frontiers | Frontiers in Big Data

ORIGINAL RESEARCH
published: 10 May 2022
doi: 10.3389/fdata.2022.878716

Wasserstein Uncertainty Estimation for Adversarial Domain Matching

Rui Wang[1]*, *Ruiyi Zhang*[2] and *Ricardo Henao*[1]

[1] Department of Electrical and Computer Engineering, Duke University, Durham, NC, United States, [2] Department of Computer Science, Duke University, Durham, NC, United States

Domain adaptation aims at reducing the domain shift between a labeled source domain and an unlabeled target domain, so that the source model can be generalized to target domains without fine tuning. In this paper, we propose to evaluate the cross-domain transferability between source and target samples by domain prediction uncertainty, which is quantified via Wasserstein gradient flows. Further, we exploit it for reweighting the training samples to alleviate the issue of domain shift. The proposed mechanism provides a meaningful curriculum for cross-domain transfer and adaptively rules out samples that contain too much domain specific information during domain adaptation. Experiments on several benchmark datasets demonstrate that our reweighting mechanism can achieve improved results in both balanced and partial domain adaptation.

Keywords: Wasserstein, domain adaptation, uncertain, optimal transport, image classification

OPEN ACCESS

Edited by:
Susu Xu,
Stony Brook University, United States

Reviewed by:
Jingxiao Liu,
Stanford University, United States
Debasmit Das,
Qualcomm, United States

***Correspondence:**
Rui Wang
rw161@duke.edu

Specialty section:
This article was submitted to Data Science, a section of the journal Frontiers in Big Data

Received: *18 February 2022*
Accepted: *31 March 2022*
Published: *10 May 2022*

Citation:
Wang R, Zhang R and Henao R (2022) Wasserstein Uncertainty Estimation for Adversarial Domain Matching. Front. Big Data 5:878716. doi: 10.3389/fdata.2022.878716

1. INTRODUCTION

Unsupervised domain adaptation transfers knowledge from a labeled source domain to an unlabeled target domain. The goal is to learn a shared latent representation of source and target samples, complemented with a classifier for accurate classification using the latent representation as input. During learning, the differences between source and target representations are minimized at a population (distribution) level, while the discriminative ability of the classifier is maximized using only the labeled source data. Subsequently, the learned classifier and representation can be used to predict on target samples without the need of manual labeling effort. With the popularity of Generative Adversarial Nets (GANs) (Goodfellow et al., 2014), recent approaches generally match the source and target latent representations (features) via adversarial training, where a domain discriminator is used to classify the source and target features while the feature encoders are trained adversarially so the the discriminator cannot tell the differences between the two domains.

However, there are several problems with the current adversarial domain matching approaches: (*i*) The datasets may include samples that contain too much domain specific information. Matching with such samples may cause unreliable gradient and deterioration during training (Wen et al., 2019). (*ii*) Most of these approaches assume that the source and target are generated from the same set of classes, i.e., the *balanced domain adaptation*. In many real applications, however, in order to for the source knowledge to cover the target but with no information on target labels, we may have to collected a much larger source dataset with classes that do not present in the target domain. Such problem is described as the *partial domain adaptation* (Cao et al., 2017, 2018). As illustrated in **Figure 1**, these scenarios are challenging because simply matching the source and target feature distributions is likely to result in *negative transfer*. This happens because distribution matching may force observations from the target to be placed nearby source observations whose label is not present in the target, thus negatively impacting the quality of the learnt target representation. As

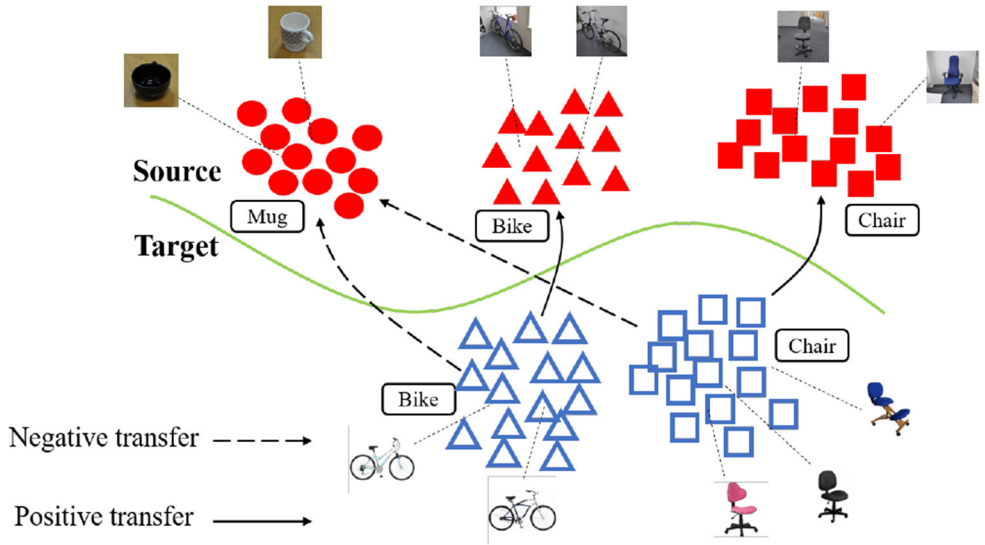

FIGURE 1 | Negative transfer in partial domain adaptation. Source and target features are colored red and blue, respectively. Target instances will be negatively transferred to source classes that are absent in the target domain (e.g., "bike" to "mug"), resulting in poorly discriminative target features.

a result, the adapted model may be sometimes worse than that trained on the source, as the target representation is poorly discriminative after adaptation.

