Discharge fertilizer rate prediction across fertilizer type for a bivariate fertilization system based on transfer learning

Jiqin Zhang[1], Qibin Zhuang[1*], Lin Xi[1], Gang Liu[2], Zhao Zhang[2], Changyuan Zhai[3], Shuo Yang[3]

(1. *School of Mechanical and Automotive Engineering, Anhui Polytechnic University, Wuhu 241000, Anhui, China*;
2. *Key Laboratory of Smart Agriculture System Integration Research of the Ministry of Education, China Agricultural University, Beijing 100083, China*;
3. *Intelligent Equipment Research Center, Beijing Academy of Agriculture and Forestry Sciences, Beijing 100097, China*)

Abstract: Accurate prediction of discharge fertilizer rate (DFR) is essential for achieving precise control in bivariate fertilizer applicators (BFA). However, most existing prediction models calibrated for specific fertilizer types ignore variations in material properties, limiting their generalizability and necessitating time-consuming recalibration for different fertilizers. Transfer learning (TL) offers a promising solution by leveraging knowledge from a source domain (D_S) to improve performance in a related target domain (D_T). This study investigated the cross-fertilizer DFR prediction in a BFA system using two transfer scenarios: from a compound fertilizer (Stanley) to another (Sakefu), and from Stanley to urea. Decision tree regression (DTR) serves as the base model. Five experimental configurations were compared: direct source domain prediction (DTR- D_S); partial target domain prediction with 20% data (DTR-D_T(20%); joint training (DTR-Recalibration); instance-based TL using transfer adaptive boosting for regression (TrAdaBoost.R2) (DTR-TrAdaBoost.R2); and full target domain prediction (DTR-D_T). The results demonstrate that: 1) DTR-Recalibration is most effective for Stanley to Sakefu, improving R^2 by 2.4% and reducing *nRMSE* by 12.5%, while DTR-TrAdaBoost.R2 performs best for Stanley to urea, increasing R^2 by 14.3% and reducing *nRMSE* by 21.5%; 2) differences in material properties significantly affect TL performance; the ranking of the importance of material characteristics is as follows: the combined difference in equivalent diameter and repose angle > bulk density > sphericity ratio; and 3) high-weight instances in TL are predominantly concentrated at medium operational intensity levels, and the stability of the discharge ratio (Q_{target}/Q_{source}) critically influences transfer effectiveness. These findings provide practical and theoretical insights for cross-fertilizer model transfer, enabling efficient adaptation of DFR prediction models with minimal target data and reducing calibration effort in precision agriculture applications.

Keywords: discharge fertilizer rate (DFR) prediction, bivariate fertilizer applicator (BFA), decision tree regression (DTR), transfer learning (TL), instance-based transfer learning

DOI: 10.25165/j.ijabe.20261904.10387

Citation: Zhang J Q, Zhuang Q B, Xi L, Liu G, Zhang Z, Zhai C Y, et al. Discharge fertilizer rate prediction across fertilizer type for a bivariate fertilization system based on transfer learning. Int J Agric & Biol Eng, 2026; 19(4): 51–66.

1 Introduction

Variable-rate fertilization technology (VRT) is a critical technique in precision agriculture, which enables site-specific application of fertilizers according to spatial heterogeneity in soil fertility[1,2]. Global adoption of VRT has been driven by its demonstrated benefits in mitigating fertilizer overuse, enhancing soil nutrient distribution, and increasing crop yield[3,4]. Concurrent with the advancement of smart agriculture, the requirements for precise control of the fertilizer discharge rate have become more stringent. In response, the bivariate fertilization applicator (BFA) has been developed in recent years[5,6]. A critical challenge in achieving precise fertilization control lies in modeling the discharge fertilizer rate (DFR), due to its highly nonlinear relationship between fertilizer discharge rate (Q) and operator parameters, including rotational speed (N) of the fluted roller shaft and the active feed-roll length (L). Consequently, developing an accurate DFR prediction model is essential.

The methodologies for constructing DFR prediction models can be broadly categorized into statistical regression and machine learning (ML) approaches[7]. Statistical regression is widely employed as a foundational method for DFR prediction owing to its simplicity and proven reliability under fixed levels of either L or N[5,8]. In recent years, ML techniques, such as Gaussian Process (GP), General Regression Neural Network (GRNN)[9], and Multilayer Perceptron (MLP)[10], have been increasingly applied to develop DFR prediction models. These methods are regarded as effective regression tools due to their capability to predict discharge rates across continuous values of L and N and to achieve superior predictive accuracy.

From a practical perspective, enhancing equipment utilization often requires a single applicator to handle multiple fertilizer types. However, most existing models proposed in previous studies are constructed using datasets obtained from indoor calibration tests specific to a particular fertilizer. Consequently, these models are

Received date: 2025-12-17 **Accepted date:** 2026-07-14

Biographies: Jiqin Zhang, Lecturer, research interest: intelligent agricultural equipment technology, Email: zhjq2010jasmine@163.com; **Lin Xi**, Associate Professor, research interest: testing technology, Email: eve_q@163.com; **Gang Liu**, Professor, research interest: agriculture information technology, Email: pac@cau.edu.cn; **Zhao Zhang**, Professor, research interest: key technology of smart agriculture, Email: zhaozhangcau@cau.edu.cn; **Changyuan Zhai**, Professor, research interest: intelligent spraying technique, Email: zhaicy@nercita.org.cn; **Shuo Yang**, Assistant Professor, research interest: intelligent spraying technique, Email: yangshuo@nercita.org.cn.

***Corresponding author: Qibin Zhuang**, Lecturer, research interest: intelligent technology and equipment for inspecting of agricultural produce. No. 54, Beijing Middle Road, Jiujiang District, Wuhu 241000, Anhui, China. Tel: +86-18154256869, Email: zhuangqibin@126.com.

difficult to transfer directly to other fertilizer types without recalibration. Moreover, prevailing DFR models primarily rely on operational parameters (L and N), while overlooking the influence of physical fertilizer characteristics. It is well established that different fertilizers exhibit distinct flow and discharge behaviors, leading to considerable variation in discharge amounts even under identical operational settings. As a result, applying a DFR model to a new fertilizer type without recalibration can substantially reduce prediction accuracy or even render the model invalid. Yet, the recalibration process is typically labor-intensive and time-consuming[11]. Therefore, investigating transfer learning methods for DFR prediction models across fertilizer types is of significant practical importance. Such research would enhance the generalization capability of models, improve modeling efficiency, increase equipment utilization, and facilitate the broader adoption of variable-rate fertilization technology.

Transfer learning (TL) presents a promising approach that adapts pre-trained models from a source domain to a related target domain, effectively addressing data scarcity issues while enhancing predictive performance and generalization capability[12,13]. This technique has been successfully applied in the agricultural field and achieved impressive results, such as in weed and seed detection[14], plant diseases classification[15–17], yield prediction[18,19], soil moisture prediction[20], etc. TL aims to improve target domain performance by leveraging knowledge acquired from the source domain, thereby eliminating the need for repeated recalibration through the reuse of models initially trained on master instruments[21]. Among TL strategies, parameter-based and instance-based methods are commonly employed to tackle inductive TL problems, especially those involving limited labeled data in the target domain[22]. Instance-based transfer learning operates under the assumption that certain source domain data can be repurposed for the target task via re-weighting. This approach facilitates transfer by assigning adjusted weights to labeled instances from the source domain to improve relevance in the target context[11]. Key techniques within this category include instance selection and instance re-weighting[22]. TrAdaBoost (Transfer Adaptive Boosting) is one widely used instance selection method that assigns higher weights to source domain samples closely aligned with the target domain, while reducing the influence of dissimilar samples through iterative re-weighting[23]. For instance, Huang et al.[24] combined the TrAdaBoost.R2 with partial least squares regression (PLSR) to estimate three key foliar nutrients in potatoes across five growth stages. Their findings indicated that incorporating TL techniques significantly improved model transferability across growth stages and enhanced overall generalizability.

Based on the literature reviewed, it can be concluded that ML methods serve as effective regression tools for constructing discharge fertilizer rate (DFR) prediction models. However, most existing models have been developed using only operational parameters (L and N), while neglecting the material characteristics of fertilizers. These material properties significantly influence the dynamic behavior of fertilizers during discharge. For example, sphericity and repose angle reflect the flowability of fertilizer particles, while equivalent diameter indicates particle size. Variations in bulk density can also lead to differing discharge amounts under identical operational settings (L and N). Moreover, the degree of divergence in material properties across fertilizer types may considerably affect the performance of TL. TL methodologies offer a promising solution to cross-domain challenges by leveraging knowledge from pre-trained models in a source domain, and have demonstrated notable success in various smart agriculture applications. Nevertheless, the prediction of DFR across different fertilizer types in a BFA remains unexplored. Thus, this study aims to investigate the generalizability and transferability of DFR prediction models across fertilizer types using TL, and to analyze the influence of both operational and material parameters on TL performance. Specifically: 1) incorporate both operational parameters (L and N) and key fertilizer material characteristics, including sphericity, equivalent diameter, repose angle, and bulk density as inputs to the ML model for DFR prediction; 2) evaluate the potential of instance-based TL methods combined with decision tree regression (DTR) for cross-fertilizer DFR prediction in a BFA; and 3) analyze the effects of fertilizer material properties and operational parameters on the efficacy of transfer learning.

2 Material and methods

2.1 Calibration platform and data preparation

To acquire calibration datasets for modeling the DFR of different fertilizer types, a bivariate fertilization calibration platform was developed. Three fertilizers were selected for indoor calibration tests: two compound fertilizers (Stanley and Sakefu) and urea. The datasets comprise both static and dynamic parameters. The static parameters, also referred to as fertilizer material characteristics, include equivalent diameter (d_e), bulk density (ρ_v), sphericity ratio (φ_s), and repose angle (α). The dynamic parameter corresponds to the discharge fertilizer rate (Q) under varying combinations of the operational parameters, feed-roll length (L) and rotational speed (N).

2.1.1 Calibration platform

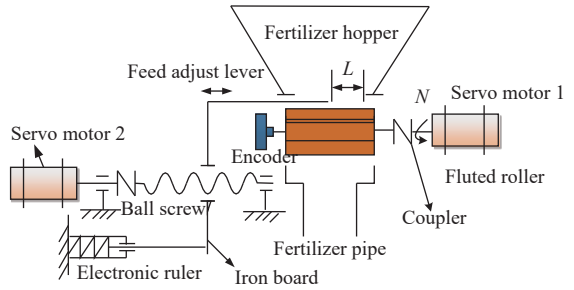
The bivariate-fertilizer calibration platform, illustrated in Figure 1[9], differs from traditional variable-rate fertilization (VRF) systems in that it enables simultaneous adjustment of the fertilization rate (Q) via two operational parameters: the rotational speed (N) of the fluted roller shaft and the active feed-roll length (L) of the hopper opener. As depicted in Figure 1a, the platform consists of four main components: a fertilizer hopper, a fluted metering roller, a rotational speed adjustment device, and an opening length adjustment mechanism. The rotational speed N is directly controlled by servo motor 1 (24 V, MoTEC, German) through a coupling assembly. The device for adjusting L includes servo motor 2 (24 V, 200 W), a ball screw, and a movable baffle affixed to the feed adjustment lever. The length L is determined based on the rotation cycles of the ball screw, which are calculated according to the thread lead (5 mm/revolution), which corresponds to specific pulse counts of servo motor 2. As the servo motor rotates, the ball screw turns synchronously, translating rotational motion into horizontal displacement of the baffle via the feed adjustment lever. As a result, the fertilizer application rate can be precisely controlled with the two servo motors.

Two sensors were integrated into the platform: an encoder to monitor the rotational speed of the fluted metering roller, and a linear displacement sensor (electronic ruler) to measure the real-time position of the baffle. One end of the displacement sensor was maintained in contact with a metal plate that moved synchronously with the feed adjustment lever. The sensor offers a detection range of 0-100 mm, with an accuracy of 0.01 mm. A photograph of the experimental platform is provided in Figure 1b. In addition, a dedicated software application developed in C# was implemented to centralize the control and data acquisition functions.

