RESEARCH ARTICLE

Effects of Adversarial Training of Word Embeddings on Chinese Agricultural Pest and Disease Named Entity Recognition with BERT-CRF

Bo Cheng[1,2], **Ziming Gao**[1,2,5], **Biao An**[1,2,5] **and Yuansheng Wang**[1,2,3,4,*]

[1] National Engineering Research Center for Information Technology in Agriculture, Beijing 100097, China
[2] Information Technology Research Center, Beijing Academy of Agriculture and Forestry Sciences, Beijing 100097, China
[3] Key Laboratory of Digital Village Technology, Ministry of Agriculture and Rural Affairs, Beijing 100097, China
[4] Beijing Research Institute of Agricultural Artificial Intelligence and Robotics, Beijing 100097, China
[5] College of Computer Science and Technology, Inner Mongolia Minzu University, Tongliao 028000, China

Abstract

Entity boundaries, confusable entity types and surface forms absent from the training data all make it harder to extract structured information from agricultural pest and disease texts. We examined whether single-step adversarial training of word embeddings with the fast gradient method (FGM) improves entity recognition when the BERT-CRF inference architecture is left unchanged. On the official 13-type AgCNER corpus, with exact matching of character span and entity type, we compared a feature-based CRF, BiLSTM-CRF, BERT-CRF and BERT-CRF+FGM, training each neural model with three fixed random seeds. BERT-CRF+FGM reached mean test precision, recall and F1 of 94.37%, 95.10% and 94.73%. Its F1 was 0.28 percentage points higher than that of the matched BERT-CRF baseline, and the paired difference was positive for every seed. F1 on unseen surface forms rose from 74.87% to 76.41%, and the overall gain remained 0.27 percentage points after test texts duplicated in other splits were removed. Mean F1 increased for 11 of the 13 types but fell for Company and Organization. FGM did not change the parameter count. The seed-42 model produced 20,670 indexed entity occurrences that can be checked against the released records and character spans. The official test set had been used in an earlier baseline study and test results informed model adoption, so these findings are exploratory; independently sourced data are needed to assess how far they generalize.

Keywords: Agricultural pests and diseases, named entity recognition, BERT-CRF, fast gradient method, unseen surface forms, entity occurrence index.

Submitted: 11 July 2026
Revised: 12 October 2026
Accepted: 20 October 2026
Published: 24 October 2026

Vol. 3, **No.** 1, 2027.
10.62762/TMI.2026.748742

***Corresponding author:**
✉ Yuansheng Wang
642634129@qq.com

Citation
Cheng, B., Gao, Z., An, B., & Wang, Y. (2026). Effects of Adversarial Training of Word Embeddings on Chinese Agricultural Pest and Disease Named Entity Recognition with BERT-CRF. *ICCK Transactions on Machine Intelligence*, 3(1), 1–18.

© 2026 by the Authors. Published by Institute of Central Computation and Knowledge. This is an open access article under the CC BY license (https://creativecommons.org/licenses/by/4.0/).

1 Introduction

Pest and disease descriptions, field experiment reports and control recommendations record which crops are affected, which pathogens or pests are involved, which plant parts show symptoms and which agents are applied. Because this information is buried in free text, it cannot be searched or cross-checked by entity. Topic classification can tell what a text is about, but finding the crops, diseases and agents it mentions requires locating each entity and assigning its type. Linking those entities to their positions in the text is the first step towards turning such material into searchable records.

Named entity recognition (NER) locates text spans that belong to predefined semantic types and underpins information extraction from specialist texts [1, 2]. Agricultural texts contain closely related terms for crops and cultivars, diseases and pests, and pathogens and treatment agents, and these names may embed identifiers, concentrations and formulations. Chinese text has no spaces between words, so boundary errors and type confusion compound each other. Whether a cultivar code belongs to the cultivar name, or whether a concentration and formulation belong to a drug entity, is decided by the annotation guidelines; a model must therefore exploit context while reproducing the annotated boundaries. AgCNER, released by Yao et al. [3], is a Chinese agricultural pest and disease corpus annotated with 13 entity types that allows such models to be compared under the same conditions.

Most agricultural NER systems are built on sequence labelling. A conditional random field (CRF) scores whole label sequences with global normalization and combines local features with label-transition information [4]. A bidirectional long short-term memory network (BiLSTM) placed before the CRF learns contextual features and reduces the need for hand-crafted ones [5]; character-level convolutions and related encoders have also been used [6], and adversarial contextual embeddings have been explored for this task [7]. Representations learned only from task data remain limited when domain vocabulary is sparse and entity types are imbalanced, so model comparisons should keep the representation source and the decoder apart.

The Transformer models token interactions through self-attention, and BERT (bidirectional encoder representations from Transformers) provides pretrained contextual representations [8, 9]. Chinese pretraining work has adapted the training objectives to properties of the Chinese language [10]. Chinese NER models such as Lattice LSTM, FLAT and MECT add lexical, positional or character-structure information [11–13]. These models change the input or the encoder. For an existing BERT-CRF system, a different question arises: can a change in training alone give a consistent gain while the inference architecture stays the same?

In agricultural NER, Guo et al. [14] combined multiscale local context features with self-attention. Tang et al. [15] used a gated fusion unit and Manhattan attention, added FGM adversarial training and evaluated their model on AgCNER, so FGM has already been applied to this corpus. Because their encoder, additional modules and training settings differ from ours, the contribution of FGM itself cannot be isolated from their reported scores. The incremental effect of FGM, and how it varies across entity types and surface forms, still has to be measured with the same corpus, scoring protocol and matched configurations.

Adversarial training adds norm-bounded perturbations in a continuous representation space, so that the training objective covers both the original representation and its neighbourhood. Gradient-based adversarial perturbations and adversarial training of text embeddings are well established [16, 17]. Single-step FGM perturbs the word-embedding parameters only during training and leaves the inference model unchanged. FreeLB uses several adversarial ascent steps when fine-tuning pretrained language models [18], SMART combines smoothness-inducing regularization with proximal optimization [19], and Ghaddar et al. [20] showed that NER models can exhibit name regularity bias, and that adversarial training sensitive to entity context can reduce it; whether such bias is present in agricultural corpora and whether single-step FGM mitigates it remains an open question. These studies motivate the training design, but the effect of FGM on an agricultural corpus has to be measured directly, and a higher F1 on clean test data is not by itself evidence of robustness to input attacks.

How a small gain is interpreted depends on the evaluation design. Sequence-labelling scores vary with random initialization and other sources of training randomness, and reporting score distributions over several runs makes this variation visible [21]. Transparent reporting of model selection, validation performance and computation budget also aids

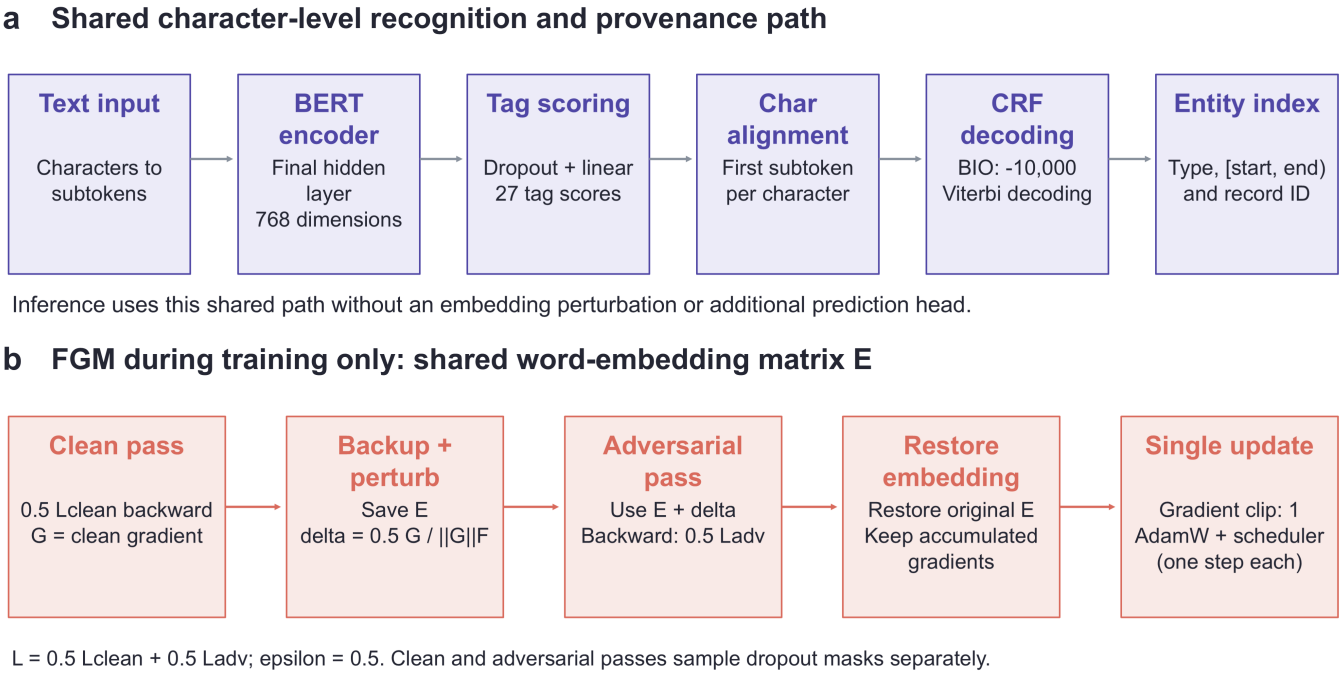

Figure 1. Shared recognition path and FGM training procedure. (a) Subtoken encoding, BERT representation, subtoken-level tag scoring, first-subtoken alignment, CRF decoding and record-level entity indexing. The BIO transition penalty is finite ($-10{,}000$) rather than a hard prohibition. (b) A single perturbation of the shared word-embedding matrix during training. Clean and adversarial CRF losses each have a weight of 0.5, and $\varepsilon = 0.5$. The original embedding is restored before gradient clipping and a single optimizer update. Dropout masks are sampled separately in the two passes. FGM adds no prediction head to the shared inference path.

comparison [22]. Aggregate scores can hide differences arising from unequal type frequencies, from surface forms seen or unseen in training, and from texts duplicated across splits; documentation frameworks for datasets likewise emphasize corpus composition, provenance and limits of use [23]. Agricultural corpus studies further show that encoder choice and adversarial training interact with entity type distribution and surface-form coverage [24]. We therefore asked three questions: Is the direction of the overall gain consistent across matched seeds? Which entity types and surface forms account for it? What training and inference costs come with it?

