# Adaptable Text Matching via Meta-Weight Regulator

Bo Zhang, Chen Zhang, Fang Ma and Dawei Song

Beijing Institute of Technology

# Background

1. Deep learning-based models

   - Data-hungry
   - Strong dependency on training data distribution (domain)

2. Text Matching Data

   - Deep semantics
   - Difficulty in collecting and labeling

# Motivation

How to train a deep text matching model in the few-shot learning setting?

Source Domain Data → Distribution Gap → Poor Performance → Adaptability

# Related Work

1. Adaptation methods
   - Merging the source and target data directly for training (Koehn et al., 2017)
   - Learning a domain classifier to confuse domains adversarially (Cohen et al., 2018)
   - A reinforcement learning-based data selector for cross-domain transfer learning (Qu et al., 2019)

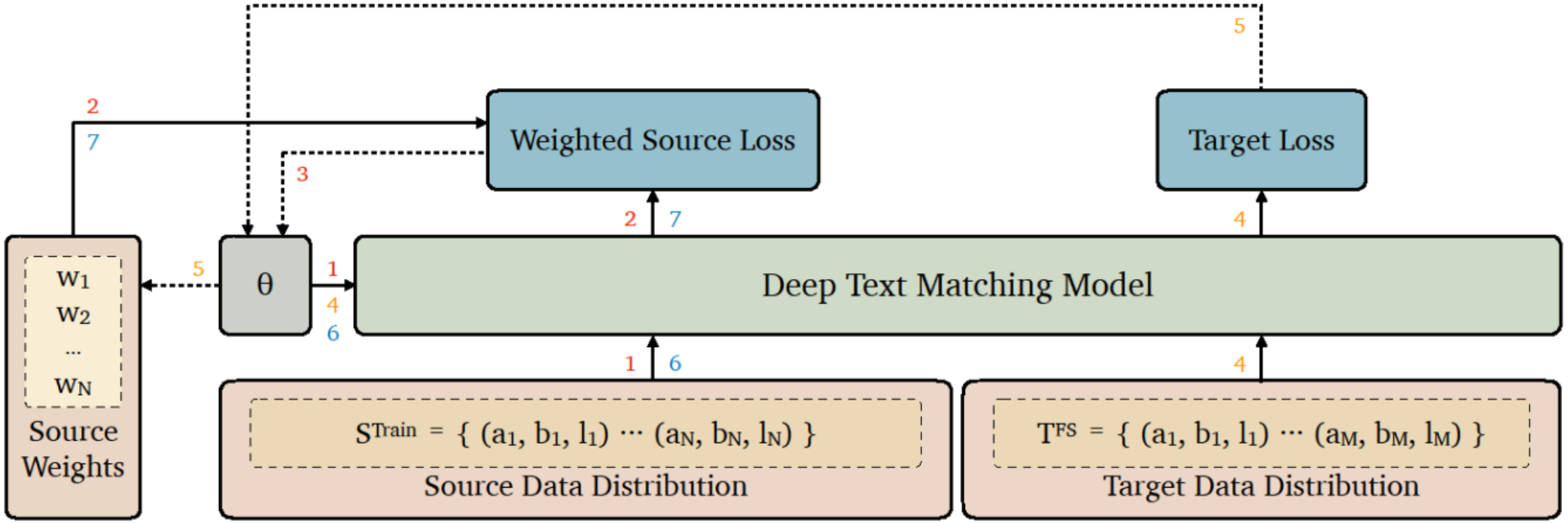
2. Weighting examples
   - Importance sampling is a classic method in statistics that assigns weights to samples to match one distribution with another (Kahn et al., 1952)
   - Focal loss adds a soft weighting scheme that emphasizes harder examples (He et al., 2017)
   - Learning to reweight examples for regularization of noisy and corrupted labels (Ren et al., 2019)

# Application Scenarios

1. Adaptation methods
   - Merging the source and target data directly for training (Koehn et al., 2017)
   - Learning a domain classifier to confuse domains adversarially (Cohen et al., 2018)
   - A reinforcement learning-based data selector for cross-domain transfer learning (Qu et al., 2019)

2. Weighting examples
   - Importance sampling is a classic method in statistics that assigns weights to samples to match one distribution with another (Kahn et al., 1952)
   - Focal loss adds a soft weighting scheme that emphasizes harder examples (He et al., 2017)
   - Learning to reweight examples for regularization of noisy and corrupted labels (Ren et al., 2019)

