A Experimental Details

As we train deep learning models, reproducibility is an issue due to the inherent randomness of the training procedure. Nevertheless, we seek to provide all possible tools for reproducing our experimental results. To do so, we attach our code and the complete set of results. Furthermore, the following sections describe our data preprocessing, the hyperparameters, the computing infrastructure, and the random seeds used in our experiments.

A.1 Data Preprocessing

We choose to use the datasets as they are, despite their imbalanced label distribution (Table 3), since such imbalance is representative of realistic applications. We preprocess the tweets by removing URLs, emails, usernames and mentions. Next, we employ the Ekphrasis package[2] to correct common spelling mistakes and unpack contractions. Finally, emojis are transformed into their respective words using the Python Emoji package[3].

A.2 Hyperparameters

To select the hyperparameters, we trained and evaluated the model on the entire MFTC corpus with 10-fold cross-validation. Table A1 shows the hyperparameters that were compared in this setting, highlighting in bold the best performing option that we then used in the experiments described in the paper. If a parameter is not present in the table, the default value supplied by the framework was used.

Table A1: Hyperparameters tested and selected

Hyperparameters	Options
Model name	**bert-base-uncased**
Number of parameters	**110M**
Max sequence length	**64**
Epochs	2, **3**, 5
Batch size	**16**, 32, 64
Dropout	0.05, **0.1**, 0.02
Optimizer	**AdamW**
Learning Rate	$\mathbf{5*10^{-5}}$
Loss function	**Binary Cross Entropy**

A.3 Computing Infrastructure

The following are the main libraries and computing environment used in our experiments.

- PyTorch: 1.8.1
- TensorFlow: 2.5.0
- FastText: 0.8.22
- Hugginface's Transformers: 4.6.0
- NVIDIA GeForce RTX 2080 Ti GPU
- CUDA: 11.2
- cuDNN: 8.1.1.33

Refer to the code base for a detailed list of the libraries we used, and their versions.

The following list details the amount of GPU hours spent for obtaining our results:

- Tables 4, B1, and B2: 44 hours
- Figures 2 and 3: 33 hours
- Tables B3, B4, and B5: 24 hours
- Table B7: 26 hours

The error analysis (Tables 5, 6, and 7) did not require additional GPU time.

A.4 Random Seeds

In our experiments, we ensured that the same train-test splits are used across different runs of each experiment. Further, to control for randomness, we fixed the random seeds in the following libraries:

- Python (`random.seed`);
- NumPy (`numpy.random.seed`);
- PyTorch (`torch.manual_seed`);
- Tensorflow (`tensorflow.random.set_seed`).

A.5 Artifacts Usage

We have mainly used two artifacts in this research: the MFTC and BERT.

The MFTC was collected with the intent of facilitating NLP research on moral values (Hoover et al., 2020). It can be downloaded[4] and used under the Creative Commons Attribution 4.0 license.

BERT (Devlin et al., 2019) was created with the intent of performing, among others, text classification. Thus, we are using it as originally intended, under its Apache 2.0 distribution license[5].

[2] https://github.com/cbaziotis/ekphrasis

[3] https://pypi.org/project/emoji/

[4] https://osf.io/k5n7y/

[5] https://github.com/google-research/bert/blob/master/LICENSE

B Extended Results

In this Appendix we extend the results presented in the paper. The following results are not crucial for supporting our conclusions. Nevertheless, they provide additional details on our experiments.

B.1 Model Comparison

We have presented the results of the transferability analysis with the BERT model. In order to evaluate whether our conclusions generalize to other model architectures, we repeat the experiment conducted in the paper (see Sections 3 and 4) with the following two additional models:

- Long Short Term Memory (**LSTM**), a category of Recurrent Neural Networks (RNN). We choose LSTM as a baseline model since it is commonly used in moral value classification (Lin et al., 2018; Mooijman et al., 2018; Rezapour et al., 2019; Hoover et al., 2020).

- **fastText**, a machine learning approach that learns character-level information, in contrast to the whole word representations LSTM employs. This flexibility makes fastText a good candidate for transfer learning. Further, we choose fastText as it attains performances on par with state-of-the-art deep learning methods, but is considerably faster.