We used BERT-CRF as a matched baseline to evaluate single-step FGM training of word embeddings on the official AgCNER splits. With a common character-span scoring protocol and three fixed seeds, we compared overall performance and all 13 entity types, and we used analyses of seen and unseen surface forms, exact duplicate texts and computational cost to delimit the observed gain. Predicted entities were also linked to released records and character spans so that every output can be checked. FGM, BERT and CRF are all established methods; our contribution is a matched evaluation of an existing training method together with analyses of the conditions under which it helps.

2 Materials and Methods

The workflow consists of parsing the AgCNER corpus, character-aligned BERT-CRF recognition, FGM training of the word embeddings and record-based checking of predicted entity occurrences (Figure 1).

2.1 Dataset and Annotation Scheme

2.1.1 Corpus Source and Data Splits

We used the official AgCNER Chinese agricultural pest and disease corpus released on Figshare [3]. The training, development and test sets correspond to release files 42188205, 42188211 and 42188208 and keep the official splits. In the original files, blank lines separate sequences and each nonempty line holds one character and its BIO label. Nonempty sequences were parsed in release order and their characters concatenated to rebuild the evaluation texts. Record IDs combine the split name with a zero-based six-digit sequence number, e.g. `test:000000`. The training, development and test sets contain 47,839, 5,963 and 5,974 records with 165,686, 20,782 and 20,524 gold entity occurrences, respectively (Table 1).

Table 1. Data splits and annotation counts.

Split	Records	Char. labels	Gold entities	Max. chars
Training	47,839	3,127,977	165,686	435
Development	5,963	394,843	20,782	380
Test	5,974	386,473	20,524	372
Filtered test	5,849	–	20,119	–

Note: Records are nonempty sequences. The filtered test subset excludes 125 test records whose complete texts also occur in the training or development set; it is used only for sensitivity analysis, without retraining. All splits contain 13 entity types.

The original data paper reports 47,840, 5,964 and 5,975 sentences in the three splits; parsing the nonempty sequences in the released files gave one record fewer in each. We report the records actually read by the models and keep the release-file identities and checksums. No formatting errors or illegal BIO transitions were found. Training records average 65.39 characters (maximum 435). The longest test record has 372 characters, or 374 subtokens with BERT special tokens. All records encode to at most 512 subtokens, so neither truncation nor sliding windows were needed.

2.1.2 *Entity Types and Character Annotations*

AgCNER defines 13 entity types: Biosystematic, Company, Crop, Cultivar, Disease, Drug, Fertilizer, Organization, Other, Pathogens, Part, Period and Pest. Table 2 lists their codes and gold counts in each split. Under BIO tagging, B-t marks the first character of an entity of type t, I-t its remaining characters and O a nonentity character, giving $2 \times 13 + 1 = 27$ labels. A single-layer BIO sequence can hold several entity types per record; nested entities are outside its scope.

Pest, Crop, Disease, Drug and Period make up most of the annotations in all three splits (Figure 2). The training set contains 36,758 Pest and 549 Company entities, a ratio of about 66.95. Nonentity O labels account for 80.68% of the training character labels, and Pest, Crop and Disease together account for 55.72% of the gold test entities. Character accuracy and micro-averaged entity scores therefore say little about individual types, so we report counts and F1 for all 13 types under the official scheme.

2.1.3 *Input Preparation and Duplicate Text Analysis*

Input preparation preserves the released characters and their label positions. Concentrations, identifiers, punctuation and special symbols are kept, and no surface form is rewritten with a hand-built dictionary. Character offsets refer to the concatenated released text, and entity spans are zero-based half-open intervals $[\mathrm{start}, \mathrm{end})$. Unicode NFC normalization is applied

Table 2. Entity types and gold entity counts by split.

Code	Entity type	Training	Dev.	Test
BIS	Biosystematic	1,480	196	165
COM	Company	549	64	96
CRO	Crop	33,788	4,323	4,074
CUL	Cultivar	9,815	1,117	1,200
DIS	Disease	22,291	2,749	2,844
DRUG	Drug	19,689	2,678	2,633
FER	Fertilizer	864	78	108
ORG	Organization	2,346	313	316
OTH	Other	1,892	236	224
PAOG	Pathogens	3,779	456	428
PART	Part	14,400	1,767	1,699
PER	Period	18,035	2,181	2,219
PET	Pest	36,758	4,624	4,518

Note: Counts are gold entity occurrences; the same surface form in different records is counted separately. Type definitions follow the AgCNER data paper [3]. Column totals equal the gold-entity counts in Table 1.

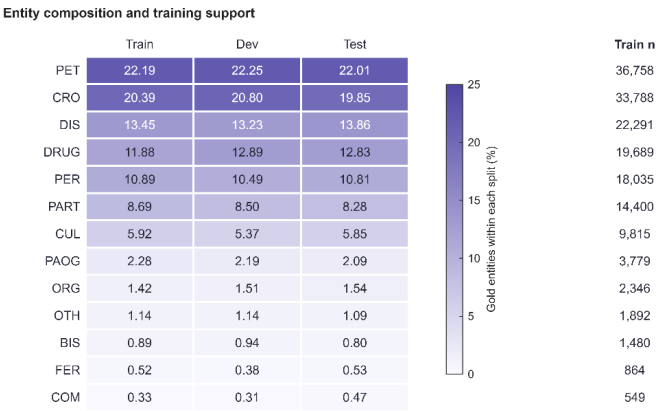

Figure 2. Entity composition across the official splits. Each cell gives the percentage of gold entities of a type within that split. Types are ordered by descending training count, and the side column gives training counts. Type codes follow Table 2. The distribution describes annotation composition, not the frequency of disease or pest occurrence in the field.

only when comparing sets of seen and unseen surface forms; it does not alter model inputs or spans.

An exact-text check found 88 development records (87 distinct texts) that also occur in the training set, and 125 test records (112 distinct texts) that match training or development texts. The main evaluation keeps the official test set. For a sensitivity analysis, the 125 test records were removed by matching SHA-256 checksums of their texts, leaving 5,849 records and 20,119 gold entities. The removal does not depend on model output, and no retraining was done. The

check covers identical complete texts only, not near duplicates, paraphrases or overlapping excerpts from the same source document.

2.2 Entity Recognition Workflow for Agricultural Pest and Disease Texts

BERT inputs are built from the released character sequences. Final-layer contextual representations pass through Dropout and a linear projection to give subtoken tag scores; character-level emission scores are taken at the first subtoken of each character, and the CRF decodes the sequence with BIO transition penalties. FGM adds one adversarial forward and backward pass during training, while inference uses the unchanged BERT-CRF. Decoded entities are linked to record IDs, text fragments and character spans, giving an automatically generated index that points back to the released text.

2.2.1 Character-to-Subtoken Alignment

Entities are scored on the original character spans, but the number of BERT subtokens can differ from the number of characters. The implementation passes the character list as pretokenized input and keeps, for each character, its subtoken group and first position $m(i)$. If tokenizer normalization leaves a character without a subtoken, `[UNK]` is inserted so that the character still receives a prediction. BERT special tokens take part in encoding but are not returned as characters. Padding is excluded from encoding by the attention mask and from decoding by the true character lengths, so it never contributes to entity counts.

Let $\mathbf{h}_j \in \mathbb{R}^{768}$ be the final-layer representation of subtoken j, and let W and $\mathbf{b}$ be the weight and bias of the linear emission layer. Dropout and the projection give the subtoken emission vector $\mathbf{z}_j$, and the emission vector of character i is taken at its first subtoken $m(i)$:

$$\mathbf{z}_j = W\,\mathrm{Dropout}(\mathbf{h}_j) + \mathbf{b}, \qquad \mathbf{z}_i^{\mathrm{c}} = \mathbf{z}_{m(i)}. \tag{1}$$

Both $\mathbf{z}_j$ and $\mathbf{z}_i^{\mathrm{c}}$ lie in $\mathbb{R}^{27}$, where the superscript c marks character-level scores; $W \in \mathbb{R}^{27\times 768}$ and $\mathbf{b} \in \mathbb{R}^{27}$. When a character maps to several subtokens, only the first-subtoken vector is used and its BIO label is not copied to the other subtokens. Gold annotations, decoded outputs and indexed occurrences therefore share the same character coordinates.