In this paper, we propose to alleviate these issues via reweighting the source and target samples by their domain prediction uncertainty, where the uncertainty is estimated by employing a probabilistic domain discriminator. The weights based on uncertainty estimation is a measure of transferability of the source and target instances. It provides an adaptive curriculum that enables domain adaptation to initially focuses on domain uncertain samples that are close to the domain classification boundary (easy to transfer), then move to domain specific samples (difficult to transfer). Such a strategy can improve the stability of the adversarial learning process (Wen et al., 2019). Furthermore, source classes that are not presented in target can be easily identified with low domain uncertainty in partial domain adaptation. These classes will be down weighted, allowing the domain matching to focus on more target-related and informative source samples. In order to accurately estimate the domain prediction uncertainty, we account for the uncertainty in model parameters, i.e., by leveraging a Bayesian neural network (BNN) as the domain discriminator. In such case, the exact posterior is intractable, therefore, we approximate the posterior following Wasserstein gradient flows (WGFs) and a numerical solution of WGFs is proposed. We further define our *Wasserstein uncertainty estimation* via the mean entropy and the variance of the posterior predictions. *Wasserstein uncertainty estimation* can be easily integrated into current methods with adversarial domain matching, enabling appropriate uncertaint reweighting. Experimental results show significant improvement, obtaining improved results on both balanced and partial domain adaptation benchmarks.

2. BACKGROUND

2.1. Adversarial Domain Matching

Assume we have a labeled *source* dataset, $\mathcal{D}_s \triangleq (X_s, Y_s)$, where X_s and Y_s represent source inputs and labels, respectively. The source label, $Y_s \in \mathcal{Y}_s$, can take one of K_s distinct labels with probability $P(Y_s)$. We seek to leverage information in the source and a set of (unlabeled) *target* inputs, X_t, to develop a target label classification model without knowing the target label Y_t. Similarly, $Y_t \in \mathcal{Y}_t$ can take one of K_t distinct labels with probability $P(Y_t)$. Here we not only consider the standard scenario, denoted as *balanced domain adaptation*, where $\mathcal{Y}_s = \mathcal{Y}_t$ and $P(Y_s) = P(Y_t)$, but also the *partial domain adaptation*, where $\mathcal{Y}_t \subset \mathcal{Y}_s$, i.e., the target labels are a true subset of the source labels, so $K_t < K_s$. This is a common scenario in practice, where we need to transfer from a large source dataset to a smaller target with fewer number of classes.

Previous methods, such as Tzeng et al. (2017), perform latent representation distribution matching in an adversarial manner. The intuition is to learn a domain invariant representation that only contains the label information. As in the concrete part of **Figure 2**, adversarial domain matching consists of three components: a source and target encoder Enc$(\cdot)$, a label predictor $C(\cdot)$, and a domain discriminator $D(\cdot)$.

The source encoder Enc$(\mathbf{x}_s; \psi_s)$ and label predictor $C_\phi(\cdot)$ are trained in a supervised-learning manner on $\mathcal{D}_s$ by minimizing the cross-entropy loss as:

$$\mathcal{L}_c = -\mathbb{E}_{(x_s,y_s)\sim\mathcal{D}_s}[y_s^\top \log\{C_\phi(\text{Enc}(\mathbf{x}_s; \psi_s); \phi_c)\}], \quad (1)$$

where $C_\phi(\cdot)$ is assumed to perform a softmax activation operation and $\mathcal{D}_s$ is the joint distribution of the source. Once trained on the source dataset, both Enc$(\cdot\,; \psi_s)$ and $C_\phi(\cdot)$ will be fixed during adaptation.

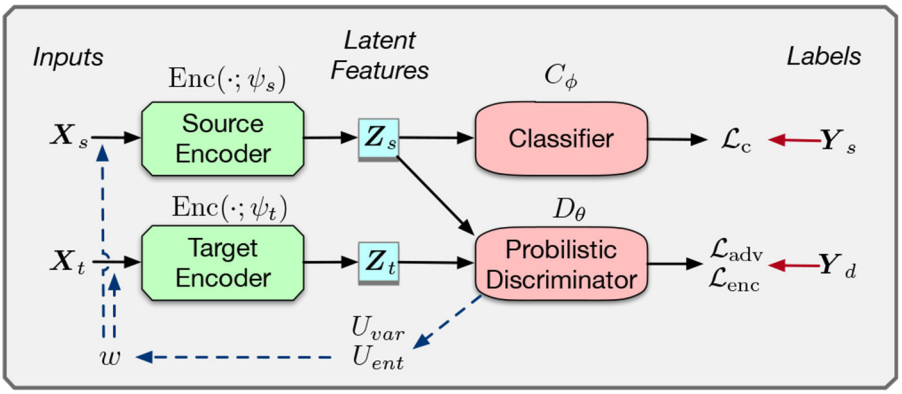

FIGURE 2 | Framework for adversarial domain matching with uncertainty reweighting. Y_s and Y_d are the source and domain labels, respectively. Model blocks are represented as rectangles and losses as ellipses. The concrete part is a general framework of of adversarial domain matching. The dash lines represents the process of our uncertainty reweighting.

To minimize the impact of the discrepancy of features between source and target domains, the discriminator is learnt to classify the source and target feature domains, while the target encoder is trained to fool D, until D can no longer set an effective boundary between the source and target. During training, the discriminator is trained by minimizing the adversarial objective, $\mathcal{L}_{\text{adv}}$,

$$\begin{aligned}\mathcal{L}_{\text{adv}} = &- \mathbb{E}_{\mathbf{x}_s\sim X_s}\ \log D_\theta(\mathrm{Enc}(\mathbf{x}_s,\psi_s)) \\ &- \mathbb{E}_{\mathbf{x}_t\sim X_t}\ \log\left(1 - D_\theta(\mathrm{Enc}(\mathbf{x}_t;\psi_t))\right)\end{aligned} \tag{2}$$

And the target encoder is separately minimized by,

$$\mathcal{L}_{\text{enc}} = -\mathbb{E}_{\mathbf{x}_t\sim X_t}\ \log(D_\theta(\mathrm{Enc}(\mathbf{x}_t;\psi_t)))\,, \tag{3}$$

where we have inverted the labels relative to Equation (2) as in Goodfellow et al. (2014), which has the same properties of the original min-max loss used in GAN but results in stronger gradients for the target encoder.