Tables B1 and B2 present the results of the transferability analysis, performed and presented analogously to Table 4, for LSTM, fastText, and BERT. We observe that BERT outperforms fastText and LSTM in most settings. This is not surprising, since BERT is state-of-the-art for text classification. Both BERT and fastText outperform LSTM, the model extensively used for predicting moral values. Further, we notice that the general trends observed in Section 4.1 hold for all three models.

Generalizability All three models achieve better average F_1-scores in the $\mathcal{C}(source, target)$ setting than the majority ($target$) baseline. However, compared to the majority ($source$) baseline, $\mathcal{C}(target, source)$ performs worse with LSTM, comparably with fastText, and much better with BERT. This suggests that a contextualized representation, as in BERT, is necessary for value classification in novel domains, especially for the novel domains with a large moral vocabulary as is the case in $\mathcal{C}(target, source)$.

Transferability From the average F_1-scores in Table B2, we observe that $\mathcal{C}(finetune, target)$ performs better than or on par with $\mathcal{C}(target, target)$ across all three models. The benefits of finetuning are most evident for LSTM (7% increase in the average m and 17% increase in M). The benefits can also be observed for fastText (similar m and 8% increase of M) and BERT (similar m and 8% increase of M), but to a lesser degree than LSTM.

Catastrophic Forgetting We observe that all three models suffer from catastrophic forgetting since finetuning on $\mathcal{T}_{target}$ reduces the performance on $\mathcal{T}_{source}$. As mentioned in the paper, the degree of catastrophic forgetting is most evident when finetuning on unbalanced datasets such as DAV than balanced datasets such as BLM.

B.1.1 Training Time

In some applications, e.g., estimating value trends on Twitter, value classifiers need to be re-trained frequently since the trends can shift fast. Similarly, to employ techniques such as active learning for value annotation requires training a classifier at every iteration to prompt for new labels. In such cases, training time is an important factor for selecting an approach and model. Figure B1 shows the average training time in logarithmic scale, for different models and scenarios (Appendix A.3 describes our computing infrastructure).

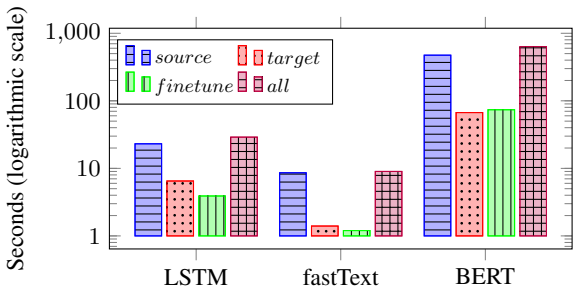

Figure B1: Average training time per model and scenario

Two considerations are evident. First, fastText trains significantly faster than the other two models. Second, for all three models, the training time is approximately proportional to the amount of data in the training set—the $target$ and $finetune$ scenarios employ a similar amount of data, which is roughly six times smaller than in the $source$ and all scenarios.

Table B1: Results of the four training scenarios and three models evaluated on $\mathcal{T}_{source}$. The columns indicate the dataset used as $\mathcal{T}_{target}$. For each experiment we report micro F_1-score (m, left-hand column) and macro F_1-score (M, right-hand column).

