \quad (14)$$

For STGN in Eq. (6), where we set $\sigma_f^t = \sigma_f$, STGN is equivalent to SLN (Eq. (6) \rightarrow Eq. (12)). Note that $\nabla_{\theta} f$ and $\nabla_{\theta} S$, playing analogous roles, are not distinguished here. \square

Theorem 2 shows that the STGN method is universal and compatible with existing methods. Formally, Eq. (11) and Eq. (14) are similar to Eq. (8) in Theorem 1. In Eq. (11), $\|\nabla_{\theta_i} f\|_2$ is in the numerator of $\sqrt{V_{i,i}}$, allowing GNMO also satisfies Property 2. Besides, in Eq. (14), $S \in \mathbb{R}^{c \times 1}$ is in the denominator of $\sqrt{V_{i,i}}$, making SLN has Property 1 as well. Note that SLN satisfies this property iff training with cross-entropy loss. However, STGN does not need to restrict the loss function form.

Model	Method	SST-5					
		20%		40%		60%	
		Peak	Average	Peak	Average	Peak	Average
BERT	Adam	51.15±0.24	48.56±0.15	49.37±0.69	41.35±0.60	40.26±2.28	30.61±0.84
	GNMP	50.12±0.11	49.56±0.10	47.56±0.36	46.97±0.33	40.32±0.51	34.84±1.16
	GNMO	49.52±1.82	47.00±0.81	47.09±0.15	43.27±2.16	36.86±1.83	29.13±1.32
	GCE	51.06±0.26	50.05±0.46	49.35±0.18	46.85±0.34	34.52±3.69	29.68±2.21
	SLN	50.32±1.42	49.37±0.44	49.40±0.30	46.71±0.43	41.09±1.09	35.77±1.50
	STGN	51.30±0.67	50.44±0.06	49.52±0.81	47.49±0.83	42.96±1.96	39.10±1.93
	GCE-STGN	51.70±0.50	50.73±0.20	49.72±0.61	48.11±0.65	39.22±2.20	34.98±2.53

Table 2: Peak accuracy (i.e., maximum test accuracy throughout the training) and Average accuracy (i.e., average test accuracy over the last five epochs) on SST-5 under different noise levels. Top-2 results are in bold.

4 Experiments

We aim to build robust NLP learning methods on noisy training sets to achieve well generalization performance on clean test sets. To this end, we experiment on three tasks (i.e., sentiment analysis, NER, and event relation reasoning) to test the effectiveness and generality of STGN. After that, we examine the theoretical analysis experimentally. We further explore the influence of hyperparameters and guidelines in STGN through an ablation study. Due to the limited space, complete experimental settings are in Appendix A.1.

4.1 Sentiment Analysis

We first test the effectiveness of STGN on the Standard Sentiment Treebank (SST-5) dataset (Socher et al., 2013). We follow the previous work (Munikar et al., 2019), adopting BERT as the base model, and compare with five baselines: 1) Adam (Kingma and Ba, 2015): the model optimized with vanilla Adam. 2) GNMP (Neelakantan et al., 2015). 3) GNMO (HaoChen et al., 2021). 4) GCE (Zhang and Sabuncu, 2018): generalized cross-entropy, a general class of noise-robust loss function, encompassing CE (i.e., cross-entropy) and MAE (i.e., mean absolute error). 5) SLN (Chen et al., 2021): the state-of-the-art (SOTA) baseline using implicit regularization.

