CROSSBOT: DRIFT-DECOUPLED ADAPTATION FOR CROSS-DOMAIN SOCIAL BOT DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

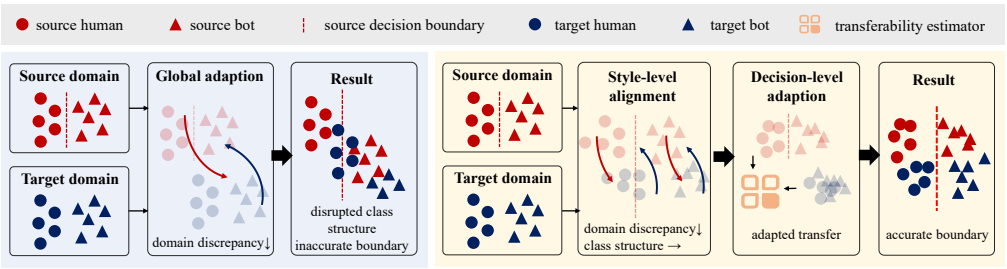
Cross-domain social bot detection suffers from substantial performance degradation due to differences in user behaviors, platform characteristics, and bot-generation strategies. Existing unsupervised domain adaptation methods generally model such variations as a single domain discrepancy and apply uniform alignment, which may remove class-discriminative cues or preserve unreliable source decisions. We propose **Drift-Decoupled Domain Adaptation (DDDA)**, a framework that decomposes cross-domain drift into representation-level style drift and decision-level drift. For style drift, DDDA introduces a *class-conditional contrastive style alignment mechanism* that reduces domain-specific expression differences while preserving discriminative information. For decision drift, DDDA develops a *transferability-guided routing mechanism* that estimates source decision reliability on unlabeled target domains and adaptively selects appropriate transfer strategies. Extensive experiments on three cross-domain social bot detection datasets demonstrate that DDDA consistently improves target-domain generalization over existing adaptation methods. Further analysis reveals diverse drift patterns across domains, highlighting the importance of drift-aware adaptation.

1 INTRODUCTION

Social bot detection plays a critical role in maintaining healthy online information ecosystems, as automated accounts can manipulate public opinion, amplify misinformation, and distort organic social interactions (Ferrara, 2023; Kaur et al., 2025). However, cross-domain deployment severely degrades detector performance due to disparities in platform environments, user behaviors, and bot-generation strategies (Ng & Carley, 2023; Qiao et al., 2026). Such discrepancies induce significant distribution shifts, making robust cross-domain knowledge transfer a fundamental challenge for social bot detection (Shi et al., 2025; Ferrara, 2023).

Unsupervised domain adaptation (UDA) provides a natural framework for addressing this challenge by transferring knowledge from labeled source domains to unlabeled target domains (Liu et al., 2022; Li et al., 2025). Recent studies have explored UDA methods for cross-domain social bot detection by reducing source-target distribution discrepancies and improving target domain generalization. For example, MGDIL (Qiao et al., 2026) combines domain-adversarial learning with cross-domain contrastive learning to learn domain-invariant and class-discriminative representations. BotSSCL (Akhtar et al., 2025) and OOD (Song et al., 2025) leverage self-supervised contrastive learning to improve representation robustness under distribution shifts. AdaBot (Sun et al., 2025) further incorporates distribution alignment, decision-boundary refinement, and source-signal annealing to alleviate transfer degradation. Despite these advances, existing approaches generally model cross-domain variation as a single domain discrepancy and apply a unified adaptation objective. This assumption overlooks that distribution shifts may arise from fundamentally different sources and therefore require different transfer strategies. As illustrated in Figure 1(a), global alignment can reduce the overall discrepancy between source and target domains, but may simultaneously remove class-discriminative information when domain-specific characteristics are also predictive of bot behaviors. Moreover, even when transferable representations are successfully learned, the decision rule that maps behaviors to labels may still vary across domains due to differences in platform policies, community norms, and bot-generation mechanisms.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Global adaption (b) Drift-decoupled adaption

Figure 1: Comparison of two adaptation paradigms for cross-domain social bot detection.

We argue that cross-domain social bot detection involves two coupled but distinct transfer factors: **representation-level style drift** and **decision-level drift**. Style drift describes changes in how user information is expressed across domains, including linguistic conventions, formatting patterns, hashtags, URLs, mentions, and other surface-level characteristics, while the underlying behavioral semantics remain relatively stable (Qiao et al., 2026). In contrast, decision drift describes changes in how observed user behaviors should be interpreted for bot detection, where the relationship between user characteristics and bot labels may shift across domains (Sun et al., 2025). These two drift factors are entangled in the observed source-target discrepancy: the same distribution difference may originate from changes in information expression or from changes in the underlying decision relationship (Alzahrani et al., 2025). Without target labels, these causes cannot be directly identified from the observed discrepancy alone. Therefore, effective adaptation requires determining *what should be aligned* and *what should be trusted*: representation variations should be reduced while preserving discriminative information, whereas source decision knowledge should be transferred according to its reliability on the target domain.

To address these challenges, we propose **Drift-Decoupled Domain Adaptation (DDDA)**, a framework that explicitly decomposes representation-level style drift from decision-level drift and enables robust cross-domain adaptation. Rather than forcing heterogeneous drift into a single adaptation objective, DDDA first introduces a *class-conditional contrastive style alignment mechanism*, which extracts style-sensitive residual representations and aligns them across domains while preserving class-discriminative structures. It then develops a *transferability-guided routing mechanism*, which estimates the reliability of source decision knowledge from different domains on unlabeled target data and adaptively selects appropriate transfer strategies. Experiments demonstrate that DDDA consistently improves target-domain generalization over existing adaptation methods. Our main contributions are summarized as follows:

- We propose DDDA, a drift-decoupled domain adaptation framework that decouples representation-level style drift from decision-level drift for robust cross-domain bot detection.
- We introduce a *class-conditional contrastive style alignment mechanism* to reduce transferable style variation while preserving discriminative information, and a *transferability-guided routing mechanism* to adaptively utilize source decision knowledge.
- We conduct extensive experiments on three datasets with diverse drift patterns to demonstrate the effectiveness of DDDA in cross-domain social bot detection.

2 Related Work

Unsupervised domain adaptation (UDA) transfers knowledge from labeled source domains to unlabeled target domains to alleviate distribution shifts (Liu et al., 2022; Wilson & Cook, 2020). Existing UDA methods mainly pursue transferable and discriminative representations through statistical distribution alignment (Long et al., 2018; Liu et al., 2019a; Zhang et al., 2024), adversarial learning (Chen et al., 2024; Pei et al., 2018), generative adaptation (Hsu et al., 2017; Hassan Pour Zonoozi et al., 2025), or contrastive learning (Oh & Bui, 2025; Wang et al., 2023).

For cross-domain social bot detection, recent studies have explored UDA from different perspectives. MGDIL (Qiao et al., 2026) combines domain-adversarial training with cross-domain

contrastive learning to learn domain-invariant and class-discriminative representations. BotSSCL (Akhtar et al., 2025) and OOD (Song et al., 2025) leverage self-supervised contrastive learning to improve representation robustness under distribution shifts. AdaBot (Sun et al., 2025) jointly performs distribution alignment, decision-boundary refinement, and source-signal annealing to alleviate transfer degradation. BotTrans (Shi et al., 2025) studies multi-source adaptation and selectively transfers knowledge from multiple source domains according to target relevance. While these methods improve cross-domain generalization, they typically do not explicitly distinguish the two sources of cross-domain drift: *representation-level style drift* and *decision-level drift*. This may distort class-discriminative representations or impose unreliable source decisions on the target domain. To address this limitation, we propose DDDA, which explicitly decouples style drift from decision drift, enabling robust cross-domain social bot detection.

3 METHOD

3.1 PROBLEM DEFINITION

We consider a multi-source UDA setting for cross-domain social bot detection. Let $\{\mathcal{D}_{S_k}\}_{k=1}^{K}$ denote K labeled source domains, where each source domain is sampled from an underlying distribution $\mathcal{D}_{S_k} \sim P_{S_k}(X, Y)$. Let $\mathcal{D}_T \sim P_T(X)$ denote an unlabeled target domain. Each user observation $x \in \mathcal{X}$ consists of the user's textual posts and their temporal organization, from which we further derive aggregated behavioral statistics. Then the objective is to learn a classifier $f : \mathcal{X} \to \mathcal{Y}$ from the labeled source domains that generalizes to the unlabeled target domain, where $\mathcal{Y} = \{0, 1\}$ and 0 and 1 denote human and bot users, respectively. Here, a domain is defined by the data-generating and collection environment represented by a dataset, which may reflect platform characteristics, user populations, bot-generation mechanisms, annotation processes, and other distributional factors.