	ALM		BLT		BLM		DAV		ELE		MT		SND		Average	
Classifier Setting	*m*	*M*	*m*	*M*	*m*	*M*	*m*	*M*	*m*	*M*	*m*	*M*	*m*	*M*	*m*	*M*
LSTM																
$\mathcal{C}(source, source)$	64.1	45.7	64.0	52.1	61.1	39.6	59.2	48.0	63.5	46.5	66.4	47.1	65.6	46.8	63.4	46.5
$\mathcal{C}(target, source)$	47.8	19.3	41.0	6.1	53.5	25.6	38.8	5.1	51.1	20.2	39.1	11.9	35.1	16.1	43.8	14.9
$\mathcal{C}(finetune, source)$	61.4	37.4	48.3	25.1	60.0	39.6	41.6	11.0	60.7	40.5	55.1	39.1	52.3	36.6	54.2	32.8
$\mathcal{C}(all, source)$	64.5	46.7	63.2	49.2	62.3	41.4	59.3	47.7	64.2	48.6	66.4	48.7	65.8	48.1	63.7	47.2
fastText																
$\mathcal{C}(source, source)$	66.8	56.0	65.9	57.8	64.4	51.5	63.1	56.9	66.6	56.7	69.5	59.5	67.8	56.8	66.3	56.5
$\mathcal{C}(target, source)$	54.5	30.9	42.7	8.5	56.4	33.1	38.7	5.1	52.2	30.0	48.9	22.0	41.3	20.3	47.8	21.4
$\mathcal{C}(finetune, source)$	62.1	48.8	54.4	39.5	62.6	46.4	52.9	39.9	61.4	50.8	57.3	45.7	56.7	49.7	58.2	45.8
$\mathcal{C}(all, source)$	66.9	56.3	66.0	57.5	64.8	52.1	63.1	56.7	66.9	57.0	68.7	58.2	67.5	56.4	66.3	56.3
BERT																
$\mathcal{C}(source, source)$	**73.9**	**65.6**	**73.9**	**68.3**	**71.2**	**61.8**	**71.1**	**66.4**	**73.3**	**66.4**	**75.7**	**68.0**	**74.5**	**66.5**	**73.4**	**66.1**
$\mathcal{C}(target, source)$	61.6	37.7	43.8	13.1	62.6	43.0	38.8	5.1	59.3	40.4	52.4	39.1	54.4	36.6	53.3	30.7
$\mathcal{C}(finetune, source)$	70.3	57.2	61.2	47.8	69.2	54.9	56.6	41.9	70.5	61.5	67.7	60.5	68.0	60.8	66.2	54.9
$\mathcal{C}(all, source)$	**73.7**	**65.6**	**73.7**	**68.0**	**71.3**	**62.1**	**71.0**	**66.4**	**73.6**	**66.7**	**75.6**	**67.7**	**74.3**	**66.6**	**73.3**	**66.2**
Majority ($source$)	47.0	6.1	42.3	5.6	49.0	6.2	38.8	5.3	46.1	6.0	49.0	6.2	48.9	6.2	45.9	5.9

Table B2: Results of the four training scenarios and three models evaluated on $\mathcal{T}_{target}$. The columns indicate the dataset used as $\mathcal{T}_{target}$. For each experiment we report micro F_1-score (m, left-hand column) and macro F_1-score (M, right-hand column).

	ALM		BLT		BLM		DAV		ELE		MT		SND		Average	
Classifier Setting	*m*	*M*	*m*	*M*	*m*	*M*	*m*	*M*	*m*	*M*	*m*	*M*	*m*	*M*	*m*	*M*
LSTM																
$\mathcal{C}(source, target)$	52.5	40.2	61.7	19.3	59.6	43.2	85.9	8.5	52.7	35.7	43.3	33.3	36.9	21.8	56.1	28.9
$\mathcal{C}(target, target)$	47.2	25.7	64.1	8.2	71.6	55.8	**92.2**	**9.0**	56.4	24.5	37.2	18.3	50.1	26.4	59.8	24.0
$\mathcal{C}(finetune, target)$	61.4	51.2	69.0	23.2	78.2	77.2	**92.2**	**9.0**	64.7	44.6	49.6	43.3	54.7	36.8	67.1	40.8
$\mathcal{C}(all, target)$	57.6	48.7	65.2	20.3	71.1	64.4	90.3	**9.1**	60.3	42.3	47.8	41.2	51.1	35.3	63.3	37.3
fastText																
$\mathcal{C}(source, target)$	57.5	46.8	57.1	23.1	62.9	54.6	83.5	**8.9**	54.1	39.5	49.2	45.5	38.5	24.9	57.5	34.8
$\mathcal{C}(target, target)$	62.4	50.4	69.2	18.3	77.6	74.2	**92.1**	**9.0**	63.8	39.5	49.4	40.8	57.4	34.0	67.4	38.0
$\mathcal{C}(finetune, target)$	62.5	57.5	68.6	30.1	77.8	78.6	88.6	**9.7**	65.8	53.3	51.4	47.6	59.0	46.7	67.7	46.2
$\mathcal{C}(all, target)$	61.8	55.3	66.8	30.4	75.2	75.3	88.1	**9.8**	63.1	51.6	52.5	49.2	57.1	45.1	66.4	45.2
BERT																
$\mathcal{C}(source, target)$	63.7	57.9	63.2	29.2	76.1	75.3	83.9	8.7	63.4	54.8	54.3	51.3	49.2	38.6	64.8	45.1
$\mathcal{C}(target, target)$	**68.0**	56.8	**71.4**	23.5	**84.4**	**84.6**	**92.2**	**9.0**	70.9	52.6	**59.4**	55.9	**65.3**	44.6	**73.1**	46.7
$\mathcal{C}(finetune, target)$	**69.4**	**67.0**	**72.1**	**37.4**	**84.6**	**85.5**	**92.2**	**9.2**	**72.9**	**65.2**	**61.4**	**59.3**	**66.7**	**55.6**	**74.2**	**54.2**
$\mathcal{C}(all, target)$	**69.9**	**67.0**	**71.2**	**34.7**	**83.9**	**85.2**	90.4	**9.3**	71.1	62.3	**61.4**	**59.3**	**66.3**	**55.6**	**73.5**	**53.3**
Majority ($target$)	37.9	5.1	64.8	7.4	28.3	4.2	**92.2**	8.7	44.5	5.7	27.9	4.4	26.4	4.0	46.0	5.6