2.2. Wasserstein Gradient Flows

Wasserstein uncertainty estimation is achieved via Wasserstein gradient flows (WGFs) (Villani, 2008). It is a generalization of gradient flows on Euclidean space. Formally, we first endow a Riemannian geometry (Carmo, 1992) on $\mathcal{P}(\Omega)$. The geometry is characterized by the length between two elements (two distributions), defined by the second-order Wasserstein distance:

$$W_2^2(\mu,\nu) \triangleq \inf_\gamma \left\{ \int_{\Omega\times\Omega} \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2^2 \mathrm{d}\gamma(\boldsymbol{\theta},\boldsymbol{\theta}') : \gamma \in \Gamma(\mu,\nu) \right\},$$

where $\Gamma(\mu,\nu)$ is the set of joint distributions over $(\boldsymbol{\theta},\boldsymbol{\theta}')$ such that the two marginals equal μ and ν, respectively. The Wasserstein distance defines an optimal-transport problem, where one wants to transform μ to ν with minimum cost (Villani, 2008). Thus, the term $\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2^2$ represents the cost to transport $\boldsymbol{\theta}$ in μ to $\boldsymbol{\theta}'$ in ν, and can be replaced by a general metric $c(\boldsymbol{\theta},\boldsymbol{\theta}')$ in a metric space. If μ is absolutely continuous w.r.t. the Lebesgue measure, there is a unique optimal transport plan from μ to ν, i.e., a mapping $T:\mathbb{R}^d \to \mathbb{R}^d$ pushing μ onto ν satisfying $T_\#\mu = \nu$. Here $T_\#\mu$ denotes the pushforward measure (Villani, 2008) of μ. The Wasserstein distance thus can be equivalently reformulated as

$$W_2^2(\mu,\nu) \triangleq \inf_T \left\{ \int_\Omega \|\boldsymbol{\theta} - T(\boldsymbol{\theta})\|_2^2 \mathrm{d}\mu(\boldsymbol{\theta}) \right\}, \tag{4}$$

Consider $\mathcal{P}(\Omega)$ with a Riemannian geometry endowed by the second-order Wasserstein metric. Let $\{\mu_\tau\}_{\tau\in[0,1]}$ be an absolutely continuous curve in $\mathcal{P}(\Omega)$ with distance between μ_τ and $\mu_{\tau+h}$ measured by $W_2^2(\mu_\tau,\mu_{\tau+h})$. We overload the definition of T to denote the underlying transformation from μ_τ to $\mu_{\tau+h}$ as $\boldsymbol{\theta}_{\tau+h} = T_h(\boldsymbol{\theta}_\tau)$. Motivated by the Euclidean-space case, if we define $\mathbf{v}_\tau(\boldsymbol{\theta}) \triangleq \lim_{h\to 0} \frac{T_h(\boldsymbol{\theta}_\tau) - \boldsymbol{\theta}_\tau}{h}$ as the *velocity of the particle*, a gradient flow can be defined on $\mathcal{P}(\Omega)$ correspondingly in Lemma 1 (Ambrosio et al., 2005).

Lemma 1. *Let $\{\mu_\tau\}_{\tau\in[0,1]}$ be an absolutely-continuous curve in $\mathcal{P}(\Omega)$ with finite second-order moments. Then for a.e. $\tau \in [0,1]$, the above vector field $\boldsymbol{v}_\tau$ defines a gradient flow on $\mathcal{P}(\Omega)$ as $\partial_\tau\mu_\tau + \nabla_{\boldsymbol{\theta}}\cdot(\boldsymbol{v}_\tau\,\mu_\tau) = 0$, where $\nabla_{\boldsymbol{\theta}}\cdot\boldsymbol{a} \triangleq \nabla_{\boldsymbol{\theta}}^\top\boldsymbol{a}$ for a vector $\boldsymbol{a}$.*

Function F above is lifted to be a functional in the space of probability measures, mapping a probability measure μ to a real value, i.e., $F:\mathcal{P}(\Omega)\to\mathbb{R}$. F is the energy functional of a gradient flow on $\mathcal{P}(\Omega)$. Consequently, it can be shown that $\mathbf{v}_\tau$ in Lemma 1 has the form $\mathbf{v}_\tau = -\nabla_{\mathbf{x}}\frac{\delta F}{\delta\mu_\tau}(\mu_\tau)$ (Ambrosio et al., 2005), where $\frac{\delta F}{\delta\mu_\tau}$ is called the *first variation* of F at μ_τ (Dougan and Nochetto, 2022). Based on this, gradient flows on $\mathcal{P}(\Omega)$ can be written in a form of partial differential equation (PDE) as

$$\partial_\tau\mu_\tau = -\nabla_{\boldsymbol{\theta}}\cdot(\mathbf{v}_\tau\,\mu_\tau) = \nabla_{\boldsymbol{\theta}}\cdot\left(\mu_\tau\nabla_{\boldsymbol{\theta}}\left(\frac{\delta F}{\delta\mu_\tau}(\mu_\tau)\right)\right). \tag{5}$$

Intuitively, an energy functional F characterizes the landscape structure of the corresponding manifold, and the gradient flow (Equation 5) defines a solution path on this manifold. Usually, by choosing appropriate F, the landscape is convex, e.g., the Itô-diffusion case (Chen C. et al., 2018). This provides a theoretical guarantee of optimal convergence of a gradient flow.

3. PROPOSED METHOD

In domain matching, instances with high prediction uncertainty from the discriminator are usually less domain specific, thus can be easily transferred. Besides, samples with low domain uncertainty might contain too much domain specific information. Such samples may cause unreliable gradient and result in instability during adversarial training. Therefore, the discriminator prediction uncertainty can serve as a measure of cross-domain transferability between the source and target samples, enabling adaptive learning from easy to difficult instances.