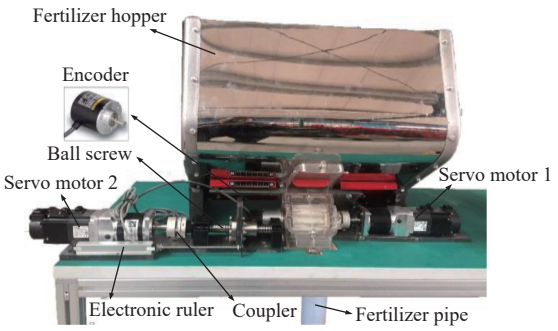
2.1.2 Data preparation

Three types of granular fertilizers commonly used for crops

such as corn, wheat, cotton, and beans were selected: two potassium sulfate compound fertilizers (Stanley and Sakefu) and urea. The datasets for modeling the DFR prediction comprise both static and dynamic feature parameters. The static feature parameters, also referred to as material characteristic parameters, remain stable for a given fertilizer type. These include sphericity ratio (φ_s), equivalent diameter (d_e), repose angle (α), and bulk density (ρ_v). In contrast, dynamic feature parameters, such as the discharge fertilizer rate (Q), vary with changes in operational parameters (L and N). Accordingly, the static parameters of the three fertilizers were measured under laboratory conditions, while the dynamic parameters were obtained through indoor calibration tests.



a. Structure diagram of the test platform



b. Real image of the test platform

Figure 1 The bivariable control test platform[9]

(1) Static feature parameters

The static feature parameters, also referred to as material characteristic parameters, reflect the motion behavior of fertilizers during application. For example, the sphericity ratio and repose angle indicate the flowability of fertilizer particles. A higher sphericity ratio generally corresponds to a lower repose angle, indicating better flowability. The equivalent diameter reflects the particle size, which affects the number of particles entering the applicator per unit time. Bulk density, defined as mass per unit volume, contributes to differences in discharge rates among fertilizers even under identical operational settings (L and N). Accordingly, these parameters, including sphericity ratio (φ_s), equivalent diameter (d_e), repose angle (α), and bulk density (ρ_v), were measured in the laboratory for the three selected fertilizers (Stanley, Sakefu, and urea), following relevant standards and using appropriate instruments[25]. Photographs of the three granular fertilizers are presented in Figure 2, which includes (a) Stanley, (b) Sakefu, and (c) urea.

The details of material characteristics of the three fertilizers, including nitrogen (N), phosphate (P_2O_5), and potassium (K_2O) content, sphericity ratio (φ_s), equivalent diameter (d_e), repose angle (α), and bulk density (ρ_v), are summarized in Table 1. The equivalent diameter (d_e) was measured as follows: first, 100 particles of each fertilizer were randomly selected. The length (l), width (w), and thickness (t) of each particle were measured using a Vernier caliper with an accuracy of 0.01 mm. The equivalent diameter was then calculated using Equation (1):

$$d_e = \sqrt[3]{lwt} \tag{1}$$

where, l, w, and t represent the length, width, and thickness (in mm) of the granule, respectively. Subsequently, the sphericity ratio (φ_s) was calculated based on Equation (2):

$$\varphi_s = \frac{d_e}{l} \tag{2}$$

(2) Dynamic feature parameters

To acquire the dynamic feature parameters for the three fertilizer types, a series of calibration experiments were carried out. Each test was conducted under static measurement conditions by systematically varying the operational parameters L (feed-roll length) and N (rotational speed). The values of L and N were determined based on the minimum operating speed required for the servo motor to maintain stable performance, as outlined in Table 2.

a. Stanley b. Sakefu c. Urea

Figure 2 Real image of the calibration fertilizer

Table 1 Summary of the material characteristic parameters

Name of the fertilizer	Compositions of Nitrogen, Phosphate, and Potassium	Sphericity ratio φ_s/%	Equivalent diameter d_e/mm	Repose angle α/°	Bulk density ρ_v/kg/m^3
Stanley	19:19:19	90.00	3.62	21.60	852.70
Sakefu	17:17:17	88.39	3.57	31.85	1115.40
Urea	—	94.84	2.19	12.83	741.67

Table 2 Basic parameters for calibration experiments

Parameters	Range	Interval	Groups
Opening length L /mm	10-60	5	11
Rotational speed N /r·min^{-1}	10-50	5	9

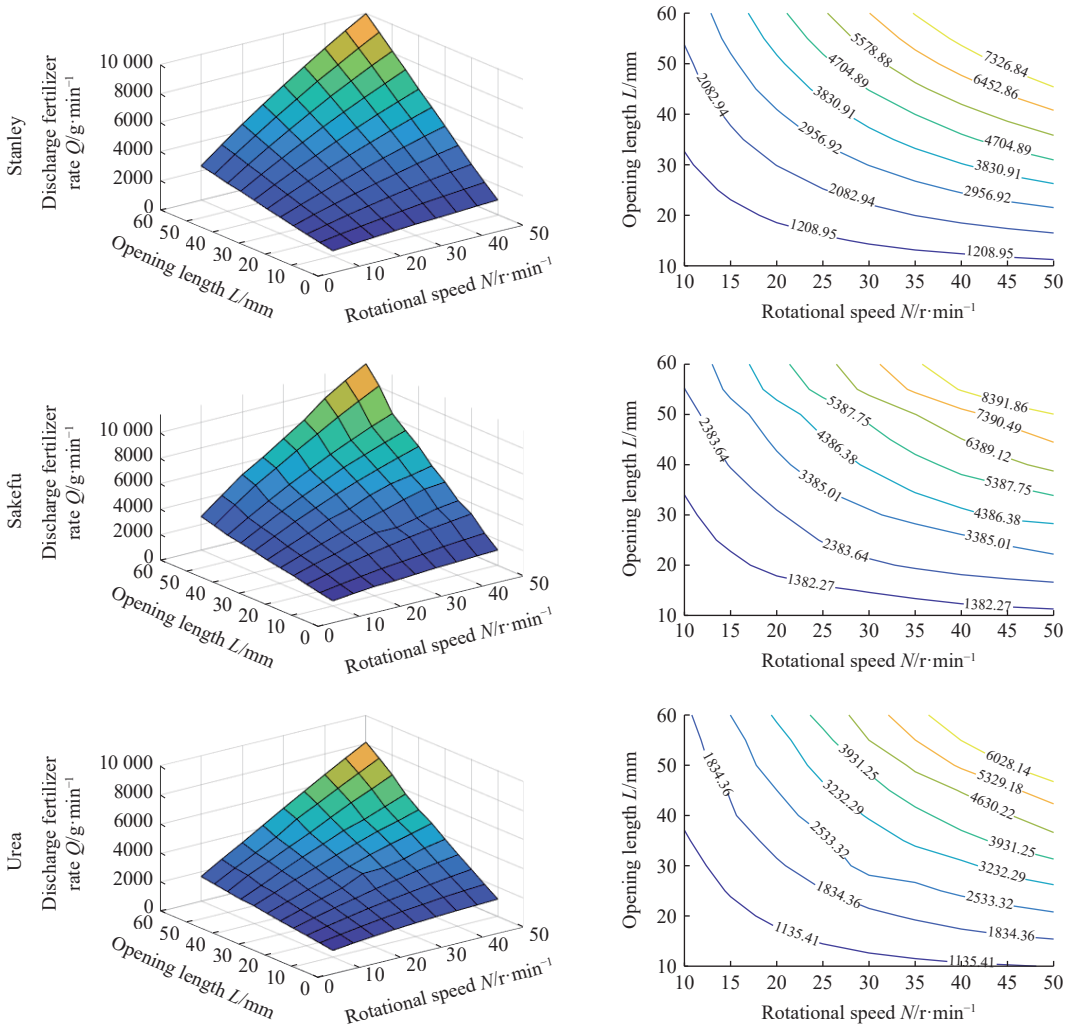
At the start of each calibration test, the fertilizer hopper was filled with one type of fertilizer. The bivariate fertilization platform was operated at a specific combination of L and N for one minute. The discharged fertilizer was collected in a container and weighed using an electronic balance. Each parameter set was replicated at least three times, and the average discharge rate per minute was computed. This process yielded 99 data groups per fertilizer type. After completing the tests for one fertilizer, the same procedure was repeated for the remaining two types to compile a comprehensive calibration dataset.

After calibration tests, each type of fertilizer got 99 groups of data, and the data were gathered and fitted by MATLAB, forming the curve fitting plot of the three types of fertilizer. As shown in Figure 3, the discharge characteristics of different fertilizers exhibit distinct distribution patterns. For the same fertilizer, the discharge rate increases with larger control parameters (L and N).

(3) Data preprocessing

Before model training, all input features (L, N, and the four static properties: sphericity, equivalent diameter, repose angle, bulk density) were standardized using z-score normalization (zero mean,

unit variance). The normalization parameters (mean μ and standard deviation σ) were computed exclusively from the target domain training set (20% of target data) to avoid information leakage from the test set. The same μ and σ were then applied to the source domain data and the target test set. This ensures that similarity calculations (e.g., Gaussian kernels used in Transfer Adaptive Boosting for Regression, TrAdaBoost.R2) are based on a consistent scale that reflects the target domain's distribution.



a. Curve fitting of discharge fertilizerrate b. Contour map of discharge fertilizerrate

Figure 3 Fertilization rate sample data from the calibration experiments

To reduce random measurement noise, each calibration point (a specific combination of L and N) was measured in triplicate, and the three replicates were averaged. If a replicate deviated from the mean by more than three standard deviations (indicating possible mechanical disturbance), it was manually excluded and the measurement was repeated. Such cases accounted for less than 0.5% of the raw data. After this quality control, no extreme outliers remained in the final dataset.

2.2 Modeling method and TL approach

To examine the transferability of the DFR prediction model across fertilizer types, a DFR prediction model was first established and subsequently integrated with a TL method. In this study, DTR was employed as the base predictive model, coupled with the Transfer Adaptive Boosting for Regression (TrAdaBoost.R2) algorithm for knowledge transfer.

2.2.1 Decision tree regression method

Decision tree regression (DTR) is a nonlinear supervised learning algorithm that operates by recursively partitioning the feature space into a tree-like structure to make predictions[26]. It has demonstrated effectiveness in various agricultural applications, including irrigation management[27], crop yield prediction[28], and soil fertility analysis[29]. DTR was selected as the base modeling approach due to its high interpretability, computational efficiency, and strong performance even with limited sample sizes[30]. Moreover, it is well-suited for capturing nonlinear relationships.

In this work, the DFR prediction model was developed using DTR. For each fertilizer type, both material characteristic parameters and operational parameters (L and N) were used as model inputs. The output was the predicted discharge fertilizer rate per minute under specific operational settings. The dataset was split into training and testing subsets. After training, the DTR model was capable of accurately estimating the discharge rate for a given fertilizer at specified values of L and N.

2.2.2 Transfer learning approach

The core objective of TL is to leverage knowledge from a source domain to improve performance in a related target domain. Formally, TL involves a source domain D_S and learning task T_S, and a target domain D_T and learning task T_T. The goal is to enhance the target predictive function $f_T(\cdot)$ in D_T using knowledge acquired from D_S and T_S[22]. Specifically, the source domain D_S comprises a

well-labeled dataset $\{X_S, y_S\}$ drawn from a joint distribution $p(X_S, y_S)$, while the target domain D_T contains limited labeled data $\{X_T, y_T\}$ from joint distribution $p(X_T, y_T)$.

TrAdaBoost.R2 is a classical instance-based TL method designed for regression tasks[31]. It operates under the assumption that some source instances may aid learning in the target domain, while others could be detrimental. The method iteratively re-weights source instances, reducing the influence of misaligned ("bad") data and reinforcing useful ("good") data to improve transfer performance[32]. In this study, TrAdaBoost.R2 was implemented as follows (Table S4):

(1) Initialization: Set the number of boosting iterations $T = 10$ (justified by sensitivity analysis, Supplementary Fig. S2) and the ensemble size $K = 5$. Initialize source domain sample weights based on their similarity to target training samples (using a Gaussian kernel on dynamic and static features), and initialize target training sample weights uniformly.