2.2.2 BERT Contextual Representations

The encoder is `google-bert/bert-base-chinese` pinned to revision `8f23c25b06e129b6c986331a13d8d025a92cf0ea`, with 12 Transformer layers, 12 attention heads and 768-dimensional hidden states. Every main run reinitializes the task model from the same pretrained weights and fine-tunes the encoder, the emission layer and the CRF jointly. Both BERT configurations use the final-layer representations and the same inference architecture; they differ only in whether the two-branch FGM scheme is switched on during training.

2.2.3 CRF Sequence Decoding

The CRF models dependencies between adjacent tags and normalizes over complete tag paths. Let $\mathcal{K}$ be the set of 27 BIO labels, let x contain $n \geq 1$ characters and let $y = (y_1, \ldots, y_n) \in \mathcal{K}^n$ be a tag path. Let E be BERT's shared word-embedding matrix and θ all other model parameters; the dependence of the path score $S(x, y)$ on θ and E is left implicit. The score sums emission, transition and start/end terms:

$$S(x,y) = \tilde{a}_{\mathrm{start},y_1} + \sum_{i=1}^{n} z_{i,y_i}^{\mathrm{c}} + \sum_{i=2}^{n} \tilde{A}_{y_{i-1},y_i} + a_{\mathrm{end},y_n}. \tag{2}$$

Here $A \in \mathbb{R}^{27\times 27}$, $a_{\mathrm{start}} \in \mathbb{R}^{27}$ and $a_{\mathrm{end}} \in \mathbb{R}^{27}$ are trainable CRF parameters, and $z_{i,k}^{\mathrm{c}}$ is the emission score of label k at character i. The effective scores $\tilde{A}$ and $\tilde{a}_{\mathrm{start}}$ equal the learned values except that a sentence-initial I-tag, or a transition into I-t from any tag other than B-t or I-t, is set to $-10{,}000$. This finite penalty discourages illegal BIO paths, while normalization still runs over all of $\mathcal{K}^n$; all gold paths are BIO-valid. Let the mini-batch $\mathcal{B}$ contain $|\mathcal{B}|$ records with input-tag pairs $(x^{(r)}, y^{(r)})$, $r = 1, \ldots, |\mathcal{B}|$. The loss $\mathcal{L}(\theta, E; \mathcal{B})$ is the mean sequence negative log-likelihood, not divided by sequence length. The clean loss is

$$\begin{aligned} p_{\theta,E}(y \mid x) &= \frac{\exp S(x,y)}{\sum_{y' \in \mathcal{K}^n} \exp S(x,y')}, \\ L_{\mathrm{c}} &= \mathcal{L}(\theta, E; \mathcal{B}) \\ &= -\frac{1}{|\mathcal{B}|} \sum_{r=1}^{|\mathcal{B}|} \log p_{\theta,E}\big(y^{(r)} \mid x^{(r)}\big). \end{aligned} \tag{3}$$

The normalizer is computed by log-sum-exp recursion, with length masking so that padding cannot extend a path. At prediction time, the Viterbi algorithm returns the highest-scoring tag sequence, from which entity spans are rebuilt by the BIO rules. A valid tag sequence does not, of course, guarantee correct boundaries or types.

2.3 Single-Step Fast Gradient Adversarial Training of Word Embeddings

2.3.1 Perturbation Construction and Loss Combination

FGM perturbs the word-embedding parameters along the gradient of the current batch's clean loss, and training combines the sequence losses under the original and perturbed embeddings [16, 17]. For a finite gradient matrix G and $\varepsilon \geq 0$, the largest linearized increase over a Frobenius-norm ball is

$$\max_{\|\delta\|_F \leq \varepsilon} \langle G, \delta \rangle_F = \varepsilon \|G\|_F. \tag{4}$$

Here G is the gradient of the weighted clean loss with respect to E (Eq. (5)), and δ has the shape of E. The Frobenius inner product is $\langle G, \delta \rangle_F = \operatorname{tr}(G^{\mathsf{T}}\delta)$, and $\|\cdot\|_F$ is the Euclidean norm of the flattened matrix. Equation (4) maximizes the linearized increment exactly (by the Cauchy-Schwarz inequality) but does not solve the full nonlinear inner maximization. After $0.5L_{\mathrm{c}}$ is backpropagated, the single-step perturbation is

$$G = \nabla_E\big(0.5L_{\mathrm{c}}\big), \qquad \varepsilon = 0.5,$$
$$\delta = \begin{cases} \varepsilon\, G/\|G\|_F, & 0 < \|G\|_F < \infty, \\ \mathbf{0}, & \text{otherwise.} \end{cases} \tag{5}$$

Scaling the gradient by the positive factor 0.5 leaves its normalized direction unchanged. The perturbation acts only on the shared embedding matrix, not on the input text, so identical token IDs share the same perturbed row. E always denotes the unperturbed matrix and E_{adv} the perturbed one (Eq. (6)). The implementation modifies the parameter tensor in place and later restores the saved value. Once built, δ is held fixed during adversarial backpropagation, and no gradient flows through its construction.

For the same batch $\mathcal{B}$, the clean and adversarial losses are evaluated with E and E_{adv}, sharing all other parameters θ, and are combined with equal weights:

$$E_{\mathrm{adv}} = E + \delta, \qquad L_{\mathrm{a}} = \mathcal{L}(\theta, E_{\mathrm{adv}}; \mathcal{B}),$$
$$L_{\mathrm{total}} = 0.5L_{\mathrm{c}} + 0.5L_{\mathrm{a}}. \tag{6}$$

The two branches share the encoder, emission layer, CRF and length masks. Both forward passes run in training mode with independently sampled Dropout masks, which are omitted from the notation.

2.3.2 Training Steps and Parameter Restoration

Each batch receives one optimizer update. After the gradients are zeroed, the clean forward pass is followed by backpropagation of $0.5L_{\mathrm{c}}$. The embedding tensor is saved and perturbed according to Eq. (5). The adversarial forward pass evaluates $0.5L_{\mathrm{a}}$ at E_{adv}, and its backward pass adds to the accumulated gradients with δ held fixed. The saved embedding is then restored without clearing the gradients, after which gradient clipping, one AdamW update and one learning-rate scheduler step are applied to the unperturbed parameters.

At inference, Dropout is disabled, no perturbation is computed and no embedding is saved or restored. The FGM model and its BERT-CRF counterpart have the same modules and parameter count; the extra forward and backward passes occur only in training. The configuration evaluated here, which differs from earlier agricultural use of FGM [15] in being isolated from other architectural changes, uses a single step with $\varepsilon = 0.5$ and weights of 0.5 for both losses.

2.4 Comparison Models and Experimental Settings

2.4.1 Baseline Models and Comparison Design

The character-level feature-based CRF uses hand-engineered features: the current character, its case variants, digit and punctuation attributes, position bins, the two characters on either side and combinations of adjacent characters. It is trained with L-BFGS with L1 and L2 coefficients of 0.1 and 0.01 and at most 100 iterations. It uses no pretrained representations and serves as a reference for a system built on local hand-engineered features.

The BiLSTM-CRF uses a character vocabulary of 3,912 entries built from the training data only and 128-dimensional randomly initialized character embeddings. One bidirectional LSTM layer with 128 hidden units per direction yields a 256-dimensional representation, which Dropout and a linear layer map to 27-dimensional CRF emission scores. Variable-length sequences are packed and length-masked so that padding does not enter the encoding of valid characters. The model follows the standard BiLSTM-CRF tagger [5, 6].

BERT-CRF and BERT-CRF+FGM share the encoder, character mapping, CRF, learning rate, numerical precision and batch size, and differ in training with a single clean loss versus the equally weighted clean/adversarial scheme. Their comparison measures the overall effect of this FGM configuration. The feature-based CRF, BiLSTM-CRF and BERT-CRF differ in representation source and training resources as well, so their comparison concerns fixed system configurations and does not isolate the contribution of pretraining or of the decoder.

Table 3. Main training settings of the comparison models.

Setting	Feature-based CRF	BiLSTM-CRF	BERT-CRF / +FGM
Representation / encoder	Character and ± 2-neighbour features	128-dim. character embeddings; 128 units per direction	Final layer of bert-base-chinese
Optimizer	L-BFGS	AdamW	AdamW
Learning rate / weight decay	Not applicable; L1 = 0.1, L2 = 0.01	1×10^{-3} / 0	3×10^{-5} / 0.01
Training batch size	Not applicable	64	16
Maximum training epochs	–	15	5
Maximum optimization iterations	100	–	–
Early-stopping patience (epochs)	–	3	2
Dropout	Not applicable	0.2	0.1
Gradient-norm clipping / warmup fraction	Not applicable	5 / none	1 / 0.1
Random seeds	42	42, 13, 21	42, 13, 21
FGM setting	Not applicable	Not applicable	$\varepsilon = 0.5$; clean/adversarial loss weights 0.5 each

Note: The two BERT configurations share the encoder, decoder, optimizer and data protocol. BiLSTM-CRF and the feature-based CRF use their own fixed settings, so cross-model differences are not attributed to individual components.

2.4.2 *Training Environment and Checkpoint Selection*

Experiments ran on a Windows workstation with an Intel Core i9-14900K CPU (24 physical cores, 32 logical processors), 128 GiB of RAM and an NVIDIA GeForce RTX 4090 GPU, using Python 3.12.2, PyTorch 2.11.0+cu128, Transformers 4.49.0, Tokenizers 0.21.4 and python-crfsuite 0.9.12. Table 3 lists the main settings. Both BERT configurations use float32 without automatic mixed precision, a training batch size of 16, a development evaluation batch size of 32 and no gradient accumulation. AdamW [25] uses a learning rate of 3×10^{-5} and a weight decay of 0.01 for all trainable parameters. The first 10% of the planned optimizer steps are a linear warmup, followed by linear decay, and gradients are clipped at an L2 norm of 1. Entity F1 on the full development set is computed after each epoch, with at most 5 epochs and an early-stopping patience of 2 epochs.