Table 2 reports peak accuracy and average accuracy under 20%, 40%, 60% noise levels, manifesting the model’s generalization performance from instantaneous peak value and stable mean perspectives. We observe that, STGN consistently outperforms all baselines under diverse noise levels on instantaneous and average test accuracy, which indicates that STGN can effectively mitigate the impact of label noise. We also find that applying an identical base model under higher noise

Method	Metric	Label Set						
		1	2	3	4	5	6	7
Adam	Pre.	63.28	72.04	48.48	73.39	57.54	63.47	70.88
	Rec.	20.05	27.52	36.07	28.88	44.90	43.53	48.32
	F1	30.46	39.82	41.37	41.45	50.44	51.64	57.47
Noise Model	Pre.	69.52	67.85	49.01	62.58	59.64	54.83	60.89
	Rec.	25.60	35.39	37.44	39.63	45.31	50.10	56.26
	F1	37.42	46.51	42.45	49.32	51.50	52.36	58.48
STGN	Pre.	72.09	68.77	48.17	64.01	56.54	60.95	65.05
	Rec.	27.58	36.48	40.52	43.46	50.03	50.10	55.92
	F1	39.90	47.67	44.01	51.77	53.09	55.00	60.14

Table 3: Performance on the test set of NoisyNER measured by precision (Pre.), recall (Rec.) and F1 scores. NoisyNER contains 7 label sets that correspond to 7 different noise levels. Top-1 F1 results are in bold.

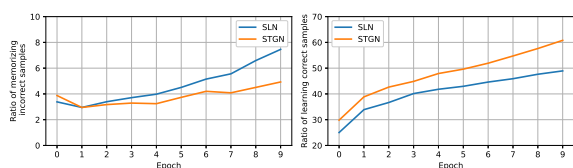
levels (e.g., under 60% label noise), other baselines are more likely to show significant overfitting, i.e., an apparent gap exists between average accuracy and peak accuracy. However, for STGN, the average accuracy is comparable with its peak accuracy, which signifies good generalization under STGN. To further validate the validity of STGN on robust approaches, we impose STGN on top of GCE (GCE-STGN). We find that STGN can further boost existing robust training methods, demonstrating its versatility. Entire optimal hyperparameters settings are provided in Table 6 of Appendix, and more experimental settings and details are in Appendix A.2.

4.2 Named Entity Recognition

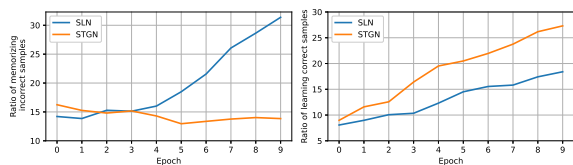
To evaluate the effectiveness of STGN when the label distribution is highly skewed, we experiment on the NoisyNER dataset (Hedderich et al., 2021), a noisy corpus in the real world with labels 3.7% persons (PER), 2.8% locations (LOC), 2.8% organizations (ORG), and 90.8% non-entity (O). Fol-

Model	Method	Step Infer.	Goal Infer.	Step Ordering
	Human	96.5	98.0	97.5
BERT		87.4	79.8	81.9
XLNet	AdamW	86.7	78.3	82.6
GPT-2		83.6	68.6	80.1
	AdamW	88.2	82.0	83.5
RoBERTa	AdamW*	88.21±0.08	79.53±0.23	83.00±0.28
	STGN	89.24±0.35	81.38±0.37	83.60±0.22

Table 4: Test accuracy (mean±std) on wikiHow of different subtasks. * indicates our reimplementation.



(a) Trained with 20% label noise.



(b) Trained with 60% label noise.

Figure 2: Ratios of memorizing incorrect samples (Left, smaller is better) and learning correct samples (Right, larger is better) during training on SST-5.

following Hedderich et al. (2021), we take BiLSTM-FC as the base model and compare STGN with two baselines: 1) Adam (Kingma and Ba, 2015): the baseline in (Hedderich et al., 2021). 2) Noise Model (Hedderich et al., 2021): variable sampling 400 clean samples to estimate the underlying noise process at seven different noise levels provided by NoisyNER. More details are in Appendix A.3.

As shown in Table 3, STGN achieves a significant gain of 9.44, 7.85, 2.64, 10.32, 2.65, 3.36, 2.67 on the F1 score over Adam based on noisy label sets 1 to 7, coming from performance promotion on recall. Recall values in Adam are too low owing to imbalanced class distribution. The results highlight that STGN efficiently enhances capacities of few positive examples recognition, suitable for imbalanced distribution scenario of categories. Besides, although Noise Model relies on additional data annotation, STGN still outperforms it, indicating the validity of STGN in addressing the class imbalance. More details can be found in Appendix A.3.