(2) Iterative training: For each iteration $t = 1$ to T:

● Train a weighted DTR base model on the combined source-target dataset using current weights.

● Compute absolute prediction errors on the target training set, normalize them by the maximum error, and calculate the weighted average error E_t. If $E_t \geq 0.5$, terminate the loop.

● Set $\beta_t = E_t/(1 - E_t)$.

● Update target weights: $v_j^{(t+1)} = v_j^{(t)} \cdot \beta_t^{1-\tilde{\epsilon}_j}$, where $\tilde{\epsilon}_j$ are normalized errors.

● Update source weights: $w_i^{(t+1)} = w_i^{(t)} \cdot \beta_t^{\tilde{\epsilon}_i}$, where $\tilde{\epsilon}_i$ are normalized errors on the source domain.

● Normalize both weight vectors to sum to 1.

(3) Ensemble prediction: After T iterations, the final prediction is the weighted average of the last K models, with each model weighted by $1/\beta_t$ (normalized).

The choice of source domain significantly affects transfer performance, as a source domain with extreme characteristics may lead to negative transfer. Therefore, to select an appropriate source domain for transfer, the normalized Euclidean distances between the static feature vectors of each fertilizer pair were computed after min-max normalization. The calculated distances are as follows: Stanley to Sakefu: 0.92, Stanley to urea: 1.37, and Sakefu to urea: 1.98. These results indicate that Stanley is not an extreme source; rather, its static properties are moderately similar to both Sakefu and urea. Consequently, Stanley is the most suitable source domain for cross-fertilizer transfer learning in this study. Based on this analysis, Stanley compound fertilizer was selected as the source domain (D_{SST}), with Sakefu compound fertilizer and urea serving as target domain 1 (D_{T1}) and target domain 2 (D_{T2}), respectively. Transfer scenarios from D_{SST} to D_{T1} (Stanley to Sakefu) and D_{SST} to D_{T2} (Stanley to urea) were designed to evaluate the transferability of the DFR prediction model and to identify effective transfer strategies.

2.3 Experiment design and parameter setting

2.3.1 Experiment design

To evaluate the transferability of the DTR model in predicting discharge fertilizer rates across fertilizer types for the BFA, several comparative models were trained. The models and data usage are summarized in Table 3.

As shown in Table 3, the first model (DTR-D_S) serves as the baseline and is trained solely on source domain data (100%) to evaluate native transferability. The second model (DTR-D_T(20%)) uses only 20% of the target data, simulating a low-data scenario. The third model (DTR-recalibration) incorporates 20% target samples into the source domain for retraining, representing a conventional recalibration approach. The fourth model (DTR-TrAdaBoost.R2) employs the TrAdaBoost.R2 algorithm to reweight source instances along with 20% target labels for transfer learning. The final model, trained on all target domain samples, provides an optimal reference for comparative analysis.

Table 3 The model used in this study

No.	Model abbreviation	Explanations	Samples utilized in modeling	
			Source domain	Target domain
1	DTR-D_S	Direct source domain prediction: DTR model trained on 100% source domain samples, applied directly to the target domain as a transferability baseline.	100%	-
2	DTR-D_T(20%)	Partial target domain prediction: DTR model trained using only 20% of target domain samples.	-	20%
3	DTR-Recalibration	Joint training: DTR model retrained on both source domain and 20% target domain samples.	100%	20%
4	DTR-TrAdaBoost.R2	Instance-based transfer learning: DTR integrated with TrAdaBoost.R2, using reweighted source data and 20% target data.	100%	20%
5	DTR-D_T	Full target domain prediction: DTR model trained on 100% target domain samples, representing the performance upper bound.	-	100%

For each target fertilizer, the calibration dataset contains 99 samples covering the full ranges of operating parameters (L and N). To simulate a low-data scenario commonly encountered in precision agriculture, where collecting a full calibration dataset for each new fertilizer is time-consuming, 20% of the target domain samples (20 out of 99) were randomly selected without replacement as the labeled training set, and the remaining 80% (79 samples) were used as the test set. Simple random sampling was employed without any bias toward specific operating conditions (e.g., moderate L-N ranges). In the main experiments, a fixed random seed was used to ensure reproducibility. To assess the stability of the results under different random splits, the entire evaluation was repeated 10 times with different random seeds and the mean ± standard deviation for each performance metric was reported (Supplementary Table S2).

2.3.2 Pruning environment and parameter setting

All experiments were performed on a workstation running Windows 11, equipped with an Intel® Core™ i9-12900H CPU (2.40 GHz) and 32 GB of RAM. The models were implemented in MATLAB R2023a.

DTR was implemented using MATLAB's fitrtree function. Its hyperparameters (minimum leaf size, maximum number of splits, split criterion, and pruning) are listed in Table 4 under the component "DTR"). For the TrAdaBoost.R2 algorithm, the number of iterations and ensemble size are also summarized in Table 4 (Component "TrAdaBoost.R2"), and a pseudocode is provided in Supplementary Algorithm S1. Other parameters were set empirically.

The iteration number T for TrAdaBoost.R2 is critical because values that are too small or too large may degrade model performance. A sensitivity analysis of the iteration number was therefore conducted on both transfer tasks using 10 independent random data splits for each T. The results (Supplementary Table S1

and Figure S1) show that $T = 10$ achieves robust and competitive performance across tasks: it gives the highest mean R^2 for Stanley to Sakefu (0.8957) and performs comparably to the best T for Stanley to urea (0.8183 vs. 0.8246 at $T = 20$, with no statistical difference). Larger T do not bring consistent gains and may increase overfitting risk. Hence, $T = 10$ was selected to balance convergence and generalization.

Table 4 Key hyperparameters of modeling framework

Component	Parameter	Symbol/value
DTR	Minimum leaf size	MinLeafSize = 5
	Maximum number of splits	MaxNumSplits = 100
	Split criterion	Mse
	Pruning	Off (no pruning)
Similarity weighting	Dynamic feature weights	α=0.7
	Static feature weights	β=0.3
	Gaussian kernel width of dynamic features	σ_{dym}=0.5
	Gaussian kernel width of static features	σ_{staic}=0.8
TrAdaBoost.R2	Number of iterations	T=10
	Ensemble size	K = 5 (last 5 models)

Note: T selected based on sensitivity analysis with 10 repeated random splits (Supplementary Figure S1).

2.3.3 Model evaluation criteria

The choice of evaluation metrics is essential for comprehensively assessing the transferability performance of the models. To evaluate the prediction accuracy of the DFR models, the coefficient of determination (R^2) and the normalized root mean square error ($nRMSE$) were selected. To facilitate a clear comparison of transferability across models, the difference percentage (DP) in R^2 and $nRMSE$ between the baseline model (DTR-D_S) and each comparative model was computed. The formulas for these metrics are given below:

(1) Coefficient of determination (R^2)

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i^{pre} - y_i^{exp}\right)^2}{\sum_{i=1}^{n}\left(y_i^{exp} - \overline{y_i^{exp}}\right)^2}, \ (i = 1,\ldots,n) \quad (3)$$

where, n is the number of test samples ($n = 79$), y_i^{pre} is the i^{th} predicted value of the trained DTR models, y_i^{exp} is the i^{th} measured value from calibration experiment, and $\overline{y_i^{exp}}$ is the mean of measured value. R^2 ranges from 0 to 1, with higher values indicating better fit.

(2) Root mean square error ($RMSE$) and normalized $RMSE$ ($nRMSE$)

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\left(y_i^{pre} - y_i^{exp}\right)^2}{n}} \times 100\% \quad (4)$$

$$nRMSE = \frac{RMSE}{y_{\max} - y_{\min}} \times 100\% \quad (5)$$

where, $y_{\max}$ and $y_{\min}$ are the maximum and minimum measured values in the test set, respectively. $RMSE$ expresses prediction error. $nRMSE$ expresses the prediction error as a percentage of the measurement range, and lower values indicate better accuracy[33].

(3) Difference percentage (DP) for comparing methods

To quantify improvement over the baseline model (DTR-D_S), we computed:

$$DP_{R^2} = \frac{(R_j^2 - R_{baseline}^2)}{R_{baseline}^2} \times 100\%, \ (j = 1,\ldots,m) \quad (6)$$

$$DP_{nRMSE} = \frac{(nRMSE_{baseline} - nRMSE_j)}{nRMSE_{baseline}} \times 100\%, \ (j = 1,\ldots,m) \quad (7)$$

where, R_j^2 and $nRMSE_j$ denote the metrics for the j^{th} model ($j = 1,\ldots,m$), and $R_{baseline}^2$ and $nRMSE_{baseline}$ correspond to the baseline model (DTR-D_S); m is the number of comparative models (m=4). Positive DP values indicate improvement over the baseline, while negative values denote degradation.

3 Results and discussion

3.1 Transferability analysis of DFR prediction from Stanley to Sakefu

3.1.1 Comparison of transfer performance across methods

The transferability performance for BFA across from Stanley to Sakefu is summarized in Table 5 and illustrated in Figure 4. A scatter plot comparing the predicted versus actual DFR for Sakefu under each model is shown in Figure 5.

Table 5 Comparison of transfer performance for different models (Stanley to Sakefu)

Models	R^2	$nRMSE$/%	DP_{R^2}/%	DP_{nRMSE}/%
DTR-D_S	0.904	7.73	—	—
DTR-D_T(20%)	0.712	13.43	–21.2	–73.7
DTR-Recalibration	0.926	6.76	2.4	12.5
DTR-TrAdaBoost.R2	0.873	8.91	–3.4	–15.3
DTR-D_T	0.927	6.77	2.5	12.4

Note: The values shown are from a single representative data split (fixed random seed). The mean ± standard deviation over 10 independent splits is provided in Supplementary Table S2.

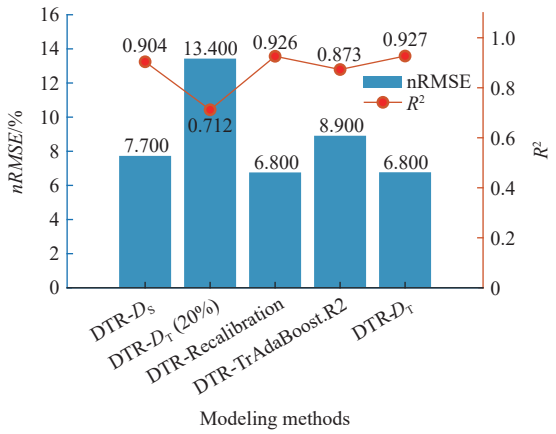


Figure 4 DFR prediction results of the samples in the Sakefu

As indicated in Table 5 and Figure 4, the DTR-D_T (full target domain training) model represents the theoretical performance upper bound, yielding an R^2 value of 0.927 and $nRMSE$ of 6.77%. Meanwhile, the DTR-D_T (20%) model exhibits the lowest performance, with an R^2 value of 0.712 and an $nRMSE$ of 13.43%. The baseline model (DTR-D_S) achieved a moderate performance, with an R^2 value of 0.904 and $nRMSE$ value of 7.73%. The key observation is that DTR-Recalibration (joint training) with an R^2 value of 0.926 closely approaches the upper bound (R^2=0.927), whereas DTR-TrAdaBoost.R2 slightly underperforms the baseline—a case of negative transfer (R^2 =0.873).

As can be observed from the scatter plots in Figure 5, DTR-D_T (20%) exhibits the most dispersed predictions with substantial

deviations from the ideal line, reflecting its poor generalization due to insufficient training data. In contrast, DTR-Recalibration shows the most concentrated distribution around the ideal line, confirming its superior predictive consistency. DTR-TrAdaBoost.R2, while maintaining a reasonable overall fit, shows slightly greater scatter compared to DTR-Recalibration, consistent with its marginal performance degradation. Paired t-tests on the 10-replicate R^2 values (Supplementary Table S3) confirm that DTR-Recalibration significantly outperforms DTR-D_S (p=0.0003) and DTR-TrAdaBoost.R2 (p=0.0305). The non-significant difference between DTR-TrAdaBoost.R2 and DTR-D_S (p=0.4990) further supports the negative transfer discussed in Section 3.1.2.