3.1. Wasserstein Uncertainty Estimation for Probabilistic Discriminator

We consider learning posterior distributions for the parameters of the discriminator for the uncertainty estimation, instead of a point estimation. We further leverage the uncertainty of domain predictions to reweight source and target instances. The posterior distribution of complex models, e.g., the neural networks, is usually intractable. For computational convenience, traditional BNN learning typically assumes fully factorized Gaussian proposals as posterior approximation when adopting variational inference (Blundell et al., 2015; Hernández-Lobato and Adams, 2015). It is obvious that the factorized Gaussian posteriors usually lead to unreasonable approximation errors and underestimate model uncertainty (underestimate variances) (Liu and Wang, 2016). Further, particle-based variational inference methods (Chen C. et al., 2018), e.g., Wasserstein gradient flows, iteratively transports a set of particles to approximate the target posterior distribution, without making explicit assumptions about the form of the posterior and avoiding the aforementioned factorization assumption.

We consider a posterior distribution $p_{\boldsymbol{\theta}} \triangleq p(\boldsymbol{\theta}|\mathcal{D}) \propto p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})$, where $\boldsymbol{\theta} \in \mathbb{R}^r$ represents the parameter of domain discriminator. The canonical form is $p(\boldsymbol{\theta}|\mathcal{D}) = (1/Z)\exp(Q(\boldsymbol{\theta}))$.

$$\begin{aligned} Q(\boldsymbol{\theta}) &\triangleq \log p(\mathcal{D}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) \\ &= \sum_{i=1}^{N} \log p(\mathbf{x}_i|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}), \end{aligned} \tag{6}$$

where potential energy is based on an i.i.d. assumption of the model, and Z is the normalizing constant, which is intractable if the discriminator model is a neural network. To apply WGFs for posterior approximation in domain discriminator, a variational (posterior) distribution for $\boldsymbol{\theta}$, denoted as $\mu(\boldsymbol{\theta})$, is learned by solving an appropriate gradient-flow problem. To make the stationary distribution of the WGF consistent with the target posterior distribution, we define an energy functional characterizing the similarity between the current variational distribution and the true distribution $p_{\boldsymbol{\theta}}$ as:

$$\begin{aligned} F(\mu) &\triangleq \underbrace{-\int Q(\boldsymbol{\theta})\mu(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}}_{E_1} + \underbrace{\int \mu(\boldsymbol{\theta})\log\mu(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}}_{E_2} \\ &= \mathsf{KL}\left(\mu\|p_{\boldsymbol{\theta}}\right). \end{aligned} \tag{7}$$

Note E_2 is the energy functional of a pure Brownian motion (e.g., $U(\boldsymbol{\theta}) = 0$ in Equation 7). According to Equation (5), the first variation of functional E_1 and E_2 can be calculated as:

$$\frac{\delta E_1}{\delta \mu} = -Q, \quad \frac{\delta E_2}{\delta \mu} = \log\mu + 1. \tag{8}$$

Substituting Equations (8) into (5) yields the specific PDE form of the WGF. The energy functional $F(\mu)$ defines a landscape determined by $\mathcal{D}_s$, whose minimum is obtained at $\mu = p_{\boldsymbol{\theta}}$.

3.2. A Numerical Solution of WGFs

To solve the above WGF problem (Equation 5) we proposed to use particles, approximating μ with M particles $\{\boldsymbol{\theta}^i\}_{i=1}^M$ as

$$\mu^{(h)} \approx \frac{1}{M}\sum_{i=1}^{M} \delta_{\boldsymbol{\theta}^i}, \tag{9}$$

where $\delta_{\boldsymbol{\theta}_k}$ is a delta function with a spike at $\boldsymbol{\theta}_k$. Consequently, solving for the optimal μ is equivalent to updating the particles. We investigate the numerical solution to solve (Equation 5) via the discrete-gradient-flow method.

Discrete gradient flows (DGFs) approximate (Equation 5) by discretizing the continuous curve μ_t into a piece-wise linear curve, leading to an iterative optimization problem to solve the intermediate points denoted as $\{\mu_k^h\}_k$, where k denotes the discrete points, and h is referred to as the stepsize parameter. The iterative optimization problem is also known as the minimizing movement scheme (MMS) (Jordan et al., 1998), where for iteration k, $\mu_{k+1}^{(h)}$ is obtained by solving the following optimization problem:

$$\mu_{k+1}^{(h)} = \arg\min_{\mu} \mathsf{KL}\left(\mu\|p_{\boldsymbol{\theta}}\right) + \frac{W_2^2(\mu, \mu_k^{(h)})}{2h}. \tag{10}$$

With particles approximating the μ in Equation (9), the evolution of distributions described by Equation (5) can be approximated with gradient descent on particles. According to Liu and Wang (2016), the gradient of the first term $F_1 \triangleq \mathsf{KL}\left(\mu\|p_{\boldsymbol{\theta}}\right)$ can be easily approximated as:

$$\frac{\partial F_1}{\partial \boldsymbol{\theta}_k^i} = \sum_{j=1}^{M}\left[-\kappa(\boldsymbol{\theta}_k^j, \boldsymbol{\theta}_k^i)\nabla_{\boldsymbol{\theta}_k^i} Q(\boldsymbol{\theta}_k^i) + \nabla_{\boldsymbol{\theta}_k^j}\kappa(\boldsymbol{\theta}_k^j, \boldsymbol{\theta}_k^i)\right], \tag{11}$$

where κ is the kernel function, which typically is the radial basis function (RBF) kernel defined as $\kappa(\boldsymbol{\theta}, \boldsymbol{\theta}') = \exp(-\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2^2/h)$.

3.2.1. Particle-Based Estimation of Wasserstein Distance

Unfortunately, the exact minimization of the Wasserstein distance $W_2^2(\cdot,\cdot)$ over γ is in general computational intractable (Genevay et al., 2018; Salimans et al., 2018). Chen C. et al. (2018) uses a Sinkhorn-style algorithm to compute the Wasserstein distance but without iterative process assuming the parallel transport (Liu et al., 2019). It renders an inexact estimation with almost equal weights in γ. To overcome this issue, we consider an efficient iterative approach to approximate the Wasserstein distance based on the particle approximation.