All three neural models use the fixed seeds 42, 13 and 21, and complete training and evaluation logs were kept. Checkpoints are selected by development-set micro-averaged entity F1, with ties going to the earlier epoch. The best BiLSTM epochs for the three seeds are 8, 12 and 11; all six BERT and FGM runs select epoch 5. Cost and case analyses use seed 42, fixed in advance rather than chosen for its test score. Identical seeds give matched starting conditions, but the extra FGM forward pass consumes additional random numbers, so the two training trajectories diverge afterwards.

2.4.3 *Evaluation Sequence and Candidate Selection*

The baseline evaluation was finished, with checkpoints frozen and the official test set already used, before FGM was added. In the FGM stage, candidates were first screened on the development set and then verified with seeds 13 and 21; the configuration, checkpoints and analysis rules were frozen before all three seeds were evaluated on the test set. Because the same test set had been used for the baseline, these results are not an independent blind test.

A candidate advanced if it reached a development F1 of at least 94.1243% in the initial screening run, a three-seed development mean of at least 94.0897% (averaged across seeds 42, 13 and 21) and an improvement for at least two matched seeds. After the test evaluation, adoption also required a three-seed mean test F1 of at least 94.5512% and improvement for at least two matched seeds. These are rules for allocating effort, not significance tests. Because test results entered the adoption decision, the test results should be read as exploratory. Neither ε, the loss weights nor the representative seed was changed after testing. Configurations that did not advance,

and the reasons for stopping them, are reported in Supplementary Material Table A1.

2.5 Evaluation Metrics and Subgroup Analyses

2.5.1 Entity-Level Precision, Recall and F1

The evaluation unit is an entity occurrence, not a character tag. Gold and predicted entities are tuples (record ID, start, end, type). A true positive (TP) requires an exact match on all four fields; an unmatched prediction is a false positive (FP), and an unmatched gold entity a false negative (FN). The micro-averaged metrics are

$$P = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \qquad R = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}, \qquad F_1 = \frac{2\mathrm{TP}}{2\mathrm{TP}+\mathrm{FP}+\mathrm{FN}}. \tag{7}$$

A prediction with shifted boundaries is not a TP: it counts as one FP, and the gold entity it fails to match counts as one FN. No partial credit is given for overlap. All models use the same scoring and offset rules. Counts are also accumulated per type, and the mean F1 and sample standard deviation (SD) over three seeds are reported. The feature-based CRF was run once, so no SD is given for it. Ratios from Eq. (7) are multiplied by 100 when reported as percentages.

For runs $s = 1, 2, 3$ (seeds 42, 13 and 21), let $F_{1,s}$ be the entity F1 in percent. The mean and sample SD are

$$\overline{F_1} = \frac{1}{m}\sum_{s=1}^{m} F_{1,s}, \qquad m = 3, \qquad \mathrm{SD} = \sqrt{\frac{1}{m-1}\sum_{s=1}^{m}\left(F_{1,s}-\overline{F_1}\right)^2}. \tag{8}$$

Mean F1 is given in percent, and SDs and differences between means in percentage points (pp). All differences are computed from unrounded values, so a difference of displayed values may differ in the last digit. The SD describes variation due to training randomness on a fixed split; it is not a confidence interval, and three seeds are not three independent datasets. The comparisons are descriptive and are not used for significance testing [21].

2.5.2 Seen and Unseen Surface Forms and Sensitivity to Exact Duplicate Texts

The surface-form set $\mathcal{V}_{\mathrm{train}}$ is built from the training gold entity texts after Unicode NFC normalization; it contains 19,806 distinct surface forms, pooled across types. Gold and predicted entities are assigned to the seen or unseen group according to whether their own surface form belongs to $\mathcal{V}_{\mathrm{train}}$, and TP, FP and FN are counted separately in each group. Because a boundary error can place a prediction and its gold counterpart in different groups, grouping gold entities alone would not be enough. Only training annotations are used: development and test forms are not added, and aliases are not merged by hand.

An unseen surface form is simply a string absent from the training annotations. It need not be a new real-world entity, and the model may still have met related text during pretraining. The duplicate-filtered subset, in turn, shows how the metrics change once exact text overlaps are removed, with the original predictions and scoring kept. Both analyses use the official split and are no substitute for cross-domain testing.

Table 4. Overall entity recognition results on the official test set.

Model	Runs	P (%)	R (%)	F1 (%)	F1 SD (pp)
CRF	1	94.38	92.32	93.3373	–
BiLSTM-CRF	3	93.80	92.09	92.9397	0.0730
BERT-CRF	3	94.03	94.88	94.4512	0.0522
BERT-CRF+FGM	3	94.37	95.10	94.7307	0.0604

Note: The feature-based CRF is a single run; other values are means over three seeds. SD is the sample SD of F1 across seeds and is not a confidence interval. P and R are rounded to two decimals; differences were computed from unrounded values.

2.5.3 Inference Cost and Error Analysis

Cost analysis uses the seed-42 BERT and FGM checkpoints on the same 256 development records, sampled with seed 42 and measured in one session. CPU measurements use one thread and batch size 1 with 10 warmup records; GPU measurements use batch size 16 with one warmup batch and CUDA synchronization after each timed batch. Timing covers tokenization, the forward pass, CRF decoding and span reconstruction, but not model loading or warmup. Each model is measured over three passes, giving 768 CPU sentence timings and 48 GPU batch timings. p50 is the median and p95 is computed with the nearest-rank method. CPU throughput is derived from the total duration of the timing windows; GPU throughput from the median elapsed time of three complete passes, which includes the overhead of prediction-consistency checks. We also report parameter count, checkpoint size and the elapsed training time of one historical seed-42 run per model.

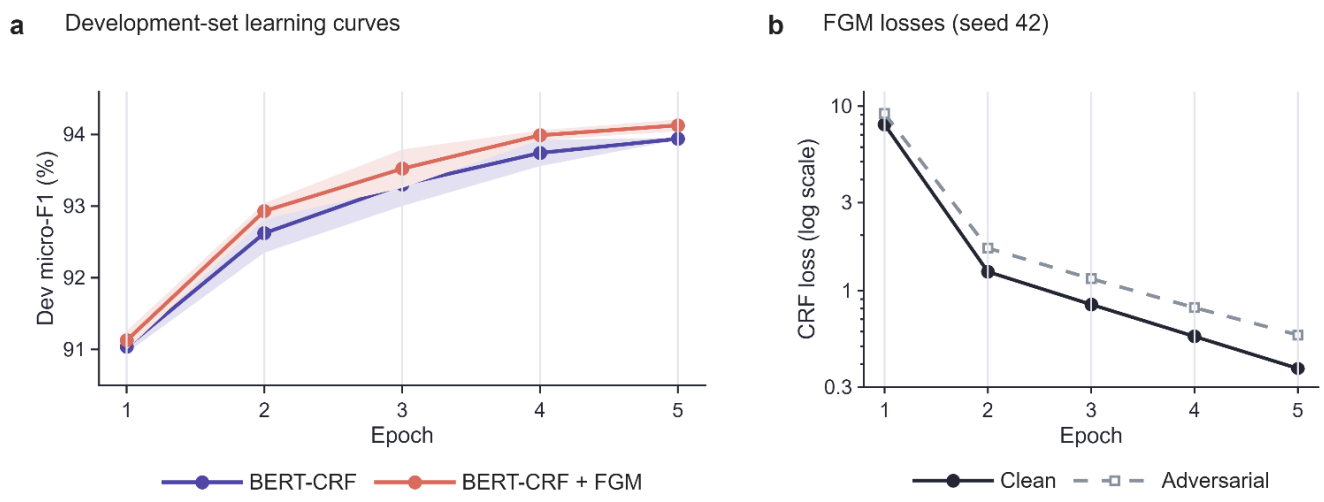

Figure 3. Training trajectories and development-set performance. (a) Micro-F1 on the full development set at each of the five epochs, shown as the mean across seeds 42, 13 and 21 with shading for the sample SD. (b) Batch-averaged clean and adversarial CRF losses of the FGM model with seed 42 on a logarithmic scale. Lines join recorded epoch values without fitting or smoothing; panel (b) has no across-seed uncertainty estimate.

CPU sentence latency and GPU batch latency refer to different timed units.

Error analysis pairs gold and predicted entities one-to-one within each record. Both lists are sorted by start, end and type, and exact matches are removed. Entities with identical boundaries but different types are paired next, followed by same-type entities with overlapping spans; at each step, gold entities are taken in order and paired with the first eligible unused prediction. What remains is counted as unmatched predictions and missed gold entities. Each boundary-error or type-error pair contributes one FP and one FN, and the totals are checked against the overall scores. The analysis only describes the composition of errors; it does not change F1 and is not an expert diagnosis of their causes.

2.6 Automatic Entity Occurrence Index and Evidence Traceability

The index is built from predicted entity occurrences. Each entry stores the model run ID, split and record ID, entity text, type, start and end positions, an evidence excerpt and the identity of the released file. The stored text is checked against the substring between the start (inclusive) and end (exclusive) positions. Occurrences of the same surface form at different positions are kept as separate entries and are not merged into one real-world object.

