Supplemental Material

This supplemental paper includes more details about some aspects of the main paper. Specifically, it provides concrete datasets' information and task orders for text classification tasks (§1) and NER tasks (§2). Moreover, the specific implementation details employed during the model's continual training and the hyper-parameters for the replay setting are provided (§3). Last, it expounds upon the Intransigence and Forgetting metrics, which are utilized, which are used to meticulously evaluate the model's learning capability and knowledge change throughout the continual learning process (§4).

Notice that the bibliographic references in this supplemental paper are about the bibliography of the main paper.

1 Datasets and Task Orders for Text Classification Task

For text classification task, we follow MBPA++ [5] and IDBR [11] to choose five most used datasets, including AG News, Yelp, DBPedia, Amazon and Yahoo! Answer. Details of the five datasets are listed in Table 1. Based on these 5 tasks, we construct 7 different task sequences, including length of 3 and 5. The specific task orders of the sequences are in Table 1:

Table 1. Dataset statistics we used for Sampled Setting in classification tasks. Type denotes the domain for task classification. The size of the validation set is the same as the size of the training set.

Dataset	Class	Type	Train	Test
AGNews	4	News	8000	7600
Yelp	5	Sentiment	10000	7600
Amazon	4	Sentiment	10000	7600
DBPedia	14	Wikipedia	28000	7600
Yahoo	10	Q&A	20000	7600

Table 2. Seven different task sequences used for experiments for classification tasks. The first 6 are used in Sampled Setting. The last 4 are used in Full Setting.

Order	Task Sequence
I	ag→yelp→yahoo
II	yelp→yahoo→ag
III	yahoo→ag→yelp
IV	ag→yelp→amazon→yahoo→dbpedia
V	yelp→yahoo→amazon→dbpedia→ag
VI	dbpedia→yahoo→ag→amazon→yelp
VII	yelp→ag→dbpedia→amazon→yahoo

2 Datasets and Class Orders for NER Task

For NER task, we follow [23] to choose two classic datasets: CoNLL-03 English NER and OntoNotes-5.0 English. To ensure enough examples for training, we select named entity types. For CoNLL-03 English, we choose 4 types: Person (PER), Location (LOC), Organization (ORG), and Miscellaneous (MISC). For OntoNotes-5.0 English, we choose 6 types: Organization (ORG), Person (PER), Geo-Political Entity (GPE), Date (DATE), Cardinal (CARD), Nationalities and Religious Political Group (NORP). Dataset statistics are shown in Table 2.

Following [23], apart from selecting entity types, to construct datasets for class-incremental learning, we divide the train/dev sets of the two datasets into 4 and 6 disjoint subsets, D^1, D^1, ..., respectively: each D^i is annotated only for the entity type e^i. And the test set up to step i is annotated for the entity type e^1, ..., e^i. As in task-incemental setting, we also sample 8 and 6 different class/type orders for CoNLL-03 and OntoNotes-5.0, respectively, as shown in Table 2.

3 Implementation Details

We set 8 for batch size, 256 for maximum sequence length, 3e-5 for learning rate and use single A100 GPU for training. We use AdamW as optimizer and constant schedule with warm up as scheduler and weight decay is 0.01. We set ρ, which decides the size of flat regions, to 0.65. The coefficient λ for Find loss, soft constraint, is 50000 and the coefficient γ for accumulating Fisher is 0.95. The number of samples used to calculate Fisher Information for each task is 128. All optimizing constraints are for the encoder and the task layer of the old tasks. All hyper-parameters are searched on the sampled version of the datasets.

For replay, we set store ratio as 0.01 and replay frequency as 20, meaning that current task replay every 20 batch updates. All baseline methods except for **Seq** are equipped with replay and their replay frequencies are identical. State-of-the-art methods except for ProgressivePrompt [24] also incorporate replay, even with a higher replaying frequency.

4 Intransigence and Forgetting

We follow [4] to measure Intransigence $\mathcal{I}_k$ after trained on task k as follows:

$$\mathcal{I}_k = a_k^* - a_{k,k}. \tag{1}$$

where a_k^* denotes the accuracy on task k of model trained in the multitask manner and $a_{k,k}$ denotes the accuracy on k-th task when trained up tp task k in C&F method. To calculate the Intransigence of a method as a whole, we average the values corresponding to each

Table 3. Distribution of entity labels in CoNLL-03 and OntoNotes-5.0.

	CoNLL-03				**OntoNotes-5.0**					
	PER	LOC	ORG	MISC	ORG	PER	GPE	DATE	CARD	NORP
Train	6,532	7,125	6,271	3,398	24,163	22,035	21,938	18,791	10,901	9,341
Dev	1,829	1,832	1,325	916	3,798	3,163	3,649	3,208	1,720	1,277
Test	1,597	1,664	1,654	698	2,002	2,134	2,546	1,787	1,005	990

Table 4. Different class orders in which entity type are added to the models for each dataset.

Order	**CoNLL-03**	**OntoNotes-5.0**
I	PER→LOC→ORG→MISC	ORG→PER→GPE→DATE→CARD→NORP
II	PER→MISC→LOC→ORG	DATE→NORP→PER→CARD→ORG→GPE
III	LOC→PER→ORG→MSC	GPE→CARD→ORG→NORP→DATE→PER
IV	LOC→ORG→MISC→PER	NORP→ORG→DATE→PER→GPE→CARD
V	ORG→LOC→MISC→PER	CARD→GPE→NORP→ORG→PER→DATE
VI	ORG→MISC→PER→LOC	PER→DATE→CARD→GPE→NORP→ORG
VII	MISC→PER→LOC→ORG	
VIII	MISC→ORG→PER→LOC	

task. So the lower the Intransigence, the better the learning ability of the model.

We follow [4] to measure Forgetting $\mathcal{F}_k$ after trained on task k as follows:

$$\begin{aligned} \mathcal{F}_k &= \mathbb{E}_{j=2\ldots t} f_j^k, \\ f_j^k &= \max_{l \in \{1\ldots k-1\}} a_{l,j} - a_{k,j}. \end{aligned} \tag{2}$$

where $a_{l,j}$ is the model's accuracy on task j after trained on task l. To calculate the Forgetting of a method as a whole, we average the values corresponding to each task. So the lower the Forgetting, the less severe the catastrophic forgetting of the model suffers.