Wasserstein Flow Networks: Forward-Pass Optimal Transport for Cross-Domain Agricultural Anomaly Segmentation

Supplementary Material

Appendix A. Complete Mathematical Formulation

A.1 Feature Measure Representation

Let the feature space be a compact subset

$$\mathcal{Z} \subset \mathbb{R}^d$$

with intrinsic dimension d. The agronomic embedding process maps each superpixel descriptor to this feature space, producing a finite set of support points

$$\{z_i\}_{i=1}^n, \quad z_i \in \mathcal{Z}.$$

Each support point is associated with a non-negative weight α_i such that

$$\alpha_i \geq 0, \qquad \sum_{i=1}^n \alpha_i = 1.$$

The resulting empirical representation of an image is therefore a probability measure

$$\mu = \sum_{i=1}^n \alpha_i \delta_{z_i},$$

where δ_{z_i} denotes the Dirac measure concentrated at location z_i.

This representation converts the spatial segmentation problem into a measure alignment problem over a finite set of agronomic feature embeddings. The number of support points n varies between images because the superpixel decomposition adapts to image complexity. Consequently, the empirical measure representation allows WFN to handle images of varying structural complexity without imposing a fixed feature resolution.

A.2 Prototype Measure Representation

To approximate the distribution of agronomic patterns observed during training, WFN maintains a prototype measure

$$\nu = \sum_{j=1}^M \beta_j \delta_{p_j},$$

where

$$p_j \in \mathcal{Z}$$

denote prototype embeddings and

$$\beta_j = \frac{1}{M}$$

define uniform weights.

The prototype support points are obtained through clustering over source-domain embeddings. Formally, let

$$\mathcal{Z}_s = \{z_k^s\}_{k=1}^N$$

be the set of embeddings obtained from the source dataset. The prototypes are defined as

$$\{p_1, \ldots, p_M\} = \mathrm{KMeans}(\mathcal{Z}_s, M).$$

Thus the prototype measure acts as a compact approximation of the source feature distribution.

A.3 Transport Map Definition

The goal of WFN is to construct a mapping

$$T : \mathcal{Z} \to \mathcal{Z}$$

that aligns the empirical image measure with the prototype distribution.

Formally, the transported measure is defined by the push-forward operator

$$T_{\#}\mu(B) = \mu(T^{-1}(B))$$

for any measurable subset $B \subset \mathcal{Z}$. The alignment objective is therefore

$$T^* = \arg\min_T W(\mu, \nu),$$

where $W(\cdot, \cdot)$ denotes the Wasserstein distance between probability measures.

A.4 Discrete Optimal Transport Formulation

For discrete measures, optimal transport reduces to solving a coupling problem. Let

$$\gamma \in \mathbb{R}^{n \times M}$$

denote the transport matrix satisfying

$$\sum_{j=1}^M \gamma_{ij} = \alpha_i, \quad \sum_{i=1}^n \gamma_{ij} = \beta_j.$$

The optimal transport problem becomes

$$\min_\gamma \sum_{i=1}^n \sum_{j=1}^M \gamma_{ij} c(z_i, p_j),$$

where $c(\cdot,\cdot)$ denotes the transport cost. WFN uses the squared Euclidean distance

$$c(z_i, p_j) = \|z_i - p_j\|_2^2.$$

The transported embedding for support point z_i is

$$\tilde{z}_i = \sum_{j=1}^{M} \frac{\gamma_{ij}}{\alpha_i} p_j.$$

Thus each embedding is mapped to a weighted barycenter of prototypes.

A.5 Projection-Based Transport Approximation

Directly solving the discrete optimal transport problem has computational complexity $O(nM)$ per iteration and becomes expensive for large support sets. WFN therefore employs projection-based transport.

Let

$$e \in \mathbb{S}^{d-1}$$

denote a unit vector on the hypersphere. A projection operator is defined as

$$\pi_e(z) = e^\top z.$$

Applying the projection transforms the high-dimensional measures into one-dimensional measures

$$\mu_e = (\pi_e)_{\#}\mu, \quad \nu_e = (\pi_e)_{\#}\nu.$$

The one-dimensional Wasserstein distance between projected measures is

$$W_1(\mu_e, \nu_e) = \int_0^1 \left| F_{\mu_e}^{-1}(t) - F_{\nu_e}^{-1}(t) \right| \, dt,$$

where F^{-1} denotes the quantile function.