# Meta-Weight Regulator

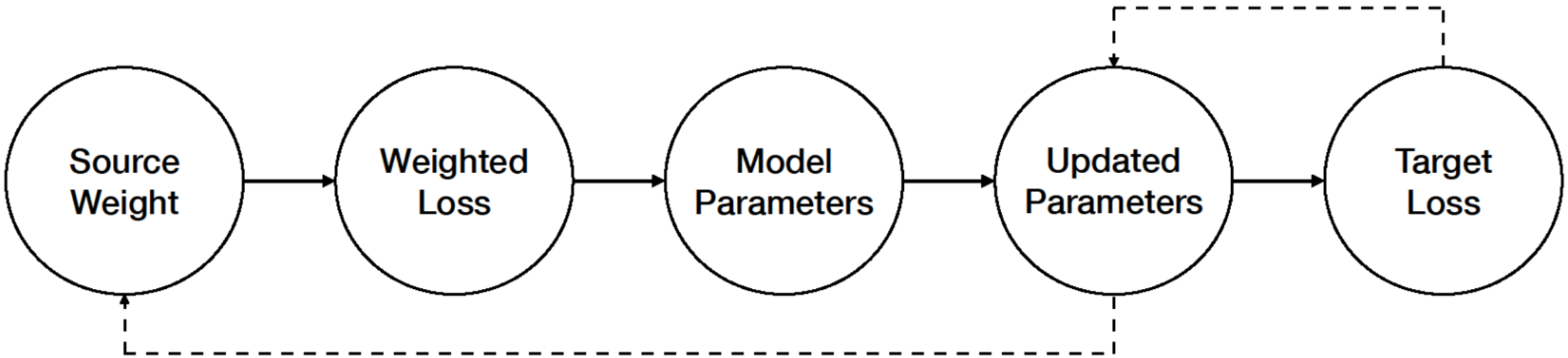



I. Establishing a connection in the computational graph (1, 2, 3)

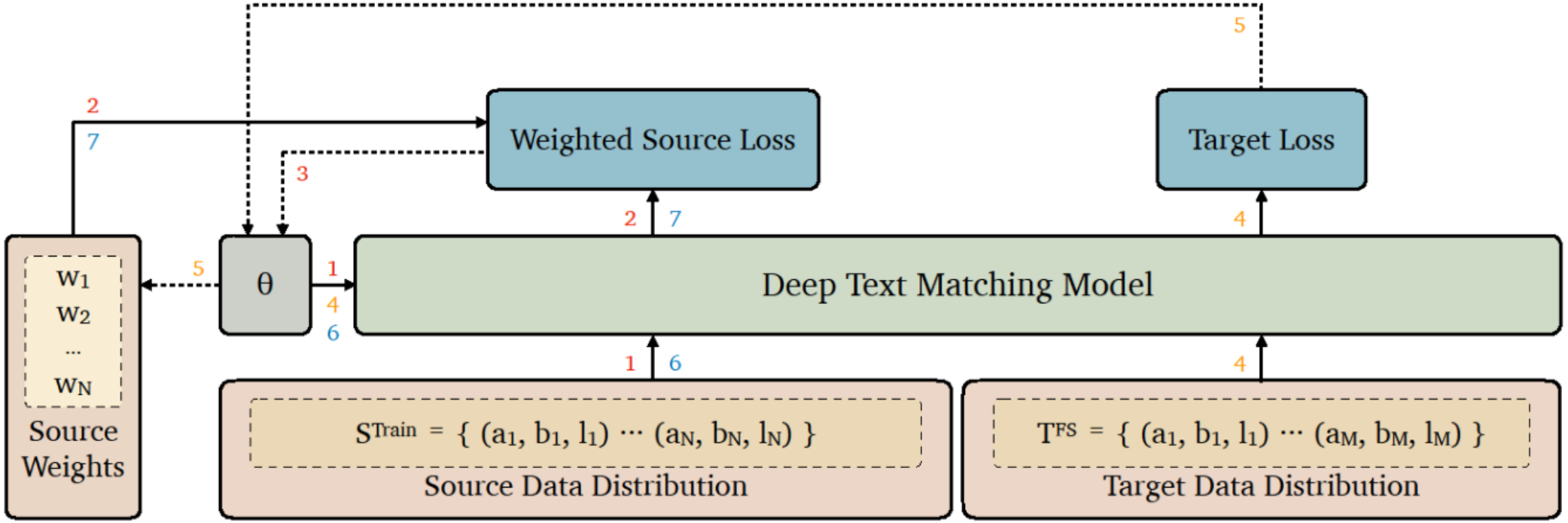
II. Regulating weights through meta-gradient descent (4, 5)

