entropy

MDPI

Article

Variational Inference via Rényi Bound Optimization and Multiple-Source Adaptation †

Dana Zalman (Oshri) [1,2,*] and Shai Fine [2]

[1] School of Computer Science, Reichman University, Herzliya 4610101, Israel
[2] Data Science Institute, Reichman University, Herzliya 4610101, Israel; shai.fine@runi.ac.il
* Correspondence: dana.oshri@post.runi.ac.il
† This paper is an extended version of our paper published in the 21st IEEE International Conference on Machine Learning and Applications (ICMLA), Nassau, Bahamas, 12–14 December, 2022.

Abstract: Variational inference provides a way to approximate probability densities through optimization. It does so by optimizing an upper or a lower bound of the likelihood of the observed data (the evidence). The classic variational inference approach suggests maximizing the Evidence Lower Bound (ELBO). Recent studies proposed to optimize the variational Rényi bound (VR) and the χ upper bound. However, these estimates, which are based on the Monte Carlo (MC) approximation, either underestimate the bound or exhibit a high variance. In this work, we introduce a new upper bound, termed the Variational Rényi Log Upper bound (VRLU), which is based on the existing VR bound. In contrast to the existing VR bound, the MC approximation of the VRLU bound maintains the upper bound property. Furthermore, we devise a (sandwiched) upper–lower bound variational inference method, termed the Variational Rényi Sandwich (VRS), to jointly optimize the upper and lower bounds. We present a set of experiments, designed to evaluate the new VRLU bound and to compare the VRS method with the classic Variational Autoencoder (VAE) and the VR methods. Next, we apply the VRS approximation to the Multiple-Source Adaptation problem (MSA). MSA is a real-world scenario where data are collected from multiple sources that differ from one another by their probability distribution over the input space. The main aim is to combine fairly accurate predictive models from these sources and create an accurate model for new, mixed target domains. However, many domain adaptation methods assume prior knowledge of the data distribution in the source domains. In this work, we apply the suggested VRS density estimate to the Multiple-Source Adaptation problem (MSA) and show, both theoretically and empirically, that it provides tighter error bounds and improved performance, compared to leading MSA methods.

Keywords: multiple-source adaptation; variational inference; Rényi divergence

Citation: Zalman, D.; Fine, S. Variational Inference via Rényi Bound Optimization and Multiple-Source Adaptation. *Entropy* **2023**, *25*, 1468. https://doi.org/10.3390/e25101468

Academic Editors: Krzysztof Grochla and Viacheslav Kovtun

Received: 25 July 2023
Revised: 10 September 2023
Accepted: 20 September 2023
Published: 20 October 2023

1. Introduction

In numerous practical situations, we encounter probability distributions that are challenging to calculate. This occurs especially when the distribution includes hidden variables. Therefore, it becomes necessary to employ approaches that can estimate or approximate such distributions. Variational inference (VI) is a technique used to accomplish this task. VI is a compelling approach for approximating posterior distributions in latent variable models [1]. It can handle intractable and possibly high-dimensional posteriors, and it makes Bayesian inference computationally efficient and scalable to large datasets. To this end, VI defines a simple distribution family, called the variational family, and then finds the optimal member of the variational family that is closest to the true posterior distribution. This transforms the posterior inference into an optimization problem concerning the variational distribution.

One of the most successful applications of VI in the deep neural network realm is the Variational Autoencoder (VAE) [2], which is a deep generative model that implements

a probabilistic model and variational Bayesian inference. Many techniques have been suggested to improve the accuracy and efficiency of variational methods (cf. [3–7]). Recent trends in variational inference have focused on the following aspects:

- Scalability: includes stochastic approximations.
- Generalization: extends the applicability of VI to a large class of otherwise intractable models, such as non-conjugate models.
- Accuracy: includes variational models beyond the mean field approximation.
- Amortization: implements the inference over local latent variables with inference networks.
- Robustness: generating a reliable representation of particular data types in the encoded space when using corrupted training data and detecting anomalies.

There are other methods for improving approximation such as Monte Carlo methods for VI and black-box methods [8].

In this work, we focus on the accuracy of the VAE models. An essential aspect of the VI methodology revolves around selecting an appropriate divergence method. This divergence measure allows us to approximate the true posterior distribution with a simpler variational distribution. Consequently, the selection of the divergence measure can have a notable impact on the accuracy of the approximation. Furthermore, using the selected divergence measure, one can devise lower and upper bounds, and estimate the true posterior.

Accordingly, we propose a new upper bound for the evidence, termed the Variational Rényi Log Upper bound (VRLU), based on the Variational Rényi (VR) bound suggested by Li and Turner [3]. Further, we devise a (sandwiched) upper–lower bound variational inference method, termed VRS, to jointly optimize the Rényi upper and lower bounds. The VRS loss function combines the VR lower bound and our new upper bound, thus providing a tighter estimate for the log evidence.

Next, we will demonstrate the practical effectiveness of VRS by applying it to the domain adaptation problem. Through this application, we aim to showcase the tangible benefits and practical relevance of our approach.

Domain adaptation is a scenario that arises when we aim to learn from a source data distribution; a well-performing model on a different (but related) target data distribution. A real-world example of domain adaptation is the common spam filtering problem. This problem consists of adapting a model from one user (the source distribution) to a new user who receives significantly different emails (the target distribution).

In the context of domain adaptation, the terms "source" and "target" domains are used to refer to the training and test sets, respectively. These sets can have distinct feature spaces, which can occur when the statistical properties of a domain change over time or when new samples are collected from various sources, resulting in domain shifts. Multiple-Source Adaptation (MSA) addresses scenarios where there are multiple source domains and one target domain. The central question is whether the learner can effectively combine relatively accurate predictors from each source domain to create an accurate predictor for any new target domain that may consist of a mixture of these sources.

In contrast to the majority of machine learning research, where models are trained and tested on data drawn from the same distribution, domain adaptation involves using data from different distributions for training and testing. When the train and test sets share the same distribution, the uniform convergence theory ensures that a model's empirical training error closely approximates its true error. This assumption is not guaranteed in the MSA problem.

In this work, we have focused on two main ideas:

- Improving the estimation of the domain distribution using VAE.
- Using the improved estimated distributions in the algorithm presented in [9] to solve the MSA problem.

The rest of the paper is organized as follows: Section 2 provides a review of variational inference for probabilistic modeling, and discusses different divergence methods such as KL Divergence, Rényi Divergence, and χ Divergence, for bounding the log evidence.

In Section 3, we present our novel approach, called Variational Rényi Log Upper bound (VRLU), which offers an improved bound for the log evidence. Additionally, we introduce an optimized technique, referred to as the Variational Rényi Sandwich (VRS), that leverages both upper and lower bounds. Section 4 offers a comprehensive overview of the domain adaptation problem and illustrates the application of the approximated distributions in calculating its loss function. Finally, in Section 5, we present a series of experiments conducted to evaluate the effectiveness of our proposed methods, VRLU and VRS, in the context of both log evidence estimation and domain adaptation.

2. Divergence Methods in Variational Inference for Probabilistic Modeling

In probabilistic modeling, we aim to devise a probabilistic model, p_θ, that best explains the data. This is commonly done by maximizing the log-likelihood of the data (also known as *log evidence*), with respect to the model's parameter θ, i.e., Maximum Likelihood Estimation (MLE). For a latent model, where we assume that the observed data, x, depend on a latent variable z, the MLE takes the following form:

$$\max_\theta \log p_\theta(x) = \max_\theta \log\left(\int p_\theta(x|z)p(z)dz\right) \tag{1}$$

For many latent models, the log evidence integral is unavailable in closed form or it is too complex to compute. A leading approach to handle such intractable cases is variational inference (VI). One of the most successful applications of VI in the deep neural network realm is the Variational Autoencoder (VAE).