A.6 Max-Sliced Wasserstein Alignment

Instead of averaging distances over random projections, WFN identifies the projection direction maximizing distribution discrepancy. Let

$$\mathcal{E} = \{e_k\}_{k=1}^{K}$$

be a set of projection directions. The Max-Sliced Wasserstein distance is defined as

$$\text{MSW}(\mu, \nu) = \max_{e \in \mathcal{E}} W_1(\mu_e, \nu_e).$$

The transport direction producing the maximal discrepancy is used to guide alignment. The transported support point becomes

$$\tilde{z}_i = z_i + \lambda \cdot e^*(q_{\nu_e}(r_i) - q_{\mu_e}(r_i)),$$

where e^* is the maximizing direction, $q(\cdot)$ denotes the quantile function, and r_i denotes the rank of projection value.

A.7 Segmentation Prediction Operator

After transport alignment, the resulting embeddings

$$\{\tilde{z}_i\}$$

are mapped to class predictions using a linear classifier

$$g(\tilde{z}_i) = W\tilde{z}_i + b.$$

The classifier outputs class probabilities through the softmax operator

$$P(y = k|\tilde{z}_i) = \frac{\exp(g_k(\tilde{z}_i))}{\sum_{l=1}^{K} \exp(g_l(\tilde{z}_i))}.$$

Predictions are then propagated to all pixels belonging to the corresponding superpixel.

A.8 End-to-End Objective

During training, WFN optimizes the following objective:

$$\mathcal{L} = \mathcal{L}_{\text{seg}} + \lambda_{\text{iso}}\mathcal{L}_{\text{iso}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}}.$$

where $\mathcal{L}_{\text{seg}}$ denotes the segmentation loss, $\mathcal{L}_{\text{iso}}$ enforces manifold-preserving embedding, and $\mathcal{L}_{\text{reg}}$ regularizes transport stability.

A.9 Complete Forward Pass

The complete WFN computation pipeline can therefore be summarized as

$$x \to \{v_i\} \to \{z_i\} \to \mu \to T(\mu) \to \{\tilde{z}_i\} \to g(\tilde{z}_i) \to \hat{y}.$$

This sequence formalizes the transformation from aerial imagery to anomaly segmentation through measure construction, optimal transport alignment, and feature decoding.

Appendix B. Theoretical Analysis of Transport-Based Domain Alignment

B.1 Domain Generalization Framework

Consider two domains:

- **Source domain:**

$$\mathcal{D}_s = \{(x_i^s, y_i^s)\}_{i=1}^{N_s}$$

- **Target domain:**

$$\mathcal{D}_t = \{x_j^t\}_{j=1}^{N_t}$$

Let $P_s(X, Y)$ and $P_t(X, Y)$ denote their joint distributions. We assume the covariate shift condition

$$P_s(Y|X) = P_t(Y|X)$$

while

$$P_s(X) \neq P_t(X).$$

The segmentation model is

$$f_\theta : \mathcal{X} \to \mathcal{Y}$$

where $\mathcal{Y} = \{1, \ldots, K\}^{H \times W}$ represents the set of pixel-wise anomaly labels.

B.2 Target Error Decomposition

The target segmentation error is defined as

$$\epsilon_t(f_\theta) = \mathbb{E}_{(x,y)\sim P_t}[\ell(f_\theta(x), y)]$$

where $\ell(\cdot)$ denotes the segmentation loss.

Following standard domain adaptation theory, the target error can be bounded as

$$\epsilon_t(f_\theta) \leq \epsilon_s(f_\theta) + d_{\mathcal{H}}(P_s, P_t) + \lambda$$

where:

- $\epsilon_s(f_\theta)$ is the source-domain error
- $d_{\mathcal{H}}(P_s, P_t)$ measures domain divergence
- λ is the optimal joint risk

$$\lambda = \min_{f\in\mathcal{H}} \big(\epsilon_s(f) + \epsilon_t(f)\big)$$

This decomposition shows that cross-domain generalization is primarily controlled by the divergence term.