III. Training text matching models on weighted samples (6, 7)

# The Path of Computational Graph



# Establishing a connection in the computational graph



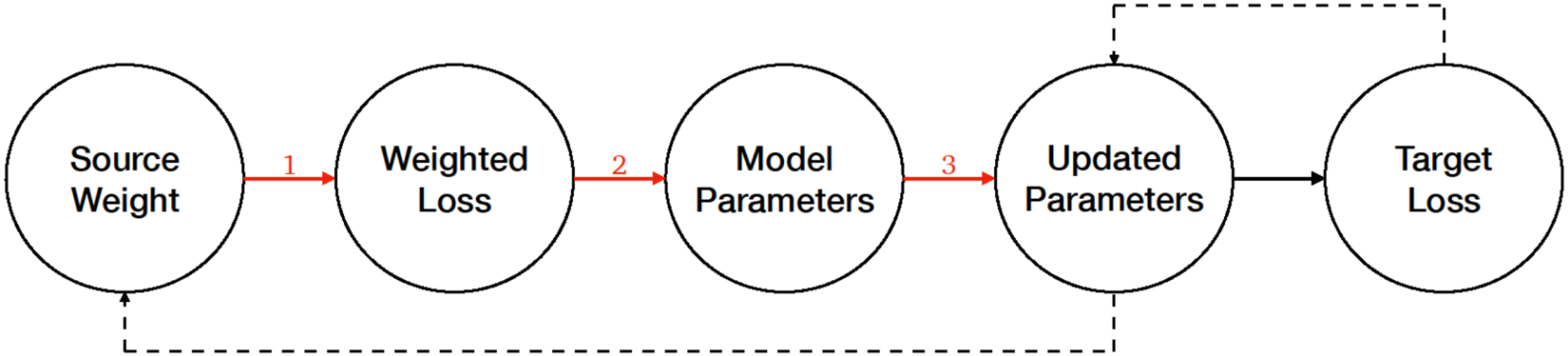

1. $y_i^s = TMM^s(a_i^s, b_i^s, \theta)$

2. $Loss^s(y^s, l^s) = \sum_{i=1}^{N} w_i^s Cost^s(y_i^s, l_i^s)$

3. $\tilde{\theta} = \theta - \alpha \cdot \frac{\partial Loss^s(y^s, l^s)}{\partial \theta}$

$= \theta - \alpha \cdot \frac{\partial \sum_{i=1}^{N} w_i^s Cost^s(y_i^s, l_i^s)}{\partial \theta}$