The index makes recognition outputs traceable to the released records, which helps in checking boundaries and types and supplies candidate entries for organizing domain information by hand. The released data contain neither article URLs nor source-document IDs, so traceability stops at the AgCNER records. No entity disambiguation, relation extraction or expert verification has been performed, so the output is an entity occurrence index, not a validated knowledge graph, and it says nothing about how often diseases and pests actually occur.

3 Results

3.1 Training Trajectories and Development-Set Performance

Development-set F1 rose at every epoch in all BERT-CRF and BERT-CRF+FGM runs, and all six runs reached their best checkpoint at epoch 5 (Figure 3(a)). The final development F1 of BERT-CRF averaged 93.9397% across seeds (SD 0.0148 pp), against 94.1284% (SD 0.0816 pp) for BERT-CRF+FGM, a mean difference of 0.1887 pp.

For BERT-CRF+FGM with seed 42, the batch-averaged clean and adversarial losses fell from 7.9516 and 9.1140 at epoch 1 to 0.3789 and 0.5757 at epoch 5 (Figure 3(b)). The adversarial loss stayed above the clean loss throughout, and the equally weighted total fell from 8.5328 to 0.4773, matching the logged

objective.

Because development F1 improved at every epoch, early stopping never triggered and every run used the full five-epoch budget. The trajectories therefore describe performance within this budget only; longer training might alter the results.

3.2 Overall Model Performance

Table 4 gives the strict entity-level results on the official test set. The single feature-based CRF run reached an F1 of 93.3373%. Mean F1 over three seeds was 92.9397% for BiLSTM-CRF, 94.4512% for BERT-CRF and 94.7307% for BERT-CRF+FGM. BERT-CRF exceeded the BiLSTM-CRF mean by 1.5115 pp, and BERT-CRF+FGM exceeded the feature-based CRF and BiLSTM-CRF by 1.3934 and 1.7910 pp.

Adding FGM to the matched BERT-CRF raised mean test precision from 94.0273% to 94.3658% and mean recall from 94.8792% to 95.0984%, so mean F1 rose by 0.2795 pp. Averaged over seeds, FP fell from 1,237.00 to 1,165.33 and FN from 1,051.00 to 1,006.00; the gain thus came from fewer errors of both kinds.

3.3 Paired Comparison of the FGM Training Scheme

Table 5 and Figure 4 give the paired results for the three seeds. The F1 differences for seeds 42, 13 and 21 were $+0.3221$, $+0.1544$ and $+0.3621$ pp, all positive. For seed 42, TP rose from 19,464 to 19,519, while FP and FN fell by 79 and 55; FP and FN also fell for seeds 13 and 21. Since both groups share the architecture, base configuration and evaluation procedure, the comparison is between complete training schemes with and without FGM.

The across-seed SD of test F1 was 0.0604 pp for BERT-CRF+FGM and 0.0522 pp for BERT-CRF, and FGM also had the larger SD on the final development score. The higher mean was therefore not accompanied by lower run-to-run variation in this experiment.

3.4 Differences in Recognition Performance Across Entity Types

BERT-CRF+FGM improved mean F1 for 11 of the 13 types, by varying amounts (Table 6 and Figure 5). For the three largest types, Pest, Crop and Disease, mean F1 reached 97.5929%, 96.5396% and 98.2009%, only 0.0371, 0.0068 and 0.0470 pp above BERT-CRF. Gains were larger for Cultivar, Part and Period (0.7000, 0.5617 and 0.9424 pp).

The largest gains were for Fertilizer, Other and Pathogens (1.5995, 1.5137 and 1.2478 pp), although Fertilizer and Other have only 108 and 224 test entities. Even after its gain, Other reached only 64.0774%, the lowest F1 of all 13 types, so a large gain and a high score describe different things.

Mean F1 fell by 1.2210 pp for Company and 0.5890 pp for Organization, to 82.8667% and 79.9832%; these types have 96 and 316 test entities. The overall improvement was therefore not shared by every type.

3.5 Seen and Unseen Surface Forms and Sensitivity to Duplicate Texts

3.5.1 Performance on Surface Forms Unseen in the Training Annotations

Of the gold test entities, 18,935 have surface forms seen in the training annotations and 1,589 (7.74%) have unseen forms. BERT-CRF reached a mean F1 of 96.2655% on seen and 74.8686% on unseen forms; BERT-CRF+FGM reached 96.4118% and 76.4077%, gains of 0.1463 and 1.5391 pp (Table 7 and Figure 6(a)). The unseen group gained more but still lagged well behind the seen group.

In the unseen group, BERT-CRF+FGM had a mean precision of 70.5977% and recall of 83.2599%, against 68.6510% and 82.3369% for BERT-CRF, so both rose. Precision stayed well below recall, meaning that many predictions in this group did not match a gold entity exactly. Because predictions are grouped by their own surface forms, a prediction with an over-extended boundary can create a new, unseen string; the unseen-group scores therefore mix surface-form recognition with boundary errors.

3.5.2 Results After Removing Exact Duplicate Texts

After the 125 test records duplicated in the training or development set were removed, mean F1 was 94.4088% for BERT-CRF and 94.6832% for BERT-CRF+FGM, a difference of 0.2744 pp (Table 8 and Figure 6(b)), close to the 0.2795 pp on the official test set. On the filtered subset, F1 for unseen surface forms rose from 74.9541% to 76.4665%, in the same direction as on the full test set.

Both models lost a little F1 after filtering, while the FGM gain stayed close to its value on the official set. This analysis addresses exact text overlap only, and the removal also changed the mix of texts and entities.

3.6 Computational Costs

BERT-CRF and BERT-CRF+FGM each have 102,289,194 parameters and a checkpoint size of

Table 5. Test results for BERT-CRF and BERT-CRF+FGM with matched seeds.

Seed	BERT-CRF F1 (%)	BERT-CRF+FGM F1 (%)	Paired difference (pp)	FGM TP	FGM FP	FGM FN
42	94.4442	94.7662	+0.3221	19,519	1,151	1,005
13	94.5066	94.6610	+0.1544	19,503	1,179	1,021
21	94.4029	94.7649	+0.3621	19,532	1,166	992

Note: Every row evaluates the same 5,974 test records and 20,524 gold entities. Seed 42 was fixed in advance as the representative model and was not chosen from test results.

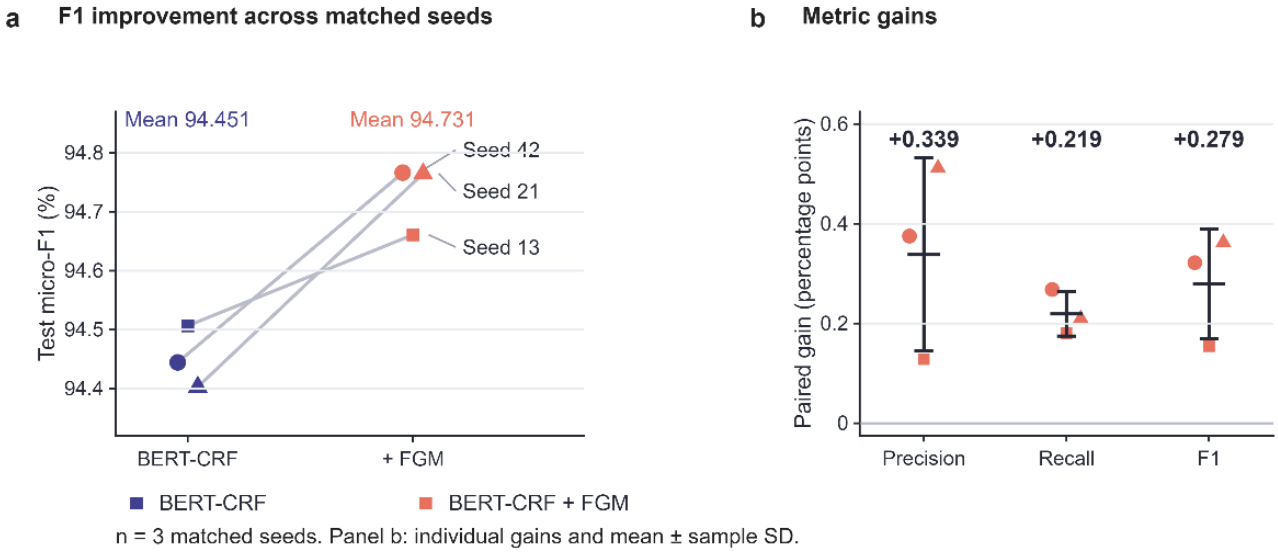

Figure 4. Paired test results. (a) Test micro-F1 for matched BERT-CRF and BERT-CRF+FGM runs for the three fixed seeds; the score axis is truncated to show small differences. (b) Paired changes in precision, recall and F1 for each seed, with the mean ± sample SD in percentage points. SD bars are not confidence intervals, and no statistical significance is implied.

Table 6. Test-set recognition results for the 13 entity types.

Type	Gold	BERT F1 (%)	FGM F1 (%)	Diff. (pp)
BIS (Biosystematic)	165	92.23 ± 0.60	92.46 ± 0.25	+0.22
COM (Company)	96	84.09 ± 0.82	82.87 ± 0.60	−1.22
CRO (Crop)	4,074	96.53 ± 0.12	96.54 ± 0.09	+0.01
CUL (Cultivar)	1,200	92.49 ± 0.38	93.19 ± 0.53	+0.70
DIS (Disease)	2,844	98.15 ± 0.09	98.20 ± 0.09	+0.05
DRUG (Drug)	2,633	92.97 ± 0.37	93.20 ± 0.14	+0.22
FER (Fertilizer)	108	94.17 ± 0.42	95.77 ± 0.24	+1.60
ORG (Organization)	316	80.57 ± 0.99	79.98 ± 1.07	−0.59
OTH (Other)	224	62.56 ± 1.10	64.08 ± 1.25	+1.51
PAOG (Pathogens)	428	91.15 ± 0.75	92.40 ± 0.23	+1.25
PART (Part)	1,699	90.91 ± 0.25	91.47 ± 0.45	+0.56
PER (Period)	2,219	91.52 ± 0.29	92.46 ± 0.33	+0.94
PET (Pest)	4,518	97.56 ± 0.15	97.59 ± 0.04	+0.04

Note: Means ± sample SD (pp) over three seeds. Gold counts are entity occurrences in the test set, not independent texts. Differences were computed from unrounded values.