We propose to use the recently introduced Inexact Proximal point method for Optimal Transport (IPOT) (Xie et al., 2018) algorithm to compute the matrix $\mathbf{T}^*$. IPOT provides a solution to the original Wasserstein distance specified in Equation (4). Specifically, IPOT iteratively solves the following optimization problem using the proximal point method (Boyd and Vandenberghe, 2004):

$$\mathbf{T}^{(t+1)} = \arg\min_{\mathbf{T}\in\Pi(\mathbf{x},\mathbf{y})} \left\{ \langle \mathbf{T}, \mathbf{C} \rangle + \beta \cdot \mathcal{B}(\mathbf{T}, \mathbf{T}^{(t)}) \right\},$$

where the proximity metric term $\mathcal{B}(\mathbf{T}, \mathbf{T}^{(t)})$ penalizes solutions that are too distant from the latest approximation, and $\frac{1}{\beta}$ is understood as the generalized stepsize. This renders a tractable iterative scheme toward the exact Wasserstein distance. In this work, we employ the generalized KL Bregman divergence $\mathcal{B}(\mathbf{T}, \mathbf{T}^{(t)}) = \sum_{i,j} \mathbf{T}_{ij} \log \frac{\mathbf{T}_{ij}}{\mathbf{T}_{ij}^{(t)}} - \sum_{i,j} \mathbf{T}_{ij} + \sum_{i,j} \mathbf{T}_{ij}^{(t)}$ as the proximity metric. Finally, we can update the particles in the k-th iteration with $\mathbf{T}$ computed based on IPOT and fixed when updating the particles:

$$\begin{aligned}\boldsymbol{\theta}_{k+1}^i = \boldsymbol{\theta}_k^i + \frac{h}{M} \sum_{j=1}^{M} \left[-\kappa(\boldsymbol{\theta}_k^j, \boldsymbol{\theta}_k^i) \nabla_{\boldsymbol{\theta}_k^i} U(\boldsymbol{\theta}_k^i) + \nabla_{\boldsymbol{\theta}_k^j} \kappa(\boldsymbol{\theta}_k^j, \boldsymbol{\theta}_k^i) \right] \\ + h \sum_{j=1}^{M} \mathbf{T}_{ij} (\boldsymbol{\theta}_k^i - \boldsymbol{\theta}_{k-1}^j). \end{aligned} \tag{12}$$

3.3. Adversarial Domain Matching via Wasserstein Uncertainty Estimation

Given particles $\{\boldsymbol{\theta}_k\}_{k=1}^M$, the prediction uncertainty of the domain discriminator can be estimated with different metrics. In our approach, we consider two uncertainty measures: entropy and variance uncertainty. Let x be a training sample, its domain prediction uncertainty can be estimated as,

$$U_{ent}(\mathbf{x}) = H\left(\frac{1}{M} \sum_{k=1}^{M} D_{\boldsymbol{\theta}_k}(\text{Enc}(\mathbf{x}; \psi)) \right) \tag{13}$$

$$U_{var}(\mathbf{x}) = \left\| \text{cov}(\{D_{\boldsymbol{\theta}_k}(\text{Enc}(\mathbf{x}; \psi))\}_{k=1}^M) \right\|_2 \tag{14}$$

where $H(\cdot)$ is the entropy function and $\text{cov}(\cdot)$ is the covariance operator. The norm of the covariance matrix corresponds to its largest eigenvalue, which is identical to the largest variance among all 1D projections of the particle outputs $\{D_{\boldsymbol{\theta}_k}(\text{Enc}(\mathbf{x}; \psi))\}_{k=1}^M$. Since (Equations 13, 14) are based in the particles solve from WGF, as in Section 3.2, $U_{ent}(\mathbf{x})$ and $U_{var}(\mathbf{x})$ are called *Wasserstein uncertainty estimation*. They describes the domain prediction uncertainty of the input samples with a domain discriminator.

For adversarial domain matching, we propose to reweight the source and target samples according to our domain prediction uncertainty. Given a training sample, its uncertainty weight can be evaluated as,

$$w(\mathbf{x}) = \lambda \frac{U_{ent}(\mathbf{x})}{\log 2} + (1 - \lambda) \frac{U_{var}(\mathbf{x})}{U_{var}^{max}} \tag{15}$$

where U_{var}^{max} is the maximum variance uncertain in the current minibatch, and $\lambda \in [0, 1]$ is a weighting parameter. Since the discriminator is a binary classifier, the largest entropy is bounded by $\log 2$. Hence we normalize the entropy-based uncertainty into $[0, 1]$ by scaling it with $\log 2$.

We integrate our uncertainty weights into the adversarial domain matching in Section 2.1. For each particle k for the discriminator, $k = 1, \ldots, M$, the adversarial objective can be modified as,

$$\begin{aligned} L_{\text{adv}_k}^w = &- \mathbb{E}_{x\sim p(X_s)} \, w(\mathbf{x}) \log D_{\theta_k}(\text{Enc}(\mathbf{x}; \psi_s)) \\ &- \mathbb{E}_{x\sim p(X_t)} \, w(\mathbf{x}) \log(1 - D_{\theta_k}(\text{Enc}(\mathbf{x}; \psi_t))). \end{aligned} \tag{16}$$

The loss for the target encoder is modified as,

$$L_{\text{enc}}^w = -\mathbb{E}_{x\sim p(X_t)} \, w(\mathbf{x}) \log(\sum_{k=0}^{M} D_{\theta_k}(\text{Enc}(\mathbf{x}; \psi_t))) \tag{17}$$

The complete procedure for the proposed adversarial domain matching is illustrated in **Algorithm 1**. Such a matching mechanism can be implemented in virtually any domain adaptation algorithm based on adversarial domain matching. Specifically, in our experiments, we modify the adversarial training process of Wang et al. (2019) with our uncertainty reweighting. The results show that our method can yield remarkable improvements and effectively alleviate negative transfer in case of partial domain adaptation.