# Establishing a connection in the computational graph



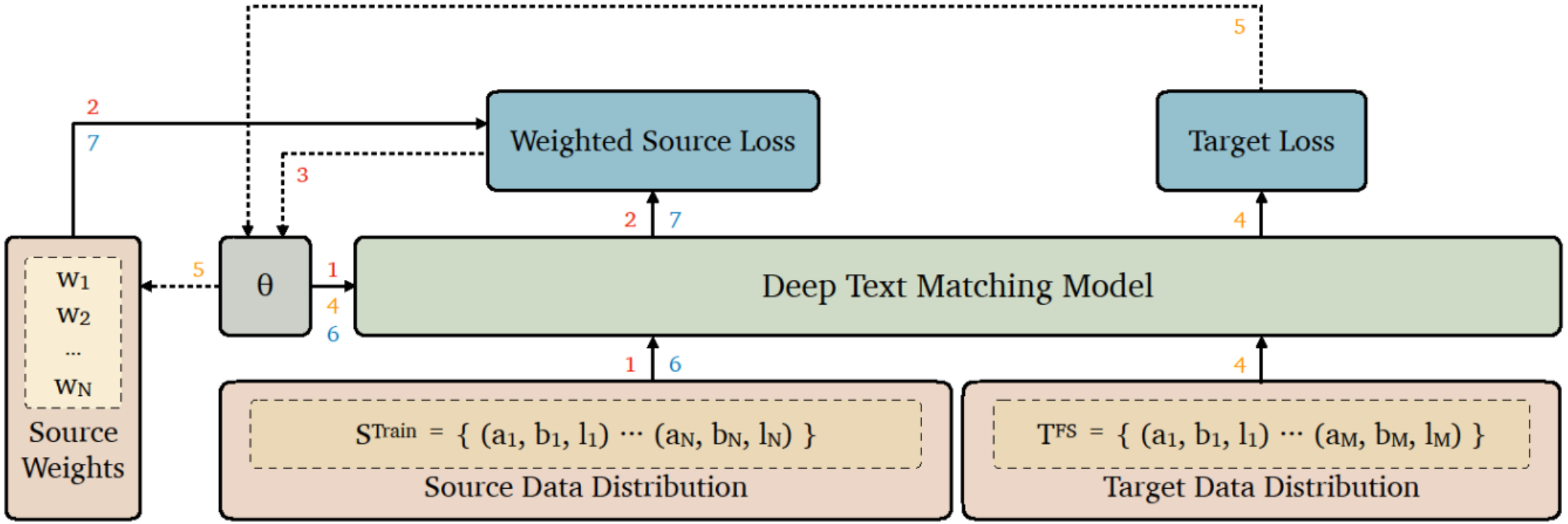

1. $y_i^s = TMM^s(a_i^s, b_i^s, \theta)$

2. $Loss^s(y^s, l^s) = \sum_{i=1}^{N} w_i^s Cost^s(y_i^s, l_i^s)$

3. $\tilde{\theta} = \theta - \alpha \cdot \frac{\partial Loss^s(y^s, l^s)}{\partial \theta}$

$= \theta - \alpha \cdot \frac{\partial \sum_{i=1}^{N} w_i^s Cost^s(y_i^s, l_i^s)}{\partial \theta}$

# Regulating weights through meta-gradient descent



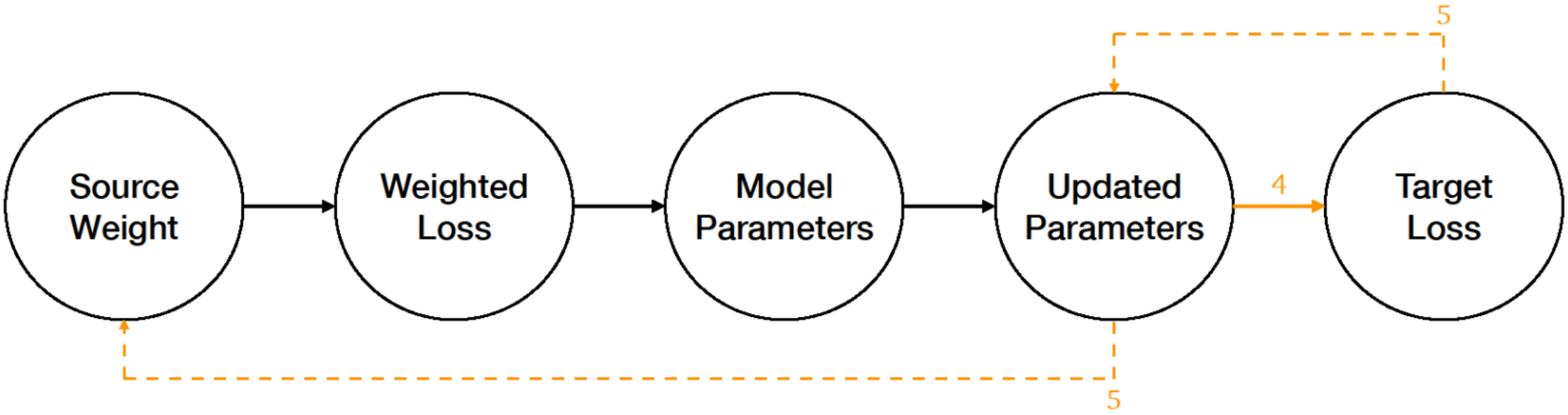

4. $Loss^t(y^t, l^t) = \sum_{j=1}^{M} Cost^t\left(y_j^t, l_j^t\right)$

$= \sum_{j=1}^{M} Cost^t\left(TMM^t\left(a_j^t, b_j^t, \tilde{\theta}\right), l_j^t\right)$