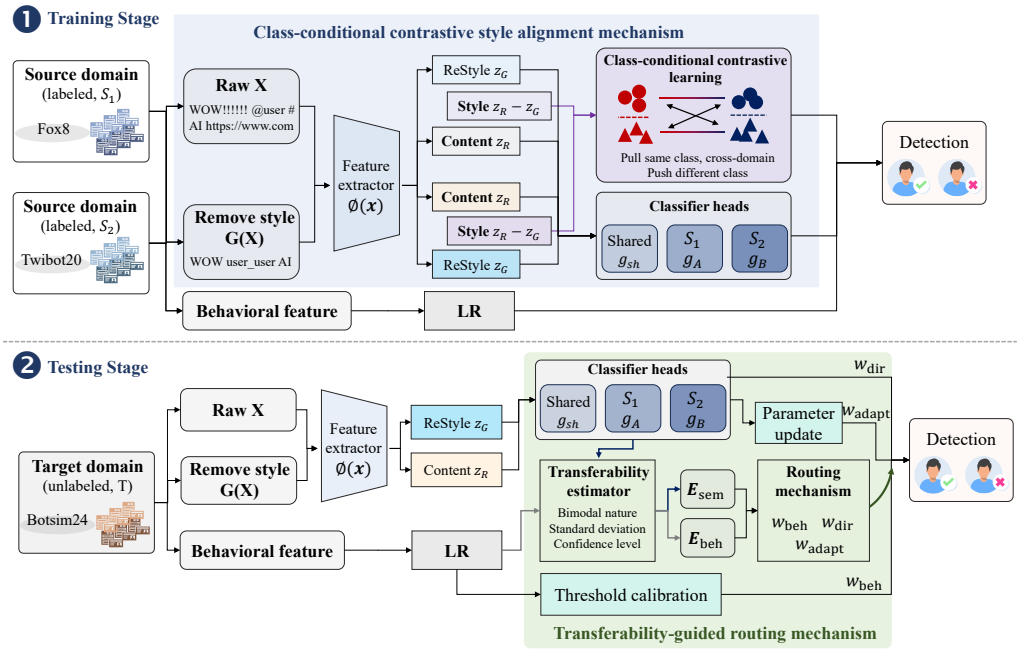


Figure 2: Overview of the DDDA framework.

3.2 FRAMEWORK OVERVIEW

We propose **Drift-Decoupled Domain Adaptation (DDDA)**, which decouples representation-level style drift from decision-level drift for robust cross-domain bot detection. As illustrated in Figure 2, DDDA addresses style drift through a *class-conditional contrastive style alignment mechanism* that aligns domain-specific variations while preserving class-discriminative information, and addresses

decision drift through a *transferability-guided routing mechanism* that adaptively selects source decision knowledge according to its estimated transferability on unlabeled target data.

3.2.1 CLASS-CONDITIONAL CONTRASTIVE STYLE ALIGNMENT

Users on different social platforms express similar behaviors through different surface patterns. Domain-specific conventions (e.g., emojis, URLs, mentions, hashtags, repeated characters) introduce representation discrepancies even when underlying behaviors are comparable. To reduce such stylistic discrepancy, we construct an intervention view via a deterministic rewriting function $G(\cdot)$ that applies Unicode normalization and removes or normalizes domain-dependent surface markers (Dawod & Yavuz, 2026). The transformation is applied identically to source and target samples and serves as a fixed perturbation.

Given posts x and transformed view $G(x)$, we encode both with a shared LoRA-adapted RoBERTa encoder $\phi(\cdot)$ (Liu et al., 2019b): $z_R = \phi(x)$, $z_G = \phi(G(x))$. The style-sensitive residual is

$$z_{\text{style}} = \text{Proj}_{\text{style}}(z_R - z_G) \in \mathbb{R}^{128} \tag{1}$$

where $\text{Proj}_{\text{style}}(\cdot)$ is a learnable projection. Since style-sensitive variations may also contain class-discriminative signals, directly enforcing domain invariance could eliminate useful bot-related cues. We therefore introduce a class-conditional contrastive objective that aligns style-sensitive residuals according to bot/human identity. For each anchor i, the positive set $\mathcal{P}_i$ contains samples from other source domains with the same label, and the negative set $\mathcal{N}_i$ contains samples with different labels. We optimize a multi-positive InfoNCE objective (Khosla et al., 2020; Lee et al., 2022):

$$\mathcal{L}_{\text{style}} = -\frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}\frac{1}{|\mathcal{P}_i|}\sum_{j\in\mathcal{P}_i}\log\frac{\exp(\bar{z}_{\text{style},i}^{\top}\bar{z}_{\text{style},j}/\tau_{\text{style}})}{\sum_{k\in\mathcal{P}_i\cup\mathcal{N}_i}\exp(\bar{z}_{\text{style},i}^{\top}\bar{z}_{\text{style},k}/\tau_{\text{style}})} \tag{2}$$

where $\mathcal{I}$ denotes anchors with non-empty positive sets and $\bar{z}_{\text{style},i} = z_{\text{style},i}/\|z_{\text{style},i}\|_2$. This objective encourages style-sensitive variations shared by the same class across source domains to become transferable while preserving bot/human discrimination.

3.2.2 DECISION KNOWLEDGE DECOMPOSITION AND TRANSFER

Decision drift reflects instability of the mapping from user information to labels across domains. As a result, even well-aligned representations may correspond to different decision rules in different domains. Therefore, DDDA does not assume that all source decision knowledge is equally transferable. Instead, it decomposes source decision knowledge into two complementary channels with different transfer characteristics: a semantic decision channel and a behavioral decision channel.

Semantic Decision Channel. Textual information provides rich semantic cues for distinguishing bots from humans. However, the corresponding decision patterns may vary across domains due to differences in platform conventions, user communities, and bot-generation mechanisms. Moreover, multiple source domains may contain complementary or conflicting decision knowledge, making a single unified classifier insufficient to capture heterogeneous source decision rules (Yang et al., 2026; Li et al., 2021). Therefore, DDDA maintains a shared prediction head g_{sh} and source-specific prediction heads $\{g_{S_k}\}_{k=1}^{K}$. The shared head captures common decision patterns across source domains, while the source-specific heads preserve domain-dependent decision characteristics. Specifically, the shared head is optimized using both original and style-normalized representations:

$$\ell_{\text{sh}} = \beta\, g_{\text{sh}}(z_G) + (1-\beta)\, g_{\text{sh}}(z_R) \tag{3}$$

where β controls the contribution of style-normalized semantics. Source-specific heads are trained only on their corresponding source domains using style-normalized representations:

$$\ell_{\text{spe}} = g_{S_k}(z_G), \qquad k \in \{1,\ldots,K\} \tag{4}$$

The semantic logit is $\ell_{\text{sem}} = \ell_{\text{sh}} + \ell_{\text{spe}}$. This separation avoids forcing heterogeneous source decision rules into a single classifier.

Behavioral Decision Channel. Behavioral regularities provide complementary signals less dependent on domain-specific linguistic styles. Under decision drift, such behavioral evidence may remain informative even when semantic decision knowledge becomes unreliable. Therefore, DDDA additionally introduces a behavioral decision channel.

Following previous observations that automated accounts often exhibit abnormal regularities in emotional consistency, topical evolution, and activity patterns (Alzahrani et al., 2025; Kušen & Strembeck, 2019; 2018), we extract two groups of behavioral features. The first group consists of aggregated behavioral statistics $\mathbf{m}$, including emotion entropy, emotion self-contradiction, and topic self-consistency, which characterize the stability and coherence of user behaviors over time. The second group captures temporal posting dynamics. Specifically, we divide each user's posts into T temporal bags and encode each bag using the shared encoder ϕ applied to style-normalized posts. This produces a sequence of embeddings $\{e_t\}_{t=1}^{T}$, and then compute consecutive semantic affinities $a_t = \cos(e_t, e_{t+1})$ for $t = 1, \ldots, T-1$. From these affinities, we extract trajectory statistics $\mathbf{a}$, including mean, standard deviation, minimum, and maximum affinity values, to characterize temporal consistency and evolution patterns. The aggregated behavioral statistics and temporal dynamics are concatenated and passed through a lightweight prediction head:

$$\ell_{\text{beh}} = W_{\text{beh}}[\mathbf{m}; \mathbf{a}] + b_{\text{beh}} \tag{5}$$

where W_{beh} and b_{beh} are learnable parameters, optimized jointly with the semantic heads using source labels via cross-entropy loss. Appendix A lists all features.

3.2.3 Transferability-guided Routing Mechanism

The reliability of semantic and behavioral channels varies across target domains. Since target labels are unavailable, we estimate transferability from the prediction distribution of each channel over the target domain. Different from conventional confidence-based approaches that rely on individual prediction certainty and often suffer from over-confidence under distribution shifts (Guo et al., 2017; Kamath et al., 2020), DDDA evaluates the global structure of target predictions from three complementary perspectives: prediction separation, prediction variation, and prediction margin.

For each target sample x_i in the unlabeled target domain $\mathcal{D}_T$, the semantic and behavioral channels produce $p_i^{\text{sem}} = \sigma(\ell_{\text{sem}}(x_i))$ and $p_i^{\text{beh}} = \sigma(\ell_{\text{beh}}(x_i))$. Then for channel $c \in \{\text{sem}, \text{beh}\}$, let $\mathbf{p}^c = \{p_i^c\}_{i=1}^{|\mathcal{D}_T|}$ denote its prediction vector over all target samples. We estimate transferability from three statistics:

$$\text{Sep}_c = \frac{|\mu_{c,1} - \mu_{c,2}|}{\sigma_{c,1} + \sigma_{c,2}}, \quad \text{Var}_c = \text{std}(\mathbf{p}^c), \quad \text{Mar}_c = 2\left|\text{mean}(\mathbf{p}^c) - \frac{1}{2}\right| \tag{6}$$

where Sep_c measures prediction separation via the bimodality of a two-component GMM, with $(\mu_{c,1}, \sigma_{c,1})$ and $(\mu_{c,2}, \sigma_{c,2})$ being the means and standard deviations of the two GMM components, Var_c measures prediction variation, and Mar_c measures prediction margin. After normalizing these statistics to $[0, 1]$, we aggregate them into a target-level transferability score:

$$E_c = \frac{1}{3}\left(\widetilde{\text{Sep}}_c + \widetilde{\text{Var}}_c + \widetilde{\text{Mar}}_c\right) \tag{7}$$

where $\widetilde{\text{Sep}}_c, \widetilde{\text{Var}}_c, \widetilde{\text{Mar}}_c$ denote the normalized counterparts of $\text{Sep}_c, \text{Var}_c, \text{Mar}_c$, respectively.