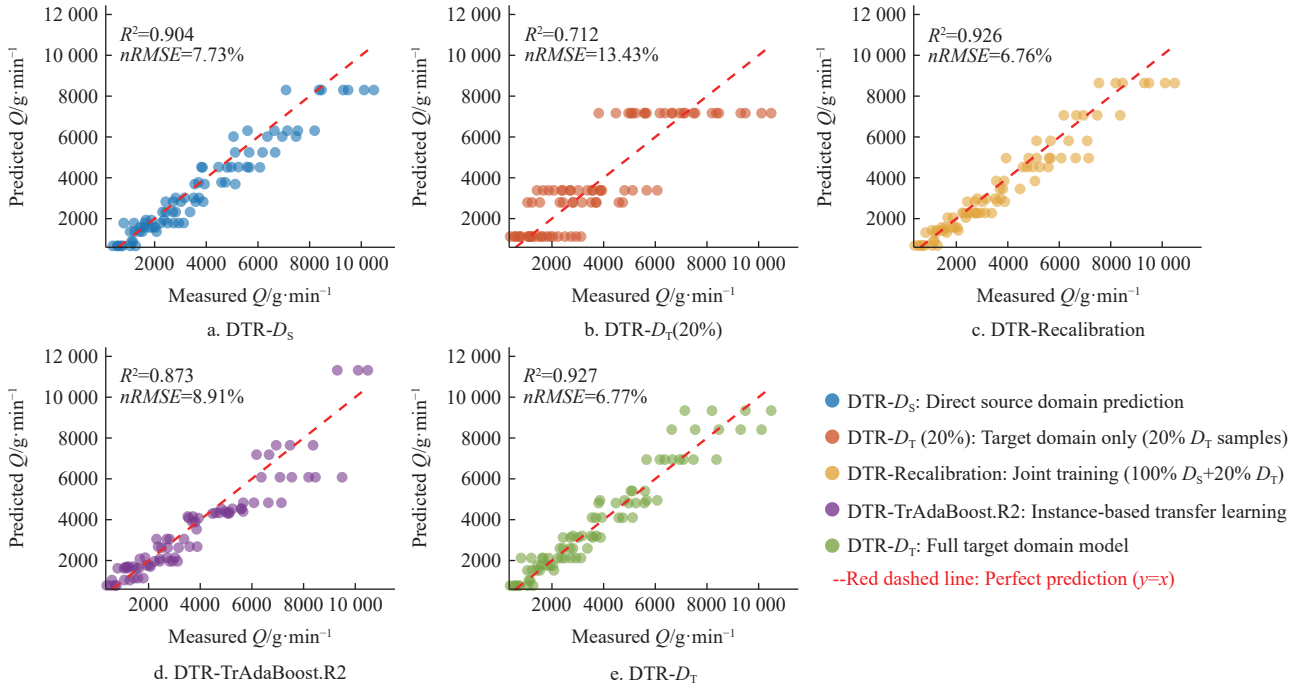


Figure 5 DFR model transfer performance comparison (Stanley to Sakefu)

To enable an intuitive comparison of transferability across modeling approaches, the percentage differences (*DP*) based on baseline in R^2 and *nRMSE* for all other models are illustrated using bar plots (as in Figure 6), where positive percentage changes are indicated in green and negative changes in red, corresponding to improved and degraded performance, respectively.

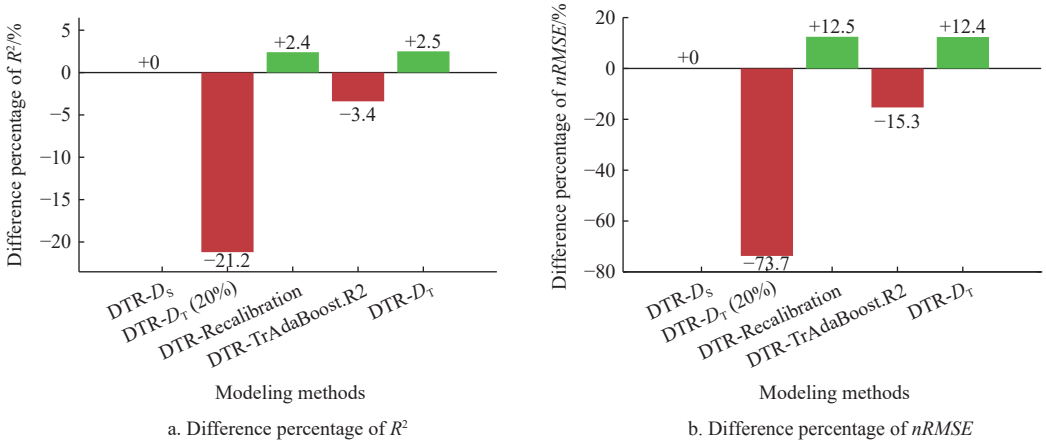


Figure 6 Difference percentage (DP) of R^2 and *nRMSE* compared with baseline model (DTR-D_S)

As shown in Figure 6, both the DTR-Recalibration (joint training) and DTR-D_T (full target domain training) methods improved the prediction accuracy of the DFR. The DTR-D_T model represents the theoretical performance upper bound, while the DTR-Recalibration model performed comparably, approaching this upper limit. Both models showed significant improvements in R^2 and *nRMSE*, with a 2.4% increase in R^2 and a greater than 12% reduction in *nRMSE*, indicating a notable enhancement in predictive accuracy and error reduction. The superiority of DTR-Recalibration over other methods was consistent across 10 independent random data splits (Supplementary Table S1). In contrast, the instance-based transfer learning method (DTR-TrAdaBoost.R2) resulted in slightly degraded performance, with a 3.4% decrease in R^2 and a 15.3% decrease in *nRMSE*, suggesting its limited effectiveness in this specific transfer scenario. The DTR-D_S (20%) model performed the poorest, exhibiting a substantial decline in accuracy, with a 21.2% reduction in R^2 and a decrease of 73.7% in *nRMSE* as shown in Figure 6b. This suggests that training with only 20% of the target data impaired model performance, likely due to insufficient data.

To further evaluate the prediction errors across the five methods, we generated boxplots of the residuals (Figure 7). DTR-D_T (full target training) yields a relatively narrow interquartile range

(IQR ≈ 450 g/min) and a median close to zero, representing the expected upper bound. DTR-Recalibration exhibits the narrowest IQR (≈430 g/min) and a median near zero, with only a few outliers, confirming its effectiveness in transferring knowledge from Stanley to Sakefu. In contrast, DTR-D_T(20%) produces an extremely wide IQR (>1300 g/min) and a median of approximately −80 g/min, indicating high uncertainty and systematic underestimation due to insufficient target data. Direct transfer (DTR-D_S) shows a moderate IQR (≈480 g/min) but its median is slightly negative (−18 g/min), suggesting a mild underestimation of the discharge rate for Sakefu.

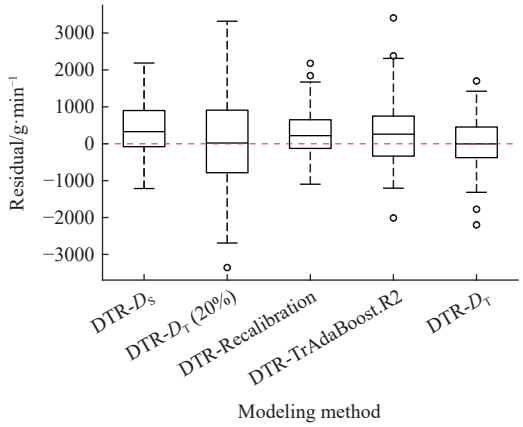


Figure 7 Boxplots of prediction residuals for the five methods (Stanley to Sakefu). The red dashed line indicates zero residual. Boxes show interquartile range (IQR), central lines denote medians, and whiskers extend to 1.5IQR. Outliers are shown as circles

Notably, DTR-TrAdaBoost.R2 not only has a wider IQR (≈ 550 g/min) than DTR-D_S but also exhibits more negative outliers (e.g., several residuals below −1000 g/min). This indicates that the instance-based transfer learning method amplifies prediction errors in certain operating conditions, leading to the negative transfer phenomenon observed in Table 5 (R^2 reduced from 0.904 to 0.873, $RMSE$ increased from 781.5 to 900.1 g/min). The residual boxplot thus provides visual evidence that the weighting mechanism of TrAdaBoost.R2 fails to improve the prediction for this fertilizer pair, consistent with the mechanism analysis presented in the following section.

3.1.2 Correlation analysis between TL effects and fertilizer characteristics (Stanley to Sakefu)

To investigate the relationship between TL performance and fertilizer properties, the influence of both static and dynamic features on TL effectiveness was examined.

(1) Static feature difference and correlation with TL effects

The similarity in material properties between two fertilizers determines the feasibility and potential effectiveness of TL. Under identical dynamic parameters (L and N), differences in static properties lead to variations in discharge volume, providing insight into the mechanistic basis of TL.

To quantify these differences, the percentage variation in static characteristics between the source fertilizer (Stanley) and the target fertilizer (Sakefu) are summarized in Table 6.

As shown in Table 6, the two fertilizers have very similar particle size and shape (differences <2%), but differ markedly in repose angle (47.5%) and bulk density (30.8%). This "partially similar, partially different" pattern leads to condition-dependent discharge behavior: under low speed or small opening length, the repose angle (flowability) dominates; under high speed or large opening, bulk density becomes more influential. Consistent with the results presented in Section 3.1.1, the source domain model (DTR-D_S) indicates that the similarity particle characteristics enable reasonable direct transfer. In the relationship between the dynamic parameters (L and N) and the discharge rate Q for the two fertilizers, the DTR-Recalibration model achieved the highest performance, confirming that adding 20% target data effectively compensates for the static differences.

Table 6 Static feature analysis of Stanley and Sakefu

Parameter	Stanley	Sakefu	Percentage difference (%)
Sphericity ratio φ_s/%	90.00	88.39	1.8
Equivalent diameter d_e/mm	3.62	3.57	1.4
Repose angle α/°	21.6	31.85	47.5
Bulk density ρ_v/kg/m³	852.70	1115.4	30.8

In contrast, the DTR-TrAdaBoost.R2 underperforms direct transfer, a typical case of negative transfer. The failure arises because the heterogeneous static pattern makes uniform instance weighting ineffective across varying operating conditions. The final source weights are nearly uniform, confirming the algorithm's inability to identify consistently useful samples. Joint training, which simply merges data without iterative reweighting, averages out the distributional differences and thus achieves robust performance.

The model trained with only 20% target domain data (DTR-D_S (20%)) gives the lowest performance (R^2=0.712), as the small sample size cannot overcome the substantial differences in repose angle and bulk density.

(2) Dynamic feature analysis and correlation with TL effects

According to the calibration tests, the dynamic characteristics (i.e., combinations of L and N) are consistent across the three fertilizers, while differences in discharge rates are primarily attributable to variations in their static properties. The instance-based transfer learning algorithm TrAdaBoost.R2 iteratively adjusts the weights of source domain samples to identify those with distributions similar to the target domain. Analyzing the dynamic characteristics (L and N) of these high-weight samples can clarify which operational parameter combinations are most influential for TL, thereby guiding the optimization of TL strategies and the design of experimental sample collection.