4.3 Event Relation Reasoning

We further test the versatility of STGN and experiment on another real-world noisy corpus wikiHow (Zhang et al., 2020). It comprises of three subtasks Step Inference, Goal Inference, and Step Ordering, devoted to relation reasoning between procedural events (i.e., GOAL-STEP relations and STEP TEMPORAL relations). As shown in Table 4, an evident gap exists between the model and human performance, demonstrating the difficulty and complexity of the task. STGN achieves consistent performance improvement across all the subtasks compared to AdamW*. More details about wikiHow, all baselines, and complete experimental settings are provided in Appendix A.4.

4.4 Property Validation

We further experimentally validate STGN’s property from two aspects. First, we validate the function of tailor-made noise. Fig. 2 demonstrates ratios of memorizing incorrect samples and learning correct ones of STGN and SLN under different noise levels. We observe more training accuracy comes from correct samples and less derives from incorrect ones of STGN relative to SLN, which further confirms the conclusion in theoretical analysis.

Second, we examine whether STGN can prevent overconfidence. Fig. 3 depicts the output probability between Adam and STGN on SST-5 under diverse noise levels. We reorder the output probability (including five probability values corresponding to five classes in the fine-grained sentiment classification task) in ascending order for each sample and focus on class predictions (x-axis in Fig. 3) for all samples. We observe that the maximum output probability in Adam (i.e., the last column is brighter) is always greater than that in STGN (i.e., the last column is darker). While the other four probability values in Adam (i.e., four-left columns are darker) are always less than those in STGN (i.e., four-left columns are brighter), suggesting that STGN can prevent overconfidence validly.

4.5 Ablation study

We investigate implications of key hyperparameters and guidelines in STGN. All experiments are conducted on the test set of SST-5.

We first explore the effect of the initial value σ_f by changing its value at a specific noise level at a time. In Fig. 4, we obtain stable and good performance concerning both peak accuracy and average

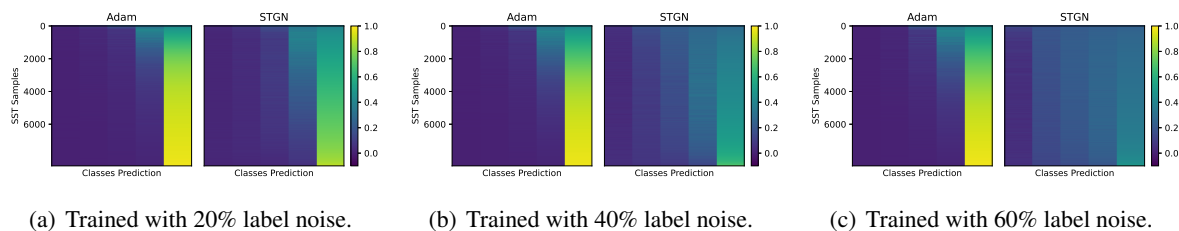


Figure 3: We visualize the output probability of Adam and STGN on SST-5 under different noise levels, where blue denotes smaller values and yellow indicates larger values. Each plot shows the predictive output probability of training samples reordered in ascending order. When given a sample, the x-axis characterizes five probability values, whose sum equals 1, corresponding to five classes in the fine-grained sentiment classification task. The y-axis depicts the number of training samples in SST-5.

Method	SST-5					
	20%		40%		60%	
	Peak	Average	Peak	Average	Peak	Average
STGN	52.22	50.44	50.59	48.32	45.11	39.57
w/o g_1	51.76	49.95	48.87	47.40	41.09	39.28
w/o g_2	50.95	49.95	49.86	47.90	44.75	39.25

Table 5: Peak accuracy and Average accuracy on SST-5 under different noise levels after removing guidelines g_1 or g_2 .