B.3 Representation-Space Alignment

Instead of aligning raw pixel distributions, WFN performs alignment in the embedding space of agronomic descriptors. Let

$$z = g_\theta(v)$$

denote the embedding produced by the AIMN network. The induced feature distributions are

$$\mu_s = g_\theta(P_s(X))$$

$$\mu_t = g_\theta(P_t(X))$$

Domain divergence therefore becomes

$$d_{\mathcal{H}}(\mu_s, \mu_t).$$

The key objective of WFN is to minimize this divergence through optimal transport.

B.4 Wasserstein Distance as Domain Divergence

Let μ and ν denote two probability measures in the embedding space. The Wasserstein-2 distance is

$$W_2(\mu,\nu) = \left(\inf_{\gamma\in\Pi(\mu,\nu)} \int \|x-y\|_2^2 \, d\gamma(x,y)\right)^{1/2}$$

where $\Pi(\mu,\nu)$ is the set of valid couplings.

The Wasserstein distance has two properties that are particularly important for domain alignment:

Property 1 — Metric structure:

$$W(\mu,\nu) = 0 \iff \mu = \nu$$

Property 2 — Sensitivity to distribution geometry: Unlike divergence measures such as KL-divergence, Wasserstein distance remains well-defined even when the two distributions have disjoint supports. This property is critical in agricultural imagery because anomalies often occupy rare and spatially isolated regions, causing feature distributions to be highly multimodal.

B.5 Transport Map Alignment

Let

$$T : \mathbb{R}^d \to \mathbb{R}^d$$

denote the optimal transport map satisfying

$$T_\# \mu_t = \mu_s$$

where $T_\#$ denotes the pushforward operator.

After applying the transport map, the aligned distribution becomes

$$\tilde{\mu}_t = T_\# \mu_t.$$

Because

$$W(\tilde{\mu}_t, \mu_s) = 0,$$

the feature distributions become aligned.

B.6 Target Error After Transport

Let

$$\tilde{x} = T(x)$$

denote the transported representation. The segmentation model is then applied as

$$f_\theta(\tilde{x}).$$

The target error becomes

$$\epsilon_t^{OT} = \mathbb{E}_{x\sim P_t}[\ell(f_\theta(T(x)), y)].$$

Substituting into the error bound gives

$$\epsilon_t^{OT} \leq \epsilon_s(f_\theta) + d_{\mathcal{H}}(\tilde{\mu}_t, \mu_s) + \lambda.$$

Since

$$d_{\mathcal{H}}(\tilde{\mu}_t, \mu_s) \approx 0,$$

the bound simplifies to

$$\epsilon_t^{OT} \lesssim \epsilon_s(f_\theta) + \lambda.$$

Thus, optimal transport alignment effectively removes the domain divergence term from the generalization bound.

B.7 Max-Sliced Wasserstein Approximation

Direct computation of high-dimensional optimal transport is expensive. WFN therefore employs the Max-Sliced Wasserstein approximation. Let

$$e \in \mathbb{S}^{d-1}$$

denote a projection direction. The projected distributions are

$$(\text{proj}_e)_\# \mu.$$

The sliced Wasserstein distance is

$$SW(\mu,\nu) = \mathbb{E}_{e\sim U(\mathbb{S}^{d-1})} W_1((\text{proj}_e)_\#\mu, (\text{proj}_e)_\#\nu).$$

WFN instead uses

$$MSW(\mu,\nu) = \max_{e\in\mathbb{S}^{d-1}} W_1((\text{proj}_e)_\#\mu, (\text{proj}_e)_\#\nu)$$

which selects the most discriminative projection direction.

B.8 Approximation Bound

Let MSW_K denote the approximation using K projection directions. Then

$$|W(\mu,\nu) - MSW_K(\mu,\nu)| \leq \mathcal{O}(1/\sqrt{K}).$$

Therefore increasing the number of projections improves the approximation accuracy. This bound explains why the MSWT layer used in WFN provides an accurate approximation of full optimal transport while remaining computationally tractable.