Remark 1. *The proposed method can be regarded as a scheme of adaptive importance sampling, which excludes the difficult instances at the earlier stage of adaptation and stabilize the adversarial training. Further, in partial domain adaptation, source classes not included in target will be predicted with very low domain uncertainty throughout training. These classes will be down weighted and ruled out during domain matching.*

4. RELATED WORK

Unsupervised domain adaptation is based on the appropriate matching between the source and target distributions (Tzeng et al., 2014; Long et al., 2015, 2016; Sun and Saenko, 2016; Sun et al., 2016). Driven by the increasing popularity of the Generative Adversarial Networks (GANs) (Goodfellow et al., 2014), recent adaptation methods resort to matching the distributions in an adversarial manner. Long et al. (2015) and Tzeng et al. (2017) added a discriminator on the output of bottleneck layer in the model to distinguish features from different domains, while the feature encoders are trained to fool the discriminator so it cannot find an effective boundary that distinguishes between source and target instances. Zhang et al. (2018b) also add domain discriminators on the lower layers, which encourage domain specific information in shallower representations.

Cao et al. (2017) and Cao et al. (2018) introduced the concept of partial domain adaptation, in which target classes are assumed to be a subset of the source. They reduce the effect of negative transfer by selecting out classes not present in the target according to the prediction frequency, however, their approaches

Algorithm 1 Adversarial domain matching with Wasserstein uncertainty reweighting.

Let ψ_s, ψ_t, be the parameters for source and target encoders. $\{\theta_k\}_{k=1}^M$ be the parameters for the samples discriminator particles.
Input:
Source and target data: $\{X_s, Y_s\}$, X_t
Learning rates $\{\gamma_{\text{adv}}, \gamma_{\text{enc}}\}$
Batch size B
Number of particles M
Training source model, Enc$(\cdot; \psi_s)$ and $C(\cdot)$, with L_c
Fix Enc$(\cdot; \psi_s)$ and $C(\cdot)$.
Initialize $\{\theta_k\}_{k=1}^M$
while not converge **do**
Draw random minibatch $\{\mathbf{x}_s^i\}_{i=1}^B$, $\{\mathbf{x}_t^i\}_{i=1}^B$
$\psi_t = \psi_t - \gamma_{\text{enc}} \nabla_{\psi_t} L_{\text{enc}}^w$
Calculate $\{\nabla_{\theta_k} L_{\text{adv}_k}^w\}_{k=1}^M$
Update $\{\theta_k\}_{k=1}^M$ according to Equation (12)
end while

are only moderate when the source and target label domains are the same. Cao et al. (2019) propose to identify samples from the redundant source classes through class-aware domain discrimination, however, they did not take into account the probabilistic domain uncertainty. In our approach, we propose to employ a probabilistic domain discriminator and reweight the source the target samples with the domain prediction uncertainty. Our uncertainty weights impose a meaningful curriculum for adversarial domain matching and can also select out samples from redundant source classes during partial domain adaptation.

5. EXPERIMENTAL RESULTS

We denote our method as Wasserstein Uncertainty Domain Matching (WUDM). We evaluate WUDM on three domain adaptation benchmark datasets: the digits datasets, Office31 and Visda2017. In order to evaluate the effectiveness of our proposed Wasserstein uncertain estimation, we conduct an ablation test by using a domain discriminator with point estimation, denoted as UDM, where the uncertainty is estimated only with its prediction entropy as in Equation (13). UDM is an ablation study of our proposed Wasserstein Uncertainty Estimation, which is equivalent to estimating the domain uncertaintyăusing the entropy uncertainty (Namdari and Li, 2019) with deterministic models. Source code will be released at https://github.com/RayWangWR?.

5.1. Datasets

5.1.1. The Digits Datasets

We consider three digits datasets with varying difficulties: MNIST, SVHN and USPS, each containing 10 classes for digits 0-9. The encoder architecture for the digits images is the modified LeNet from Tzeng et al. (2017). For the domain classification, each sampled particle of the adversarial discriminator consists of 3 fully connected layers with 500 hidden units for the first two layers and 2 for the output. All images are converted to grayscale and rescaled to 28 × 28 pixels. Following the experiments of Tzeng et al. (2017), we consider three directions of transfer: SVHN→MNIST, USPS→MNIST and MNIST→USPS.

5.1.2. VisDA2017

This is a dataset for the Visual Domain Adaptation Challenge from synthetic 2D renderings of 3D models to real images. It consists of 12 classes of objects shared by both domains, each with a very large number of instances. The architecture of the encoder for images in Visda2017 is a Resnet-50 (He et al., 2016) pre-trained on ImageNet. All the images are first resized to 256 × 256 pixels RGB images, then random cropped during training and central cropped during testing into 224 × 224 RGB images for the model input.

5.1.3. Office31

This is a standard benchmark for domain adaptation widely used in computer vision, it consists of 4,652 images from 31 classes. These images are collected from three distinct domains: Amazon

TABLE 1 | Balanced domain adaptation on the digits datasets.

Method	SVHN→MNIST	USPS→MNIST	MNIST→USPS
LeNet LeCun et al., 1998	0.598	0.634	0.771
ADDA Tzeng et al., 2017	0.760	0.901	0.894
MCD Saito et al., 2018b	0.962	0.941	0.942
AdDropout Saito et al., 2018a	0.950	0.931	0.932
RAAN Chen Q. et al., 2018	0.892	0.921	0.890
JDDA-I Chen et al., 2019	0.931	0.970	-
EntroDA Wen et al., 2019	0.915	0.981	0.957
RUDA Wang et al., 2019	0.965	0.979	0.952
UDM	0.969	0.953	0.945
WUDM	**0.971**	**0.985**	**0.961**

The values in bold means it is the highest in each column.

TABLE 2 | Partial domain adaptation on VisDA2017.

Method	Syn-12→Real-6	Real-12→Syn6	Average
ResNet He et al., 2016	0.421	0.568	0.494
DANN Ganin et al., 2016	0.327	0.605	0.466
RTN Long et al., 2016	0.279	0.500	0.390
ADDA Tzeng et al., 2017	0.545	0.562	0.554
ADDA-mix Tzeng et al., 2017	0.543	0.605	0.574
PADA Cao et al., 2018	0.535	0.765	0.650
ENT Cao et al., 2019	0.706	0.708	0.707
RUDA Wang et al., 2019	0.700	0.846	0.773
UDM	**0.754**	0.711	0.733
WUDM	0.750	**0.864**	**0.807**

The values in bold means it is the highest in each column.