2.1. Variational Autoencoder and the Kulback–Leibler Divergence

A Variational Autoencoder is a deep generative model that implements a probabilistic model and variational Bayesian inference. Introduced by Kingma and Welling [2], a VAE model is an autoencoder, designed to stochastically encode the input data into a constrained multivariate latent space (encoding), and then to reconstruct it as accurately as possible (decoding). To turn the intractable posterior inference into a solvable problem, we use a parametric inference model $q_\phi(z|x)$ which is also called an encoder. We optimize the variational parameters ϕ such that $q_\phi(z|x) \sim p_\theta(z|x)$. The VAE loss function is composed of a "reconstruction term" (to ensure the decoded data are close to the original data) and a "regularisation term". The goal of the regularisation term is to ensure that the distributions returned by the encoder are close to a standard normal distribution. That is expressed as the Kulback–Leibler divergence between the returned distribution and a standard Gaussian.

Definition 1. *Kulback–Leibler (KL) divergence [10,11]. For discrete probability distributions p and q, defined on the same probability space, the KL divergence from q to p is defined to be:*

$$D_{KL}(p||q) = \sum_x p(x) \log\left(\frac{p(x)}{q(x)}\right) \tag{2}$$

Since the true posterior $p_\theta(z|x)$ is intractable, we aim to approximate it with a Gaussian distribution $q_\phi(z|x)$, in the KL divergence sense. It follows that:

$$\log p_\theta(x) = D_{KL}(q_\phi(z|x)||p_\theta(z|x)) + \mathbf{ELBO} \tag{3}$$

Definition 2. *Evidence Lower Bound (ELBO):*

$$\mathbf{ELBO} := E_{z\sim q_\phi(z|x)}\left[\log(p_\theta(z,x)) - \log(q_\phi(z|x))\right] \tag{4}$$

We note that the KL divergence is non-negative, thus maximizing the ELBO results with the minimization of the KL divergence between $q_\phi(z|x)$ and the true posterior $p_\theta(z|x)$.

ELBO optimization is a well-known method that has been studied in depth, and is applicable in many models, especially in VAE [12]. Nevertheless, using the ELBO can give rise to some drawbacks. First, the ELBO is not always very tight, and maximizing the bound instead of the actual likelihood can lead to bias. Typically this leads to a simpler model q_ϕ, which approximates the real posterior. Second, the $D_{KL}(q_\phi(z|x)||p_\theta(z|x))$ does not always lead to the best results—it tends to favor approximate distributions q_ϕ that underestimate the entropy of the true posterior ("zero-forcing"). Namely, $D_{KL}(q_\phi(z|x)||p_\theta(z|x))$ is infinite when $p_\theta(z|x) = 0$ and $q_\phi(z|x) > 0$. Therefore, the optimal variational distribution q will be 0 when $p_\theta(z|x) = 0$. This "zero-forcing" behavior leads to degenerate solutions during optimization.

2.2. Rényi Divergence

One of the core parts of probabilistic models is the selection of the method for estimating the approximation of the distribution. In the previous section, we introduced Kulback–Leibler (KL) divergence. In this section, we will present the Rényi divergence (also known as α divergence), which measures the difference between two distributions p and q, and is defined by:

$$\begin{aligned} D_\alpha(p||q) =& \frac{1}{\alpha-1}\log\left(E_p\left[\left(\frac{p(x)}{q(x)}\right)^{\alpha-1}\right]\right) \\ =& \frac{1}{\alpha-1}\log\left(\sum_{x\in X}\frac{p(x)^\alpha}{q(x)^{\alpha-1}}\right) \end{aligned} \tag{5}$$

Rényi divergence was initially defined for $\alpha \in \{\alpha > 0\,, \alpha \neq 1\}$. The definition was extended to $\alpha = 0, 1, +\infty$ by continuity. There are certain α values for which Rényi divergence has a wider application than the others. Of particular interest are the values 0, $\frac{1}{2}$, 1, 2, and ∞, presented in Table 1. We note that for $\alpha \to 1$: $\lim_{\alpha\to 1} D_\alpha(p||q) = D_{KL}(p||q)$, the KL divergence is recovered.

Table 1. Special cases in the Rényi divergence family.

α	*Definition*	*Notes*
$\alpha \to 0$	$-\log q(\{p > 0\})$	Not a divergence
$\alpha \to 1$	$E_{x\sim p(x)}[\log \frac{p(x)}{q(x)}]$	KL divergence
$\alpha = \frac{1}{2}$	$-2\log\left(1 - \mathrm{Hel}^2(p||q)\right)$	Rényi divergence symmetric in its arguments
$\alpha = 2$	$-\log(1 - \chi^2(p||q))$	Correlated to the χ^2 divergence
$\alpha \to \infty$	$\log\max\left(\frac{p}{q}\right)$	Worst-case regret

2.2.1. Selected Properties of Rényi Divergence

Theorem 1. *(Positivity): For any order* $\alpha \in [0, \infty]$*:* $D_\alpha(p||q) \geq 0$ *, and* $D_\alpha(p||q) = 0 \iff p = q$

Theorem 2. *(Convexity): For any order* $\alpha \in [0, 1]$ *Rényi divergence is jointly convex in its arguments. That is, for any two pairs of probability distributions* (p_0, q_0) *and* (p_1, q_1)*, and any* $0 < \lambda < 1$*:*

$$D_\alpha((1-\lambda)p_0 + \lambda p_1||(1-\lambda)q_0 + \lambda q_1) \leq (1-\lambda)D_\alpha(p_0||q_0) + \lambda D_\alpha(p_1||q_1) \tag{6}$$

For any order $\alpha \in [0, \infty]$ *Rényi divergence is convex in its second argument. That is, for any probability distributions* p, q_0 *and* q_1*:*

$$D_\alpha(p||(1-\lambda)q_0 + \lambda q_1) \leq (1-\lambda)D_\alpha(p||q_0) + \lambda D_\alpha(p||q_1) \tag{7}$$

Theorem 3. *(Continuity in the Order): The Rényi divergence is continuous in* α *on* $A = \{\alpha \in [0, \infty] | 0 \leq \alpha \leq 1$ *or* $D_\alpha(p||q) < \infty\}$.

The definition of Rényi divergence was extended to $\alpha < 0$ as well. However, not all properties are preserved, and some are inverted. For example, Rényi divergence for negative orders is *non-positive* and *concave* in its first argument (cf. Figure 1). The extended definition of Rényi divergence to all $\alpha \in \mathbb{R}$ has some interesting properties:

Theorem 4. *(Monotonicity) [3]: Rényi divergence, extended to negative* α*, is continuous and non-decreasing on* $\alpha \in \{\alpha : -\infty < D_\alpha < +\infty\}$.

Lemma 1. *The Skew Symmetry property:*

- *For any* $\alpha \in (-\infty, \infty), \alpha \notin \{0, 1\}$

$$D_\alpha(p||q) = \frac{\alpha}{1-\alpha} D_{1-\alpha}(q||p)$$
$$D_{-\infty}(p||q) = -D_\infty(q||p)$$

- *For any* $\alpha \in (-\infty, \infty), \alpha \notin \{0, 1\}$

$$D_\alpha(p||q) \leq \frac{\alpha}{1-\alpha} D_{1-\alpha}(p||q)$$

Definition 3. *We will denote by* $d_\alpha(p||q)$ *the exponential:*

$$d_\alpha(p||q) = e^{D_\alpha(p||q)} = \left(\sum_{x \in X} \frac{p(x)^\alpha}{q(x)^{\alpha-1}} \right)^{\frac{1}{\alpha-1}} \tag{8}$$

Figure 1 illustrates d_α and D_α. One can see that d_α achieves high values very quickly. $D_\alpha(p||q)$ and $d_\alpha(p||q)$ are non-decreasing as functions of α, and:

$$d_\alpha(p||q) \leq d_\infty(p||q) = \sup_{x \in X} \left[\frac{p(x)}{q(x)} \right] \tag{9}$$

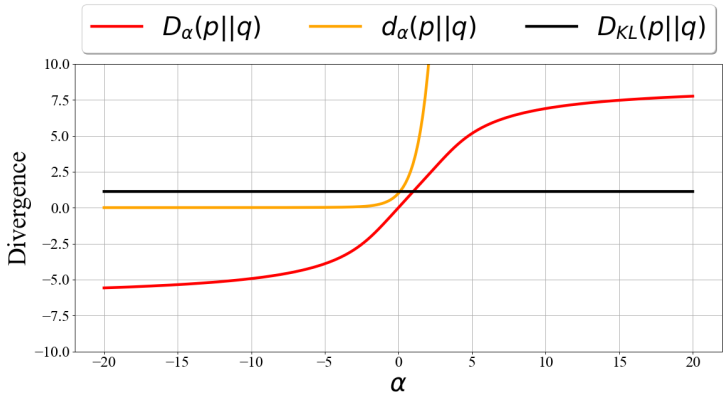

Figure 1. Illustration of $d_\alpha(p||q)$ vs. $D_\alpha(p||q)$ for fixed distributions p and q over different α values. $p \sim N(0, 2)$, $q \sim N(3, 2)$.