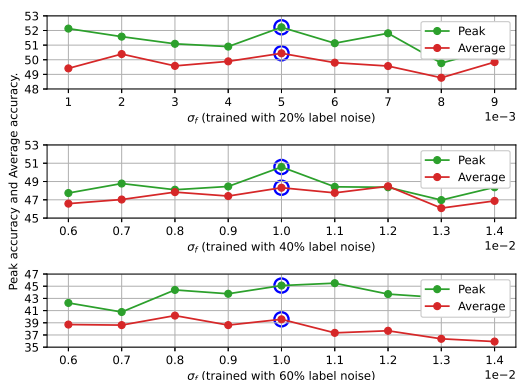


Figure 4: Peak accuracy and Average accuracy w.r.t. σ_f when training on SST-5 under different noise levels. The results in the blue circle are reported in Table 2.

accuracy regardless of different σ_f and noise levels, manifesting the robustness of STGN, whose performance is insensitive to the selection of hyperparameters. Table 5 shows the model performance after removing guidelines g_1 or g_2 . Removing either guideline may entail apparent performance degradation, indicating the validity of g_1 and g_2 . Moreover, it confirms multiple guidelines benefit more precise judgment on correct or incorrect data.

5 Related Work

Noisy Labels in NLP. No matter manual or automatic annotation, labeling errors would inevitably be introduced when labeling large-scale corpora. To address this issue, existing work has studied a series of robust methods to mitigate the impact of noisy labels for NLP tasks. Meng et al. (2021) propose a noise-robust learning step followed by a self-training step to train a robust NER model, utilizing pretrained language models to improve model generalization. Jindal et al. (2019) introduce a non-linear processing layer on top of the CNN model to solve labeling errors using extra clean data in text classification tasks. In entity linking, Le and Titov (2019) apply a noise detection component to disregard noisy samples. Jia et al. (2019) propose an attention regularization-based method to combat label noise in relation classification. However, these approaches are developed specifically for a given task or model architecture, and hardly generalize to other tasks. Up to now, few works have concentrated on all-purpose methods to tackle label noise in NLP. Zhou and Chen (2021) realize denoising by developing a general co-regularization framework, which optimizes several models jointly with identical structures but diverse parameters, and thus occupies massive computing resources. In contrast, STGN is versatile and suitable for many NLP tasks without introducing much overhead or model complexity.

Implicit Regularization. Although DNNs involve far more learnable parameters than training samples, DNNs still have good generalization performance even without explicit regularization. It prompts researchers to investigate implicit regularization, whose regularization effect comes from model optimization. Among many factors of im-

licit regularization (like batch size and learning rate (Smith et al., 2018)), SGD noise is viewed as a critical source (Neelakantan et al., 2015; HaoChen et al., 2021; Chen et al., 2021). Chen et al. (2021) first exploit SGD noise to combat sample mislabeling and propose an SGD variant SLN. However, SLN relies on a specific loss function and may cause side effects of underfitting when mitigating overfitting. In this paper, STGN (also a variant of SGD noise) is not limited to given loss, and exerts tailor-made noise for correct samples and incorrect ones, which fulfills the consistent targets between optimization and generalization.

6 Conclusion

In this paper, STGN exerts different benign noise intensities on correct and incorrect samples, ensuring the goals of optimization and generalization are always the same to resist inherent label noise. We empirically validate the universality and effectiveness of STGN under synthetic label noise and real-world label noise, demonstrating that STGN can be applied to many NLP tasks without dedicated framework or excessive clean corpus.

Limitations

We combine two guidelines (g_1 and g_2) to facilitate correct and incorrect sample discrimination. Nevertheless, unmistakably filtering out data with label noise remains challenging. Since our method supports the extension of more guidelines (Eq. (4)), we plan to investigate more efficient guidelines from different perspectives in future work to achieve a more accurate recognition of noise labels.