Based on the estimated transferability, DDDA adaptively routes source decision knowledge through three complementary strategies: *direct transfer*, *behavioral transfer*, and *semantic adaptation*. Direct transfer preserves transferable semantic decision knowledge while incorporating behavioral information as complementary evidence; behavioral transfer reduces reliance on semantic decision knowledge when its transferability is low; and semantic adaptation refines the semantic decision rule when its transferability is uncertain.

Direct transfer. When semantic decision knowledge exhibits high transferability, DDDA directly transfers the semantic prediction to the target domain while incorporating behavioral information as complementary evidence: $p_{\text{dir}} = \alpha_{\text{dir}}\text{rank}(p_{\text{sem}}) + (1 - \alpha_{\text{dir}})\text{rank}(p_{\text{beh}})$, where $\alpha_{\text{dir}} \in [0, 1]$ controls the contribution of semantic decision knowledge. We apply the rank transformation to reduce the influence of cross-domain probability calibration differences, while preserving the relative ordering of target samples.

Behavioral transfer. When semantic transferability is low but behavioral decision knowledge remains relatively transferable, DDDA reduces reliance on the semantic channel and transfers decision knowledge through the behavioral channel. We adjust the behavioral prediction using an adaptive decision threshold (Li et al., 2026): $p^*_{\mathrm{beh}} = \mathrm{clip}\left(p_{\mathrm{beh}} - \tau_{\mathrm{ada}} + 0.5, 0, 1\right)$, where τ_{ada} denotes the estimated decision threshold for the behavioral channel and $\mathrm{clip}(\cdot, 0, 1)$ constrains the resulting prediction to the valid probability range. This compensates for potential threshold shifts across domains while retaining the behavioral channel as an alternative source of decision knowledge.

Semantic adaptation. When semantic transferability lies in an uncertain regime, directly applying the source semantic decision may be unreliable, whereas discarding it may lose transferable information. DDDA therefore refines the semantic decision rule through test-time adaptation on unlabeled target samples $p_{\mathrm{adapt}} = \sigma\left(\ell^{\mathrm{TTA}}_{\mathrm{sem}}\right)$, where $\ell^{\mathrm{TTA}}_{\mathrm{sem}}$ denotes the semantic logit after test-time adaptation using unlabeled target data (Song et al., 2025). This strategy allows the semantic decision to be adjusted to the target domain rather than being directly transferred without modification.

To adaptively balance the three transfer paths, we first estimate whether the target domain lies in an intermediate regime where semantic adaptation is potentially beneficial. We define an intermediate-regime score as $E_{\mathrm{mid}} = \mathrm{clip}\left(1 - \frac{|E_{\mathrm{sem}} - \sigma_{\mathrm{mid}}|}{\sigma_{\mathrm{mid}}}, 0.02, 0.98\right)$, which reaches its maximum when semantic transferability is around σ_{mid} and decreases as E_{sem} moves away from this intermediate regime, σ_{mid} is the center of the intermediate regime and also serves as the normalization scale. The lower and upper clipping bounds prevent degenerate routing scores while preserving the relative differences between transfer regimes. We then construct a compatibility score for each transfer path based on the estimated transferability of the semantic and behavioral channels: $q_{\mathrm{beh}} = (1 - E_{\mathrm{sem}})(1 - E_{\mathrm{mid}})E_{\mathrm{beh}}$, $q_{\mathrm{dir}} = E_{\mathrm{sem}}(1 - E_{\mathrm{mid}})E_{\mathrm{beh}}$, and $q_{\mathrm{adapt}} = E_{\mathrm{sem}}E_{\mathrm{mid}}$. The routing weights are obtained by normalizing these compatibility scores:

$$w_k = \frac{q_k}{q_{\mathrm{beh}} + q_{\mathrm{dir}} + q_{\mathrm{adapt}}}, \qquad k \in \{\mathrm{beh}, \mathrm{dir}, \mathrm{adapt}\} \tag{8}$$

These scores correspond to three complementary transfer regimes. When semantic transferability is low and the intermediate-regime signal is weak, q_{beh} increases with behavioral transferability, favoring behavioral transfer. When semantic transferability is high but the target is outside the intermediate adaptation regime, (q_{dir} is favored, provided that behavioral transferability is also sufficiently reliable. In contrast, q_{adapt} increases when semantic knowledge remains transferable while the target exhibits a strong intermediate-regime signal, favoring semantic adaptation. Thus, high semantic transferability does not automatically imply direct transfer; instead, the routing mechanism jointly considers semantic transferability, behavioral transferability, and the estimated adaptation regime to balance direct transfer, behavioral transfer, and semantic adaptation.

Finally, DDDA aggregates the predictions produced by the three strategies:

$$p_T = w_{\mathrm{beh}}p^*_{\mathrm{beh}} + w_{\mathrm{dir}}p_{\mathrm{dir}} + w_{\mathrm{adapt}}p_{\mathrm{adapt}} \tag{9}$$

We can obtain the final target-domain prediction as $\hat{y}_T = \mathbf{1}[p_T \geq 0.5]$. So DDDA does not impose a fixed transfer strategy across target domains. Instead, it adaptively adjusts the utilization of source decision knowledge according to its estimated transferability, allowing reliable semantic knowledge to be directly transferred, unreliable semantic knowledge to be complemented by behavioral knowledge, and uncertain semantic knowledge to be refined through target-specific adaptation.

3.2.4 Training Objective

DDDA is trained using only labeled source domains, while target-domain labels are not involved during optimization. The training objective aims to jointly optimize three goals: maintaining discriminative bot detection ability, preserving the predictive capability of individual decision channels, and reducing transferable style variation across source domains.

Given the semantic logit ℓ_{sem} and behavioral logit ℓ_{beh}, we first learn a fused classifier for source-domain supervision: $\ell_{\mathrm{fuse}} = \eta_{\mathrm{sem}}\ell_{\mathrm{sem}} + \eta_{\mathrm{beh}}\ell_{\mathrm{beh}}$, where η_{sem} and η_{beh} are learnable fusion weights. The primary classification objective adopts label smoothing with $\epsilon = 0.1$: $\mathcal{L}_{\mathrm{fuse}} = \mathrm{CE}_{\epsilon}(\ell_{\mathrm{fuse}}, y)$. To prevent one decision channel from dominating the fused prediction and to preserve complementary decision knowledge, we introduce an auxiliary classification objective. Specifically, the shared

semantic head, source-specific semantic heads, and behavioral head are optimized independently:

$$\mathcal{L}_{\text{aux}} = \text{CE}(\ell_{\text{sh}}, y) + \text{CE}(\ell_{\text{spe}}, y) + \text{CE}(\ell_{\text{beh}}, y) \tag{10}$$

The training objective is: $\mathcal{L} = \mathcal{L}_{\text{fuse}} + \lambda_{\text{aux}}\mathcal{L}_{\text{aux}} + \lambda_{\text{style}}\mathcal{L}_{\text{style}}$, where λ_{aux} and λ_{style} control the contributions of auxiliary decision supervision and class-conditional style alignment, respectively.

4 Experiments

4.1 Experimental Settings

Evaluation Setup We evaluate DDDA on three publicly available social bot detection datasets: BotSim24 (Qiao et al., 2025), Fox8 (Yang & Menczer, 2023), and TwiBot20 (Feng et al., 2021). BotSim24 is a Reddit-based dataset, and Fox8 and TwiBot20 are both Twitter-based datasets. Detailed dataset statistics are provided in Appendix B.

To evaluate cross-domain generalization under UDA, we construct three leave-one-domain-out transfer scenarios. Specifically, each dataset is selected as the unlabeled target domain in turn, while the remaining two datasets are used as labeled source domains. For each scenario, source-domain data are split into training, validation, and test sets with a ratio of 7:1:2, while target labels are only used for final evaluation. All experiments are repeated three times with different random seeds, and we report the average performance. Besides, following previous cross-domain social bot detection studies (Song et al., 2025; Qiao et al., 2026), Accuracy and Macro-F1 are adopted as the primary evaluation metrics because they jointly measure overall classification performance and class-balanced detection capability.

Baselines We compare DDDA with three groups of baselines. **Direct transfer methods.** These methods are trained on the combined source domains and directly applied to the target domain without adaptation. They include conventional classifiers, including MLP, Random Forest (RF), and AdaBoost, as well as pretrained language-model based methods, including RoBERTa-Full and Qwen0.5B-Full. These baselines measure the performance degradation caused by cross-domain distribution shifts without adaptation. **Foundation model baselines.** To investigate whether general-purpose LLMs can directly transfer social bot detection knowledge across domains, we additionally evaluate several LLMs under zero-shot inference, including Qwen3-4B, Qwen3-14B-Instruct, and DeepSeek-V3.2 (Liu et al., 2025). **UDA methods.** We compare with representative UDA approaches designed to improve target-domain generalization by reducing source-target discrepancies. These include MGDIL (Qiao et al., 2026), AdaBot (Sun et al., 2025), OOD (Song et al., 2025), BotTrans (Shi et al., 2025), PoCoMT (Cao et al., 2026), UASA (Fang et al., 2025), and SSA (Li et al., 2026). These methods represent different adaptation paradigms, including domain alignment, contrastive representation learning, test-time adaptation, and multi-source knowledge transfer. Detailed baselines and implementation details are provided in Appendix B.