B.2 Composition of $\mathcal{T}_{source}$

In Section 3.1, we mention that in our experiments $\mathcal{T}_{target}$ is always composed of one dataset of the MFTC, while we test with $\mathcal{T}_{source}$ being composed of one, three, or six datasets. In the main paper we present the results where $\mathcal{T}_{source}$ is composed of six datasets. Here, we present the results where it is composed of one or three datasets, using BERT.

B.2.1 One Dataset as $\mathcal{T}_{source}$

Not all the settings described in Section 3.1 can be meaningfully replicated when $\mathcal{T}_{source}$ is composed of just one dataset. For instance, $\mathcal{C}(source, source)$ and $\mathcal{C}(target, target)$ would coincide, as well as $\mathcal{C}(source, target)$ and $\mathcal{C}(target, source)$. Thus, in Tables B3, B4, and B5 we present the results along the lines of generalizability, transferability, and catastrophic forgetting, respectively. When possible, we compare the results to the results pre-

Table B3: Generalizability: the model is trained on $\mathcal{T}_{source}$ and evaluated on $\mathcal{T}_{target}$.

$\mathcal{T}_{target} \rightarrow$	**ALM**		**BLT**		**BLM**		**DAV**		**ELE**		**MT**		**SND**	
$\mathcal{T}_{source} \downarrow$	*m*	*M*	*m*	*M*	*m*	*M*	*m*	*M*	*m*	*M*	*m*	*M*	*m*	*M*
ALM	-	-	**65.6**	21.3	72.0	55.4	87.2	8.5	58.4	30.3	45.1	33.1	44.8	24.2
BLT	33.4	11.4	-	-	36.0	17.6	**90.9**	8.6	44.9	8.4	26.9	9.2	30.3	7.3
BLM	**64.1**	53.6	64.2	21.6	-	-	86.4	8.4	**65.2**	49.7	49.7	43.3	44.5	30.4
DAV	35.8	4.9	63.0	7.3	25.3	3.9	-	-	46.6	6.0	27.8	4.5	25.2	3.9
ELE	53.7	35.2	63.5	22.7	60.8	49.8	85.8	**9.6**	-	-	48.4	41.3	47.3	30.8
MT	47.9	43.8	58.8	20.5	54.9	48.3	49.9	6.0	54.7	41.9	-	-	41.5	29.2
SND	47.7	33.5	54.8	22.6	50.6	37.2	79.1	8.6	48.9	33.6	42.8	35.1	-	-
Six	**63.7**	**57.9**	63.2	**29.2**	**76.1**	**75.3**	83.9	8.7	63.4	**54.8**	**54.3**	**51.3**	**49.2**	**38.6**

Table B4: Transferability: the model is trained on $\mathcal{T}_{source}$, retrained on $\mathcal{T}_{target}$, and evaluated on $\mathcal{T}_{target}$.