B.9 Interpretation for Agricultural Vision

The theoretical results above imply three important conclusions:

1. **Transport reduces domain divergence:** By aligning empirical feature measures, optimal transport directly minimizes the divergence term in the domain adaptation bound.
2. **Spectral feature manifolds improve transport stability:** Because agronomic spectral indices encode physically meaningful vegetation properties, the embedding manifold is smoother and easier to align across domains.
3. **Forward-pass transport is sufficient:** Since the transport map operates directly on feature representations, no gradient updates are required at inference time, enabling efficient deployment in real-time agricultural monitoring systems.

Appendix C. Implementation Details

C.1 Model Architecture Configuration

The Wasserstein Flow Network consists of four computational stages executed sequentially during inference:

1. Agronomic-Informed Measure Network (AIMN)
2. Deep Isometric Embedding Network
3. Max-Sliced Wasserstein Transport Layer
4. Transported Feature Decoder

Each module is implemented as described below.

C.1.1 Agronomic-Informed Measure Network (AIMN)

The AIMN module performs spectral index computation, adaptive superpixel segmentation, and descriptor extraction.

Spectral Index Computation

The following vegetation indices are computed for every pixel in the input image:

$$\text{NDVI} = \frac{NIR - R}{NIR + R + \epsilon}$$

$$\text{ExG} = 2G - R - B$$

$$\text{GNDVI} = \frac{NIR - G}{NIR + G + \epsilon}$$

All channels are normalized to the range $[0, 1]$ before index computation. The constant $\epsilon = 10^{-6}$ is used to avoid division instability.

Superpixel Generation

Superpixels are generated using the SLIC (Simple Linear Iterative Clustering) algorithm.

Table 10. SLIC Implementation Parameters

Parameter	Value
Superpixel compactness	10
Initial grid size	adaptive
Maximum iterations	10
Distance metric	Euclidean in spectral index space

The algorithm operates on the stacked spectral index representation:

$$I_{\text{index}} = [\text{NDVI}, \text{ExG}, \text{GNDVI}]$$

instead of raw RGB channels.

Adaptive Superpixel Count

The number of superpixels n is determined using entropy of the NDVI distribution:

$$n = 200 + 160 \cdot H_{\text{NDVI}}$$

where

$$H_{\text{NDVI}} = -\sum_{k=1}^{256} p_k \log p_k$$

and p_k denotes the normalized NDVI histogram. This adaptive allocation ensures that images with complex anomaly structure receive finer spatial segmentation.

Superpixel Descriptor Extraction

For each superpixel S_i, the feature descriptor $v_i \in \mathbb{R}^8$ is computed as

$$v_i = [\mu_{NDVI}, \sigma_{NDVI}, \mu_{ExG}, \sigma_{ExG}, \mu_{GNDVI}, \sigma_{GNDVI}, a_i, e_i]$$

where:

Table 11. Superpixel Descriptor Features

Feature	Description
μ_{NDVI}	mean NDVI value
σ_{NDVI}	NDVI variance
μ_{ExG}	mean ExG
σ_{ExG}	ExG variance
μ_{GNDVI}	mean GNDVI
σ_{GNDVI}	GNDVI variance
a_i	normalized superpixel area
e_i	elongation ratio

Area normalization is defined as

$$a_i = \frac{|S_i|}{HW}$$

Elongation ratio is computed from the eigenvalues of the covariance matrix of pixel coordinates.

C.2 Deep Isometric Embedding Network

The embedding network g_θ maps superpixel descriptors to the latent embedding space.

Network Structure

Table 12. Embedding Network Architecture

Layer	Dimension	Activation
FC1	$8 \to 64$	ReLU
FC2	$64 \to 32$	ReLU
FC3	$32 \to 8$	Linear

The output embedding dimension is therefore

$$d = 8$$

which matches the intrinsic dimensionality estimate of the superpixel feature manifold.

Embedding Normalization

Each embedding is L2-normalized:

$$z_i \leftarrow \frac{z_i}{\|z_i\|_2}$$

This ensures all support points lie on the unit hypersphere $\mathbb{S}^{d-1}$, stabilizing the transport computation.