4.2 Experimental Results

Comparison with Baselines Table 1 compares DDDA with direct transfer, foundation model, and UDA methods under three transfer scenarios. DDDA (RoBERTa+LoRA) achieves target Macro-F1 scores of 0.788, 0.819, and 0.571 on BotSim24, Fox8, and TwiBot20, respectively, demonstrating strong performance across all three target domains. Furthermore, DDDA maintains good performance across different backbone architectures.

Conventional feature-based classifiers vary strongly across target domains: MLP reaches 0.763 Macro-F1 on Fox8 but only 0.358 on TwiBot20, with RF and AdaBoost showing similar trends, indicating that shallow features capture domain-specific patterns that transfer poorly. Pretrained language models fail to close this gap (RoBERTa-Full and Qwen0.5B-Full: 0.328 and 0.582 Macro-F1), and LoRA only helps Qwen0.5B on BotSim24 ($0.387 \rightarrow 0.539$) but barely on TwiBot20 (0.334), suggesting that backbone adaptation alone is insufficient under heterogeneous cross-domain shifts.

The foundation model results further show that model scale does not translate directly into more reliable cross-domain transfer. Qwen3-14B-Instruct and DeepSeek-V3.2 achieve Macro-F1 scores of 0.623 and 0.713 on Fox8 and 0.546 and 0.530 on TwiBot20, but drop to 0.268 and 0.372 on

Table 1: Comparison of different methods on BotSim24, Fox8, and TwiBot20 datasets.

Method	BotSim24		Fox8		TwiBot20	
	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
MLP	0.547	0.529	0.775	0.763	0.505	0.358
RF	0.581	0.438	0.697	0.677	0.500	0.344
Adaboost	0.554	0.510	0.583	0.572	0.503	0.361
RoBERTa-Full	0.590	0.379	0.641	0.569	0.484	0.328
Qwen0.5B-Full	0.630	0.387	0.637	0.582	0.493	0.338
Qwen0.5B-LoRA	0.574	0.539	0.557	0.449	0.500	0.334
Qwen3-4B	0.475	0.461	0.555	0.527	0.489	0.452
Qwen3-14B-Instruct	0.351	0.268	0.644	0.623	0.583	0.546
DeepSeek-V3.2(Liu et al., 2025)	0.411	0.372	0.715	0.713	0.536	0.530
MGDIL(Qiao et al., 2026)	0.649	0.394	0.802	0.788	0.501	0.338
AdaBot(Sun et al., 2025)	0.618	0.582	0.541	0.435	0.508	0.469
OOD(Song et al., 2025)	0.745	0.642	0.565	0.523	0.515	0.483
BotTrans(Shi et al., 2025)	0.652	0.395	0.629	0.570	0.504	0.347
PoCoMT(Cao et al., 2026)	0.529	0.515	0.788	0.778	0.561	0.527
UASA(Fang et al., 2025)	0.679	0.404	0.539	0.367	0.505	0.355
SSA(Li et al., 2026)	0.604	0.595	0.810	0.806	0.484	0.481
DDDA (RoBERTa + LoRA)	0.788	0.783	0.819	0.819	0.581	0.571
DDDA (Qwen0.5B + LoRA)	0.575	0.573	0.809	0.808	0.581	0.572
DDDA (Qwen1.5B + LoRA)	0.764	0.758	0.846	0.846	0.582	0.572

Table 2: Ablation study on BotSim24, Fox8, and TwiBot20 datasets.

Variant	BotSim24		Fox8		TwiBot20	
	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
DDDA	0.788	0.783	0.819	0.819	0.581	0.571
w/o $\mathcal{L}_{\text{style}}$	0.771	0.764	0.845	0.844	0.531	0.530
w/o style alignment	0.729	0.721	0.816	0.816	0.565	0.565
w/o routing mechanism (direct transfer)	0.696	0.689	0.850	0.850	0.587	0.587
w/o semantic adaptation	0.627	0.619	0.850	0.850	0.582	0.573
w/o behavioral channel	0.767	0.761	0.726	0.725	0.561	0.560
w/o semantic channel	0.620	0.612	0.706	0.704	0.584	0.574

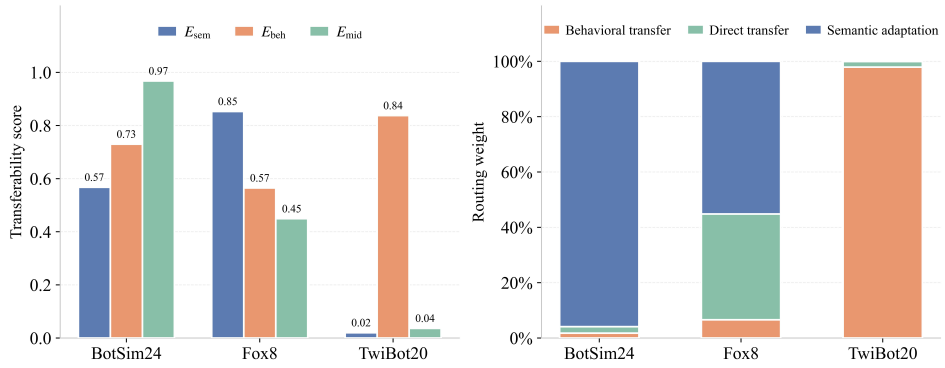
BotSim24. Thus, although LLMs provide richer semantic representations, their decision knowledge can remain mismatched with target-domain bot-generation and behavioral patterns.

Existing UDA methods reduce part of the cross-domain transfer gap, but their effectiveness remains strongly target-dependent. MGDIL, PoCoMT, and SSA perform well on Fox8, while OOD and AdaBot obtain their strongest results on BotSim24. This may be related to different transfer conditions across targets: Fox8 is Twitter-based and contains ChatGPT-generated bots with relatively regular textual patterns, which may preserve transferable semantic representations from source domains. In contrast, BotSim24 is Reddit-based and contains GPT-4o-mini-generated bots, where larger differences in textual patterns may make direct semantic transfer less reliable, favoring methods with target adaptation or decision refinement. On TwiBot20, most adaptation methods perform relatively poorly, suggesting that more complex variations in bot behaviors and textual characteristics reduce representation transferability and increase decision uncertainty. Overall, no single adaptation strategy consistently dominates across different targets. DDDA addresses this limitation by combining class-conditional contrastive style alignment with transferability-guided routing, achieving consistent improvements over existing methods across BotSim24, Fox8, and TwiBot20.

Ablation Study Table 2 evaluates the contribution of each DDDA component by removing the class-conditional contrastive style loss $\mathcal{L}_{\text{style}}$, the class-conditional contrastive style alignment mechanism, the transferability-guided routing mechanism, the semantic adaptation strategy, the behavioral channel, and the semantic channel. Overall, the contribution of each component is strongly target-dependent, which supports the central motivation of DDDA.

From this table, it is observed that removing style alignment generally degrades performance, with substantial drops on BotSim24 and TwiBot20, while Fox8 changes little. This supports selectively reducing stylistic variation while preserving class structure, and indicates that style drift is target-dependent. Third, removing semantic adaptation causes the largest degradation on BotSim24, while Fox8 improves and TwiBot20 remains stable. This indicates that target-specific semantic refinement

is essential under moderate semantic transferability, but can be unnecessary when source semantics are already reliable. Fourth, removing the behavioral channel consistently degrades performance on all targets, most notably on Fox8, confirming that behavioral signals provide complementary evidence when semantic transfer is unreliable. Finally, removing the semantic channel severely degrades BotSim24 and Fox8 but slightly improves TwiBot20, indicating that semantic decision knowledge is indispensable on some targets but may be less transferable or even noisy on others. In summary, different target domains rely on different components. These target-dependent patterns justify DDDA's drift-decoupled and transferability-aware transfer, while also suggesting that routing and semantic adaptation require more reliable gating mechanisms.



(a) Target-level transferability score **(b) Transferability-guided routing**

Figure 3: Target-level transferability estimates and the resulting transferability-guided routing.

Analysis of Transferability-Guided Routing Figure 3 visualizes the target-level transferability scores and routing weights of DDDA on three target domains. It can be observed that TwiBot20 exhibits the lowest semantic transferability but a substantially higher behavioral transferability, so DDDA assigns the largest weight to behavioral transfer. This indicates that source semantic knowledge is largely unreliable on TwiBot20, while behavioral signals provide a more reliable basis for prediction. This observation is consistent with the ablation results, suggesting that forcing additional semantic adaptation is less beneficial when the semantic channel exhibits extremely low transferability. For BotSim24, semantic transferability is moderate: source semantics remain partially transferable but still mismatched, so DDDA assigns the largest decision weight to semantic adaptation rather than direct transfer, consistent with the ablation result that refining semantic decisions is important in this intermediate regime. For Fox8, the semantic channel achieves the highest transferability. The routing assigns the higher decision weight to semantic adaptation than direct transfer, with negligible weight assigned to behavioral transfer, indicating that source semantics remain largely applicable. Notably, the ablation performs better when routing and semantic adaptation are removed, suggesting that additional test-time adaptation can be unnecessary when the source decision rule is already highly transferable. This highlights the importance of adapting the degree of knowledge utilization rather than uniformly applying additional adaptation. Overall, because source decision reliability varies substantially across targets, DDDA adapts source knowledge utilization according to estimated target-level transferability, providing a unified mechanism for heterogeneous decision drift.