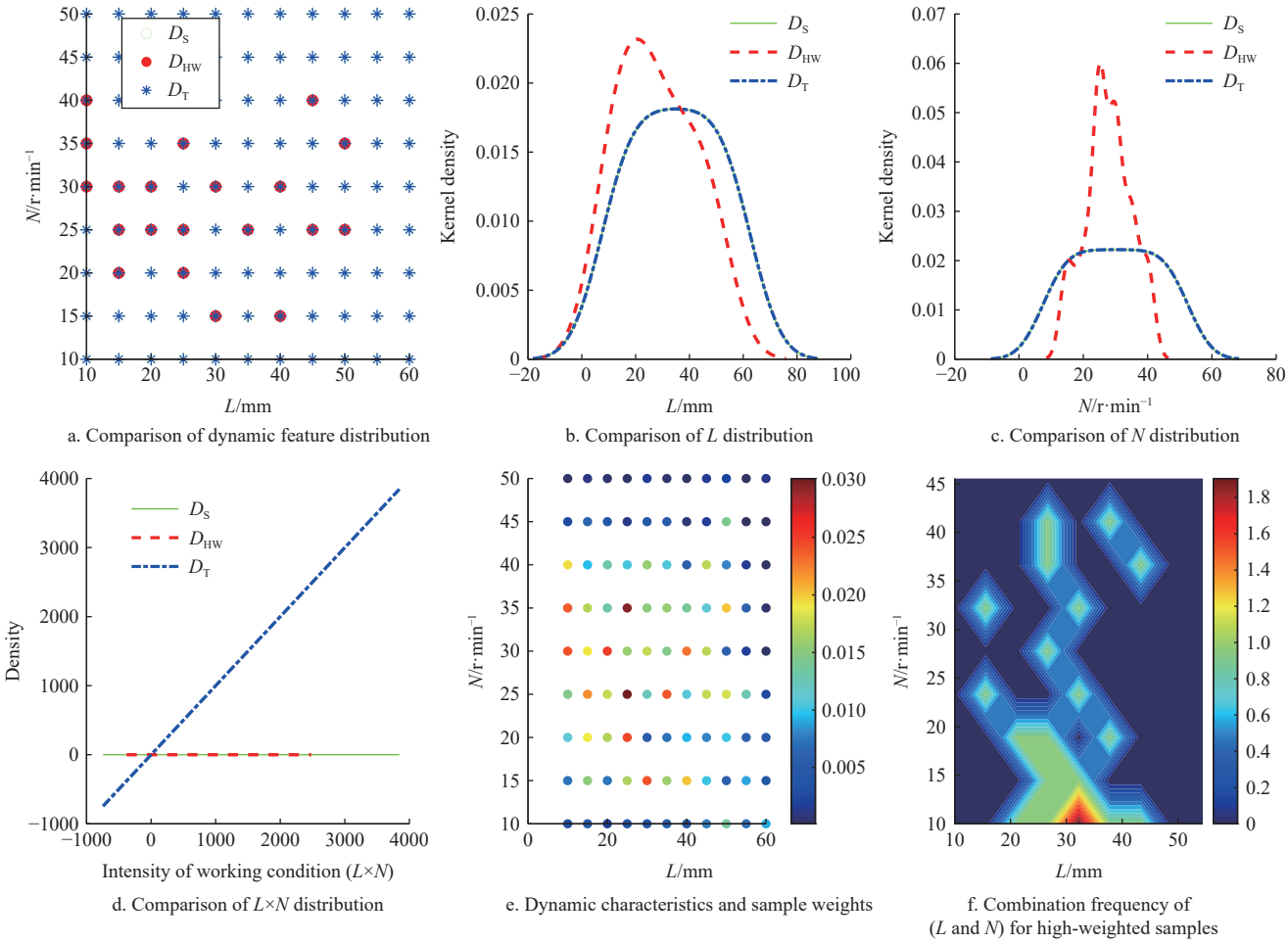
To this end, we extracted high-weight source domain samples (denoted as D_{HW}) from the TrAdaBoost.R2 transfer process from Stanley to Sakefu and visualized their dynamic characteristic distributions, specifically, L, N, and the product L×N, on the L-N plane, as shown in Figure 8.

For a more detailed analysis, we computed summary statistics, including the opening length (L), rotational speed (N), operational intensity (L×N), and discharge ratio (($Q_{Sakefu}/Q_{stanley}$), for the top 20% of high-weight samples. The results are listed in Table 7 and illustrated in Figure 9.

As shown in Table 7, the high-weight samples are predominantly distributed around medium operating ranges (L ≈ 28 mm, N≈ 28 r/min). However, the standard deviation of the discharge ratio (Q_{Sakefu} to $Q_{stanley}$) for these high-weight samples (0.0545) is actually larger than that of the full dataset (0.0500), indicating that their discharge behavior is less stable. Thus, TrAdaBoost.R2 mistakenly assigns high weight to unstable samples, introducing noise into the model. In contrast, joint training (DTR-Recalibration) avoids iterative reweighting and directly merges source and target data, effectively averaging out

distributional differences and achieving robust performance as discussed in Section 3.1.1.

This spatial pattern is further corroborated by Figure 9, which visualized the weight distribution across the *L*-*N* plane using a red-to-blue colormap with contour lines indicating distribution boundaries. The heatmap also highlights individual samples with weights exceeding 0.02, marking the most influential source domain instances, which are located at (*L*=25, *N*=25), (*L*=25, *N*=35). Consistent with Table 7, the high-weight regions are predominantly concentrated around medium operating ranges (*L*≈28 mm, *N*≈28 r/min), while extreme *L*-*N* combinations receive substantially lower weights.



a. Comparison of dynamic feature distribution

b. Comparison of *L* distribution

c. Comparison of *N* distribution

d. Comparison of *L*×*N* distribution

e. Dynamic characteristics and sample weights

f. Combination frequency of (*L* and *N*) for high-weighted samples

Note: D_s represents the samples from Stanley; D_{HW} represents high-weighted samples; D_T represents samples from Sakefu.

Figure 8 Distribution of dynamic characteristics (*L*, *N*, and *L*×*N*) of high-weight samples on the *L*-*N* plane (Stanley to Sakefu)

Table 7 Statistical characteristics of the top 20% high-weight samples (Stanley to Sakefu)

Parameter	High-weight samples (top 20%)			All samples		
	Distribution range	Mean value	Standard deviation	Distribution range	Mean value	Standard deviation
L/mm	10-50	27.8	-	10-60	35	-
N/r·min⁻¹	15-40	27.8	-	10-50	30	-
L×*N*/mm·r·min⁻¹	300-1800	761.2	-	100-3000	1050	-
$Q_{Sakefu}/Q_{stanley}$	1.035-1.217	1.122	0.0545	1.026-1.261	1.111	0.0500

3.2 Transferability analysis of DFR prediction from Stanley to urea

3.2.1 Comparison of transfer performance across methods

The transfer performance metrics of various models for the BFA from Stanley to urea are summarized in Table 8 and illustrated in Figure 10. Figure 11 presents the scatter plots of predicted versus actual DFR for urea under each model.

As shown in Table 8, the DTR-D_T model, trained on the full target domain dataset, achieved the highest prediction accuracy, with an R^2 value of 0.944 and *nRMSE* of 6.05%. In contrast, the DTR- D_T (20%) model, trained using only 20% of the target data, showed the lowest performance, with an R^2 value of 0.436 and *nRMSE* of 19.23%. The baseline model (DTR-D_S), trained solely on the source domain (Stanley), demonstrated moderate performance, with an R^2 value of 0.728 and *nRMSE* of 13.36%. The remaining methods improved transferability to varying degrees.

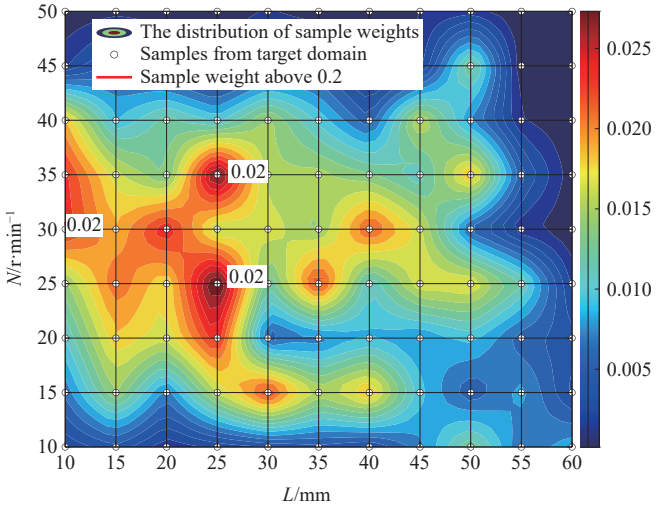


Figure 9 Distribution of sample weights on the *L*-*N* plane

Table 8 Comparison of transfer performance for different models (Stanley to urea)

Models	R^2	*nRMSE*/%	DP_{R^2}/%	DP_{nRMSE}/%
DTR-D_S	0.728	13.36	-	-
DTR-D_T(20%)	0.436	19.23	–40.1	–43.9
DTR-Recalibration	0.745	12.93	2.3	3.2
DTR-TrAdaBoost.R2	0.832	10.49	14.3	21.5
DTR-D_T	0.944	6.05	29.7	54.7

*The values shown are from a single representative data split (fixed random seed). The mean ± standard deviation over 10 independent splits is provided in Supplementary Table S2.

Paired t-tests based on the 10 independent random data splits (Supplementary Table S3) show that DTR-TrAdaBoost.R2 significantly outperforms DTR-Recalibration (p=0.0033), DTR-D_S (p=8.3×10^{-6}), and DTR-D_T (20%) (p=2.2×10^{-5}). DTR-Recalibration also performs significantly better than DTR-D_S (p=0.000 80). These results confirm that the observed advantages of instance-based transfer learning for the urea task are statistically robust.

To quantitatively assess the transferability improvement, DTR-D_S model was utilized as the benchmark. The percentage differences in R^2 and *nRMSE* of other models relative to this baseline were computed and are summarized in Table 8. Figure 12 presents bar plots illustrating these difference percentages, with green bars indicating performance improvement and red bars indicating degradation.

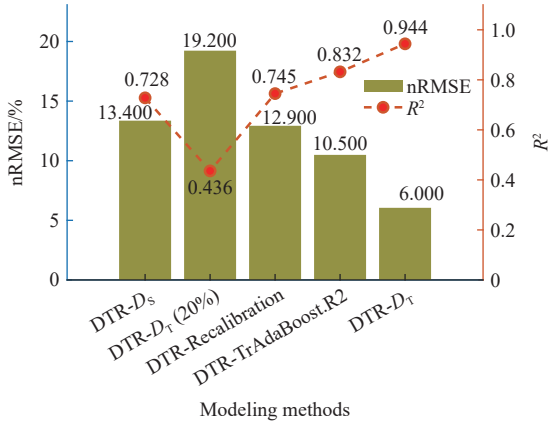


Figure 10 DFR prediction results of the samples in the urea

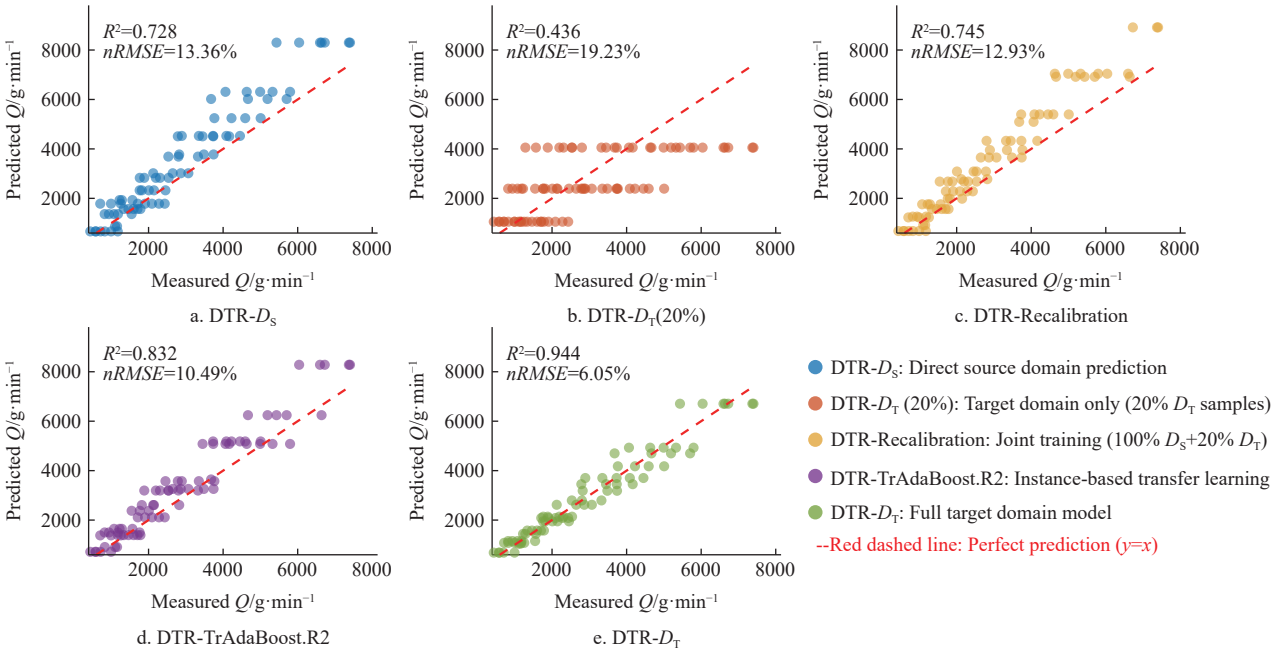


Figure 11 DFR model transfer performance comparison (Stanley to urea)