Many other properties described in [3,13].

2.2.2. Rényi Divergence Variational Inference

To estimate the evidence $p_\theta(x)$, we employ a minimization approach using Rényi divergence between the variational distribution $q_\phi(z|x)$ and the true posterior distribution $p_\theta(z|x)$, where α is a selected positive value. Extending the posterior $p_\theta(z|x)$ and using Bayes' theorem, we obtain:

$$\begin{aligned} D_\alpha(q_\phi(z|x)||p_\theta(z|x)) &= \frac{1}{\alpha-1}\log\left(E_{z\sim q_\phi(z|x)}\left[\left(\frac{q_\phi(z|x)}{p_\theta(z|x)}\right)^{\alpha-1}\right]\right) \\ &= \frac{1}{\alpha-1}\log\left(E_{z\sim q_\phi(z|x)}\left[\left(\frac{p_\theta(z,x)}{q_\phi(z|x)\cdot p_\theta(x)}\right)^{1-\alpha}\right]\right) \\ &= \log p_\theta(x) + \frac{1}{\alpha-1}\log\left(E_{z\sim q_\phi(z|x)}\left[\left(\frac{p_\theta(z,x)}{q_\phi(z|x)}\right)^{1-\alpha}\right]\right) \end{aligned} \tag{10}$$

It follows that:

$$\log p_\theta(x) = D_\alpha(q_\phi(z|x)||p_\theta(z|x)) + \mathbf{VR}_\alpha \tag{11}$$

Definition 4. *Variational Rényi (VR) bound [3]:*

$$\mathbf{VR}_\alpha := \frac{1}{1-\alpha}\log\left(E_{z\sim q_\phi(z|x)}\left[\left(\frac{p_\theta(z,x)}{q_\phi(z|x)}\right)^{1-\alpha}\right]\right) \tag{12}$$

The variational Rényi (VR) bound can be extended for $\alpha < 0$ as well. Since $D_\alpha(p||q) \geq 0$ for $\alpha \geq 0$ and $D_\alpha(p||q) \leq 0$ for $\alpha \leq 0$ (see Figure 1) , then, for $\alpha \geq 0$, $\mathbf{VR}_\alpha$ is a lower bound for $\log p_\theta(x)$, and for $\alpha \leq 0$, $\mathbf{VR}_\alpha$ is an upper bound for $\log p_\theta(x)$.

2.3. χ Divergence

Similarly to the KL divergence and the Rényi divergence, one can use the χ^2-divergence (or in general the χ^n-divergence) and develop a bound of the log evidence [14].

Definition 5. *χ^2-divergence:*

$$D_{\chi^2}(p||q) = \mathbb{E}_q\left[\left(\frac{p(x)}{q(x)}\right)^2 - 1\right] \tag{13}$$

Now, our objective is to approximate the evidence $p_\theta(x)$ by using χ^2-divergence between the true posterior $p_\theta(z|x)$ and $q_\phi(z|x)$.

$$\begin{aligned} D_{\chi^2}(p_\theta(z|x)||q_\phi(z|x)) &= \mathbb{E}_{z\sim q_\phi(z|x)}\left[\left(\frac{p_\theta(z|x)}{q_\phi(z|x)}\right)^2 - 1\right] \\ &= \mathbb{E}_{z\sim q_\phi(z|x)}\left[\left(\frac{p_\theta(z,x)}{p_\theta(x)q_\phi(z|x)}\right)^2\right] - 1 \\ &= \frac{1}{p_\theta(x)^2}\mathbb{E}_{z\sim q_\phi(z|x)}\left[\left(\frac{p_\theta(z,x)}{q_\phi(z|x)}\right)^2\right] - 1 \end{aligned} \tag{14}$$

After rearranging the equation we will obtain:

$$\mathbb{E}_{z\sim q_\phi(z|x)}\left[\left(\frac{p_\theta(z,x)}{q_\phi(z|x)}\right)^2\right] = p_\theta(x)^2\left[1 + D_{\chi^2}(p_\theta(z|x)||q_\phi(z|x))\right] \tag{15}$$

Taking logarithms on both sides:

$$\begin{aligned}\log\left(\mathbb{E}_{z\sim q_\phi(z|x)}\left[\left(\frac{p_\theta(z,x)}{q_\phi(z|x)}\right)^2\right]\right) &= 2\log p_\theta(x) + \log\Big(\Big[1 + D_{\chi^2}(p_\theta(z|x)||q_\phi(z|x))\Big]\Big)\\ \log p_\theta(x) &= \frac{1}{2}\log\left(\mathbb{E}_{z\sim q_\phi(z|x)}\left[\left(\frac{p_\theta(z,x)}{q_\phi(z|x)}\right)^2\right]\right) - \frac{1}{2}\log\Big(\Big[1 + D_{\chi^2}(p_\theta(z|x)||q_\phi(z|x))\Big]\Big)\end{aligned} \tag{16}$$

By monotonicity of log and non-negativity of the χ^2-divergence, this quantity is an upper bound of the log evidence:

$$\log p_\theta(x) \leq \frac{1}{2}\log\left(\mathbb{E}_{z\sim q_\phi(z|x)}\left[\left(\frac{p_\theta(z,x)}{q_\phi(z|x)}\right)^2\right]\right) \tag{17}$$

Definition 6. *χ upper bound (CUBO):*

$$\mathbf{CUBO}_2 := \frac{1}{2}\log\left(\mathbb{E}_{z\sim q_\phi(z|x)}\left[\left(\frac{p_\theta(z,x)}{q_\phi(z|x)}\right)^2\right]\right) \tag{18}$$

$$\mathbf{CUBO}_n := \frac{1}{n}\log\left(\mathbb{E}_{z\sim q_\phi(z|x)}\left[\left(\frac{p_\theta(z,x)}{q_\phi(z|x)}\right)^n\right]\right) \tag{19}$$

Using χ^n-divergence for general n, $\mathbf{CUBO}_n$ provides a family of bounds. We note the strong connection between the $\mathbf{CUBO}_n$ and the Rényi bound $\mathbf{VR}_\alpha$: when $n = 1 - \alpha$, the VR bound is revealed.

Theorem 5. *(Sandwich Theorem [14]) For* $\mathbf{CUBO}_n$ *the following holds:*

1. $\forall n > 1 :$ $\mathbf{ELBO} \leq \log p_\theta(x) \leq \mathbf{CUBO}_n$.
2. $\forall n > 1 :$ $\mathbf{CUBO}_n$ *is a non-decreasing function of the n order χ-divergence.*
3. $\lim_{n\to 0} \mathbf{CUBO}_n = \mathbf{ELBO}$.

Using Theorem 5, one can estimate $\log p_\theta(x)$ with both upper and lower bounds, which may provide a better approximation for the log evidence.

The χ upper bound has many advantages: It is a black-box inference algorithm in that it does not need model-specific derivations and it is easy to apply to a wide class of models. In addition, it is useful when the KL divergence is not a good objective, and it is guaranteed to converge [14].

2.4. Monte Carlo Approximation

So far, we have discussed KL divergence, Rényi divergence, and χ divergence, and have demonstrated how each of these measurements can be used to construct a bound for the log evidence. However, calculating these bounds is computationally intractable, due to the stochastic nature of the latent space and the exponential number of random variables. In real-world situations, where datasets are typically limited and contain a finite number of data points, empirical estimations become necessary. A popular method for estimating these bounds is the Monte Carlo (MC) approximation [15,16]. Typically, the MC method involves random sampling from certain probability distributions.