$\mathcal{T}_{target} \rightarrow$	**ALM**		**BLT**		**BLM**		**DAV**		**ELE**		**MT**		**SND**	
$\mathcal{T}_{source} \downarrow$	*m*	*M*	*m*	*M*	*m*	*M*	*m*	*M*	*m*	*M*	*m*	*M*	*m*	*M*
ALM	-	-	**74.3**	31.8	**85.3**	**86.0**	89.8	8.6	72.4	62.7	**61.1**	**58.8**	67.4	54.5
BLT	**69.4**	58.0	-	-	82.9	83.6	91.7	8.7	72.1	62.7	**58.4**	55.4	65.2	47.2
BLM	66.9	60.8	72.6	33.4	-	-	**92.5**	8.8	72.4	**66.9**	**61.0**	59.1	**68.8**	**62.6**
DAV	23.7	13.8	68.1	16.9	56.2	43.9	-	-	46.9	33.1	29.6	16.2	46.6	25.5
ELE	68.6	61.1	72.1	**36.2**	82.9	83.5	**92.5**	8.8	-	-	60.0	**58.7**	66.9	53.6
MT	66.7	60.2	72.9	**36.4**	83.8	84.1	90.1	8.6	**73.4**	61.1	-	-	65.7	52.5
SND	**69.6**	**66.7**	**73.9**	**34.7**	83.6	85.1	91.9	8.7	68.7	58.8	**60.7**	56.4	-	-
Six	**69.4**	**67.0**	**72.1**	**37.4**	**84.6**	**85.5**	**92.2**	**9.2**	**72.9**	**65.2**	**61.4**	**59.3**	66.7	55.6

Table B5: Catastrophic forgetting: the model is trained on $\mathcal{T}_{source}$, retrained on $\mathcal{T}_{target}$, and evaluated on $\mathcal{T}_{source}$.

$\mathcal{T}_{target} \rightarrow$	**ALM**		**BLT**		**BLM**		**DAV**		**ELE**		**MT**		**SND**		**No retrain**	
$\mathcal{T}_{source} \downarrow$	*m*	*M*	*m*	*M*	*m*	*M*	*m*	*M*	*m*	*M*	*m*	*M*	*m*	*M*	*m*	*M*
ALM	-	-	48.4	34.6	**67.0**	**64.3**	49.2	24.3	60.6	55.2	57.2	52.5	60.1	57.7	**68.0**	56.8
BLT	66.0	24.0	-	-	65.7	25.8	67.6	12.7	64.8	**28.6**	62.5	**28.9**	57.6	25.7	**71.4**	23.5
BLM	79.4	79.8	60.2	55.4	-	-	52.2	40.5	77.7	78.1	74.9	74.5	74.5	76.6	**84.4**	**84.6**
DAV	45.1	4.3	**91.5**	**8.7**	70.3	6.9	-	-	59.9	6.3	45.0	4.9	63.1	6.6	**92.2**	**9.0**
ELE	67.6	48.0	57.8	33.1	**70.0**	**55.3**	46.5	8.4	-	-	63.8	**56.7**	59.8	52.8	**70.9**	52.6
MT	51.3	45.0	40.2	28.2	55.4	50.8	28.4	5.1	55.8	52.2	-	-	54.3	51.0	**59.4**	**55.9**
SND	54.0	37.5	39.9	20.4	55.8	41.7	26.9	4.4	55.0	43.3	57.4	**47.3**	-	-	**65.3**	44.6

sented in the paper (where $\mathcal{T}_{source}$ is composed of six datasets). As in the paper, we highlight in bold the best result and the results that are not significantly different from it.

Generalizability To evaluate generalizability (Table B3), the model is trained on $\mathcal{T}_{source}$ and evaluated on $\mathcal{T}_{target}$, akin to the $\mathcal{C}(source, target)$ setting described in the paper. Thus, at the end of the table, we append the results of $\mathcal{C}(source, target)$ from Table 4 (where $\mathcal{T}_{source}$ is composed of six datasets). First, we notice that the results are generally better when $\mathcal{T}_{source}$ is composed of six datasets. Further, there is no dataset that stands out as clearly better than the other six in generalizability.

Transferability To evaluate transferability (Table B4), the model is trained on $\mathcal{T}_{source}$, retrained on $\mathcal{T}_{target}$, and evaluated on $\mathcal{T}_{target}$, akin to the $\mathcal{C}(finetune, target)$ setting described in the paper. Thus, at the end of the table, we append the results of $\mathcal{C}(finetune, target)$ from Table 4 (where $\mathcal{T}_{source}$ is composed of six datasets). First, we notice that the results are generally better or on par to the results where $\mathcal{T}_{source}$ is composed of six datasets. Further, there is no dataset that stands out as clearly better than the other six in transferability. These two aspects suggest that a combination of the six datasets as $\mathcal{T}_{source}$ consistently leads to better transferability results.