5 Conclusion

We propose Drift-Decoupled Domain Adaptation (DDDA) for cross-domain social bot detection. DDDA explicitly decomposes representation-level style drift from decision-level drift: a class-conditional contrastive style alignment mechanism reduces domain-specific surface variation while preserving class-discriminative cues, and a transferability-guided routing mechanism adaptively selects among direct transfer, behavioral transfer, and semantic adaptation according to unlabeled target predictions. Experiments on three datasets demonstrate DDDA's effectiveness across diverse transfer scenarios, and analyses reveal distinct domain drift patterns that highlight the need for adaptive source-knowledge utilization in robust transfer.

AI USE STATEMENT

In this work, we used generative AI tools for **language polishing and improving the readability of the manuscript**. We also used generative AI tools to assist with the formal presentation, specifically for **checking notational consistency and formatting equations**. Generative AI tools were not used to generate research ideas, design the methodology, or produce experimental results. We have reviewed and verified all AI-assisted content and have checked the final manuscript, and we take full responsibility for the content of this work, including all text, claims, and artifacts produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

All datasets used in this work are publicly available. Our code is available at `https://anonymous.4open.science/r/CrossBot-3C27/`. We provide comprehensive method details in Appendix A and experimental settings in Appendix B.

REFERENCES

Mohammad Majid Akhtar, Navid Shadman Bhuiyan, Rahat Masood, Muhammad Ikram, and Salil S Kanhere. Botsscl: Social bot detection with self-supervised contrastive learning. *Online Social Networks and Media*, 48:100318, 2025.

Ohoud Alzahrani, Russell Beale, and Bob Hendley. Bots don't sit still: A longitudinal study of bot behaviour change, temporal drift, and feature-structure evolution. *arXiv preprint arXiv:2512.17067*, 2025.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

Qi Cao, Jianwen Tao, Yufang Dan, and Di Zhou. Prototype-oriented contrastive mean-teacher for unsupervised domain adaptive object detection. *Scientific Reports*, 2026.

Jianting Chen, Ling Ding, Yunxiao Yang, Zaiyuan Di, and Yang Xiang. Domain adversarial active learning for domain generalization classification. *IEEE Transactions on Knowledge and Data Engineering*, 37(1):226–238, 2024.

Sabah Dawod and Burcu Çarkli Yavuz. Accuracy–efficiency trade-offs in deep learning for twitter emotion classification: A hardware-aware benchmark of transformer and neural models. *IEEE Access*, 2026.

Xiang Fang, Arvind Easwaran, Blaise Genest, and Ponnuthurai Nagaratnam Suganthan. Your data is not perfect: Towards cross-domain out-of-distribution detection in class-imbalanced data. *Expert Systems with Applications*, 267:126031, 2025.

Shangbin Feng, Herun Wan, Ningnan Wang, Jundong Li, and Minnan Luo. Twibot-20: A comprehensive twitter bot detection benchmark. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pp. 4485–4494, 2021.

Emilio Ferrara. Social bot detection in the age of chatgpt: Challenges and opportunities. *First Monday*, 2023.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Mahta Hassan Pour Zonoozi, Vahid Seydi, and Mahmood Deypir. An unsupervised adversarial domain adaptation based on variational auto-encoder. *Machine Learning*, 114(5):128, 2025.

Wei-Ning Hsu, Yu Zhang, and James Glass. Unsupervised domain adaptation for robust speech recognition via variational autoencoder-based data augmentation. In *2017 IEEE automatic speech recognition and understanding workshop (ASRU)*, pp. 16–23. IEEE, 2017.

Amita Kamath, Robin Jia, and Percy Liang. Selective question answering under domain shift. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 5684–5696, 2020.

Amandeep Kaur et al. A comprehensive study of bot detection in twitter: Evolution, comparisons, and challenges. In *2025 International Conference on Electronics, AI and Computing (EAIC)*, pp. 1–6. IEEE, 2025.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

Ema Kušen and Mark Strembeck. Why so emotional? an analysis of emotional bot-generated content on twitter. In *Proceedings of the 3rd international conference on complexity, future information systems and risk*, volume 1, pp. 13–22, 2018.

Ema Kušen and Mark Strembeck. Something draws near, i can feel it: An analysis of human and bot emotion-exchange motifs on twitter. *Online Social Networks and Media*, 10:1–17, 2019.

Janghyeon Lee, Jongsuk Kim, Hyounguk Shon, Bumsoo Kim, Seung Hwan Kim, Honglak Lee, and Junmo Kim. Uniclip: Unified framework for contrastive language-image pre-training. *Advances in Neural Information Processing Systems*, 35:1008–1019, 2022.

Xinyao Li, Jingjing Li, Zhekai Du, Lei Zhu, and Heng Tao Shen. Unified modality separation: A vision-language framework for unsupervised domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Yongguang Li, Jindong Li, Qi Wang, QianLi Xing, Runliang Niu, Shengsheng Wang, and Menglin Yang. Beyond retraining: Training-free unknown class filtering for source-free open set domain adaptation of vision–language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 23265–23273, 2026.

Yunsheng Li, Lu Yuan, Yinpeng Chen, Pei Wang, and Nuno Vasconcelos. Dynamic transfer for multi-source domain adaptation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10993–11002. IEEE, 2021.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025.

Xiaofeng Liu, Yang Zou, Tong Che, Peng Ding, Ping Jia, Jane You, and BVK Kumar. Conservative wasserstein training for pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8262–8272, 2019a.

Xiaofeng Liu, Chaehwa Yoo, Fangxu Xing, Hyejin Oh, Georges El Fakhri, Je-Won Kang, and Jonghye Woo. Deep unsupervised domain adaptation: A review of recent advances and perspectives. *arXiv preprint arXiv:2208.07422*, 2022.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019b.

Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *Advances in neural information processing systems*, 31, 2018.

Lynnette Hui Xian Ng and Kathleen M Carley. Botbuster: Multi-platform bot detection using a mixture of experts. In *Proceedings of the international AAAI conference on web and social media*, volume 17, pp. 686–697, 2023.

YongKyung Oh and Alex Bui. Multi-view contrastive learning for robust domain adaptation in medical time series analysis. *arXiv preprint arXiv:2506.22393*, 2025.

Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Boyu Qiao, Kun Li, Wei Zhou, Shilong Li, Qianqian Lu, and Songlin Hu. Botsim: Llm-powered malicious social botnet simulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 14377–14385, 2025.

Boyu Qiao, Yunman Chen, Kun Li, Wei Zhou, Songlin Hu, and Yunya Song. Mgdil: Multi-granularity summarization and domain-invariant learning for cross-domain social bot detection. *arXiv preprint arXiv:2603.27928*, 2026.

Boshen Shi, Yongqing Wang, Fangda Guo, Jiangli Shao, Huawei Shen, and Xueqi Cheng. Bottrans: A multi-source graph domain adaptation approach for social bot detection. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 228–243. Springer, 2025.

Yuxuan Song, Qiudan Li, Shu Wu, Zheng Xie, and Daniel Dajun Zeng. A self-supervised adaptive tuning approach for out-of-distribution social media bot detection. In *Companion Proceedings of the ACM on Web Conference 2025*, pp. 1312–1315, 2025.

Hao Sun, Huailiang Peng, Yanan Cao, Qiong Dai, and Xu Bai. Adaptive social bot detection through bridging the feature bias between source and target users. In *Proceedings of the 2025 International Conference on Multimedia Retrieval*, pp. 1228–1236, 2025.

Shanshan Wang, Lei Zhang, Pichao Wang, MengZhu Wang, and Xingyi Zhang. Bp-triplet net for unsupervised domain adaptation: A bayesian perspective. *Pattern Recognition*, 133:108993, 2023.

Garrett Wilson and Diane J Cook. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(5):1–46, 2020.

Jianing Yang, Huibin Tan, Xihuai He, Wanrong Huang, Hengzhu Liu, and Huan Chen. Role-specialist and confidence-selective multi-teacher collaborative distillation for multi-source domain adaptation. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3686–3690. IEEE, 2026.

Kai-Cheng Yang and Filippo Menczer. Anatomy of an ai-powered malicious social botnet. *arXiv preprint arXiv:2307.16336*, 2023.

Canyu Zhang, Feiping Nie, and Rong Wang. Anchor guided unsupervised domain adaptation. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

A METHOD DETAILS

A.1 BEHAVIORAL FEATURES

The behavioral decision channel captures complementary bot-related regularities beyond textual representations. Following previous observations that automated accounts may exhibit distinctive patterns in emotional expression, topical organization, and temporal activity (Alzahrani et al., 2025; Kušen & Strembeck, 2019; 2018), we extract behavioral features from two complementary aspects: **emotional and topical patterns** and **temporal content dynamics**. The former characterizes the stability, diversity, and internal consistency of a user's emotional and topical expressions, while the latter captures how the semantic content of a user's posts evolves over time.