The Monte Carlo (MC) approximation of the Kullback–Leibler (KL) divergence is unbiased, guaranteeing the convergence of the optimization process for the Evidence Lower Bound (ELBO). However, the MC approximation for the Rényi bound introduces bias, leading to an underestimation of the true expectation. In the case of positive values of α, this implies a relatively looser bound, but it should still be effective. On the other hand,

for negative values of α, this becomes a significant issue as it underestimates an upper bound. More precisely, the MC approximation for the Rényi bound is:

$$\widehat{\mathbf{VR}}_\alpha = \frac{1}{1-\alpha}\log\left(\frac{1}{K}\sum_{i=1}^{K}\left(\frac{p_\theta(z_i,x)}{q_\phi(z_i|x)}\right)^{1-\alpha}\right) \tag{20}$$

For this to be unbiased, the expectation should be equal to the true value,

$$E_{q_\phi}\left[\widehat{\mathbf{VR}}_\alpha\right] = \frac{1}{1-\alpha}E_{q_\phi}\left[\log\left(\frac{1}{K}\sum_{i=1}^{K}\left(\frac{p_\theta(z_i,x)}{q_\phi(z_i|x)}\right)^{1-\alpha}\right)\right] \tag{21}$$

By Jensen's inequality:

$$\begin{aligned}&\leq \frac{1}{1-\alpha}\log\left(E_{q_\phi}\left[\frac{1}{K}\sum_{i=1}^{K}\left(\frac{p_\theta(z_i,x)}{q_\phi(z_i|x)}\right)^{1-\alpha}\right]\right)\\ &= \mathbf{VR}_\alpha\end{aligned} \tag{22}$$

Thus, the approximation is actually an underestimate of the true bound. This characteristic was also discussed in [3], where the authors suggested improving the approximation quality by using more samples and using negative α values to improve the accuracy, at the cost of losing the upper-bound guarantee.

Other papers have suggested different approaches to keep the upper bounding property intact [8,14,17]. Of particular interest is the generic χ upper bound, $\mathbf{CUBO}_n$, which also suffers from the same problem of biased estimation using MC approximation. In [14], the authors suggested an approach to avoid the biased approximation, by exponentiation:

$$\mathbf{L} = e^{n\cdot\mathbf{CUBO}_n} \tag{23}$$

Applying MC approximation to $\mathbf{L}$ provides an unbiased upper bound. However, this change affects the variance of the gradients, which may damage the quality of the approximation result. It may result in high variance estimates and requires a large number of samples in order to serve as a reliable upper bound [18].

3. Improved VR Bound and Upper–Lower Bound Optimization

3.1. Variational Rényi Log Upper Bound (VRLU)

We suggest a different approach for estimating the upper bound while preserving the upper bound property. Consider the following inequalities:

$$1-\frac{1}{x} \leq \log x \leq x-1 \tag{24}$$

Where equality holds on both sides if and only if $x = 1$.

Definition 7. *Variational Rényi Log Upper bound (VRLU):*

$$\mathbf{VRLU}_\alpha := \frac{1}{1-\alpha}\left(E_{z\sim q_\phi(z|x)}\left[\left(\frac{p_\theta(z,x)}{q_\phi(z|x)}\right)^{1-\alpha}\right]-1\right) \tag{25}$$

$$\widehat{\mathbf{VRLU}}_\alpha := \frac{1}{1-\alpha}\left(\left(\frac{1}{K}\sum_{i=1}^{K}\left(\frac{p_\theta(z_i,x)}{q_\phi(z_i|x)}\right)^{1-\alpha}\right)-1\right) \tag{26}$$

For negative α, $\widehat{\mathbf{VRLU}}_\alpha$ is an estimation of the Rényi upper bound, and an upper bound of the log evidence:

$$
\begin{aligned}
E_{q_\phi}\left[\widehat{\mathbf{VRLU}}_\alpha\right] &= E_{q_\phi}\left[\frac{1}{1-\alpha}\left(\left(\frac{1}{K}\sum_{i=1}^{K}\left(\frac{p_\theta(z_i,x)}{q_\phi(z_i|x)}\right)^{1-\alpha}\right)-1\right)\right] \\
&\geq \frac{1}{1-\alpha}\log\left(E_{q_\phi}\left[\frac{1}{K}\sum_{i=1}^{K}\left(\frac{p_\theta(z_i,x)}{q_\phi(z_i|x)}\right)^{1-\alpha}\right]\right) \\
&= \frac{1}{1-\alpha}\log\left(E_{q_\phi}\left[\left(\frac{p_\theta(z_i,x)}{q_\phi(z_i|x)}\right)^{1-\alpha}\right]\right)
\end{aligned}
\tag{27}
$$

Note that the inequalities in (24) become tighter as the argument of the log is closer to 1. In the Rényi bound approximation (20), this argument is $1/k\sum(p_\theta(z,x)/q_\phi(z|x))^{1-\alpha}$. Thus, the approximation becomes tighter as the variational distribution, q_ϕ, is getting closer to the true distribution p_θ (the lower the divergence, the tighter the approximation), which is exactly the goal of the optimization.

We evaluated the bias of MC approximations for both bounds, $\mathbf{VR}_\alpha$ and $\mathbf{VRLU}_\alpha$, over a range of negative α values. To this end, we fixed the distributions p and q to both be Gaussian: $p \sim N(0,1)$, $q \sim N(1.5,1)$. The bounds $\mathbf{VR}_\alpha$ and $\mathbf{VRLU}_\alpha$ were estimated using the MC approximation (cf. (26) and (20)) and we evaluated the quality of the approximation for different values of MC samples, denoted by K.

Figure 2 shows the empirical results. We can see that the MC approximations for $\mathbf{VR}_\alpha$ are biased and get better as the sample size K increases. Furthermore, the bias results in an underestimation of $\mathbf{VR}_\alpha$ for $\alpha \leq 0$, which makes it unattractive to be used as an upper bound at the negative α range. On the other hand, the MC approximation for $\mathbf{VRLU}_\alpha$ preserves the upper bound property and has a relatively low variance. As a result, $\mathbf{VRLU}_\alpha$ is a more suitable choice as an upper bound for negative α and may be used as an objective for risk minimization.

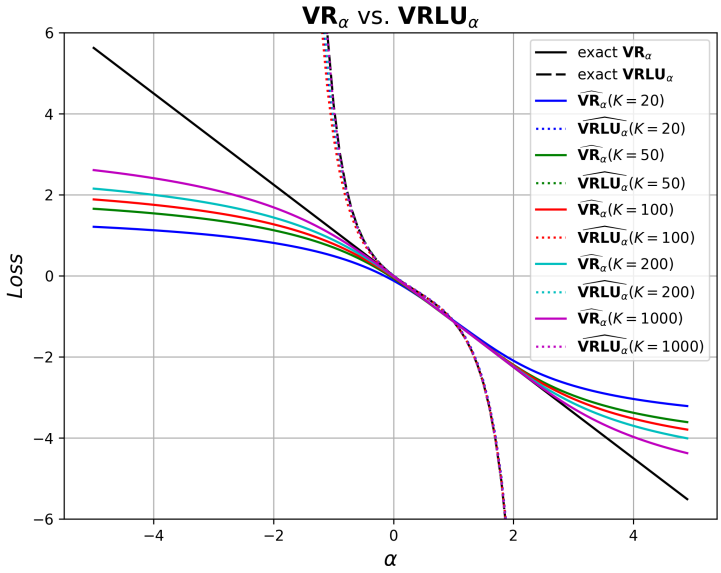

Figure 2. $\mathbf{VR}_\alpha$ and $\mathbf{VRLU}_\alpha$, vs. their Monte Carlo approximations with different number of samples K, over a range of α values, using fixed distributions: $p \sim N(0,1)$ and $q \sim N(1.5,1)$.

Figure 3 presents the comparison between $\mathbf{VR}_\alpha(p||q)$ and $\mathbf{VRLU}_\alpha(p||q)$ over different values of q. To this end, we fixed $p \sim N(1,2)$ and set $q \sim N(\mu,2)$ while varying μ in the range $[-5,10]$. We can see that as closer q is to p, both $\mathbf{VR}_\alpha(p||q)$ and $\mathbf{VRLU}_\alpha(p||q)$ values are decreasing, and for $p=q$, $\mathbf{VR}_\alpha(p||q) = \mathbf{VRLU}_\alpha(p||q) = 0$ for all α values. Furthermore, as α is farther away from 0, the steeper the graph becomes.