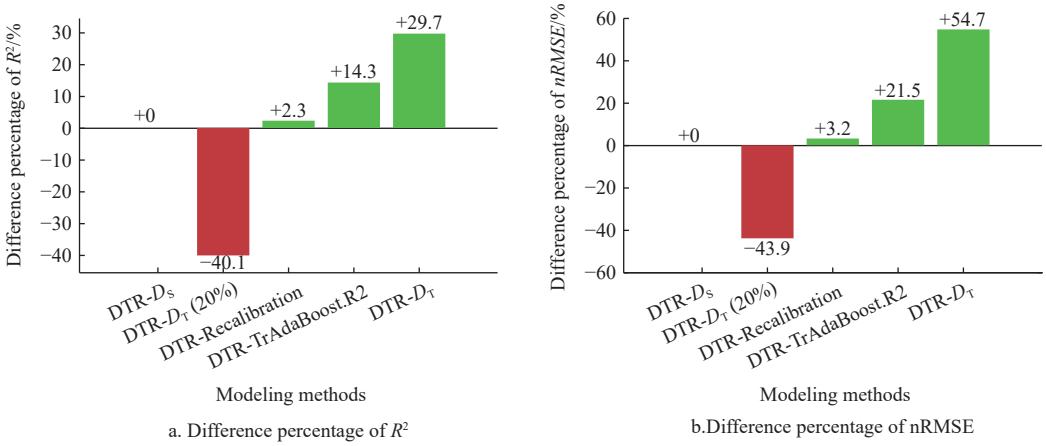
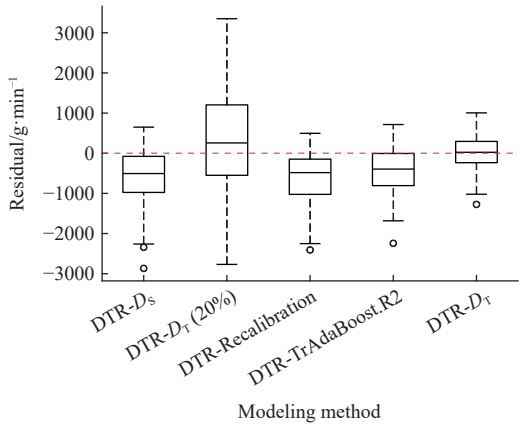


Figure 12 Difference percentage (DP) of R^2 and *nRMSE* compared with baseline model (DTR-D_S)

Apart from the DTR-D_T (20%) model, which exhibited a significant performance degradation, with a 40.1% decrease in R^2 and a 43.9% decrease in *nRMSE*, all other methods showed improvements in transferability. The DTR-Recalibration model achieved moderate enhancement, reflected in a 2.3% increase in R^2 and a 3.2% in *nRMSE*. Aside from the theoretical upper bound represented by DTR-D_T, trained by full target domain dataset, the DTR-TrAdaBoost.R2 model demonstrated the most substantial gains, with a 14.3% improvement in R^2 and a 24.5% reduction in *nRMSE*. The superiority of DTR-TrAdaBoost.R2 over other methods was consistent across 10 independent random data splits (Supplementary Table S1).

Following the same visualization approach as in Section 3.1.1, we generated boxplots of prediction residuals for the five methods (Figure 13). In contrast to the Stanley to Sakefu task where TrAdaBoost.R2 exhibited negative transfer, here TrAdaBoost.R2 shows the narrowest interquartile range (≈780 g/min) and the median closest to zero (≈−370 g/min) among all transfer methods. Its residual distribution is even slightly better than that of DTR-D_T (full target training) in terms of central tendency. This confirms that instance-based transfer learning is highly effective when source and target domains exhibit large and consistent static feature differences (the "globally different" pattern). DTR-D_T (20%) still performs worst, with an extremely wide IQR (>2000 g/min) and a median far from zero, highlighting the necessity of TL when target data are scarce.



Note: The red dashed line indicates zero residual. Boxes show interquartile range (IQR), central lines denote medians, and whiskers extend to 1.5×IQR. Outliers are shown as circles.

Figure 13 Boxplots of prediction residuals for the five methods (Stanley to urea)

3.2.2 Correlation analysis between TL effects and fertilizer characteristics (Stanley to urea)

(1) Static feature difference and correlation with TL effects

Following the methodology outlined in Section 3.1.2, the influence of static feature parameters on TL performance from Stanley to urea was analyzed. The percentage differences in static characteristics between the source domain (Stanley) and target domain (urea) are summarized in Table 9.

Table 9 Static feature analysis of Stanley and urea

Parameter	Stanley	Urea	Percentage difference/%
Sphericity ratio φ_s/%	90.00	94.84	5.1
Equivalent diameter d_e/mm	3.62	2.19	39.5
Repose angle α/°	21.6	12.83	40.6
Bulk density ρ_v/kg/m³	852.70	741.67	13.02

As shown in Table 9, the sphericity values of Stanley and urea differ by only 5.1%, suggesting a moderate similarity in particle shape. In contrast, a pronounced difference of 40.6% is observed in the repose angle, indicating substantially distinct flow behaviors between the two fertilizers. The equivalent diameter of urea is significantly smaller than that of Stanley. Given urea's high sphericity (94.84%) and low repose angle (12.83°), it can be inferred that urea exhibits superior flowability. This results in a considerably higher discharge rate for urea under identical operational parameters (*L* and *N*), thereby increasing the challenge of effective knowledge transfer. Additionally, the bulk density differs by 13.2%, influencing the mass per unit volume and introducing systematic bias in discharge rates under the same *L* and *N*.

Consistent with the results in Section 3.2.1, the substantial discrepancies in equivalent diameter and repose angle contribute to the relatively poor performance of direct transfer (DTR-D_S), which achieved an R^2 of only 0.728. The limited improvement from DTR-Recalibration (a 2.3% increase in R^2) further suggests that simple parameter adjustment is insufficient to overcome major static property differences. In comparison, the instance-based TL method (DTR-TrAdaBoost.R2) yielded a significant enhancement, with a 14.3% improvement in R^2, demonstrating its ability to identify and leverage source samples resembling the target domain. Ultimately, the full target-domain model (DTR-D_T) achieved optimal performance, confirming that adequate target data enable effective learning of urea-specific discharge behavior.

(2) Dynamic feature analysis and correlation with TL effects

To analyze the dynamic characteristics influencing the TL process from Stanley to urea and their effect on the performance of the instance-based TL method (DTR-TrAdaBoost.R2), high-weight samples (D_{HW}) were extracted from the source domain. The distribution of dynamic features, opening length (*L*), rotational speed (*N*), and operational intensity (*L*×*N*) for these samples is visualized on the *L*–*N* plane in Figure 14. Furthermore, a statistical analysis was conducted on the top 20% of high-weight samples, including the discharge ratio ($(Q_{urea}/Q_{stanley})$). The results are summarized in Table 10 and visualized in Figure 15.

As shown in Table 10, the high-weight samples are predominantly concentrated within medium ranges of opening length (*L*=28.8 mm), rotational speed (*N*=27.8 r/min), and operational intensity (781.2 mm·r/min). A similar trend is evident in the *L*–*N* heatmap presented in Figure 13. Additionally, the standard deviation of the discharge ratio for high-weight samples is 0.114, which is 20% lower than that of the full dataset (0.143). This indicates a more stable discharge relationship in the high-weight samples, contributing positively to the effectiveness of transfer learning.

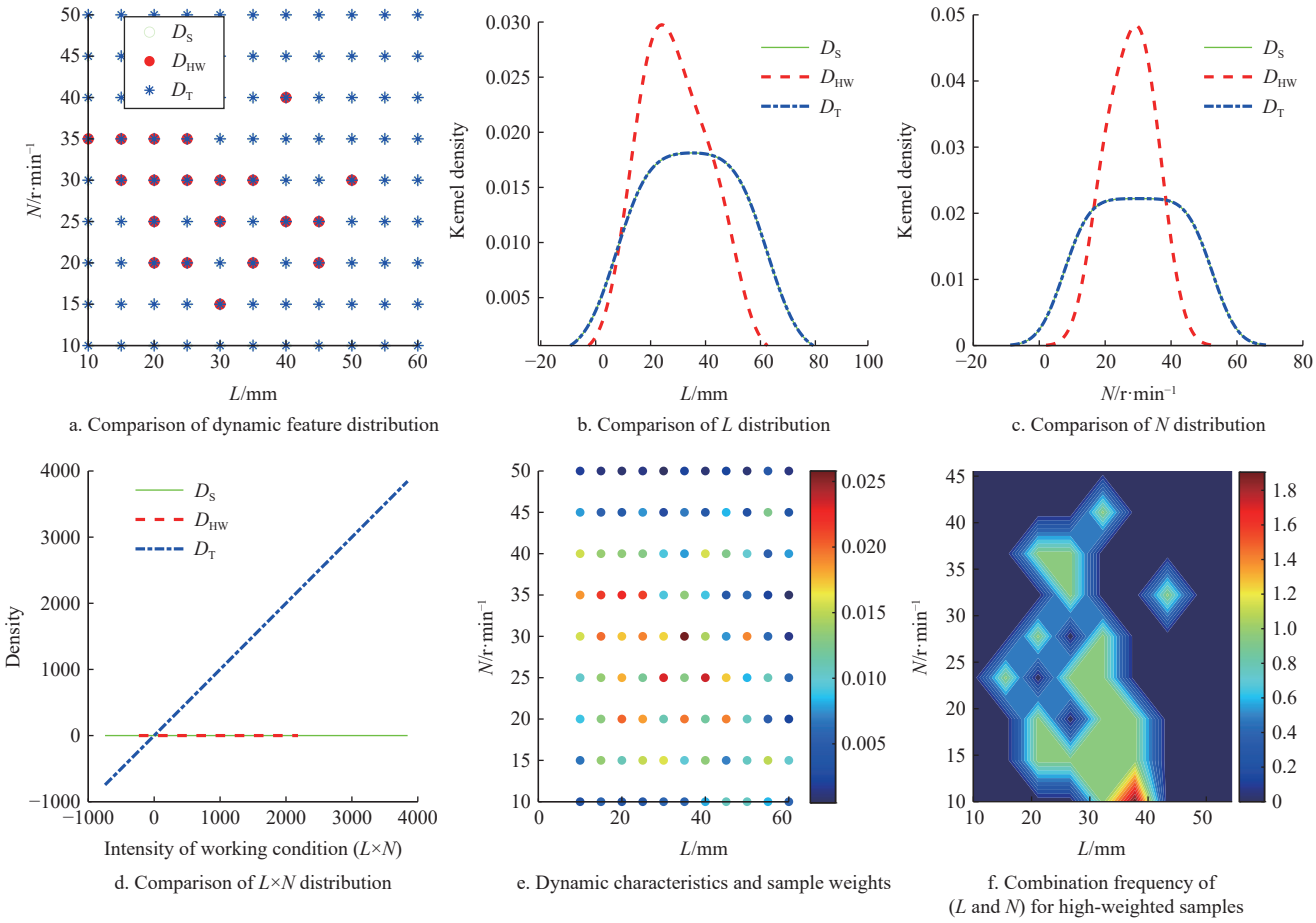
This spatial pattern is further corroborated by the *L*-*N* heatmap in Figure 15, which visualizes the weight distribution using a red-to-blue colormap (red: high concentration, blue: low/dispersed) with contour lines indicating distribution boundaries. The heatmap also highlights individual samples with weights exceeding 0.2, marking the most influential source domain instances, which are located at (*L*=25, *N*=35), (*L*=30, *N*=25), (*L*=35, *N*=30), and (*L*=40, *N*=25). Consistent with Table 10, the high-weight regions are predominantly concentrated around medium operating ranges, while extreme *L*-*N* combinations receive substantially lower weights.