409,241,571 bytes (Table 9). Measured in the same session, their p50 per-sentence latencies on a single CPU thread at batch size 1 were 80.30 and 80.60 ms, and both processed 10.25 sentences/s. At GPU batch size 16, p50 batch latencies were 38.80 and 36.77 ms.

The historical training times of BERT-CRF and BERT-CRF+FGM with seed 42 were 1.1215 and 1.2993 GPU·h, so the FGM run took about 15.9% longer; each figure comes from a single run.

3.7 Entity Occurrence Indexing and Error Composition

3.7.1 *Tracing Predictions to Records and Representative Cases*

The seed-42 BERT-CRF+FGM model produced 20,670 indexed test entity occurrences, 19,519 TP and 1,151 FP. Every occurrence can be traced to the released text through its record ID and character span, and field checks found no mismatch between the stored entity text and the corresponding substring. Table 10 shows a correct recognition, a boundary error, a type error and a correctly recognized unseen surface form, each selected by the predefined first-occurrence rule.

In the boundary-error case, the gold entity “穗” (“ear”) occupies $[34, 35)$, whereas the model predicted “穗病叶” over $[34, 37)$. The context “穗病叶不病” means that the ears were diseased while the leaves were not, so the predicted Part span wrongly absorbed the characters for “diseased” and “leaf”; the type is correct,

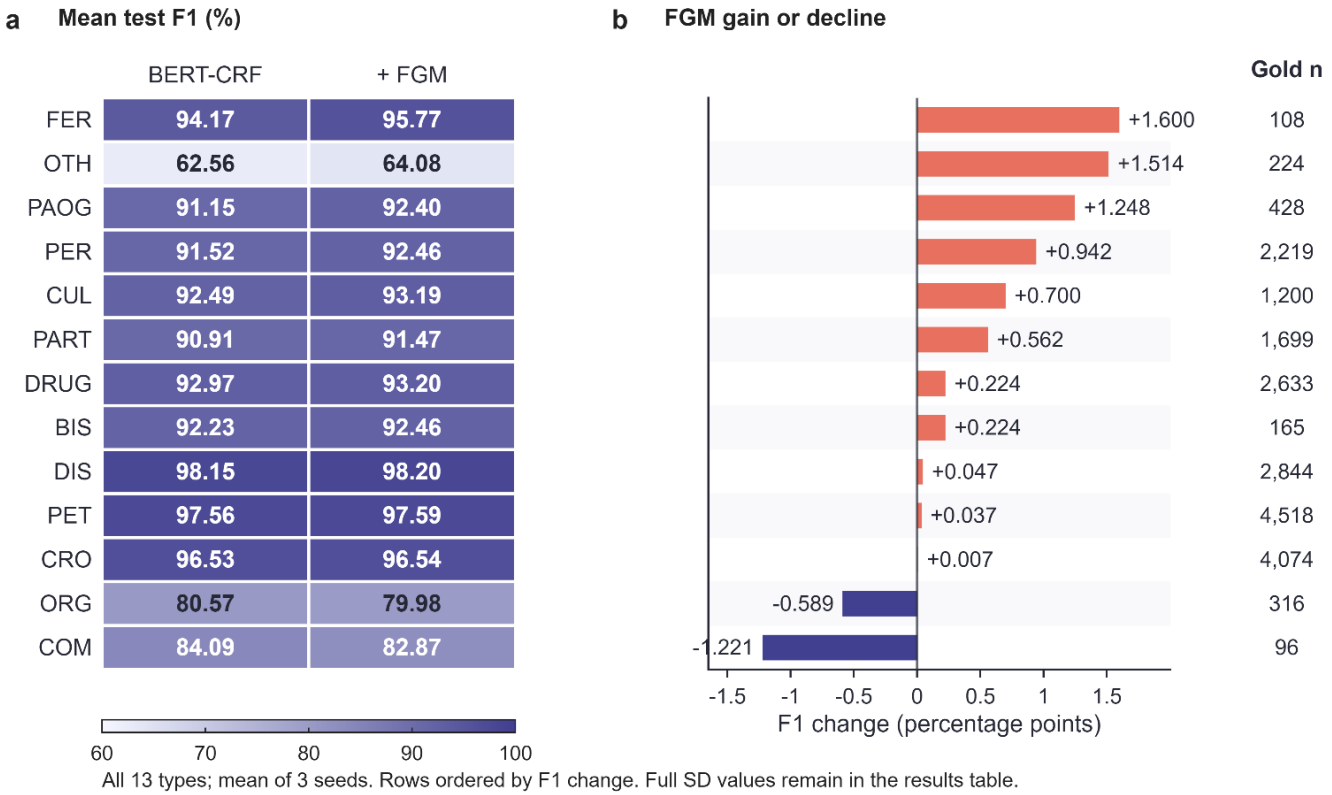

Figure 5. Type-specific test performance and signed FGM gains. (a) Heatmap of mean F1 over three seeds for both models; sample SDs are given in Table 6. (b) Mean F1 of BERT-CRF+FGM minus BERT-CRF in percentage points, with rows ordered by signed change and test gold counts in the right column. No uncertainty interval is derived for the differences, and no multiple-comparison test is reported.

Table 7. Test-set results for surface forms seen and unseen in the training annotations.

Group	Model	Gold count	Mean P (%)	Mean R (%)	Mean F1 (%) ± SD (pp)
Seen surface forms	BERT-CRF	18,935	96.60	95.93	96.27 ± 0.05
Seen surface forms	BERT-CRF+FGM	18,935	96.73	96.09	96.41 ± 0.08
Unseen surface forms	BERT-CRF	1,589	68.65	82.34	74.87 ± 0.96
Unseen surface forms	BERT-CRF+FGM	1,589	70.60	83.26	76.41 ± 0.28

Note: The surface-form set was built only from training gold entity texts after Unicode NFC normalization. Predicted and gold entities were grouped separately by their own surface forms. Three seeds per model; SD in percentage points.

Table 8. Sensitivity to removal of exact duplicate texts.

Model	Official test F1 (%)	Filtered test F1 (%)
BERT-CRF	94.4512 ± 0.0522	94.4088 ± 0.0550
BERT-CRF+FGM	94.7307 ± 0.0604	94.6832 ± 0.0609

Note: Means ± sample SD (pp) over three seeds. The official test set has 5,974 records and the filtered test subset 5,849. Records were removed when the SHA-256 checksum of their complete text matched that of a training or development record. Near duplicates and overlap in source articles were not examined.

but under strict scoring the span mismatch makes the prediction an error. In the type-error case “南通染化厂” (a dye-chemical plant in Nantong), the span is correct but the gold type is Company (COM) and the predicted type Organization (ORG). The cultivar “连粳11号” (Lianjing 11) does not appear among the training surface forms, yet the model recovered both its span and its Cultivar type; aggregate results for unseen forms are in Table 7.

3.7.2 *Boundary, Type and Unpaired Errors*

Under the fixed pairing rules, seed 42 produced 421 same-type boundary-error pairs, 44 same-boundary type-error pairs, 686 unmatched predictions and 540 missed gold entities (Figure 7). Since each boundary- or type-error pair contributes one FP and one FN, $\mathrm{FP} = 421+44+686 = 1{,}151$ and $\mathrm{FN} = 421+44+540 = 1{,}005$, consistent with the overall scores. Paired errors appear in both the FP and FN bars to show their contribution