Emotional and Topical Patterns. We extract six statistics to characterize users' emotional and topical patterns, covering emotional diversity, emotional variation, topical variation, and topic-emotion coherence. Let $\mathcal{E}$ and $\mathcal{Q}$ denote the emotional and topical statistics, respectively.

Emotional entropy measures the diversity of a user's emotional expressions: $H_{\text{emo}} = -\sum_k p_k \log p_k$, where p_k denotes the empirical frequency of emotional state k.

Emotional conflict score measures the co-occurrence of opposing sentiment polarities. For a text d_i, let C_{pos}, C_{neg}, and C_{all} denote the numbers of positive words, negative words, and total words, respectively. We compute

$$R_{\text{emo}} = \frac{(C_{\text{pos}} + \sigma)(C_{\text{neg}} + \sigma)}{C_{\text{all}}}, \qquad \sigma = 0.5 \tag{11}$$

To capture temporal variation, we compute the coefficient of variation of emotional entropy and topical entropy over successive temporal segments:

$$V_{\text{emo}} = \text{CV}\left(H_{\text{emo}}^{(1:T)}\right), \qquad V_{\text{topic}} = \text{CV}\left(H_{\text{topic}}^{(1:T)}\right) \tag{12}$$

We additionally measure topical variability across posts using the standard deviation of topical representations, denoted by S_{topic}. Finally, C_{te} measures the coefficient of variation of topic-emotion coherence across temporal segments, capturing the stability of the relationship between topical and emotional expressions.

The resulting statistics are concatenated as

$$\mathbf{m} = [H_{\text{emo}}, R_{\text{emo}}, V_{\text{emo}}, S_{\text{topic}}, V_{\text{topic}}, C_{\text{te}}] \tag{13}$$

Temporal Content Dynamics. In addition to aggregated emotional and topical statistics, we characterize the temporal dynamics of a user's textual content. Given T temporal bags, we encode the style-normalized posts within each bag using the shared encoder ϕ and aggregate their representations to obtain a temporal embedding e_t for each bag, resulting in a sequence $e_t * t = 1^T$. The semantic affinity between two consecutive temporal bags is then defined as their cosine similarity:

$$a_t = \cos(e_t, e_{t+1}), \qquad t = 1, \ldots, T-1 \tag{14}$$

We summarize the resulting affinity trajectory using statistics that capture its overall level, dispersion, discontinuity, volatility, and boundary behavior. Specifically,

$$\mathbf{a} = [\text{mean}(a_t), \text{std}(a_t), \min(a_t), \max(a_t), N_{<0.5}, \text{mean}(|a_{t+1} - a_t|), a_1, a_{T-1}, \text{mean}(c_t), \text{std}(c_t)] \tag{15}$$

Here, $N_{<0.5}$ measures the frequency of low-affinity transitions ($a_t < 0.5$), while $\text{mean}(|a_{t+1} - a_t|)$ captures local affinity volatility. To further characterize deviations from a user's typical semantic profile, we compute the cosine similarity between each temporal embedding and the user's semantic centroid e_{cent}: $c_t = \cos(e_t, e_{\text{cent}})$ The mean and standard deviation of c_t characterize the overall semantic similarity to the user's typical profile and its variation across temporal segments, respectively. These two statistics, together with the eight affinity-based statistics, form the temporal feature vector $\mathbf{a}$.

Finally, the emotional, topical, and temporal features are concatenated as $[\mathbf{m}; \mathbf{a}]$, where $\mathbf{m}$ denotes the aggregated emotional and topical statistics. The resulting behavioral feature vector is fed into the behavioral prediction head to obtain the behavioral decision logit ℓ_{beh}.

Algorithm 1 Drift-Decoupled Domain Adaptation (DDDA)

Input: labeled source domains $\mathcal{D}_{S_1}, \mathcal{D}_{S_2}$ and unlabeled target domain $\mathcal{D}_T$
Source training:
for each source minibatch from $\mathcal{D}_{S_1} \cup \mathcal{D}_{S_2}$ **do**
  Construct original and style-normalized views x and $G(x)$.
  Encode both views with the shared encoder: $z_R = \phi(x)$ and $z_G = \phi(G(x))$.
  Compute style-sensitive residuals $z_{\text{style}} = \text{Proj}_{\text{style}}(z_R - z_G)$.
  Compute the shared semantic logit ℓ_{sh} and the corresponding source-specific logit $\ell_{\text{spe}} = g_{S_k}(z_G), k \in \{1, \ldots, K\}$.
  Obtain the semantic logit $\ell_{\text{sem}} = \ell_{\text{sh}} + \ell_{\text{spe}}$.
  Compute the behavioral logit ℓ_{beh}.
  Compute $\mathcal{L}_{\text{fuse}}$, $\mathcal{L}_{\text{aux}}$, and $\mathcal{L}_{\text{style}}$.
  Update all trainable parameters using $\mathcal{L} = \mathcal{L}_{\text{fuse}} + \lambda_{\text{aux}}\mathcal{L}_{\text{aux}} + \lambda_{\text{style}}\mathcal{L}_{\text{style}}$.
end for
Target transferability estimation:
Obtain semantic and behavioral predictions p_{sem} and p_{beh} for all samples in $\mathcal{D}_T$.
Estimate E_{sem} and E_{beh} from the unlabeled target prediction distributions.
Compute $E_{\text{mid}} = \text{clip}\left(1 - \frac{|E_{\text{sem}} - \sigma_{\text{mid}}|}{\sigma_{\text{mid}}}, 0.02, 0.98\right)$.
Compute routing weights $(w_{\text{beh}}, w_{\text{dir}}, w_{\text{adapt}})$.
Target inference:
for each target sample or batch **do**
  Compute the direct-transfer prediction p_{dir} using rank-transformed p_{sem} and p_{beh}.
  Calibrate the behavioral prediction to obtain p^*_{beh}.
  Apply test-time adaptation to the semantic channel and obtain p_{adapt} when the adaptation route is used.
  Aggregate the three transfer paths: $p_T = w_{\text{beh}}p^*_{\text{beh}} + w_{\text{dir}}p_{\text{dir}} + w_{\text{adapt}}p_{\text{adapt}}$.
  Predict $\hat{y}_T = \mathbf{1}[p_T \geq 0.5]$.
end for
Return: target predictions $\{\hat{y}_T\}$.

A.2 DDDA Algorithm

Algorithm 1 summarizes the training and inference procedure of DDDA. The model is first trained using labeled source domains only, where the class-conditional style alignment objective is jointly optimized with the semantic and behavioral decision objectives. After training, the target-domain prediction distributions are obtained without using target labels and are used to estimate the transferability of the two decision channels. The resulting transferability scores determine the routing weights for target inference.

A.3 Theoretical Motivation for Drift Decoupling

Our motivation for drift decoupling can be understood from the standard domain adaptation bound (Ben-David et al., 2010). For the effective source mixture P_S and an unlabeled target domain P_T, the target risk of a hypothesis $h \in \mathcal{H}$ satisfies

$$\epsilon_T(h) \leq \epsilon_S(h) + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(P_S, P_T) + \lambda \tag{16}$$

where ϵ_S and ϵ_T denote source and target risks, $d_{\mathcal{H}\Delta\mathcal{H}}$ measures the domain discrepancy, and λ denotes the ideal joint risk.

In cross-domain social bot detection, the observed domain discrepancy reflects heterogeneous sources of variation. In particular, style drift concerns changes in how user information is expressed across domains, whereas decision drift concerns changes in how the same underlying information is mapped to bot or human labels. These two drifts therefore correspond to different transfer objectives: style drift calls for selective alignment of transferable stylistic variation, while decision drift requires selective transfer of reliable source decision knowledge.

When both objectives are optimized jointly, their gradients and learned representations may interact, creating an interference cost. We denote this cost abstractly by Γ. A monolithic adaptation objective can therefore be conceptually written as

$$C_{\text{mono}} = C_{\text{style}} + C_{\text{decision}} + \Gamma \tag{17}$$

where C_{style} and C_{decision} denote the costs associated with handling style and decision drift, respectively.

DDDA separates these two transfer processes and allows them to be handled by different mechanisms. Its corresponding conceptual cost can be written as

$$C_{\text{DDDA}} = C'_{\text{style}} + C'_{\text{decision}} + \Gamma' \tag{18}$$

where Γ' denotes the residual interference after decoupling. Consequently, decoupling is beneficial whenever

$$\Gamma - \Gamma' > \left(C'_{\text{style}} - C_{\text{style}}\right) + \left(C'_{\text{decision}} - C_{\text{decision}}\right) \tag{19}$$

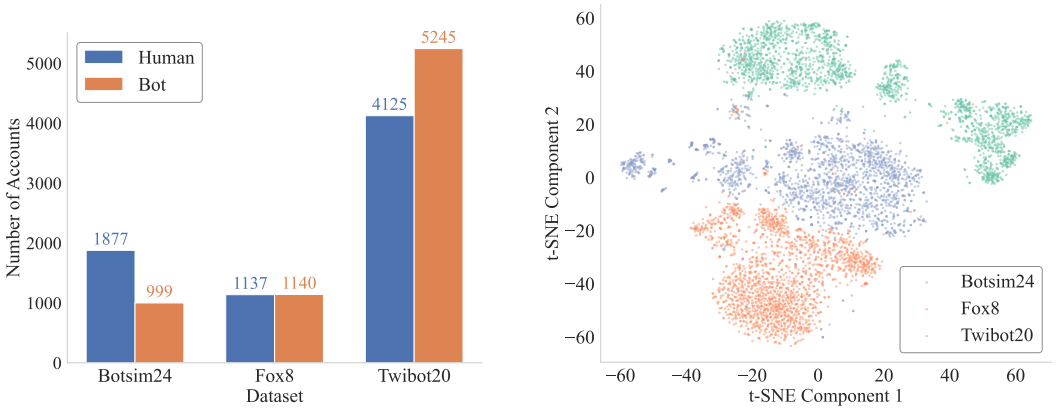
This condition formalizes the design principle of DDDA: decoupling is advantageous when the reduction in interference between heterogeneous transfer objectives outweighs the additional cost introduced by treating them separately. Importantly, this analysis does not assume that the $\mathcal{H}\Delta\mathcal{H}$ divergence decomposes exactly into style and decision components, nor does it claim that DDDA universally yields a tighter target-risk bound. Instead, it provides a conceptual explanation for why explicitly separating heterogeneous drift can improve cross-domain transfer.