Acknowledgments

We thank the anonymous reviewers for their constructive comments, and gratefully acknowledge the support of the Technological Innovation “2030 Megaproject” - New Generation Artificial Intelligence of China (2020AAA0106501), and the National Natural Science Foundation of China (62176079, 61976073, 62106061).

References

Alessandro Achille and Stefano Soatto. 2018. Emergence of invariance and disentanglement in deep representations. *The Journal of Machine Learning Research*, 19(1):1947–1980.

Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. 2020. Tacred revisited: A thorough evaluation of the tacred relation extraction task. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1558–1569.

Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. 2019. Implicit regularization in deep matrix factorization. *Advances in Neural Information Processing Systems*, 32.

Piotr Bojanowski, Édouard Grave, Armand Joulin, and Tomáš Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Pengfei Chen, Guangyong Chen, Junjie Ye, Pheng-Ann Heng, et al. 2021. Noise against noise: stochastic label noise helps combat inherent label noise. In *International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*.

Bo Han, Gang Niu, Xingrui Yu, Quanming Yao, Miao Xu, Ivor Tsang, and Masashi Sugiyama. 2020. Sigua: Forgetting may make learning with noisy labels more robust. In *International Conference on Machine Learning*, pages 4006–4016. PMLR.

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

Jeff Z HaoChen, Colin Wei, Jason Lee, and Tengyu Ma. 2021. Shape matters: Understanding the implicit bias of the noise covariance. In *Conference on Learning Theory*, pages 2315–2357. PMLR.

Michael A Hedderich, Dawei Zhu, and Dietrich Klakow. 2021. Analysing the noise model error for realistic noisy label data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7675–7684.

Wei Jia, Dai Dai, Xinyan Xiao, and Hua Wu. 2019. Arnor: Attention regularization based noise reduction for distant supervision relation classification. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 1399–1408.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. 2018. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International conference on machine learning*, pages 2304–2313. PMLR.

Ishan Jindal, Daniel Pressel, Brian Lester, and Matthew Noleby. 2019. An effective label noise model for dnn text classification. 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)*, pages 3246–3256.
- Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. 2017. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*.
- Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.
- Phong Le and Ivan Titov. 2019. Distant learning for entity linking with automatic noise detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4081–4090.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv: Computation and Language*.
- Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.
- Michal Lukasik, Srinadh Bhojanapalli, Aditya Menon, and Sanjiv Kumar. 2020. Does label smoothing mitigate label noise? In *International Conference on Machine Learning*, pages 6448–6458. PMLR.
- Yu Meng, Yunyi Zhang, Jiaxin Huang, Xuan Wang, Yu Zhang, Heng Ji, and Jiawei Han. 2021. Distantly-supervised named entity recognition with noise-robust learning and language model augmented self-training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10367–10378.
- Aditya Krishna Menon, Ankit Singh Rawat, Sashank J Reddi, and Sanjiv Kumar. 2019. Can gradient clipping mitigate label noise? In *International Conference on Learning Representations*.
- Manish Munikar, Sushil Shakya, and Aakash Shrestha. 2019. Fine-grained sentiment classification using bert. In *2019 Artificial Intelligence for Transforming Business and Society (AITB)*, volume 1, pages 1–5. IEEE.
- Arvind Neelakantan, Luke Vilnis, Quoc V Le, Ilya Sutskever, Lukasz Kaiser, Karol Kurach, and James Martens. 2015. Adding gradient noise improves learning for very deep networks. *arXiv preprint arXiv:1511.06807*.
- Bo Pang, Lillian Lee, et al. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in information retrieval*, 2(1–2):1–135.
- Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. 2017. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1944–1952.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Shai Shalev-Shwartz and Shai Ben-David. 2014. *Understanding machine learning: From theory to algorithms*. Cambridge university press.
- Samuel L Smith, Pieter-Jan Kindermans, Chris Ying, and Quoc V Le. 2018. Don’t decay the learning rate, increase the batch size. In *ICLR (Poster)*.
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.
- Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. 2019. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Neural Information Processing Systems*.
- Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2017. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*.
- Li Zhang, Qing Lyu, and Chris Callison-Burch. 2020. Reasoning about goals, steps, and temporal ordering with wikihow. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4630–4639.
- Zhilu Zhang and Mert Sabuncu. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31.
- Wenxuan Zhou and Muhao Chen. 2021. Learning from noisy labels for entity-centric information extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5381–5392.