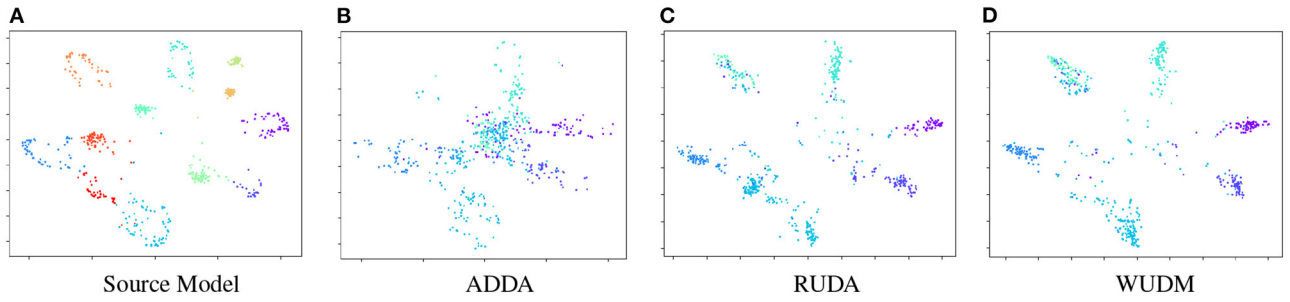

FIGURE 3 | *t*-SNE embeddings of the representations for Syn12→Real6. The features are embedded into the same subspace and visualized within the same scope. **(A)** is the source representation with 12 classes. **(B–D)** are the adapted target representation with 6 classes. The embedding from WUDM consists of 6 disentangled clusters, while others tend to be more entangled or contains excessive number of clusters.

TABLE 3 | Partial domain adaptation on Office31.

Method	A31→D10	D31→ W10	D31→ A10	Average
ResNet50 He et al., 2016	0.701	0.980	0.690	0.793
DANN Ganin et al., 2016	0.529	0.314	0.468	0.437
ADDA Tzeng et al., 2017	0.675	0.705	0.686	0.689
SAN Cao et al., 2017	0.813	0.986	0.806	0.868
IWAN Zhang et al., 2018a	0.790	0.990	0.895	0.892
PADA Cao et al., 2018	**0.865**	0.993	0.927	0.928
RUDA Wang et al., 2019	0.847	0.997	0.919	0.921
UDM	0.815	0.990	0.877	0.894
WUDM	0.860	**1.000**	**0.930**	**0.930**

The values in bold means it is the highest in each column.

(A), Webcam (W) and DSLR (D). This is a relatively difficult dataset since the Webcam and DSLR contains very small amount of images, i.e., less than 10 for some classes, which may easily lead to overfitting during the adaptation process. In order to explore different combinations of large and small datasets for the source and target, we consider three transfer directions: A→D (large to small), W→A (small to large) and W→D (small to small).

The data pre-processing and experiment setting are the same as above for Visda2017 except that we use ResNet-50 with 31-dimensional output instead of 12. Due to the small size of Office31, we approach the task as fully transductive, where all labeled instances from the source and all unlabeled instances from the target are used during training and adaptation. This is the same for the experiments in Long et al. (2015), Ganin et al. (2016), and Tzeng et al. (2017). Complementary to Visda2017, Office31 will validate the performance of our method on small-scale datasets.

5.2. Balanced Domain Adaptation

5.2.1. The Digit Datasets

We conduct experiments with all the 10 digits in the balanced setting. The results are shown in **Table 1**. Our method outperforms all the other baselines in all three directions, which demonstrates the effectiveness of our method in the standard balanced domain adaptation.

Note that the proposed method outperforms simple model (point estimation) with a large margin. This indicated that our uncertainty weights with Equation (15) is more accurate in representing sample uncertainty, demonstrating the effectiveness of our Wasserstein uncertainty estimation.

5.3. Partial Domain Adaptation

5.3.1. VisDA2017

Following Cao et al. (2018), we only reserve images of the first 6 classes of VisDA2017 in alphabetic order in the target domain (REAL-6, SYN-6), and all the images of the 12 classes are kept in the source domain (REAL-12, SYN-12). The results for SYN12→REAL6 and REAL12→SYN6 are shown in **Table 2**. Our method outperforms RUDA and the other baselines by a large margin. This validates the usefulness of our uncertainty reweighting for partial domain adaptation.

In **Figure 3**, we visualize the source and target features of SYN12→REAL6 in the same subspace with *t*-SNE. It can be shown that other methods tend to negatively transfer the target samples toward the redundant source classes. These samples will be misclassified by the source classifier and caused degraded performance for the adaptation. Our method promotes transfer within the same class, which preserves intra-class structure of target representation during domain adaptation. The resulting target representation is more discriminative and less entangled.

5.3.2. Office31

We select the 10 classes shared by Office31 and Caltech-256 as our target labels. For each direction of adaptation, we use all the images of these 10 classes in the target split as the target domain (denoted as A10, W10, D10), and images from all the 31 classes in the source split as the source domain (denoted as A31, W31, D31). In **Table 3**, our method is better than RUDA and UDM in all three directions. These experiments validate the effectiveness of our uncertainty weighting in alleviating negative target transfer on small datasets as Office31.

Combined with the experiments of Visda2017, our method tend to produce larger performance improvement compared with the results from balanced domain adaptation. This is because partial domain adaptation suffers from larger degree of negative transfer, and our method can alleviate such effect

by ruling out the irrelevant source classes and focusing on source samples that are more informative for target classification. The average accuracies of WUDM are higher then UDM in both Visda2017 and Office31. This validates the usefulness of our Wasserstein uncertainty estimation in the case of partial domain adaptation.