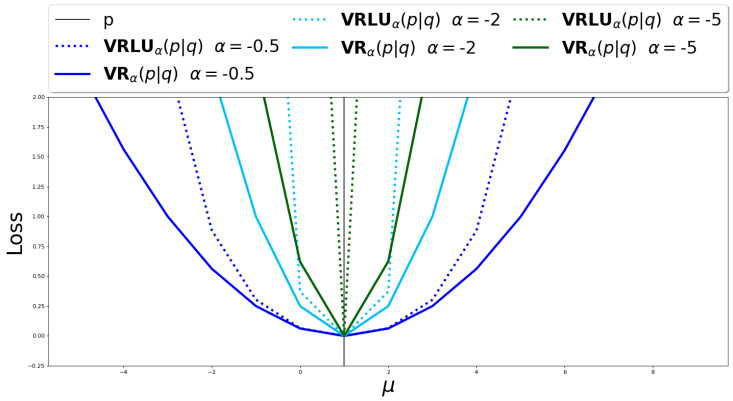

Figure 3. Comparison between $\mathbf{VR}_\alpha(p||q)$ and $\mathbf{VRLU}_\alpha(p||q)$ over different q distributions divergent from fixed distribution p. $p = N(1,2)$ and $q = N(\mu,2)$ where $-5 \leq \mu \leq 10$.

In conclusion, we empirically evaluated the $\mathbf{VRLU}_\alpha$ upper bound and matched it against the $\mathbf{VR}_\alpha$ upper bound, for varying values of negative α. The divergence curve of the $\mathbf{VRLU}_\alpha$ upper bound is steeper than the $\mathbf{VR}_\alpha$ upper bound, and the variance is much lower, suggesting a higher convergence as the variational distribution is getting closer to the true posterior.

3.2. Upper–Lower Bound Optimization

Using the new upper bound, $\mathbf{VRLU}_\alpha$, we devised $\mathbf{VRS}_{\alpha_+,\alpha_-}$; a (sandwiched) upper–lower bound variational inference algorithm for jointly minimizing the Rényi upper and lower bounds. $\mathbf{VRS}_{\alpha_+,\alpha_-}$ combined both the upper and lower Rényi bounds, where the lower bound $\mathbf{VR}_\alpha$ is computed as in Equation (20) for a constant positive α, and the upper bound $\mathbf{VRLU}_\alpha$ is computed as in Equation (26) for a constant negative α. The overall $\mathbf{VRS}_{\alpha_+,\alpha_-}$ loss is the average of both terms, i.e.,

$$\mathbf{VRS}_{\alpha_+,\alpha_-} := \frac{1}{2} \cdot (\mathbf{VRLU}_{\alpha_-} + \mathbf{VR}_{\alpha_+}) \tag{28}$$

$$\widehat{\mathbf{VRS}}_{\alpha_+,\alpha_-} = \frac{1}{2} \cdot \left(\widehat{\mathbf{VRLU}}_{\alpha_-} + \widehat{\mathbf{VR}}_{\alpha_+}\right) \tag{29}$$

Since $\mathbf{VR}_{\alpha_+} \leq \log p_\theta(x) \leq \mathbf{VR}_{\alpha_-} \leq \mathbf{VRLU}_{\alpha_-}$, the $\mathbf{VRS}_{\alpha_+,\alpha_-}$ loss provides a useful estimate for the log-likelihood of the evidence.

3.3. Probability Approximation

Recall that our objective is to develop a probabilistic model, denoted as p_θ, that effectively captures and explains the underlying data. In variational inference (VI), we tackle an optimization problem that seeks to find a simpler distribution that closely approximates the original data distribution, also known as the evidence. In this section, we will inspect the approximate distribution, denoted as $\hat{p_\theta}$, that minimizes the divergence $d_\alpha(\hat{p_\theta}||p_\theta)$. Our aim is to find an approximation that accurately represents the true data distribution.

We will evaluate two methods of approximating p_θ. One using VR bound:

$$\hat{p_\theta}(x) = \frac{e^{\mathbf{VR}_\alpha(x)}}{\sum_{x \in X} e^{\mathbf{VR}_\alpha(x)}} \tag{30}$$

and one using our VRS method:

$$\hat{p_\theta}(x) = \frac{e^{\mathbf{VRS}_{\alpha_+,\alpha_-}(x)}}{\sum_{x \in X} e^{\mathbf{VRS}_{\alpha_+,\alpha_-}(x)}} \tag{31}$$

We notice that for both estimators, $\hat{p_\theta}$ is indeed a probability. Given that both $\mathbf{VR}_\alpha$ and $\mathbf{VRS}_{\alpha_+,\alpha_-}$ estimate the log evidence, we will use the exponent of these estimates to approximate p_θ.

Let us denote $\alpha_+ > 0$. Using Equation (11),

$$\begin{aligned} e^{\mathbf{VR}_{\alpha_+}(x)} &= e^{\log p_\theta(x) - D_{\alpha_+}(q_\phi(z|x)||p_\theta(z|x))} \\ &= \frac{e^{\log p_\theta(x)}}{e^{D_{\alpha_+}(q_\phi(z|x)||p_\theta(z|x))}} \\ &= \frac{p_\theta(x)}{d_{\alpha_+}(q_\phi(z|x)||p_\theta(z|x))} \end{aligned} \tag{32}$$

Let us denote $\alpha_- < 0$.

$$\begin{aligned} e^{\mathbf{VR}_{\alpha_-}(x)} &= e^{\log p_\theta(x) - D_{\alpha_-}(q_\phi(z|x)||p_\theta(z|x))} \\ &= e^{\log p_\theta(x)} e^{-D_{\alpha_-}(q_\phi(z|x)||p_\theta(z|x))} \\ &= \frac{p_\theta(x)}{d_{\alpha_-}(q_\phi(z|x)||p_\theta(z|x))} \end{aligned} \tag{33}$$

Using both upper and lower bounds we will find that:

$$\begin{aligned} e^{\mathbf{VRS}_{\alpha_+,\alpha_-}(x)} &= e^{\frac{1}{2}(\mathbf{VR}_{\alpha_+}(x) + \mathbf{VR}_{\alpha_-}(x))} \\ &= \left(e^{\mathbf{VR}_{\alpha_+}(x)} e^{\mathbf{VR}_{\alpha_-}(x)}\right)^{\frac{1}{2}} \\ &= \sqrt{\frac{p_\theta(x)^2}{d_{\alpha_+}(q_\phi(z|x)||p_\theta(z|x)) d_{\alpha_-}(q_\phi(z|x)||p_\theta(z|x))}} \\ &= p_\theta(x) \sqrt{\frac{1}{d_{\alpha_+}(q_\phi(z|x)||p_\theta(z|x)) d_{\alpha_-}(q_\phi(z|x)||p_\theta(z|x))}} \end{aligned} \tag{34}$$

We will define multiplication factors for both our approximations as follows:

$$\mathbf{VRS}_{MF} := \frac{1}{\sqrt{d_{\alpha_+}(q_\phi(z|x)||p_\theta(z|x)) d_{\alpha_-}(q_\phi(z|x)||p_\theta(z|x))}} \tag{35}$$

$$\mathbf{VR}_{MF} := \frac{1}{d_{\alpha_+}(q_\phi(z|x)||p_\theta(z|x))} \tag{36}$$

Note that $e^{\mathbf{VRS}_{\alpha_+,\alpha_-}(x)} = p_\theta(x) \cdot \mathbf{VRS}_{MF}$ and $e^{\mathbf{VR}_\alpha(x)} = p_\theta(x) \cdot \mathbf{VR}_{MF}$. Thus, our goal is to achieve a multiplication factor as close to one as possible. We examine these values using the fixed distribution $p \sim N(0, 2)$, and distribution $q \sim N(\mu, 2)$, where $-3 < \mu < 3$. When $\mu = 0$, $p = q$. We used different α_+ and α_- values. The results are presented in Figure 4.

We can see that for every α_+, $\mathbf{VRS}_{MF}$ is closer to one for all different α_- values compared to $\mathbf{VR}_{MF}$ with the same α_+ value. In addition, when α_- and α_+ are symmetric around zero, the multiplication factor of $\mathbf{VRS}_{\alpha_+,\alpha_-}$ is closest to one. This indicates that the $\hat{p_\theta}(x)$ approximation calculated using $\mathbf{VRS}_{\alpha_+,\alpha_-}$ is more accurate among the two methods.