5. $\widetilde{w^s} = w^s - \alpha \cdot \frac{\partial^2 Loss^t(y^t, l^t)}{\partial^2 w^s}$

$= w^s - \alpha \cdot \frac{\partial\left(\frac{\partial Loss^t(y^t, l^t)}{\partial \tilde{\theta}} \cdot \frac{\partial \tilde{\theta}}{\partial w^s}\right)}{\partial w^s}$

# Regulating weights through meta-gradient descent



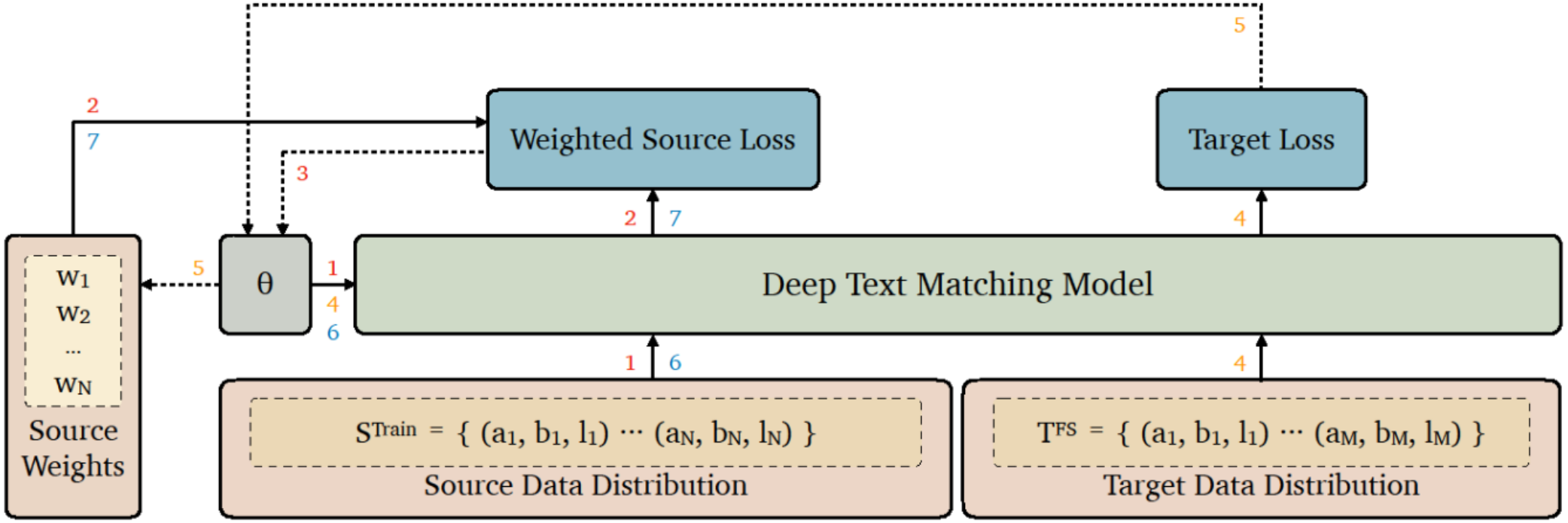

4. $Loss^t(y^t, l^t) = \sum_{j=1}^{M} Cost^t(y_j^t, l_j^t)$

$= \sum_{j=1}^{M} Cost^t(TMM^t(a_j^t, b_j^t, \tilde{\theta}), l_j^t)$

5. $\widetilde{w^s} = w^s - \alpha \cdot \frac{\partial^2 Loss^t(y^t, l^t)}{\partial^2 w^s}$

$= w^s - \alpha \cdot \frac{\partial \left( \frac{\partial Loss^t(y^t, l^t)}{\partial \tilde{\theta}} \cdot \frac{\partial \tilde{\theta}}{\partial w^s} \right)}{\partial w^s}$