Isometric Loss Implementation

The isometric loss described in the main paper is implemented using sampled descriptor pairs within each mini-batch. Let $d_G(v_i, v_j)$ denote graph geodesic distance. The loss is computed as

$$\mathcal{L}_{\text{iso}} = \frac{1}{|P|} \sum_{(i,j)\in P} \left(\|z_i - z_j\|_2 - d_G(v_i, v_j)\right)^2$$

where P denotes the set of sampled pairs. The loss weight used in training is

$$\lambda_{\text{iso}} = 0.1$$

C.3 Source Prototype Bank

The prototype bank stores representative embeddings of source-domain superpixels.

Prototype Initialization

Initial prototypes are obtained via k-means clustering on source-domain embeddings.

Table 13. Prototype Bank Initialization

Parameter	Value
Clusters	512
Initialization	k-means++
Iterations	50

Prototype Update

Prototypes are updated using an exponential moving average:

$$p_k \leftarrow \beta p_k + (1 - \beta)\bar{z}_k$$

where p_k is the prototype vector, $\bar{z}_k$ is the batch centroid, and $\beta = 0.99$. The update is detached from gradient flow to prevent instability.

C.4 Max-Sliced Wasserstein Transport Layer

The MSWT layer computes transport between target embeddings and the prototype distribution.

Projection Direction Learning

Projection directions $e_k \in \mathbb{S}^{d-1}$ are implemented as normalized linear parameters.

Table 14. Projection Direction Configuration

Parameter	Value
Number of projections	64
Optimization	Adam
Learning rate	10^{-3}

Projection vectors are updated every five training iterations.

Two-Stage Transport Implementation

Transport is computed using two stages.

Stage 1: Approximate Transport

For each projection direction e_k: project embeddings, sort projected values, and compute quantile mapping.

Complexity:

$$\mathcal{O}(Kn \log n)$$

where K is the number of projections.

Stage 2: Sparse Refinement

Support points with transport residuals exceeding threshold τ are refined using local Sinkhorn iterations.

Table 15. Sparse Refinement Parameters

Parameter	Value
Sinkhorn iterations	10
Regularization	10^{-2}

C.5 Transported Feature Decoder

The decoder maps transported embeddings to anomaly predictions.

Decoder Architecture where K denotes the number of anomaly classes.

Table 16. Segmentation Decoder Architecture

Layer	Dimension	Activation
FC	$8 \rightarrow 64$	ReLU
FC	$64 \rightarrow$ K	—

Pixel-Level Reconstruction
Superpixel logits are propagated to pixels via the SLIC assignment map. Final segmentation maps are obtained using bilinear upsampling.

C.6 Optimization Procedure

Training minimizes the combined objective

$$\mathcal{L} = \mathcal{L}_{\text{seg}} + \lambda_{\text{iso}} \mathcal{L}_{\text{iso}}$$

where $\mathcal{L}_{\text{seg}}$ is the cross-entropy segmentation loss and $\mathcal{L}_{\text{iso}}$ is the manifold preservation loss.

Table 17. Optimization Settings

Parameter	Value
Optimizer	AdamW
Learning rate	3×10^{-4}
Weight decay	10^{-4}
Batch size	8
Training epochs	100
Gradient clipping	5

C.7 Training Schedule

The learning rate follows cosine annealing:

$$\eta_t = \eta_0 \frac{1}{2} \left(1 + \cos \left(\frac{\pi t}{T} \right) \right)$$

with warmup during the first five epochs.

Transport regularization follows exponential annealing:

$$\epsilon(t) = 0.5 e^{-3t/T}$$

where T denotes total training steps.

C.8 Inference Configuration

During inference:
- Prototype bank is frozen.
- Projection directions are fixed.
- No parameter updates occur.

Total inference time per 512x512 image is approximately 47 ms, including spectral index computation, superpixel segmentation, transport alignment, and decoding.

C.9 Software Environment

The implementation uses the following environment:

Table 18. Software Environment

Component	Version
Python	3.10
PyTorch	2.1
CUDA	12.1
scikit-image	0.21
NumPy	1.25