A Appendix

A.1 Experimental Settings

We focus on NLP classification tasks and adopt cross-entropy as the loss function. Regarding evaluation metrics, we apply accuracy in sentiment analysis and event relation reasoning, and F1 score in named entity recognition (NER).

We conduct experiments on SST-5 dataset (with manually corrupted labels) and two real-world noisy corpora: NoisyNER and wikiHow. The former facilitates inspecting the specific efficacy of STGN under different noise levels. We manually inject symmetric label noise into the corpus, similar to previous works (Jindal et al., 2019; Chen et al., 2021). The latter is more practical and complicated. Moreover, we select *diversified model architectures* (e.g., BiLSTM, pretrained models) to be base models and *varied SGD variants* (e.g., Adam (Kingma and Ba, 2015), AdamW (Loshchilov and Hutter, 2018)) to be baselines to justify the universality of STGN.

During training, we initialize σ_f^t as σ_f and empirically set $\sigma_{max} = 2\sigma_f$, $\gamma = 0.1\sigma_f$ to keep down the number of hyperparameters. Although we allocate each instance tailor-made perturbation, STGN is only involved in training. At inference, we take SGD or its variants as an alternative and hence it has no effect on model complexity.

Method	Optimal Hyperparameters Settings		
	20%	40%	60%
STGN	$\sigma_f=5e-3,$ $\lambda=2$	$\sigma_f=1e-2,$ $\lambda=1$	$\sigma_f=1e-2,$ $\lambda=3$
GCE-STGN	$\sigma_f=5e-3,$ $\lambda=2$	$\sigma_f=1e-2,$ $\lambda=2$	$\sigma_f=5e-3,$ $\lambda=2$

Table 6: Optimal hyperparameters settings on SST-5 under different noise levels, where STGN or GCE-STGN denote integrating benign noise induced by STGN into Adam trained with CE or GCE loss function.

A.2 Sentiment Analysis

Task and Dataset. Sentiment analysis is a significant task in NLP. The objective of this task is to predict the polarity of subjective information from a given source (Pang et al., 2008). This task is always evaluated on the Stanford Sentiment Treebank (SST-5) (Socher et al., 2013). SST-5 is a fine-grained sentiment classification dataset, including very negative, negative, neutral, positive, and very

Label Set	1	2	3	4	5	6	7
Precision	67	73	37	75	48	53	59
Recall	18	27	31	27	41	41	49
F1	28	39	34	40	44	46	54

Table 7: Percentages of correct labels in different label sets of NoisyNER. Owing to severely skewed label distribution, precision, recall, and F1 score are reported.

positive labels, which has labels for 11,855 sentences extracted from movie reviews. It comprises 8,544/1,101/2,210 samples as training, validation, and test set, respectively. We evaluate the proposed method with manually corrupted labels. The noise levels are set to 20%, 40% and 60%, respectively.

Experimental Settings. We follow the experiment settings in Munikar et al. (2019) and finetune the pretrained model BERT of the base-sized version for 10 epochs using Adam with batchsize 32, learning rate $1e-5$. Entire optimal hyperparameters settings are provided in Table 6.

A.3 Named Entity Recognition

Task and Dataset. Named entity recognition (NER), which aims at detecting real-world entity mentions from texts, is a fundamental task in NLP. To evaluate STGN, we experiment on a real-world noisy dataset NoisyNER (Hedderich et al., 2021), which uses varying amounts of heuristics during the automatic annotation process and provides seven sets of noisy labels. NoisyNER, whose labels involve persons, locations, and organizations, depicts the NER task based on low-resource Estonian. We divide the data into 80/10/10 ratios as training, validation, and test set. Moreover, NoisyNER contains clean labels, including persons, locations, and organizations, annotated by experts. The percentage of correct labels in different label sets of NoisyNER are indicated in Table 7.