Catastrophic Forgetting To evaluate catastrophic forgetting (Table B5), the model is trained on $\mathcal{T}_{source}$, retrained on $\mathcal{T}_{target}$, and evaluated on $\mathcal{T}_{source}$, akin to the $\mathcal{C}(finetune, source)$ setting described in the paper. However, we cannot compare the results with the $\mathcal{C}(finetune, source)$

setting, as the evaluation sets differ (one dataset in Table B5, six datasets in $\mathcal{C}(finetune, source)$ in Table 4). However, we compare to the case where the model is only trained on $\mathcal{T}_{source}$. Differently from the previous tables, the evaluation sets (i.e., $\mathcal{T}_{source}$) are consistent in every row, not in every column. Thus, we highlight the best results per row. It is evident that catastrophic forgetting happens even when $\mathcal{T}_{source}$ is composed of one dataset. Further, there is no dataset that stands out as better than the other six in mitigating forgetting.

B.2.2 Three Datasets as $\mathcal{T}_{source}$

When employing three datasets as $\mathcal{T}_{source}$, the settings described in Section 3.1 can be meaningfully reproduced. However, the selection of the three datasets (out of the six available at each experiment) that compose $\mathcal{T}_{source}$ is not trivial. Experimenting with all possible combinations would result in $\frac{6!}{3!(6-3)!} = 20$ experiments per setting. In order to simplify the experiments, we decide to test with only one combination of three datasets, selected as the best performing combination from the experiments in Section B.2.1. We average the results of Tables B3, B4, and B5, and for each dataset used as $\mathcal{T}_{target}$, we select the three datasets that led to the best average performance. Due to the class imbalance of all datasets, one of the biggest challenges is to achieve good performances across all values. Thus, we decide to consider only the average macro F_1-scores. We report the best resulting datasets in Table B6—for each dataset that we use as $\mathcal{T}_{target}$ in the following experiments, we use the indicated three datasets as $\mathcal{T}_{source}$.

Table B6: The three datasets used as $\mathcal{T}_{source}$ in Table B7

$\mathcal{T}_{target}$	$\mathcal{T}_{source}$
ALM	BLM, MT, SND
BLT	ELE, MT, SND
BLM	ALM, ELE, MT
DAV	BLT, BLM, ELE
ELE	BLM, MT, SND
MT	BLM, ELE, SND
SND	BLM, ELE, MT

Table B7 reports the complete cross-domain evaluation results, analogously to Table 4. For further comparison, we add the results from Table 4 (where $\mathcal{T}_{source}$ is composed of six datasets). The results in the bottom half of the table can be directly compared, as in each column the model is evaluated on the same test set. However, the results on the top half cannot be directly compared, as the model is evaluated on different test sets (three and six datasets, respectively).

It is evident that the results are consistent with the results presented in the main paper. In the top half of the table, the best performing settings are $\mathcal{C}(source, source)$ and $\mathcal{C}(all, source)$, both when $\mathcal{T}_{source}$ is composed of three and six datasets. In the bottom half, where the results can be directly compared, we notice that the best performing settings are consistent, and lead to comparable results.

We conclude that selecting the three best performing datasets as $\mathcal{T}_{source}$ has neither advantage nor disadvantage over selecting all six datasets. However, selecting all six allows for a consistent evaluation, where all MFTC datasets are used in all evaluation settings, thus avoiding the arbitrary choice of datasets to be used as $\mathcal{T}_{source}$ that we described at the beginning of this section.

Table B7: Results of the four training scenarios evaluated on $\mathcal{T}_{source}$ and $\mathcal{T}_{target}$, when $\mathcal{T}_{source}$ is composed of three or six datasets. The columns indicate the dataset used as $\mathcal{T}_{target}$. We report both micro F_1-score (m, left column) and macro F_1-score (M, right column).