B Experiments Settings

B.1 Datasets

We conduct our experiments on three publicly available social bot detection datasets: BotSim24 (Qiao et al., 2025), Fox8 (Yang & Menczer, 2023), and TwiBot20 (Feng et al., 2021). These datasets cover two major social platforms, Reddit and Twitter, and differ substantially in their account composition, posting behavior, and bot-generation mechanisms. Such diversity provides a challenging setting for evaluating cross-domain transfer and the robustness of domain adaptation methods. BotSim24 is a Reddit-based dataset containing 2,876 accounts, including 1,877 human accounts and 999 bot accounts. The bot accounts are generated using GPT-4o-mini, providing a controlled collection of LLM-generated social bots. Compared with naturally occurring bots, these accounts exhibit language-generation patterns associated with contemporary large language models, making BotSim24 particularly relevant for evaluating whether detection knowledge learned from one domain can generalize to different forms of bot-generated content. Fox8 is a Twitter-based dataset containing 2,277 accounts, including 1,140 accounts suspected of posting ChatGPT-generated content and 1,137 human accounts. The human accounts are collected from several established Twitter datasets, including Botometer-Feedback, Gilani-17, Midterm-2018, and Varol-ICWSM. Thus, Fox8 combines LLM-related bot activity with human accounts drawn from heterogeneous Twitter sources, introducing substantial variation in both account characteristics and textual behavior. TwiBot20 is a large-scale Twitter-based social bot detection benchmark containing 9,370 accounts, including 4,125 human accounts and 5,245 bot accounts. The dataset covers multiple topical communities, including business, entertainment, politics, and sports, and therefore contains a broader range of user behaviors and content characteristics than the two datasets described above. Unlike BotSim24 and Fox8, whose bot populations are explicitly associated with LLM-generated or suspected ChatGPT-generated content, TwiBot20 contains a mixture of bot types, providing a complementary testbed for evaluating cross-dataset generalization under heterogeneous bot-generation patterns. These differences create heterogeneous source-target conditions for cross-domain transfer. Table 3 summarizes the main statistics of the three datasets.

Beyond these aggregate statistics, we further examine the distributional characteristics of the three datasets. Figure 4(a) presents the class distribution, while Figure 4(b) visualizes the user representations obtained from the datasets. The class distributions show that the proportion of human and bot accounts varies across datasets, whereas the feature visualization reveals substantial differences in the representation spaces across domains and datasets. These observations illustrate that source and target domains cannot be assumed to share identical empirical distributions, motivating the use of adaptation methods for cross-domain social bot detection.



(a) Class distribution

(b) Feature distribution

Figure 4: Class and feature distributions across BotSim24, Fox8, and TwiBot20 datasets.

B.2 BASELINES

We compare DDDA with three groups of baselines, covering direct source-to-target transfer, general-purpose foundation models, and adaptation-based methods.

Direct transfer methods. These methods are trained using labeled source-domain data and directly applied to the target domain without exploiting target-domain information during training or inference. We include conventional classifiers, namely MLP, RF, and AdaBoost, as well as pretrained language-model based classifiers, including RoBERTa-Full and Qwen0.5B-Full. The conventional classifiers operate on pretrained user representations, while RoBERTa-Full and Qwen0.5B-Full are fully fine-tuned on the labeled source domains. These baselines measure the extent to which detection models trained on source domains can generalize to unseen target domains without explicit adaptation.

Foundation model baselines. We additionally evaluate general-purpose instruction-tuned language models under zero-shot inference, including Qwen3-4B, Qwen3-14B-Instruct, and DeepSeek-V3.2 (Liu et al., 2025). Unlike the source-trained classifiers above, these models do not receive task-specific fine-tuning on the source domains. They therefore assess whether general-purpose language models can directly transfer social bot detection knowledge across domains through prompting alone.

UDA methods. We further compare with representative UDA methods that exploit source-target adaptation signals or unlabeled target data. MGDIL (Qiao et al., 2026) is a cross-domain social bot detection method that combines LLM-based multi-granularity summarization with domain-adversarial and cross-domain contrastive learning to learn domain-invariant representations. AdaBot (Sun et al., 2025) addresses feature bias between source and target users through feature-space alignment, unsupervised decision-boundary refinement, and source-signal annealing. OOD (Song et al., 2025) performs self-supervised test-time adaptation by dynamically tuning model parameters

Table 3: Statistics of BotSim24, Fox8, and TwiBot20 datasets.

Attribute	**BotSim24**	**Fox8**	**TwiBot20**
Platform	Reddit	Twitter	Twitter
User Count	2,876	2,277	9,370
Human Count	1,877	1,137	4,125
Bot Count	999	1,140	5,245
Post Count (avg±std)	18±43	116±58	159±61
Post Length (avg±std)	98±72	109±37	118±75
Bot Type	LLM-generated (GPT-4o-mini)	LLM-generated (suspected ChatGPT)	Mixed

on unlabeled target data through contrastive learning. BotTrans (Shi et al., 2025) performs multi-source graph domain adaptation by constructing cross-source topologies, aggregating cross-domain neighborhood information, and selectively transferring knowledge according to source-target relevance. PoCoMT (Cao et al., 2026) is a prototype-oriented UDA method that combines contrastive learning, prototype alignment, and mean-teacher self-training to align class structures across domains. UASA (Fang et al., 2025) is designed for class-imbalanced cross-domain OOD detection and combines prototype-guided domain alignment with adaptive threshold generation and uncertainty-aware target clustering. SSA (Li et al., 2026) performs training-free source-free open-set adaptation for vision-language models through unknown-subspace filtering and adaptive score-based unknown detection.

All baselines are evaluated under the same source-target splits as DDDA. Direct transfer methods use only labeled source data, while UDA methods additionally use unlabeled target data according to their respective adaptation settings. Foundation-model baselines are evaluated through zero-shot prompting without source-domain fine-tuning.

B.3 Implementation Details

In our experiments, we instantiate DDDA with K=2 source domains, following the leave-one-domain-out protocol. DDDA adopts RoBERTa-base with LoRA adaptation as the backbone encoder. The LoRA rank and scaling factor are set to $r = 8$ and $\alpha = 32$, respectively, with a dropout rate of 0.1. We optimize the model using AdamW with a batch size of 8 and a learning rate of $2e^{-5}$.

For the proposed components, the auxiliary supervision weight is $\lambda_{\text{aux}} = 0.2$, and the style alignment weight is $\lambda_{\text{style}} = 0.3$. The contrastive temperature scaling parameter is set to $\tau_{\text{style}} = 0.35$, and the normalized-view coefficient is $\beta = 0.6$. The center of the intermediate regime is $\sigma_{\text{mid}} = 0.55$. At test time, the direct transfer mixes semantic and behavioral ranks with $\alpha_{\text{dir}} = 0.6$.

Table 4: Source and target domain evaluation results on BotSim24 dataset.

Method	Source				Target			
	Accuracy	Precision	Recall	Macro-F1	Accuracy	Precision	Recall	Macro-F1
MLP	0.7850	0.7879	0.7850	0.7844	0.5466	0.5363	0.5393	0.5291
RF	0.7863	0.7920	0.7863	0.7853	0.5811	0.4445	0.4715	0.4379
Adaboost	0.7629	0.7635	0.7629	0.7628	0.5535	0.5111	0.5105	0.5100
RoBERTa-Full	0.8321	0.8338	0.8315	0.8317	0.5903	0.3560	0.4806	0.3789
Qwen0.5B-Full	0.8250	0.8251	0.8250	0.8250	0.6302	0.3247	0.4776	0.3866
Qwen0.5B-LoRA	0.8043	0.8051	0.8053	0.8043	0.5744	0.5393	0.5407	0.5395
Qwen3-4B	–	–	–	–	0.4749	0.4726	0.4698	0.4612
Qwen3-14B-Instruct	–	–	–	–	0.3511	0.5257	0.5011	0.2675
DeepSeek-V3.2(Liu et al., 2025)	–	–	–	–	0.4105	0.5893	0.5357	0.3723
MGDIL(Qiao et al., 2026)	0.8767	0.8763	0.8768	0.8765	0.6494	0.3566	0.4986	0.3941
AdaBot(Sun et al., 2025)	0.6444	0.6745	0.6514	0.6346	0.6176	0.5817	0.5893	0.5823
OOD(Song et al., 2025)	0.7717	0.7743	0.7734	0.7716	0.7448	0.8102	0.6408	0.6418
BotTrans(Shi et al., 2025)	0.7963	0.8080	0.7999	0.7955	0.6516	0.3261	0.4992	0.3945
PoCoMT(Cao et al., 2026)	0.8130	0.8134	0.8121	0.8125	0.5292	0.5245	0.5269	0.5155
UASA(Fang et al., 2025)	0.8170	0.8167	0.8169	0.8168	0.6788	0.3400	0.4987	0.4043
SSA(Li et al., 2026)	0.7613	0.7684	0.7642	0.7608	0.6043	0.6042	0.6149	0.5948
DDDA (RoBERTa + LoRA)	0.7462	0.7481	0.7476	0.7462	0.7879	0.7865	0.8160	0.7825
DDDA (Qwen0.5B + LoRA)	0.7462	0.7464	0.7468	0.7462	0.5751	0.6036	0.6110	0.5731
DDDA (Qwen1.5B + LoRA)	0.7995	0.8015	0.8010	0.7995	0.7636	0.7626	0.7896	0.7577

C Experiments Results

C.1 Comparison with Baselines

This section provides the complete source- and target-domain evaluation results for all compared methods. For direct transfer and UDA methods, the source-domain results are computed on the source test sets, while target-domain results are obtained using the corresponding unlabeled target data during adaptation and the target ground-truth labels only for final evaluation. Foundation-model

Table 5: Source and target domain evaluation results on Fox8 dataset.