3.3 Comprehensive analysis of DTR prediction model transferability

3.3.1 Comparison of transfer effects across fertilizer types and TL strategy

A comparison of transfer performance from Stanley to Sakefu and from Stanley to urea reveals two distinct patterns (detailed in

Sections 3.1 and 3.2). In the case of Stanley to Sakefu, joint training (DTR-Recalibration) achieves the highest accuracy, whereas instance-based TL (DTR-TrAdaBoost.R2) yields limited improvement (negative transfer), while in the case of Stanley to urea, the opposite holds: DTR-TrAdaBoost.R2 significantly outperforms joint training.



a. Comparison of dynamic feature distribution

b. Comparison of L distribution

c. Comparison of N distribution

d. Comparison of L×N distribution

e. Dynamic characteristics and sample weights

f. Combination frequency of (L and N) for high-weighted samples

Note: D_s represents the samples from Stanley, D_{HW} represents high-weighted samples, D_T represents samples from Sakefu.

Figure 14 Distribution of dynamic characteristics (L, N, and L×N) of high-weight samples on the L-N plane (Stanley to urea)

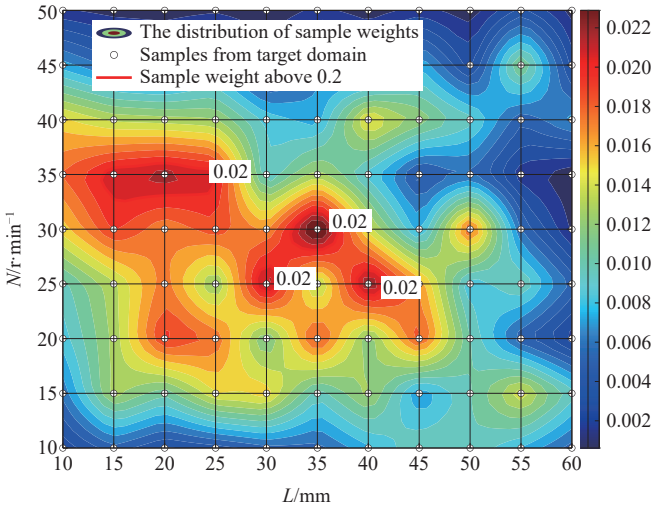


Figure 15 Distribution of sample weights on the L-N plane (Stanley to urea)

This discrepancy originates from the pattern of static feature differences between source and target fertilizers, not merely their magnitude. The two transfer scenarios represent two fundamentally different patterns: between Stanley and Sakefu, there is a "partially similar, partially different" pattern (small differences in equivalent diameter and sphericity, large differences in repose angle and bulk density), which leads to condition-dependent discharge behavior. In contrast, Stanley and urea exhibit a "globally different" pattern (large differences across most static features), which creates a systematic offset in discharge rate. The mechanisms underlying these patterns and their quantitative impact on TL effectiveness are analyzed in detail in Section 3.3.2.

Table 10 Statistical characteristics of the top 20% high-weight samples (Stanley to urea)

Parameter	High-weight samples (top 20%)			All samples		
	Distribution range	Mean value	Standard deviation	Distribution range	Mean value	Standard deviation
L/mm	10--50	28.8	-	10-60	35	-
N/r·min^{-1}	15-40	27.8	-	10-50	30	-
L×N/mm·r·min^{-1}	350-1600	781.2	-	100-3000	1050	-
Q_{urea}/Q_{stanly}	0.765-1.234	0.891	0.114	0.730-1.303	0.877	0.143

3.3.2 Quantitative correlation between static feature patterns and TL performance

The effectiveness of TL depends not only on the magnitude of static discrepancies but also on their pattern. To quantify this dependence, the static feature differences were correlated with the performance of different TL methods.

In the "partially similar, partially different" pattern (Stanley to Sakefu), the heterogeneous feature differences lead to condition-dependent discharge behavior: repose angle dominates at low speed,

while bulk density governs at high speed. Consequently, TrAdaBoost.R2's uniform instance weighting fails, causing negative transfer. In contrast, joint training (DTR-Recalibration) averages out these variations and achieves the best performance. In the "globally different" pattern (Stanley to urea), the consistent divergence across all operating conditions creates a systematic offset in discharge rate. Here, TrAdaBoost.R2 effectively corrects this offset via iterative reweighting, outperforming joint training.

The performance of direct transfer (DTR-D_S) is inversely correlated with the degree of static characteristic differences. Specifically, the equivalent diameter is the most influential factor for transfer efficacy. Smaller differences in equivalent diameter (e.g., Stanley vs. Sakefu) enable high direct transfer performance, while larger differences lead to poor direct transfer. The repose angle, which reflects flow characteristics, is the second most influential. When equivalent diameter is similar, its effect can be partially compensated, but when both differ substantially, transfer becomes challenging. Bulk density has a moderate influence, while sphericity plays a minor role.

These quantitative findings provide a practical guideline: joint training is preferred when static feature differences are heterogeneous across operating conditions (partial similarity), whereas instance-based TL is more suitable when differences are large and consistent across most features.

3.3.3 Correlation analysis of dynamic characteristics and TL effects

Statistical analysis of high-weight samples from the transfer learning processes (Stanley to Sakefu and Stanley to urea) indicates that fertilizer discharge under medium operating conditions (L and N) exhibits greater stability. The stability of the discharge ratio significantly influences the effectiveness of TL, with higher stability correlating with improved performance in instance-based transfer learning.

Notably, when the TrAdaBoost.R2 algorithm is applied for TL, the high-weight samples in both transfer tasks (Stanley to Sakefu and Stanley to urea) were predominantly concentrated under medium operating conditions. This pattern may be explained by the higher variability in fertilizer discharge under extreme values of L and N, which introduces greater uncertainty in discharge rate prediction. As a result, the algorithm assigns lower weights to samples obtained under non-medium operating conditions.

Notable differences were observed in the stability of the discharge ratio between the two transfer scenarios. In the case of Stanley to Sakefu, high-weight samples showed less stability in discharge behavior compared to the full dataset, which limited TL efficacy. Conversely, during transfer from Stanley to urea, high-weight samples exhibited more stable discharge relationships than the full dataset, contributing positively to TL performance.

4 Conclusions

This study addresses the challenge of cross-fertilizer discharge rate prediction in bivariate fertilization systems using transfer learning. The key findings are as follows:

(1) Transfer learning enables effective cross-fertilizer knowledge transfer, achieving over 85% of the performance obtained with full target data using only 20% of target domain samples. This substantially reduces calibration effort while maintaining prediction accuracy.

(2) Static fertilizer properties critically influence transfer effectiveness. The choice of TL method should be guided by the specific static feature pattern: joint training is preferred when differences are heterogeneous across operating conditions (e.g., Stanley to Sakefu), whereas instance-based TL is more suitable when differences are large and consistent (e.g., Stanley to urea).

(3) Discharge stability under medium operating conditions (L and N) enhances TL performance, as stable discharge ratios lead to better transfer outcomes. Practical calibration should prioritize collecting target data in these medium ranges.

These findings validate the effectiveness of transfer learning for cross-fertilizer DFR prediction and provide practical guidelines for selecting appropriate TL strategies based on fertilizer properties and operating conditions. The main limitation is that the study uses only three fertilizer types; future work should validate these strategies on a more diverse set of fertilizers and applicator configurations.

Acknowledgements

The authors acknowledge that this work was financially supported by Anhui Polytechnic University Horizontal Research Project (Grant No. HX-2025-11-083), Research Fund for Talent Introduction of Anhui Polytechnic University (Grant No. 2023YQQ005), Open Fund of Anhui Provincial Key Laboratory of Advanced CNC and Servo Drive Technology (Grant No. XJSK202507), Wuhu Science and Technology Plan Project (Grant No. 2023ly05), and Science Research Project of Higher Education in Anhui Province (Grant No. 2023AH050921).

[References]

[1] Wang X W, Jiang R, Zhou Z Y, Song C C, Luo X W, Bao R F, et al. Discharge rate consistency of each channel for UAV-based pneumatic granular fertilizer spreader. Int J Agric & Biol Eng, 2023; 16(4): 20–28.

[2] Toselli M, Baldi E, Ferro F, Rossi S, Cillis D. Smart farming tool for monitoring nutrients in soil and plants for precise fertilization. Horticulturae. 2023; 9(9): 1–14. doi: 10.3390/horticulturae9091011

[3] Pawase P P, Nalawade S M, Bhanage G B, Walunj A A, Kadam P B, Durgude A G, et al. Variable rate fertilizer application technology for nutrient management: A review. Int J Agric & Biol Eng, 2023; 16(4): 11–19.

[4] Beneduzzi H M, de Souza E G, Moreira W K O, Sobjak R, Bazzi CL, Rodrigues M. Fertilizer recommendation methods for precision agriculture-a systematic literature study. Engenharia Agricola. 2022; 42(1). doi: 10.1590/1809-4430-eng.agric.v42n1e20210185/2022

[5] Alameen A A, Al-Gaadi K A, Tola E K. Development and performance evaluation of a control system for variable rate granular fertilizer application. Computers and Electronics in Agriculture, 2019; 160: 31–39.

[6] Zhang J Q, Liu G. Effects of control sequence optimisation on the performance of bivariate fertiliser applicator. Computers and Electronics in Agriculture, 2022; 192(17): 106594.

[7] Zhang J Q, Zhuang Q B, Liu G. Analysis of the multi-objective control sequence optimization problem in bivariate fertilizer applicators. Symmetry, 2025; 17(6): 926.

[8] Su N, Xu T S, Song L T, Wang R J, Wei Y Y. Variable rate fertilization system with adjustable active feed-roll length. Int J Agric & Biol Eng, 2015; 8(4): 19–26.

[9] Zhang J, Liu G, Luo C, Hu H, Huang J. MOEA/D-DE based bivariate control sequence optimization of a variable- rate fertilizer applicator. Computers and Electronics in Agriculture, 2019; 167: 105063.

[10] Shi Y Y, Xin Y P, Wang X C, Zheng E L, Shen C, Zhang Z. Research on fertilizer application strategy for rice-wheat dual-variable precision fertilizer applicator based on MLP. Transactions of the CSAE, 2025; 41(10): 51–60. (in Chinese) doi: 10.11975/j.issn.1002-6819.202408134

[11] Hossen M I, Awrangjeb M, Pan S, Mamun A A. Transfer learning in agriculture: A review. Artificial Intelligence Review, 2025; 58(4). doi: 10.1007/s10462-024-11081-x

[12] Narmadha R P, Sengottaiyan N, Kavitha R J. Deep transfer learning based rice plant disease detection model. Intelligent Automation and Soft Computing, 2022; 31(2): 1257–1271.

[13] Panigrahi S, Nanda A, Swarnkar T. A survey on transfer learning. Smart Innovation, Systems and Technologies, 2021; 194: 781–789.

[14] Wang Y C, Zhang S Q, Dai B S, Yang S S, Song H C. Fine-grained weed recognition using Swin Transformer and two-stage transfer learning. Frontiers in Plant Science, 2023; 14: 1–13.

[15] Krishnamoorthy N, Narasimha Prasad L V, Pavan Kumar C S, Subedi B, Abraha H B, Sathishkumar V E. Rice leaf diseases prediction using deep neural networks with transfer learning. Environmental Research, 2021; 198: 111275.

[16] Yin X, Li W H, Li Z, Yi L L. Recognition of grape leaf diseases using MobileNetV3 and deep transfer learning. Int J Agric & Biol Eng, 2022; 15(3): 184–194. Int J Agric & Biol Eng, 2022; 15(3): 184–194. doi: 10.25165/j.ijabe.20221503.7062

[17] Ma Z, Wang Y, Zhang T S, Wang H G, Jia Y J, Gao R, et al. Maize leaf disease identification using deep transfer convolutional neural networks. Int J Agric & Biol Eng, 2022; 15(5): 187–195.