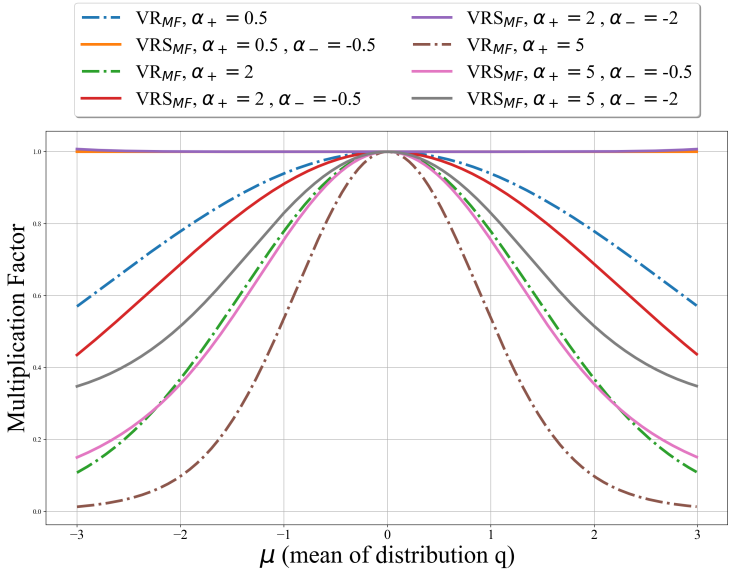

Figure 4. Comparison between $\mathbf{VR}_\alpha$ and $\mathbf{VRS}_{\alpha_+,\alpha_-}$ multiplication factors over fixed distribution p and different q distributions.

4. Multiple-Source Adaptation (MSA)

In statistical learning, there are numerous settings that require an accurate estimation of the data distribution to find effective solutions. One such task is known as domain adaptation. In the preceding section, we introduced VRS as an enhanced method to obtain accurate approximations of the data distribution. In this section, we will apply these estimated distributions to the domain adaptation objective, thus demonstrating the effectiveness and practicality of the VRS method to yield accurate solutions.

Domain adaptation is a scenario where we aim to train a classifier on one dataset (referred to as the source domain) for which labels or annotations are available and achieve good performance on another dataset (referred to as the target domain) for which labels or annotations are not available. A common example of a domain adaptation application is spam filtering, where a model trained on one user's emails (the source domain) is adapted and used to filter spam for a different user who receives distinct emails (the target domain).

In this work, our focus is on the Multi-Source Domain Adaptation (MSA) problem, where there are multiple source domains available in addition to only one target domain. The target domain can be considered as either an exact mixture of the source domains, or it might be well approximated by such a mixture. The goal is to leverage the information provided by the source domains to improve the performance on the target domain, where annotations or labels are not available.

In many real-world scenarios, the learner may not have access to all of the source data at once, due to privacy or storage constraints. Therefore, the learner cannot simply combine all of the source data together to train a predictor. A possible solution to this problem is the Mixture of Experts (MOE) approach. MOE is an ensemble learning technique that involves training multiple experts on different sub-tasks of a predictive modeling problem. Each expert concentrates on a specific part of the modeling problem space. A gating network then combines the outputs of the various experts. In the domain adaptation problem, this concept can be applied by modeling the domain relationship with an MOE approach.

The MSA problem was theoretically analyzed by Mansour, Mohri, and Rostamizadeh in [19]. In their paper, the authors presented the domain adaptation problem setup and proved that for any target domain, there exists a hypothesis, referred to as the distribution weighted combining rule, which is capable of achieving a low error rate with respect to the target domain. However, it should be noted that the authors did not provide a method for determining or learning the aforementioned hypothesis.

In the paper by Hoffman, Mohri and Zhang [9], the authors extended the definition of the weighted combination rule to solve probabilistic models as well, using cross-entropy loss. Additionally, the authors introduced an iterative algorithm based on Difference of Convex (DC) programming, that constructs the weighted combination rule. Nonetheless, the algorithm proposed in the paper assumes either prior knowledge of the probabilities associated with the data samples or relies on accurate estimates of these probabilities. The authors evaluated the performance of their model by employing the Rényi divergence, which quantifies the discrepancy between the true distribution and the approximated distribution. As a result, the effectiveness of their model is contingent upon the accuracy of the probability approximations as well.

In order to circumvent the need for good estimates of the data distribution, Cortes et al. [20] proposed a discriminative technique using an estimate of the conditional probabilities $p(i|x)$ for each source domain $i \in \{1, ..., k\}$ (that is, the probability that an instance x belongs to source i). To this end, they had to modify the DC algorithm proposed in [9], in order to adapt to their new distribution calculation.

In this study, we will build upon the algorithm introduced by Hoffman, Mohri, and Zhang [9], and enhance it with a refined approximation of the source distribution via variational inference.

4.1. MSA Problem Setup

We refer to a probability model where there is a distribution over the input space X. Each data point $x \in X$ has a corresponding label $y \in Y$, where Y denotes the space of labels. Our objective function describes the correspondence between the data point and its label $f : X \to Y$. We will focus on the adaptation problem with k source domains and a single target domain. For each domain $i \in \{1, ..., k\}$, we have a source distribution p_i and corresponding hypotheses $h_i(x, y) \to [0, 1]$. More precisely, h_i returns the probability that $f(x) = y$.

Definition 8. *Let $L : \mathbb{R} \to \mathbb{R}$ be a loss function penalizing errors with respect to f. The loss of hypothesis h with respect to the objective function f and a distribution p is denoted by $\mathcal{L}(h, p, f)$ and defined as:*

$$\mathcal{L}(h, p, f) := E_p[L(h, f)] = \sum_{x \in X} p(x) L(h(x, f(x))) \tag{37}$$

For simplicity, we will denote $L(h(x, f(x)))$ as $L(h, f)$ throughout this paper. We will assume that the following properties hold for the loss function L:

- L is non-negative: $L(x) \geq 0 \quad \forall x \in \mathbb{R}$
- L is convex.
- L is bounded: $\exists M \geq 0$ *s.t.* $\forall x \in \mathbb{R} : L(x) \leq M$.
- L is continuous in both arguments.
- L is symmetric.

Proposition 1. *For each domain i, the hypothesis h_i is a relatively accurate predictor for domain i with the distribution p_i; i.e., there exists $\epsilon > 0$ such that:*

$$\forall i \in \{1, ..., k\}, \quad \mathcal{L}(h_i, p_i, f) \leq \epsilon \tag{38}$$

Proposition 2. *We will denote the simplex: $\Delta = \{\lambda : \lambda_i \geq 0 \wedge \sum_{i=1}^{k} \lambda_i = 1\}$. The distribution of the target domain p_T is assumed to be a mixture of the k source distributions $p_1, ..., p_k$, that is:*

$$p_T(x) = \sum_{i=1}^{k} \lambda_i p_i(x) \quad (\textit{for } \lambda \in \Delta) \tag{39}$$

4.2. Existence of a Good Hypothesis

The goal of solving the MSA problem is to establish a good predictor (a good predictor: a predictor that provides a small error with respect to the target domain) for the target domain, given the source domain's predictors. A common assumption is that there exists some relationship between the target domain and the distributions of the source domains (See Proposition 2). It can be demonstrated that conventional convex combinations of source predictors may yield suboptimal results in certain scenarios. In particular, studies have indicated that even if the source predictors possess zero loss, no convex combination can attain a loss lower than a specific constant for a uniform mixture of the source distributions.

Alternately, Mansour, Mohri, and Rostamizadeh [19] proposed a distribution-weighted solution and defined the distribution-weighted combination hypothesis for a regression model. Hoffman and Mohri [9] extended the distribution-weighted combination hypothesis to a probabilistic model, as follows:

Definition 9. *Distribution-weighted combination hypothesis.*
For any $\lambda \in \Delta, \eta > 0$ *and* $(x, y) \in X \times Y$*:*

$$h_w^{\eta}(x, y) = \sum_{i=1}^{k} \frac{w_i p_i(x) + \eta \frac{U(x)}{k}}{\sum_{j=1}^{k}(w_j p_j(x)) + \eta U(x)} h_i(x, y) \tag{40}$$

where $U(x)$ *is the uniform distribution over* X*.*

In the probabilistic model case, we will use L as the binary cross entropy loss:

$$L(h, f) = -\log h(x, f(x)) \tag{41}$$

which maintain all of the required properties stated in Section 4.1.