# Training text matching models on weighted samples




6. $y_i^s = TMM^s(a_i^s, b_i^s, \theta)$

7. $Loss^s(y, l) = \sum_{i=1}^{N} \widetilde{w_i^s} \cdot Cost^s(y_i^s, l_i^s)$

# Experiments

1. Tasks and datasets
   - Question Answering (TrecQA, WikiQA)
   - Natural Language Inference (SNLI, SciTail)
2. The Categories of Experiments
   - Adaptation distribution gap (cross-dataset, cross-task)
   - Target domain data size (10-shot, 50-shot, 100-shot and 1000-shot)
3. Baselines
   - Backbone model (BERT)
   - Data Merging, Fine Tuning
   - Adversarial Domain Confusion, Reinforced Transfer Learning

# Intra-task Cross-dataset Adaptation Experiment

| Target Size | Methods | Natural Language Inference | | Question Answering | |
|---|---|---|---|---|---|
| | | SciTail → SNLI | SNLI → SciTail | WikiQA → TrecQA | TrecQA → WikiQA |
| 10-shot | Backbone | 0.5391 | 0.5502 | 0.5307 | 0.5197 |
| | Fine Tuning | 0.5390 | 0.5505 | 0.5379 | 0.5306 |
| | Data Merging | 0.5392 | 0.5499 | 0.5304 | 0.5299 |
| | Reinforced Transfer Learning[27] | 0.5675 | 0.5796 | 0.5603 | 0.5538 |
| | Adversarial Domain Confusion[7] | 0.5634 | 0.5426 | 0.5406 | 0.5495 |
| | Meta-Weight Regulator | **0.5769**$^{\dagger}$ | **0.5887**$^{\dagger}$ | **0.5768**$^{\dagger}$ | **0.5796**$^{\dagger}$ |
| 50-shot | Backbone | 0.5456 | 0.5551 | 0.5376 | 0.5285 |
| | Fine Tuning | 0.5463 | 0.5579 | 0.5380 | 0.5396 |
| | Data Merging | 0.5406 | 0.5288 | 0.5377 | 0.5297 |
| | Reinforced Transfer Learning[27] | 0.5725 | 0.5801 | 0.5632 | 0.5673 |
| | Adversarial Domain Confusion[7] | 0.5642 | 0.5536 | 0.5411 | 0.5502 |
| | Meta-Weight Regulator | **0.5865**$^{\dagger}$ | **0.5927**$^{\dagger}$ | **0.5858**$^{\dagger}$ | **0.5833**$^{\dagger}$ |
| 100-shot | Backbone | 0.5697 | 0.5761 | 0.5678 | 0.5658 |
| | Fine Tuning | 0.5701 | 0.5763 | 0.5676 | 0.5669 |
| | Data Merging | 0.5698 | 0.5759 | 0.5697 | 0.5663 |
| | Reinforced Transfer Learning[27] | 0.5909 | 0.5976 | 0.5837 | 0.5885 |
| | Adversarial Domain Confusion[7] | 0.5797 | 0.5857 | 0.5703 | 0.5812 |
| | Meta-Weight Regulator | **0.6066**$^{\dagger}$ | **0.6095**$^{\dagger}$ | **0.6011**$^{\dagger}$ | **0.6003**$^{\dagger}$ |
| 1000-shot | Backbone | 0.5854 | 0.5971 | 0.5867 | 0.5883 |
| | Fine Tuning | 0.5906 | 0.6051 | 0.5970 | 0.5997 |
| | Data Merging | 0.5835 | 0.5976 | 0.5879 | 0.5895 |
| | Reinforced Transfer Learning[27] | 0.6169 | 0.6206 | 0.6007 | 0.6053 |
| | Adversarial Domain Confusion[7] | 0.6078 | 0.6103 | 0.6202 | **0.6205** |
| | Meta-Weight Regulator | **0.6309**$^{\dagger}$ | **0.6397**$^{\dagger}$ | **0.6278**$^{\dagger}$ | **0.6205** |