[18] Ma Y C, Zhang Z, Yang H L, Yang Z W. An adaptive adversarial domain adaptation approach for corn yield prediction. Computers and Electronics in Agriculture, 2021; 187: 106314.

[19] Lim H-J, Kim Y-J, Kim S-M, Kim S-J. Convolutional Neural Network (CNN)-based transfer learning framework for cherry tomato production. Int J Agric & Biol Eng, 2025; 18(5): 90–101.

[20] Chen Y, Li L, Whiting M, Chen F, Sun Z C, Song K S, et al. Convolutional neural network model for soil moisture prediction and its transferability analysis based on laboratory Vis-NIR spectral data. International Journal of Applied Earth Observation and Geoinformation, 2021; 104: 102550.

[21] Yang J, Li J T, Hu J, Yang W J, Zhang X L, Xu J F, et al. An interpretable deep learning approach for calibration transfer among multiple near-infrared instruments. Computers and Electronics in Agriculture, 2022; 192: 106584.

[22] Pan S J, Yang Q. A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineeringngineering, 2009; 22: 1345–1359.

[23] Wu S T, Zhu Y S, Zhang Q. A new robust visual tracking algorithm based on transfer adaptive boosting. Mathematical Methods in the Applied Sciences, 2012; 35(17). doi: 10.1002/mma.2643

[24] Huang Y R, Chen W Q, Tan W, Deng Y J, Yang C H, Zhu X G, et al. Transfer learning for enhancing the generality of leaf spectroscopic models in estimating crop foliar nutrients across growth stages. International Journal of Applied Earth Observation and Geoinformation, 2025; 139. doi: 10.1016/j.jag.2025.104481

[25] Bu H R, Yu S Y, Dong W C, Wang Y Q, Zhang L X, Xia Y Q. Calibration and testing of discrete element simulation parameters for urea particles. Processes, 2022; 10(3). doi: 10.3390/pr10030511

[26] Miller T, Mikiciuk G, Kisiel A, Mikiciuk M, Paliwoda D, Sas-Paszt L, et al. Machine learning approaches for forecasting the best microbial strains to alleviate drought impact in agriculture. Agriculture (Switzerland), 2023; 13(8): 1–16.

[27] Torres-Sanchez R, Navarro-Hellin H, Guillamon-Frutos A, San-Segundo R, Ruiz-Abellón M C, Domingo-Miguel R. A decision support system for irrigation management: Analysis and implementation of different learning techniques. Water (Switzerland). 2020;12(2): 548. doi:10.3390/w12020548

[28] Biddapp.C B, Srikanth V. Crop yield prediction on agriculture using machine learning. International Journal of Research Publication and Reviews, 2024; 5(3): 165–167.

[29] Chen L, Cui Y P, Wang M, Niu Y C, Xu A G, Liu K Y, et al. A study of relationships between cultivated land fertility and corn head smut using a decision tree model. Journal of Agricultural Resources and Environment, 2021; 38: 928–36.

[30] Paudel D, de Wit A, Boogaard H, Marcos D, Osinga S, Athanasiadis I N. Interpretability of deep learning models for crop yield forecasting. Computers and Electronics in Agriculture, 2023; 206: 107663.

[31] Tang D J, Yang X H, Wang X S. Improving the transferability of the crash prediction model using the TrAdaBoost. R2 algorithm. Accident Analysis and Prevention, 2020; 141: 105551.

[32] Dai W Y, Yang Q, Xue G R, Yu Y. Boosting for transfer learning. In: Proceedings of the 24th international conference on Machine learning. Corvalis, Oregon, USA; 2007; pp.193–200. doi: 10.1145/1273496.1273521

[33] Skobalski J, Sagan V, Alifu H, Al Akkad O, Lopes F A, Grignola F. Bridging the gap between crop breeding and GeoAI: Soybean yield prediction from multispectral UAV images with transfer learning. ISPRS Journal of Photogrammetry and Remote Sensing, 2024; 210: 260–281.

Supplementary Tables

Table S1 Sensitivity tests of T on both transfer tasks using 10 independent random data splits

Task iteration T	Stanley to Sakefu		Stanley to urea	
	R^2	*nRMSE*/%	R^2	*nRMSE*/%
T = 1	0.891±0.0188	7.73 ± 0.69	0.799±0.0436	10.62 ± 1.31
T = 5	0.895±0.0314	**7.48 ± 1.38**	0.781±0.0722	10.86 ± 1.90
T = 10	**0.896±0.0407**	7.57 ± 1.58	0.818±0.0321	10.23 ± 0.75
T = 20	0.895±0.0226	7.57 ± 0.91	**0.825±0.0298**	**9.90 ± 0.93**
T = 50	0.891±0.0184	7.74 ± 0.98	0.820±0.0212	10.09 ± 0.75

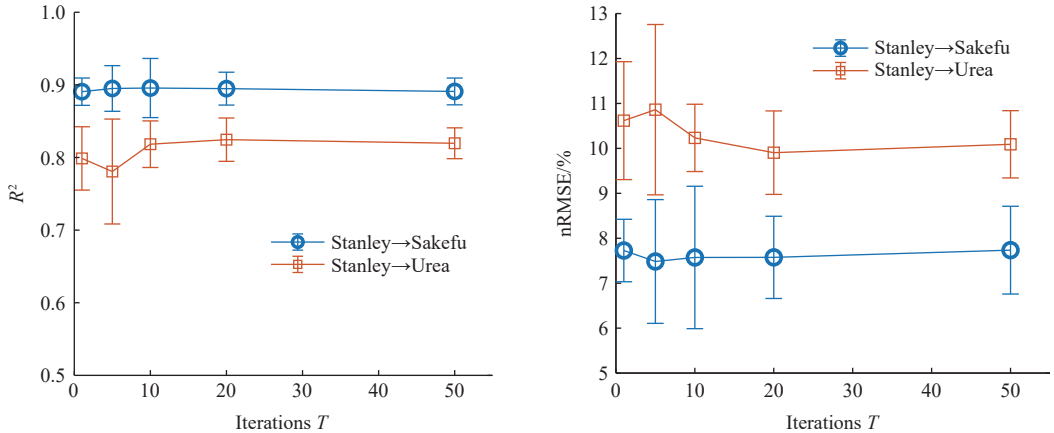
Table S2 Mean ± standard deviation of R^2 and *nRMSE* for five methods over 10 independent random data splits

Task method	Stanley to Sakefu		Stanley to urea	
	R^2	*nRMSE*/%	R^2	*nRMSE*/%
DTR-D_S	0.9010 ± 0.0070	7.36 ± 0.30	0.7082 ± 0.0330	12.87 ± 0.89
DTR-D_T(20%)	0.5039 ± 0.1220	**16.38 ± 2.13**	0.3398 ± 0.1962	19.19 ± 3.05
DTR-Recalibration	**0.9195 ± 0.0115**	6.63 ± 0.49	0.7579 ± 0.0285	11.71 ± 0.70
DTR-TrAdaBoost.R2	0.8959 ± 0.0245	7.50 ± 1.03	**0.8063 ± 0.0358**	**10.45 ± 1.06**
DTR-D_T	0.9268 ± 0.0066	6.33 ± 0.36	0.9409 ± 0.0040	5.80 ± 0.28

Note: Values are rounded to four decimal places for R^2 and two decimal places for *nRMSE*.

Table S3 Full pairwise *p*-value matrix for R^2 based on 10 replicates

	Stanley to Sakefu				
	DTR-D_S	DTR-D_T(20%)	DTR-Recalibration	DTR-TrAdaBoost.R2	DTR-D_T
DTR-D_S	0	3.0391e-06	2.6417e-04	4.9935e-01	3.3747e-06
DTR-D_T(20%)	3.0391e-06	0	1.4672e-06	6.1875e-06	1.5861e-06
DTR-Recalibration	**2.6417e-04**	1.4672e-06	0	**3.0462e-02**	5.7489e-02
DTR-TrAdaBoost.R2	4.9935e-01	6.1875e-06	3.0462e-02	0	3.8530e-03
DTR-D_T	3.3747e-06	1.5861e-06	5.7489e-02	3.8530e-03	0
	Stanley to urea				
	DTR-D_S	DTR-D_T(20%)	DTR-Recalibration	DTR-TrAdaBoost.R2	DTR-D_T
DTR-D_S	0	1.5994e-04	7.9825e-04	8.3464e-06	1.3193e-09
DTR-D_T(20%)	1.5994e-04	0	4.7346e-05	2.2274e-05	4.3346e-06
DTR-Recalibration	**7.9825e-04**	4.7346e-05	0	3.3361e-03	4.6851e-09
DTR-TrAdaBoost.R2	**8.3464e-06**	2.2274e-05	**3.3361e-03**	0	**6.1215e-07**
DTR-D_T	1.3193e-09	4.3346e-06	4.6851e-09	6.1215e-07	0



a. Mean±std for R^2 b. Mean±std for nRMSE

Figure S1 Sensitivity analysis of iteration number (mean±std)

Table S4 Algorithm S1: TrAdaBoost.R2 for cross-fertilizer DFR prediction

Input:

Source data (X_s, y_s);

target training data (X_t, y_t);

test data X_{te};

max iterations T;

ensemble size K;

similarity parameters $\alpha, \beta, \sigma_{dyn}, \sigma_{static}$.

Output:

Predictions $\hat{y}_{te}$.

Step 1 - Initialization

1.1 Compute initial source weights $w_i^{(1)}$ based on similarity: $w_i^{(1)} \propto \alpha \cdot K_{dyn}(x_s^i, X_t) + \beta \cdot K_{static}(x_s^i, X_t)$, where K is Gaussian kernel with width σ.

1.2 Normalize $w_i^{(1)}$ to sum to 1. $w_i^{(1)} \leftarrow w_i^{(1)} / \sum_i w_i^{(1)}$

1.3 Set target training weights $v_j^{(1)} = 1/n_t$, where $(n_t = |X_t|)$

1.4 Set ensemble weight list $\beta_{list} = []$.

Step 2 – Boost iteration

For $i = 1, \ldots, T$, do

2.1 Train weighted DTR h_t on $(X_s \cup X_t)$ with weights $[w^{(t)}; v^{(t)}]$.

2.2 Compute absolute errors on target training set: $e_i = \left| h_t\left(x_t^j - y_t^j\right) \right|$, for $j = 1 \ldots n_t$;

2.3 Normalize errors: $\tilde{e}_j = e_j / \max(e)$.

2.4 Compute weighted average error: $E_t = \sum_{j=1}^{n_t} v_j^{(t)} \tilde{e}_j$.

2.5 If $E_t \geq 0.5$, break the loop (stop to avoid negative transfer).

2.6 Set $\beta_t = E_t/(1 - E_t)$, append β_t to β_{list}

2.7 Update target weights: $v_j^{(t+1)} = v_j^{(t)} \cdot \beta_t^{1-\tilde{e}_j}$

2.8 Compute source absolute errors: $e'_i = \left| h_t\left(x_s^i - y_s^i\right) \right|$, for $i = 1 \ldots n_s$; Normalize: $\tilde{e}'_i = e'_i / \max(e')$

2.9 Update source weights: $w_i^{(t+1)} = w_i^{(t)} \cdot \beta_t^{\tilde{e}'_i}$

2.10 Normalize $w_i^{(t+1)}$ and $v_i^{(t+1)}$ to sum 1.

End for

Step 3 -Ensemble prediction

3.1 Effective iterations: T_{eff} = number of completed iterations $(\leq T)$.

3.2 Select models: Keep the last K models: (indices $t = T_{eff} - K + 1, \ldots, T_{eff}$)

3.3 For each selected model h_k, compute its ensemble weight: $\omega_k = 1/\beta_k$, where β_k is corresponding value.

3.4 Normalize ω_k to sum to 1.

3.5 For each test sample x $\in X_{te}$, compute: $\hat{y}(x) = \sum_k \omega_k \cdot h_k(x)$.

3.6 $\hat{y}_{te} = \{\hat{y}(x) \mid x \in X_{te}\}$.