Theorem 6. *For any target function* $f \in \{f : \forall i \in \{1, ..., k\}, \mathcal{L}(h_i, p_i, f) \leq \epsilon\}$ *and for any* $\delta > 0$*, there exist* $\eta > 0$ *and* $w \in \Delta$ *such that* $\mathcal{L}(h_w^{\eta}, p_\lambda, f) \leq \epsilon + \delta$ *for any mixture parameter* λ*.*

The proof of Theorem 6 is detailed in [19]. From this Theorem, it can be inferred that for any fixed target function f, the distribution-weighted combination hypothesis is a good hypothesis for the target domain.

4.3. A Good Hypothesis with Estimated Probabilities

On closer inspection of Definition 9, it is evident that constructing h_w^{η} requires access to the distributions of all domains, represented by $p_i(x) \ \ \forall i \in 1, ..., k$. Yet, in practical settings, the true distributions p_i may not be directly available to the learner. Instead, the learner relies on estimates $\hat{p}_i$ derived from the available data. Thus, addressing the application of domain adaptation becomes essential for real-world scenarios where the true distributions remain unknown.

Our objective is to minimize the value of $\mathcal{L}(h_i, \hat{p}_i, f)$. To accomplish this, we will develop an upper bound for this loss function (similar to previous research [9,21]). By doing so, we can examine the impact of utilizing estimated distributions $\hat{p}_i$ on the efficacy of our model and gain insights into the application of domain adaptation in real-world scenarios. First, let us recall Holder's inequality:

Theorem 7. *Holder's inequality: For any* s *and* t *in the open interval* $(1, \infty)$ *with* $\frac{1}{s} + \frac{1}{t} = 1$*, and for* $\{x_j\}$ *and* $\{y_j\}$ $j \in \{1, ..., k\}$ *be certain sets of real numbers, we have:*

$$\sum_{j=1}^{n} |x_j y_j| \leq \left(\sum_{j=1}^{n} |x_j|^s \right)^{\frac{1}{s}} \left(\sum_{j=1}^{n} |y_j|^t \right)^{\frac{1}{t}} \tag{42}$$

Corollary 1. *Let $\hat{p}_i$ be an estimation of the original domain distribution p_i. The following inequality holds for any $\alpha > 1$:*

$$\mathcal{L}(h_i, \hat{p}_i, f) \leq (d_\alpha(\hat{p}_i||p_i)\epsilon)^{\frac{\alpha-1}{\alpha}} M^{\frac{1}{\alpha}} \tag{43}$$

Proof of Corollary 1. For any hypothesis h and any distributions p, q, and for any $\alpha > 1$, the following holds (the proof is based on a similar corollary proven in [9]):

$$\begin{aligned}
\mathcal{L}(h,q,f) &= \sum_{x\in X} q(x)L(h,f) \\
&= \sum_{x\in X} \left(\frac{q(x)}{p(x)^{\frac{\alpha-1}{\alpha}}}\right) p(x)^{\frac{\alpha-1}{\alpha}} L(h,f) \\
&\leq \left(\sum_{x\in X} \frac{q(x)^\alpha}{p(x)^{\alpha-1}}\right)^{\frac{1}{\alpha}} \left(\sum_{x\in X} p(x)L(h,f)^{\frac{\alpha}{\alpha-1}}\right)^{\frac{\alpha-1}{\alpha}} && \text{By Holder's inequality for } s=\alpha, \text{ and } t=\frac{\alpha}{\alpha-1} \\
&= \left(\left(\sum_{x\in X} \frac{q(x)^\alpha}{p(x)^{\alpha-1}}\right)^{\frac{1}{\alpha-1}}\right)^{\frac{\alpha-1}{\alpha}} \left(\sum_{x\in X} p(x)L(h,f)L(h,f)^{\frac{1}{\alpha-1}}\right)^{\frac{\alpha-1}{\alpha}} \\
&= (d_\alpha(q||p))^{\frac{\alpha-1}{\alpha}} \left(\sum_{x\in X} p(x)L(h,f)L(h,f)^{\frac{1}{\alpha-1}}\right)^{\frac{\alpha-1}{\alpha}} && \text{By Definition 3} \\
&\leq (d_\alpha(q||p))^{\frac{\alpha-1}{\alpha}} \left(\sum_{x\in X} p(x)L(h,f)M^{\frac{1}{\alpha-1}}\right)^{\frac{\alpha-1}{\alpha}} && \text{Since } M \geq |L(h,f)| \text{ and } \frac{1}{\alpha-1} > 0 \\
&= (d_\alpha(q||p)\mathcal{L}(h,p,f))^{\frac{\alpha-1}{\alpha}} M^{\frac{1}{\alpha}}
\end{aligned}$$

For each $i \in \{1, ..., k\}$, by setting $p := p_i, q := \hat{p}_i$ and $h := h_i$, we will find that:

$$\begin{aligned}
\mathcal{L}(h_i, \hat{p}_i, f) &\leq (d_\alpha(\hat{p}_i||p_i)\mathcal{L}(h_i, p_i, f))^{\frac{\alpha-1}{\alpha}} M^{\frac{1}{\alpha}} \\
&\leq (d_\alpha(\hat{p}_i||p_i)\epsilon)^{\frac{\alpha-1}{\alpha}} M^{\frac{1}{\alpha}} && \text{By Proposition 1}
\end{aligned}$$

□

Corollary 1 provides us an upper bound of the loss using the estimated distributions $\hat{p}_i$. When $\hat{p}_i \to p_i$, $d_\alpha(\hat{p}_i||p_i) \to 1$ and we will remain with $\epsilon^{\frac{\alpha-1}{\alpha}} M^{\frac{1}{\alpha}}$. We will set $M = 1$, since we use the loss function $L(h,f) = -\log(h(x, f(x)))$ as the cross-entropy loss (log-loss). Thus, when $\hat{p}_i \to p_i$, $(d_\alpha(\hat{p}_i||p_i)\epsilon)^{\frac{\alpha-1}{\alpha}} M^{\frac{1}{\alpha}} \to \epsilon^{\frac{\alpha-1}{\alpha}}$.

By performing the aforementioned calculation with $\alpha < 1$, it is possible to derive a lower bound for $\mathcal{L}(h_i, \hat{p}_i, f)$. This lower bound serves as a confirmation that the utilization of approximated probabilities does not lead to significant errors. For instance, if the lower bound exhibits a considerably large value, it indicates that our approximation is inadequate. Conversely, if the lower bound demonstrates a small value, it signifies the effectiveness of

our approximation. Moreover, by employing both upper and lower bounds, we can obtain a more precise estimation of the loss.