Experimental Settings. In the main paper, we take BiLSTM-FC as the base model, comprising a BiLSTM model, a fully connected layer, and a softmax classification layer. We follow Hedderich et al. (2021) and train a BiLSTM-FC model for 80 epochs using Adam with a learning rate $1e-3$, state size 300 for each direction of BiLSTM, size 100 for fully connected layer, fixed FastText embeddings (Bojanowski et al., 2017) to embed tokens. We evaluate with the micro-average F1 score. Entire optimal hyperparameters settings are provided in Table 8.

Method	Label Set						
	1	2	3	4	5	6	7
STGN	$\sigma_f=5e-4,$ $\lambda=10,$ $\epsilon=0.35$	$\sigma_f=5e-4,$ $\lambda=10,$ $\epsilon=0.3$	$\sigma_f=5e-4,$ $\lambda=10,$ $\epsilon=0.3$	$\sigma_f=5e-4,$ $\lambda=10,$ $\epsilon=0.3$	$\sigma_f=5e-4,$ $\lambda=7,$ $\epsilon=0.3$	$\sigma_f=1e-3,$ $\lambda=7,$ $\epsilon=0.3$	$\sigma_f=5e-4,$ $\lambda=10,$ $\epsilon=0.25$

Table 8: Optimal hyperparameters settings under label sets 1 to 7 on NoisyNER.

	Train Size	Val Size	Test Size
Step Infer.	336,851	37,427	2,250
Goal Infer.	116,708	18,523	1,703
Step Ordering	752,516	83,612	3,100

Table 9: Statistics of wikiHow dataset.

A.4 Event Relation Reasoning

Task and Dataset. Procedural event relation reasoning can be considered as a classification task in NLP. This reasoning task can be further partitioned into three subtasks, i.e., reasoning on GOAL-STEP relation, STEP-GOAL relation, and STEP TEMPORAL relation, respectively. For instance, “learn poses” is a step in the larger goal of “doing yoga” and “buy a yoga mat” typically precedes “learn poses” (Zhang et al., 2020). The training set for each subtask is crawled from the wikiHow website² and is automatically generated, which brings in noisy labels unavoidably. The original wikiHow dataset only comprises the training set and the test set. We randomly divide a part of the examples from the training set as the validation set, with a ratio of 9:1. The statistics of wikiHow are indicated in Table 9.

Experimental Settings. We reproduce the strongest baseline with the implementation by Zhang et al. (2020).³ We finetune the RoBERT model of the base-sized version using AdamW for 3 epochs (for Step Infer. subtask), 2 epochs (for Step Ordering subtask), and 5 epochs (for Goal Infer. subtask) with learning rate $5e-5$, weight decay 0. Because of the large scale of the dataset, we set the batchsize as 48 and define 1000 steps as an iteration. Besides, we set $\sigma=5e-4$, $\lambda=0$, $\epsilon=0.15$ for three subtasks.

Baselines. We compare STGN with the following baselines: (1) Human (Zhang et al., 2020), which reports the human performance on these subtasks. There is a gap of about 10% to 20% between other

baselines and humans, which indicates the task is challenging. (2) BERT (Devlin et al., 2019), which finetunes pretrained BERT model on the training set and reports test accuracy. (3) XLNet (Yang et al., 2019), which finetunes pretrained XLNet model and others are the same as BERT. (4) GPT-2 (Radford et al., 2019), which is identical to BERT besides finetuning pretrained GPT-2 model. (5) RoBERTa (Liu et al., 2019), which finetunes pretrained RoBERTa model, and the remainder is similar to BERT.

²<https://www.wikihow.com/>

³<https://github.com/zharry29/wikihow-goal-step>