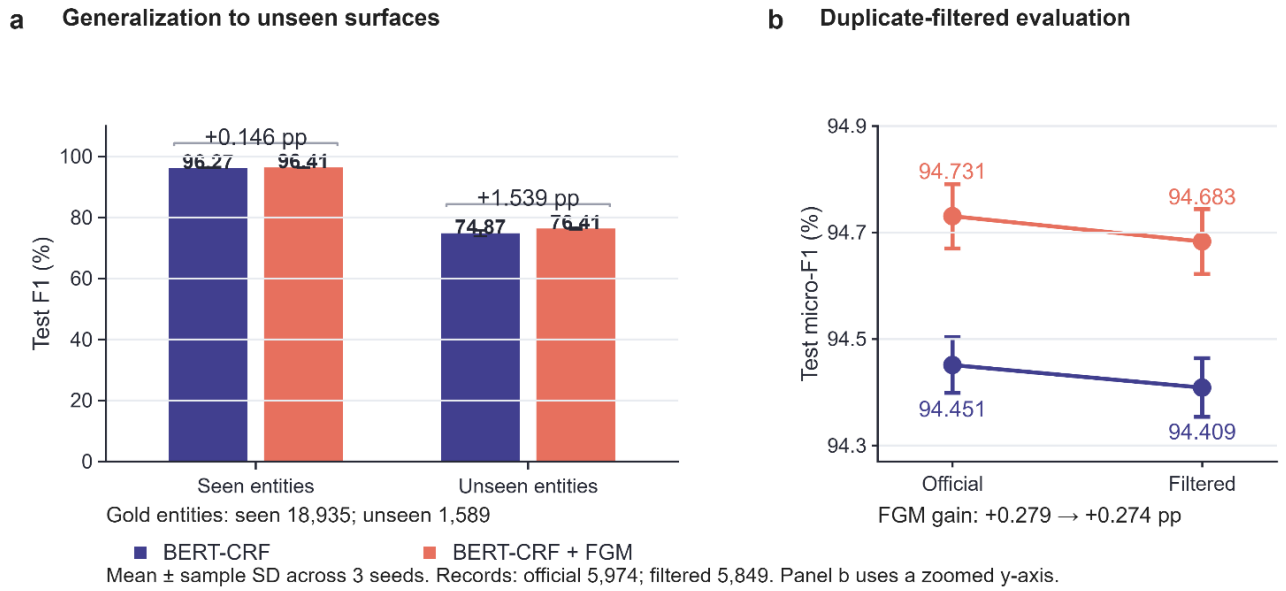

Figure 6. Seen and unseen surface forms and sensitivity to removal of exact duplicate texts. (a) F1 for surface forms seen and unseen among training gold entities, on a percentage axis starting at zero; gains are mean differences in percentage points. (b) Official test micro-F1 and the results after removing test records whose complete text exactly matches a training or development record; the F1 axis is truncated. Bars, points and error bars show means and sample SDs over three seeds. The official and filtered sets contain 5,974 and 5,849 records.

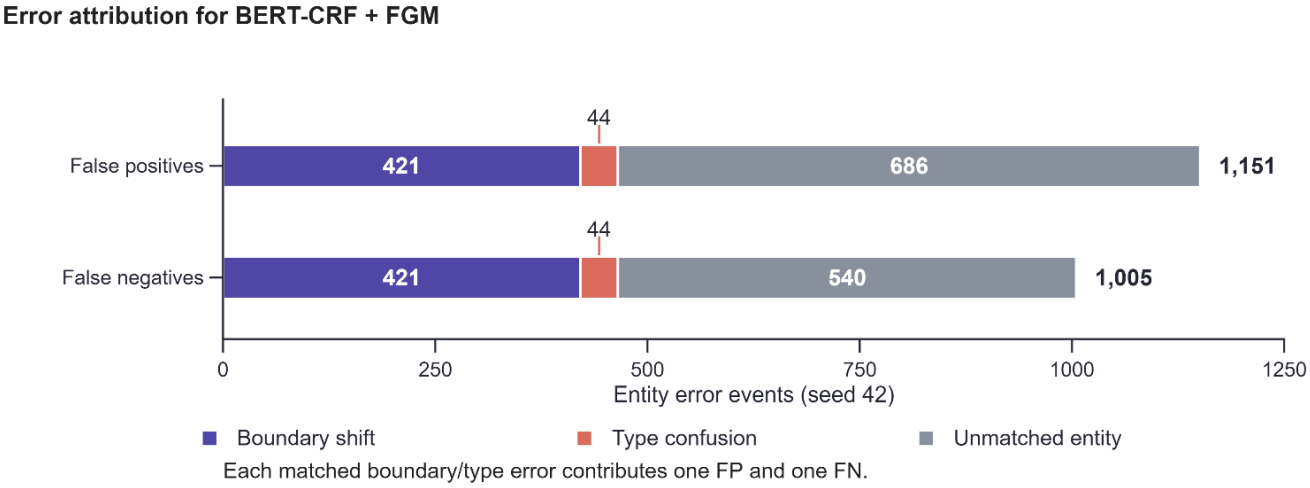

Figure 7. Error composition of the FGM model with seed 42. Absolute FP and FN counts after removing exact matches, then pairing same-boundary type errors and finally overlapping same-type boundary errors. Each paired boundary or type error contributes one FP and one FN, whereas unmatched predictions and missed gold entities contribute only an FP or an FN. Totals are 1,151 FP and 1,005 FN; adding them counts each paired error twice and does not give the number of distinct error events. Error categorization leaves strict entity-level F1 unchanged.

to the counts; they are not two separate error cases.

4 Discussion

This study gives a paired evaluation of an existing FGM training method on the official AgCNER corpus and shows where its gains do and do not occur. Earlier agricultural NER work improved term representations with multiscale local context and attention [14], and Tang et al. [15] combined feature fusion with FGM; Chinese NER models have also incorporated lexical lattices and character-structure or radical information [11–13]. Building on the earlier agricultural use of FGM, we measured its incremental effect under matched BERT-CRF conditions, using consistent character-span scoring, matched seeds, subgroup analyses and traceable outputs. We did

Table 9. Inference costs measured in one session and historical training times for the seed-42 models.

Measurement	BERT-CRF	BERT-CRF+FGM
Parameters	102,289,194	102,289,194
Checkpoint size (bytes)	409,241,571	409,241,571
CPU, 1 thread, batch 1: p50 (ms)	80.30	80.60
CPU, 1 thread, batch 1: p95 (ms)	201.77	203.34
CPU throughput (sentences/s)	10.25	10.25
GPU, batch 16: p50 (ms)	38.80	36.77
GPU, batch 16: p95 (ms)	61.61	56.89
GPU throughput (sentences/s)	371.9	400.5
Historical training, seed 42 (GPU·h)	1.1215	1.2993

Note: p50 and p95 are the 50th and 95th latency percentiles. Inference was measured over three passes through the same 256 development records in one session (768 per-sentence CPU and 48 GPU batch observations). Each training time is a single historical observation, not a repeated controlled measurement.

not reproduce the complete systems cited above, so ranking our results against them would require controlled comparisons.

The feature-based CRF scored higher than the three-seed BiLSTM-CRF mean, so the neural encoder trained from scratch offered no advantage in the present configuration. The two systems used different representations and settings and were not tuned systematically under equal budgets, so this ranking applies only to the configurations tested. Both BERT systems rely on a pretrained model, so their advantage over the other two also reflects the representation source and pretraining resources, and cross-model scores cannot isolate the contribution of the CRF or of attention.

FGM combines losses on clean embeddings and on locally perturbed ones in each update, which is the rationale for the scheme [17]. The paired results compare complete training schemes, however. We ran no compute-matched control with two forward passes, equally weighted losses and $\varepsilon = 0$, so the effects of the perturbation, the second Dropout pass and the altered random-number trajectory cannot be separated, and name regularity bias was not tested directly. The score increase supports a practical benefit under the present settings; identifying its mechanism will need dedicated controls. Evaluation on unperturbed test data, even grouped by unseen surface forms, is also not a robustness evaluation under input perturbations.

FGM adds forward and backward computation during training but leaves the deployed BERT-CRF architecture and parameter count unchanged. For applications that must keep an existing inference pipeline, it can therefore offer a modest gain, although it does nothing to reduce encoder size. Training time was observed once per model and inference cost was measured in one session, without isolating system load, caching or timing effects. The 15.9% difference is thus not a general estimate of FGM overhead, and the GPU latency difference does not show a consistent speed-up. Training time, single-request latency and batch throughput measure different things and should be interpreted separately [22]; surveys of efficient NLP methods further distinguish hardware utilisation, memory footprint and throughput as complementary dimensions of deployment cost [26].

Record IDs and character spans make each prediction traceable to the released text, which supports retrieval by type and surface form and makes errors easy to locate. The index provides candidate entries for organizing agricultural text material, but its field-consistency checks say nothing about entity semantics or the reliability of the source. Further organizing information on drugs, pathogens, companies and organizations would require expert review, entity disambiguation and relation confirmation, and tracing to the released records does not replace checking the original sources.

During method selection we also tried layer fusion, grouped learning rates, a residual character CNN and MacBERT [10]; none met the predefined continuation criteria. A later single-seed screen with $\varepsilon = 0.3$ reached a development F1 of 94.0926%, below the 94.1687% obtained with $\varepsilon = 0.5$, so it was not run with further seeds or on the test set. Scores and stopping rules for these explorations are given in Supplementary Material Table A1. Reporting configurations that were dropped makes the development process more transparent, but single-seed screening and multistage comparison can still introduce selection bias, and failing a continuation criterion does not show that a method is ineffective.

The main limits on inference are the use of a single corpus and the earlier exposure of its test set (Section 2.4.3). The unseen-surface-form groups and the filtered subset both come from the same official split; they describe behaviour within this corpus but not across sources, regions or a second corpus. Unequal type frequencies, the small number of seeds, single-session cost measurements and the lack of expert review of the index further restrict the statistical and practical conclusions.

Three follow-up steps would address these limits: evaluation on independently sourced corpora to test whether the mean gain holds; annotation and context

Table 10. Cases traced to records for the fixed representative model.

Case	Record ID	Gold entity	FGM prediction
Correct recognition	test:000000	[8, 10) PART 「叶片」	[8, 10) PART 「叶片」
Boundary error	test:000005	[34, 35) PART 「穗」	[34, 37) PART 「穗病叶」
Type error	test:000308	[107, 112) COM 「南通染化厂」	[107, 112) ORG 「南通染化厂」
Correct unseen surface form	test:000055	[19, 24) CUL 「连粳11号」	[19, 24) CUL 「连粳11号」

Note: The representative model was fixed in advance to seed 42. For each case category, the first eligible occurrence was taken in test-record order, checking entities in character order. Spans are zero-based half-open intervals. Error types were assigned automatically and have not been reviewed by agricultural experts.

review for the types that lost performance, such as Company and Organization, to separate type confusion from small-sample effects; and repeated, controlled timing to estimate training overhead and deployment cost. Any new module should likewise be tested against matched controls under the same data and evaluation conditions.