# Cross-task Adaptation Experiment

| Target Size | Methods | NLI → QA | | | | QA → NLI | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | S → W | S → T | N → W | N → T | W → S | W → N | T → S | T → N |
| 10-shot | Backbone | 0.5302 | 0.5323 | 0.5412 | 0.5436 | 0.5252 | 0.5238 | 0.5150 | 0.5178 |
| | Fine Tuning | 0.5365 | 0.5318 | 0.5465 | 0.5512 | 0.5201 | 0.5289 | 0.5189 | 0.5269 |
| | Data Merging | 0.5356 | 0.5281 | 0.5496 | 0.5423 | 0.5275 | 0.5285 | 0.5104 | 0.5207 |
| | Reinforced Transfer Learning[27] | 0.5525 | 0.5601 | 0.5652 | 0.5665 | 0.5501 | 0.5573 | 0.5592 | 0.5590 |
| | Adversarial Domain Confusion[7] | 0.5521 | 0.5599 | 0.5525 | 0.5536 | 0.5395 | 0.5410 | 0.5421 | 0.5496 |
| | Meta-Weight Regulator | **0.5721**† | **0.5710** | **0.5855**† | **0.5867**† | **0.5755**† | **0.5732** | **0.5725** | **0.5759**† |
| 50-shot | Backbone | 0.5321 | 0.5356 | 0.5499 | 0.5472 | 0.5271 | 0.5298 | 0.5199 | 0.5210 |
| | Fine Tuning | 0.5375 | 0.5217 | 0.5412 | 0.5524 | 0.5203 | 0.5299 | 0.5168 | 0.5277 |
| | Data Merging | 0.5376 | 0.5210 | 0.5510 | 0.5322 | 0.5279 | 0.5291 | 0.5201 | 0.5215 |
| | Reinforced Transfer Learning[27] | 0.5652 | 0.5699 | 0.5719 | 0.5705 | 0.5519 | 0.5602 | 0.5657 | 0.5612 |
| | Adversarial Domain Confusion[7] | 0.5647 | 0.5691 | 0.5601 | 0.5599 | 0.5397 | 0.5413 | 0.5425 | 0.5501 |
| | Meta-Weight Regulator | **0.5807** | **0.5821** | **0.5911**† | **0.5904**† | **0.5897**† | **0.5845**† | **0.5823** | **0.5929**† |
| 100-shot | Backbone | 0.5521 | 0.5507 | 0.5676 | 0.5689 | 0.5563 | 0.5621 | 0.5669 | 0.5687 |
| | Fine Tuning | 0.5537 | 0.5523 | 0.5670 | 0.5702 | 0.5615 | 0.5645 | 0.5719 | 0.5697 |
| | Data Merging | 0.5536 | 0.5492 | 0.5563 | 0.5613 | 0.5578 | 0.5601 | 0.5691 | 0.5539 |
| | Reinforced Transfer Learning[27] | 0.5755 | 0.5815 | 0.5896 | 0.5905 | 0.5776 | 0.5791 | 0.5756 | 0.5801 |
| | Adversarial Domain Confusion[7] | 0.5779 | 0.5796 | 0.5834 | 0.5826 | 0.5779 | 0.5793 | 0.5757 | 0.5812 |
| | Meta-Weight Regulator | **0.5965** | **0.6017**† | **0.5979** | **0.6061**† | **0.6003**† | **0.6013**† | **0.5999**† | **0.6198**† |
| 1000-shot | Backbone | 0.5619 | 0.5625 | 0.5897 | 0.5901 | 0.5768 | 0.5724 | 0.5779 | 0.5791 |
| | Fine Tuning | 0.5634 | 0.5637 | 0.5870 | 0.6002 | 0.5914 | 0.5845 | 0.5896 | 0.5799 |
| | Data Merging | 0.5674 | 0.5729 | 0.5856 | 0.5997 | 0.5779 | 0.5809 | 0.5876 | 0.5693 |
| | Reinforced Transfer Learning[27] | 0.5976 | 0.5997 | 0.6103 | 0.6201 | 0.6059 | 0.6032 | 0.6075 | 0.6012 |
| | Adversarial Domain Confusion[7] | 0.6077 | 0.6089 | 0.6021 | 0.6187 | **0.6125** | 0.6017 | 0.6142 | 0.6099 |
| | Meta-Weight Regulator | **0.6146**† | **0.6157**† | **0.6275**† | **0.6296**† | 0.6112 | **0.6180**† | **0.6145** | **0.6301**† |

# Ablation Study

1. Backbone Models
   - Interaction-based text matching model
   - Representation-based text matching model
2. Source Weight Initialization
   - Random Initialization (vs **Zero Initialization**, vs One Initialization)
   - One Initialization (vs **Zero Initialization**, vs Random Initialization)
3. Effectiveness of MAML
   - MAML vs Reptile

# Summary

- The adaptability of the deep text matching model is improved significantly
- No constraints on source domain data
- No hyperparameters are introduced

# Thanks for listening