6. CONCLUSIONS

In this paper, we propose to reweight the source and target samples in domain adaptation with domain prediction uncertainty. For estimation of domain uncertainty, we employ a probabilistic domain discriminator and develop the Wasserstein uncertainty estimation, which can be easily integrated into concurrent adversarial domain matching. The resulting uncertainty weights impose an adaptive curriculum on domain adaptation that stabilize adversarial training and alleviate the effect of negative transfer in the case of partial domain adaptation. Experiments on several benchmarks show that our method achieves improved results on both balanced and partial domain adaptation.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: http://yann.lecun.com/exdb/mnist/, https://paperswithcode.com/dataset/office-31, http://ai.bu.edu/visda-2017/.

AUTHOR CONTRIBUTIONS

All authors contributed to the article and approved the submitted version.

FUNDING

It is funded by ECE, Duke University.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fdata.2022.878716/full#supplementary-material

REFERENCES

Ambrosio, L., Gigli, N., and Savaré, G. (2005). *Gradient Flows in Metric Spaces and in the Space of Probability Measures*. Lectures in Mathematics ETH Zürich.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015). "Weight uncertainty in neural networks," in *ICML*.

Boyd, S., and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge university press.

Cao, Z., Long, M., Wang, J., and Jordan, M. I. (2017). Partial transfer learning with selective adversarial networks.

Cao, Z., Ma, L., Long, M., and Wang, J. (2018). Partial adversarial domain adaptation. *arXiv preprint arXiv:1808.04205*.

Cao, Z., You, K., Long, M., Wang, J., and Yang, Q. (2019). "Learning to transfer examples for partial domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Long Beach, CA: IEEE), 2985–2994.

Carmo, M. P. d. (1992). *Riemannian Geometry*. Birkhäuser.

Chen, C., Chen, Z., Jiang, B., and Jin, X. (2019). Joint domain alignment and discriminative feature learning for unsupervised deep domain adaptation. *Proc. AAAI Conf. Artif. Intell.* 33, 3296–3303. doi: 10.1609/aaai.v33i01.33013296

Chen, C., Zhang, R., Wang, W., Li, B., and Chen, L. (2018a). " A unified particle-optimization framework for scalable bayesian sampling," in *UAI*.

Chen, Q., Liu, Y., Wang, Z., Wassell, I., and Chetty, K. (2018b). "Re-weighted adversarial adaptation network for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT: IEEE), 7976–7985.

Dougan, G., and Nochetto, R. H. (2022). "First variation of the general curvature-dependent surface energy," in *ESAIM: Mathematical Modelling and Numerical Analysis*.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., et al. (2016). Domain-adversarial training of neural networks. *J. Mach. Learn. Res.* 17, 2096–2030. Available online at: https://jmlr.org/papers/v17/15-239.html

Genevay, A., Peyré, G., and Cuturi, M. (2018). "Learning generative models with sinkhorn divergences," in *AISTATS*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). "Generative adversarial nets," In *Advances in Neural Information Processing Systems*, 2672–2680.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV: IEEE), 770–778.

Hernández-Lobato, J. M., and Adams, R. (2015). "Probabilistic backpropagation for scalable learning of bayesian neural networks," in *ICML*.

Jordan, R., Kinderlehrer, D., and Otto, F. (1998). "The variational formulation of the fokker–planck equation," in *SIMA*.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. (1998). Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 2278–2324.

Liu, C., Zhuo, J., Cheng, P., Zhang, R., Zhu, J., and Carin, L. (2019). Understanding and accelerating particle-based variational inference. In *ICML*.

Liu, Q., and Wang, D. (2016). "Stein variational gradient descent: A general purpose bayesian inference algorithm," in *NIPS*.

Long, M., Cao, Y., Wang, J., and Jordan, M. I. (2015). Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*.

Long, M., Zhu, H., Wang, J., and Jordan, M. I. (2016). "Unsupervised domain adaptation with residual transfer networks," in *Advances in Neural Information Processing Systems*, 136–144.

Namdari, A., and Li, Z. (2019). A review of entropy measures for uncertainty quantification of stochastic processes. *Adv. Mech. Eng.* 11, 1687814019857350. Available online at: https://journals.sagepub.com/doi/pdf/10.1177/1687814019857350

Saito, K., Ushiku, Y., Harada, T., and Saenko, K. (2018a). "Adversarial dropout regularization," in *Proceedings of International Conference on Learning Representations*.

Saito, K., Watanabe, K., Ushiku, Y., and Harada, T. (2018b). "Maximum classifier discrepancy for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3723–3732.

Salimans, T., Zhang, H., Radford, A., and Metaxas, D. (2018). "Improving GANs using optimal transport," in *ICLR*.

Sun, B., Feng, J., and Saenko, K. (2016). "Return of frustratingly easy domain adaptation," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16), Vol. 6*, 8.

Sun, B., and Saenko, K. (2016). "Deep coral: correlation alignment for deep domain adaptation," in *European Conference on Computer Vision* (Springer), 443–450.

Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T. (2017). "Adversarial discriminative domain adaptation," in *Computer Vision and Pattern Recognition (CVPR), Vol. 1*, 4.

Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., and Darrell, T. (2014). Deep domain confusion: Maximizing for domain invariance. *CoRR*, abs/1412.3474.

Villani, C. (2008). *Optimal Transport: Old and New*. Springer Science & Business Media.

Wang, R., Wang, G., and Henao, R. (2019). Discriminative clustering for robust unsupervised domain adaptation. *arXiv preprint arXiv:1905.13331*.

Wen, J., Zheng, N., Yuan, J., Gong, Z., and Chen, C. (2019). Bayesian uncertainty matching for unsupervised domain adaptation. *arXiv preprint arXiv:1906.09693*.

Xie, Y., Wang, X., Wang, R., and Zha, H. (2018). A fast proximal point method for Wasserstein distance. *arXiv:1802.04307*.

Zhang, J., Ding, Z., Li, W., and Ogunbona, P. (2018a). "Importance weighted adversarial nets for partial domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8156–8164.

Zhang, W., Ouyang, W., Li, W., and Xu, D. (2018b). "Collaborative and adversarial network for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT: IEEE),3801–3809.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

The handling editor SX is currently organizing a Research Topic with the author(s) RZ.