5 Conclusion

We evaluated BERT-CRF with single-step FGM training of word embeddings on AgCNER and linked its predictions to the released records, using the same character-span scoring for model comparisons, same-seed paired comparisons and subgroup analyses. The main findings are as follows.

1) On the official AgCNER test set, BERT-CRF+FGM reached mean precision, recall and F1 of 94.3658%, 95.0984% and 94.7307% over three seeds. Its mean F1 exceeded that of BERT-CRF by 0.2795 pp, and the difference was positive for all three matched seeds.

2) The gain varied by entity type and surface form. Mean F1 rose for 11 of the 13 types but fell for Company and Organization. F1 on surface forms unseen in training rose from 74.8686% to 76.4077% but stayed well below that on seen forms. After exact duplicate texts were removed, BERT-CRF+FGM reached 94.6832%, 0.2744 pp above the baseline.

3) FGM kept the BERT-CRF inference architecture and its 102,289,194 parameters, with the extra computation confined to training. The seed-42 model produced 20,670 indexed entity occurrences whose record IDs and spans allow every prediction to be checked and every error to be located.

These results come from an exploratory evaluation of an existing training method on a single agricultural corpus. Whether they carry over to other sources and to practical use remains to be tested on independent data and with expert review.

Data Availability Statement

The AgCNER corpus is publicly available on Figshare (collection DOI: 10.6084/m9.figshare.c.6807873.v1, https://figshare.com/collections/6807873). Processed predictions, the entity occurrence index and analysis scripts are available from the corresponding author upon reasonable request.

Funding

This work was supported by the National Major Agricultural Science and Technology Project under Grant 0304.

Conflicts of Interest

The authors declare no conflicts of interest.

AI Use Statement

The authors declare that no generative AI was used in the preparation of this manuscript.

Ethical Approval and Consent to Participate

Not applicable. This study used only a publicly released text corpus and involved no human participants or animals.

References

[1] Li, J., Sun, A., Han, J., & Li, C. (2020). A survey on deep learning for named entity recognition. *IEEE transactions on knowledge and data engineering, 34*(1), 50-70. [CrossRef]

[2] Nadeau, D., & Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes, 30*(1), 3-26. [CrossRef]

[3] Yao, X., Hao, X., Liu, R., Li, L., & Guo, X. (2024). AgCNER, the first large-scale Chinese named entity recognition dataset for agricultural diseases and pests. *Scientific Data, 11*(1), 769. [CrossRef]

[4] Lafferty, J., McCallum, A., & Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning* (pp. 282-289). Morgan Kaufmann. https://repository.upenn.edu/cis_papers/159

[5] Huang, Z., Xu, W., & Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991*. [CrossRef]

[6] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 260-270). [CrossRef]

[7] Guo, X., Hao, X., Tang, Z., Diao, L., Bai, Z., Lu, S., & Li, L. (2021). Ace-adp: adversarial contextual embeddings based named entity recognition for agricultural diseases and pests. *Agriculture, 11*(10), 912. [CrossRef]

[8] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems 30* (pp. 5998-6008).

[9] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019, June). 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). [CrossRef]

[10] Cui, Y., Che, W., Liu, T., Qin, B., Wang, S., & Hu, G. (2020, November). Revisiting pre-trained models for Chinese natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020* (pp. 657-668). [CrossRef]

[11] Zhang, Y., & Yang, J. (2018). Chinese NER using lattice LSTM. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1554-1564). [CrossRef]

[12] Li, X., Yan, H., Qiu, X., & Huang, X. J. (2020, July). FLAT: Chinese NER using flat-lattice transformer. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 6836-6842). [CrossRef]

[13] Wu, S., Song, X., & Feng, Z. (2021, August). MECT: Multi-metadata embedding based cross-transformer for Chinese named entity recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (pp. 1529-1539). [CrossRef]

[14] Guo, X., Zhou, H., Su, J., Hao, X., Tang, Z., Diao, L., & Li, L. (2020). Chinese agricultural diseases and pests named entity recognition with multi-scale local context features and self-attention mechanism. *Computers and electronics in agriculture, 179*, 105830. [CrossRef]

[15] Tang, W., Wen, X., & Hu, Z. (2024). Named entity recognition for crop diseases and pests based on gated fusion unit and Manhattan attention. *Agriculture, 14*(9), 1565. [CrossRef]

[16] Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*. [CrossRef]

[17] Miyato, T., Dai, A. M., & Goodfellow, I. (2016). Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*. [CrossRef]

[18] Zhu, C., Cheng, Y., Gan, Z., Sun, S., Goldstein, T., & Liu, J. (2019). Freelb: Enhanced adversarial training for natural language understanding. *arXiv preprint arXiv:1909.11764*. [CrossRef]

[19] Jiang, H., He, P., Chen, W., Liu, X., Gao, J., & Zhao, T. (2020). SMART: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 2177-2190). [CrossRef]

[20] Ghaddar, A., Langlais, P., Rashid, A., & Rezagholizadeh, M. (2021). Context-aware adversarial training for name regularity bias in named entity recognition. *Transactions of the Association for Computational Linguistics, 9*, 586-604. [CrossRef]

[21] Reimers, N., & Gurevych, I. (2017). Reporting score distributions makes a difference: Performance study of LSTM-networks for sequence tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (pp. 338-348). [CrossRef]

[22] Dodge, J., Gururangan, S., Card, D., Schwartz, R., & Smith, N. A. (2019). Show your work: Improved reporting of experimental results. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 2185-2194). [CrossRef]

[23] Augenstein, I., Derczynski, L., & Bontcheva, K. (2017). Generalisation in named entity recognition: A quantitative analysis. *Computer Speech & Language, 44*, 61-83. [CrossRef]

[24] Liang, J., Li, D., Lin, Y., Wu, S., & Huang, Z. (2023). Named entity recognition of Chinese crop diseases and pests based on RoBERTa-wwm with adversarial training. *Agronomy, 13*(3), 941. [CrossRef]

[25] Loshchilov, I., & Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*. [CrossRef]

[26] Treviso, M., Lee, J. U., Ji, T., Van Aken, B., Cao, Q., Ciosici, M. R., ... & Schwartz, R. (2023). Efficient methods for natural language processing: A survey. *Transactions of the Association for Computational Linguistics, 11*, 826-860. [CrossRef]

Appendix

Completed Exploratory Configurations and Selection Process

Table A1 reports the completed NER training runs, all on the AgCNER development set with seed 42. The configurations were examined at different stages of a bounded exploration and do not form a complete ablation matrix. These single-seed development results are kept separate from the three-seed test results in the main text.

Table A1. Completed exploratory configurations and subsequent decisions.

Configuration	Seed	Dev F1 (%)	Subsequent decision
Learnable fusion of the last four BERT layers	42	93.9643	Did not meet its stage-specific continuation criterion; stopped without test evaluation.
Grouped learning rates for BERT-CRF	42	94.0387	Below the 94.1243% threshold; stopped without test evaluation.
Character residual CNN	42	93.8237	Below the 94.1243% threshold; stopped without test evaluation.
MacBERT-CRF	42	93.8732	Below the 94.1243% threshold; stopped without test evaluation.
FGM, $\varepsilon = 0.5$	42	94.1687	Advanced to three-seed verification and test evaluation; results reported in the main text.
Start/end boundary auxiliary task + FGM	42	94.1112	Did not meet the applicable continuation criterion; stopped without test evaluation.
Boundary-frequency-weighted embedding-row FGM	42	94.1002	Did not meet the applicable continuation criterion; stopped without test evaluation.
FGM, $\varepsilon = 0.3$	42	94.0926	Below the 94.2687% threshold; stopped without test evaluation.

The layer-fusion run was stopped under its own first-stage rule. Grouped learning rates, the character residual CNN and MacBERT did not reach the 94.1243% development threshold that applied at their stage, whereas FGM with $\varepsilon = 0.5$ did. A later, separate single-seed screen with $\varepsilon = 0.3$ used a threshold of 94.2687%; its F1 of 94.0926% was also below the 94.1687% obtained with $\varepsilon = 0.5$, so it was neither run with seeds 13 and 21 nor evaluated on the test set. These thresholds governed the allocation of further runs and do not represent statistical significance.

The boundary auxiliary-task and boundary-frequency-weighted FGM configurations also have complete training records. Some further ideas were examined only with CPU diagnostics, which produce no NER scores and are not treated as evidence for any model change. Table A1 therefore summarizes the main selection path leading to the adopted FGM configuration rather than ranking all diagnostic work.

Because the official test set had already been used in the baseline study (Section 2.4.3), the test evaluation is not an independent blind test, and reporting the unsuccessful explorations does not remove the selection bias inherent in multistage model development.

Bo Cheng is with the Information Technology Research Center, Beijing Academy of Agriculture and Forestry Sciences, Beijing 100097, China. [Degree and research interests to be added by the author.] (Email: nercita_zswj@126.com)

Biao An is with the College of Computer Science and Technology, Inner Mongolia Minzu University, Tongliao 028000, China. [Degree and research interests to be added by the author.] (Email: 420865541@qq.com)

Ziming Gao is with the College of Computer Science and Technology, Inner Mongolia Minzu University, Tongliao 028000, China. [Degree and research interests to be added by the author.] (Email: 2424176145@qq.com)

Yuansheng Wang is with the Information Technology Research Center, Beijing Academy of Agriculture and Forestry Sciences, Beijing 100097, China. [Degree and research interests to be added by the author.] (Email: 642634129@qq.com)