Method	Source				Target			
	Accuracy	Precision	Recall	Macro-F1	Accuracy	Precision	Recall	Macro-F1
MLP	0.8173	0.8184	0.8101	0.8122	0.7752	0.8368	0.7750	0.7633
RF	0.8036	0.8014	0.7978	0.7992	0.6972	0.7610	0.6969	0.6773
Adaboost	0.7858	0.7823	0.7829	0.7824	0.5825	0.5888	0.5826	0.5720
RoBERTa-Full	0.8438	0.8426	0.8402	0.8412	0.6411	0.7970	0.6221	0.5690
Qwen0.5B-Full	0.8307	0.8286	0.8296	0.8291	0.6368	0.7823	0.6360	0.5823
Qwen0.5B-LoRA	0.8212	0.8179	0.8176	0.8177	0.5573	0.7654	0.5567	0.4486
Qwen3-4B	–	–	–	–	0.5547	0.5708	0.5542	0.5268
Qwen3-14B-Instruct	–	–	–	–	0.6440	0.6844	0.6434	0.6228
DeepSeek-V3.2(Liu et al., 2025)	–	–	–	–	0.7145	0.7179	0.7145	0.7134
MGDIL(Qiao et al., 2026)	0.8447	0.8439	0.8450	0.8443	0.8025	0.7960	0.7831	0.7881
AdaBot(Sun et al., 2025)	0.7573	0.7834	0.7765	0.7570	0.5409	0.6472	0.5384	0.4352
OOD(Song et al., 2025)	0.6599	0.6524	0.6496	0.6505	0.5648	0.5989	0.5644	0.5230
BotTrans(Shi et al., 2025)	0.8343	0.8332	0.8288	0.8306	0.6289	0.7806	0.6284	0.5704
PoCoMT(Cao et al., 2026)	0.8394	0.8360	0.8376	0.8367	0.7879	0.8475	0.7876	0.7783
UASA(Fang et al., 2025)	0.8343	0.8312	0.8312	0.8312	0.5395	0.7677	0.5093	0.3671
SSA(Li et al., 2026)	0.8023	0.8045	0.7905	0.7945	0.8103	0.8388	0.8101	0.8061
DDDA (RoBERTa + LoRA)	0.7660	0.7773	0.7789	0.7660	0.8191	0.8195	0.8190	0.8190
DDDA (Qwen0.5B + LoRA)	0.8074	0.8083	0.8142	0.8067	0.8085	0.8099	0.8085	0.8083
DDDA (Qwen1.5B + LoRA)	0.7733	0.7839	0.7859	0.7732	0.8458	0.8461	0.8459	0.8459

Table 6: Source and target domain evaluation results on TwiBot20 dataset.

Method	Source				Target			
	Accuracy	Precision	Recall	Macro-F1	Accuracy	Precision	Recall	Macro-F1
MLP	0.9919	0.9917	0.9915	0.9916	0.5050	0.5684	0.5050	0.3575
RF	0.9827	0.9827	0.9818	0.9822	0.4999	0.4957	0.4999	0.3435
Adaboost	0.9866	0.9865	0.9860	0.9862	0.5028	0.5280	0.5028	0.3606
RoBERTa-Full	0.9981	0.9976	0.9984	0.9980	0.4838	0.7416	0.5012	0.3281
Qwen0.5B-Full	0.9971	0.9976	0.9964	0.9970	0.4925	0.7450	0.5049	0.3385
Qwen0.5B-LoRA	0.9932	0.9938	0.9924	0.9931	0.4998	0.4499	0.4998	0.3341
Qwen3-4B	–	–	–	–	0.4885	0.5436	0.5266	0.4524
Qwen3-14B-Instruct	–	–	–	–	0.5834	0.5702	0.5567	0.5461
DeepSeek-V3.2(Liu et al., 2025)	–	–	–	–	0.5364	0.5585	0.5529	0.5304
MGDIL(Qiao et al., 2026)	0.9933	0.9942	0.9914	0.9928	0.5013	0.5965	0.5013	0.3379
AdaBot(Sun et al., 2025)	0.9545	0.9523	0.9590	0.9541	0.5083	0.5168	0.5116	0.4686
OOD(Song et al., 2025)	0.7628	0.7652	0.7485	0.7522	0.5148	0.5196	0.5148	0.4826
BotTrans(Shi et al., 2025)	0.9758	0.9776	0.9734	0.9753	0.5035	0.5804	0.5035	0.3475
PoCoMT(Cao et al., 2026)	0.9923	0.9916	0.9926	0.9921	0.5610	0.5852	0.5610	0.5274
UASA(Fang et al., 2025)	0.9874	0.9864	0.9881	0.9872	0.5050	0.7165	0.5107	0.3555
SSA(Li et al., 2026)	0.9748	0.9727	0.9773	0.9745	0.4843	0.4838	0.4843	0.4805
DDDA (RoBERTa + LoRA)	0.9681	0.9659	0.9702	0.9676	0.5805	0.5881	0.5805	0.5713
DDDA (Qwen0.5B + LoRA)	0.9293	0.9269	0.9305	0.9284	0.5810	0.5887	0.5809	0.5716
DDDA (Qwen1.5B + LoRA)	0.9255	0.9231	0.9261	0.9244	0.5815	0.5897	0.5815	0.5717

baselines are evaluated only on the target domain under zero-shot inference and therefore do not have source-domain results.

Tables 4, 5, and 6 provide the complete results when BotSim24, Fox8, and TwiBot20 are used as the target domain, respectively. These detailed results complement the main-table comparison by reporting all evaluation metrics and source-domain performance. They also expose the substantial variation in transfer performance across target domains, motivating the need for adaptive knowledge utilization under heterogeneous cross-domain drift.

C.2 Sensitivity Analysis

We further investigate the sensitivity of DDDA to its key hyperparameters. Figure 5 reports the target-domain performance under different hyperparameter settings. Overall, DDDA is relatively insensitive to λ_{style}, λ_{aux} and τ_{style} over a broad range of values, demonstrating good robustness. In contrast, β, σ_{mid} and α_{dir} have a more pronounced impact on performance, with their optimal values varying across datasets. Specifically, BotSim24 performs well with $\beta \approx 0.6$ and $\alpha_{\text{dir}} \approx 0.6$, while Fox8 reaches its peak performance at $\beta \approx 0.5$ and $\alpha_{\text{dir}} \approx 0.7$. TwiBot20 is overall insensitive to variations in both parameters. And $\sigma_{\text{mid}} = 0.5$ is optimal on three scenarios.

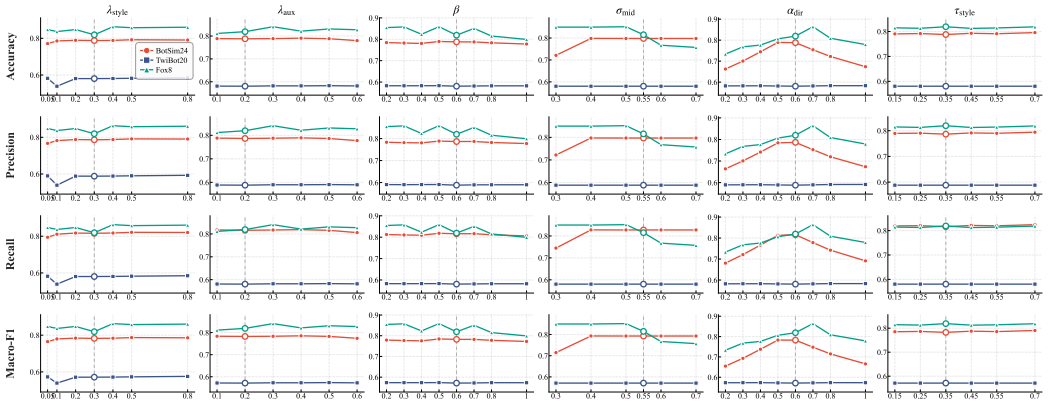


Figure 5: Sensitivity of DDDA to key hyperparameters, including λ_{style}, λ_{aux}, β, σ_{mid}, α_{dir}, and τ_{style}.