	ALM		**BLT**		**BLM**		**DAV**		**ELE**		**MT**		**SND**		**Average**	
Classifier Setting	*m*	*M*	*m*	*M*	*m*	*M*	*m*	*M*	*m*	*M*	*m*	*M*	*m*	*M*	*m*	*M*
Three datasets as $\mathcal{T}_{source}$																
$\mathcal{C}(source, source)$	**70.9**	**68.8**	**66.1**	**62.5**	**67.2**	**63.4**	**76.1**	**70.3**	**71.2**	**69.1**	**75.0**	**71.8**	**72.4**	**69.6**	**71.3**	**67.9**
$\mathcal{C}(target, source)$	52.8	40.7	34.2	8.8	59.1	49.4	46.3	6.0	52.9	44.3	50.6	43.8	48.3	37.3	49.2	32.9
$\mathcal{C}(finetune, source)$	64.1	59.4	50.9	38.5	65.0	58.8	58.7	34.6	66.9	63.7	68.2	65.7	65.0	62.7	62.7	54.8
$\mathcal{C}(all, source)$	**70.9**	**69.1**	**66.3**	**62.6**	**67.1**	**63.4**	**75.9**	**69.8**	**70.9**	**68.9**	**74.6**	**71.5**	**72.5**	**69.7**	**71.2**	**67.9**
Six datasets as $\mathcal{T}_{source}$																
$\mathcal{C}(source, source)$	**73.9**	**65.6**	**73.9**	**68.3**	**71.2**	**61.8**	**71.1**	**66.4**	**73.3**	**66.4**	**75.7**	**68.0**	**74.5**	**66.5**	**73.4**	**66.1**
$\mathcal{C}(target, source)$	61.6	37.7	43.8	13.1	62.6	43.0	38.8	5.1	59.3	40.4	52.4	39.1	54.4	36.6	53.3	30.7
$\mathcal{C}(finetune, source)$	70.3	57.2	61.2	47.8	69.2	54.9	56.6	41.9	70.5	61.5	67.7	60.5	68.0	60.8	66.2	54.9
$\mathcal{C}(all, source)$	**73.7**	**65.6**	**73.7**	**68.0**	**71.3**	**62.1**	**71.0**	**66.4**	**73.6**	**66.7**	**75.6**	**67.7**	**74.3**	**66.6**	**73.3**	**66.2**
Three datasets as $\mathcal{T}_{source}$																
$\mathcal{C}(source, target)$	64.8	58.9	61.4	26.6	77.1	74.5	85.3	**8.8**	60.0	54.7	54.9	51.7	51.3	41.1	65.0	45.2
$\mathcal{C}(target, target)$	68.1	56.8	**71.1**	23.3	**83.8**	**84.2**	**92.2**	8.7	71.0	53.6	59.1	54.9	65.2	44.7	**72.9**	46.6
$\mathcal{C}(finetune, target)$	**70.1**	**67.4**	**72.6**	**37.4**	**84.9**	**85.4**	**92.2**	8.7	**72.9**	**64.7**	**61.2**	**59.6**	**68.0**	**58.3**	**74.5**	**54.5**
$\mathcal{C}(all, target)$	**69.6**	**66.2**	71.2	**35.0**	**84.0**	**85.1**	91.0	**9.3**	**71.7**	**64.2**	**61.0**	**59.2**	**67.8**	**58.3**	**73.7**	**53.9**
Six datasets as $\mathcal{T}_{source}$																
$\mathcal{C}(source, target)$	63.7	57.9	63.2	29.2	76.1	75.3	83.9	8.7	63.4	54.8	54.3	51.3	49.2	38.6	64.8	45.1
$\mathcal{C}(target, target)$	**68.0**	56.8	**71.4**	23.5	**84.4**	**84.6**	**92.2**	**9.0**	70.9	52.6	**59.4**	55.9	**65.3**	44.6	**73.1**	46.7
$\mathcal{C}(finetune, target)$	**69.4**	**67.0**	**72.1**	**37.4**	**84.6**	**85.5**	**92.2**	**9.2**	**72.9**	**65.2**	**61.4**	**59.3**	**66.7**	**55.6**	**74.2**	**54.2**
$\mathcal{C}(all, target)$	**69.9**	**67.0**	**71.2**	**34.7**	**83.9**	**85.2**	90.4	**9.3**	71.1	62.3	**61.4**	**59.3**	**66.3**	**55.6**	**73.5**	**53.3**
Majority ($target$)	37.9	5.1	64.8	7.4	28.3	4.2	**92.2**	8.7	44.5	5.7	27.9	4.4	26.4	4.0	46.0	5.6