Theorem 8. *Generalization of Holder's inequality [22]: Let* $0 < s < 1$ *and* $t \in \mathbb{R}$ *with* $\frac{1}{s} + \frac{1}{t} = 1$, *and for* $\{x_j\}$ *and* $\{y_j\}$ $j \in \{1, ..., n\}$ *be certain sets of real numbers, we have:*

$$\sum_{j=1}^{n} |x_j y_j| \geq \left(\sum_{j=1}^{n} |x_j|^s \right)^{\frac{1}{s}} \left(\sum_{j=1}^{n} |y_j|^t \right)^{\frac{1}{t}} \tag{44}$$

Corollary 2. *Let* $\hat{p}_i$ *be an estimation of the original domain distribution* p_i. *The following inequality holds for any* $\alpha < 1$:

$$\mathcal{L}(h_i, \hat{p}_i, f) \geq (d_\alpha(\hat{p}_i || p_i))^{\frac{\alpha-1}{\alpha}} \psi \tag{45}$$

where $\psi = \left(\sum_{x \in X} p_i(x) L(h_i, f)^{\frac{\alpha}{\alpha-1}} \right)^{\frac{\alpha-1}{\alpha}}$

Proof of Corollary 2. First, we will prove for $0 < \alpha < 1$, and then for $\alpha < 0$. Let us set $0 < \alpha < 1$, $s = \alpha$ and $t = \frac{\alpha}{\alpha-1}$. For any hypothesis h and any distributions p, q, the following holds:

$$\begin{aligned}
\mathcal{L}(h, q, f) &= \sum_{x \in X} q(x) L(h, f) \\
&= \sum_{x \in X} \left(\frac{q(x)}{p(x)^{\frac{\alpha-1}{\alpha}}} \right) p(x)^{\frac{\alpha-1}{\alpha}} L(h, f) \\
&\geq \left(\sum_{x \in X} \frac{q(x)^\alpha}{p(x)^{\alpha-1}} \right)^{\frac{1}{\alpha}} \left(\sum_{x \in X} p(x) L(h, f)^{\frac{\alpha}{\alpha-1}} \right)^{\frac{\alpha-1}{\alpha}} && \text{By the generalization of Holder's inequality for } s = \alpha, t = \frac{\alpha}{\alpha-1} \\
&= \left(\left(\sum_{x \in X} \frac{q(x)^\alpha}{p(x)^{\alpha-1}} \right)^{\frac{1}{\alpha-1}} \right)^{\frac{\alpha-1}{\alpha}} \left(\sum_{x \in X} p(x) L(h, f)^{\frac{\alpha}{\alpha-1}} \right)^{\frac{\alpha-1}{\alpha}} \\
&= (d_\alpha(q||p))^{\frac{\alpha-1}{\alpha}} \left(\sum_{x \in X} p(x) L(h, f)^{\frac{\alpha}{\alpha-1}} \right)^{\frac{\alpha-1}{\alpha}} && \text{By Definition 3}
\end{aligned}$$

Next, let us set $\alpha < 0$, $t = \alpha$ and $s = \frac{\alpha}{\alpha-1}$ (notice that $\alpha < 0 \rightarrow 0 < s < 1$).
For any hypothesis h and any distributions p, q, the following holds:

$$\begin{aligned}
\mathcal{L}(h, q, f) &= \sum_{x \in X} q(x) L(h, f) \\
&= \sum_{x \in X} \left(\frac{q(x)}{p(x)^{\frac{\alpha-1}{\alpha}}} \right) p(x)^{\frac{\alpha-1}{\alpha}} L(h, f) \\
&= \sum_{x \in X} p(x)^{\frac{\alpha-1}{\alpha}} L(h, f) \left(\frac{q(x)}{p(x)^{\frac{\alpha-1}{\alpha}}} \right) \\
&\geq \left(\sum_{x \in X} p(x) L(h, f)^{\frac{\alpha}{\alpha-1}} \right)^{\frac{\alpha-1}{\alpha}} \left(\sum_{x \in X} \frac{q(x)^\alpha}{p(x)^{\alpha-1}} \right)^{\frac{1}{\alpha}} && \text{By the generalization of Holder's inequality for } t = \alpha, s = \frac{\alpha}{\alpha-1}
\end{aligned}$$

$$= \left(\sum_{x\in X} \frac{q(x)^{\alpha}}{p(x)^{\alpha-1}}\right)^{\frac{1}{\alpha}} \left(\sum_{x\in X} p(x)L(h,f)^{\frac{\alpha}{\alpha-1}}\right)^{\frac{\alpha-1}{\alpha}}$$

$$= \left(\left(\sum_{x\in X} \frac{q(x)^{\alpha}}{p(x)^{\alpha-1}}\right)^{\frac{1}{\alpha-1}}\right)^{\frac{\alpha-1}{\alpha}} \left(\sum_{x\in X} p(x)L(h,f)^{\frac{\alpha}{\alpha-1}}\right)^{\frac{\alpha-1}{\alpha}}$$

$$= (d_{\alpha}(q||p))^{\frac{\alpha-1}{\alpha}} \left(\sum_{x\in X} p(x)L(h,f)^{\frac{\alpha}{\alpha-1}}\right)^{\frac{\alpha-1}{\alpha}} \quad \textit{By Definition 3}$$

For each $i \in \{1, ..., k\}$, by setting $p := p_i, q := \hat{p}_i$ and $h := h_i$, we will find that:

$$\mathcal{L}(h_i, \hat{p}_i, f) \geq (d_{\alpha}(\hat{p}_i||p_i))^{\frac{\alpha-1}{\alpha}} \left(\sum_{x\in X} p_i(x)L(h_i,f)^{\frac{\alpha}{\alpha-1}}\right)^{\frac{\alpha-1}{\alpha}} = (d_{\alpha}(\hat{p}_i||p_i))^{\frac{\alpha-1}{\alpha}} \psi$$

□

We contend that the value of $\psi = \left(\sum_{x\in X} p_i(x)L(h_i,f)^{\frac{\alpha}{\alpha-1}}\right)^{\frac{\alpha-1}{\alpha}}$ can be disregarded when examining the loss bound. As previously mentioned, we assume that $L(h_i, f) \leq M$, where we have set $M = 1$. Consequently, we are left with $\left(\sum_{x\in X} p_i(x)\right)^{\frac{\alpha-1}{\alpha}}$. Since p_i is a distribution, the sum equals 1.

Let us set $\mathcal{L}_{\alpha}(\hat{p}, p) := (d_{\alpha}(\hat{p}||p))^{\frac{\alpha-1}{\alpha}}$. We would like to present an example of different $\mathcal{L}_{\alpha}(\hat{p}, p)$ values calculated with a constant distribution $p \sim N(3, 10)$, and a distribution $\hat{p} \sim N(\mu, 10)$, where $0 < \mu < 6$. When $\mu = 3$, $p = \hat{p}$. The results are shown in Figure 5.

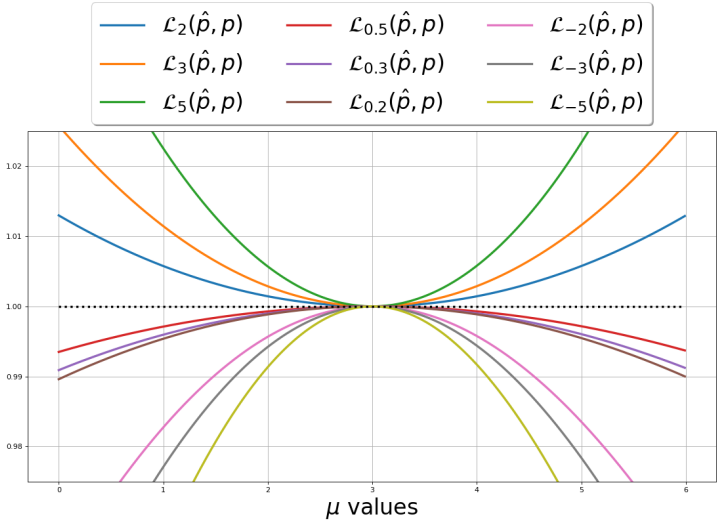

Figure 5. Comparison between $\mathcal{L}_{\alpha}(\hat{p}, p)$ with different α values over fixed distribution $p \sim N(3, 10)$, and distribution $\hat{p} \sim N(\mu, 10)$, where $0 < \mu < 6$.

As we can observe, as the estimated distribution $\hat{p}$ approaches the true distribution p (i.e., as μ approaches 3), the bounds on the loss function become increasingly similar. We can also see that the value of the lower bounds is not significantly large, which means that we can consider using the probability approximation to solve the MSA problem. It is also worth noting that when α deviates significantly from 1, the bounds move away from the actual value.

Theorem 9. *Let p_T be an arbitrary target distribution. For any $\delta > 0$, there exists $\eta > 0$ and $w \in \Delta$, such that the following inequality holds for any $\alpha > 1$ and any mixture parameter λ:*

$$\mathcal{L}(h_w^\eta, p_T, f) \leq ((\epsilon + \delta) d_\alpha(p_T||p_\lambda))^{\frac{\alpha-1}{\alpha}} M^{\frac{1}{\alpha}} \tag{46}$$

Proof of Theorem 9. Let $\delta > 0$. In the proof for Corollary 1, we showed that for any hypothesis h and any distributions p, q, and for any $\alpha > 1$, the following holds:

$$\mathcal{L}(h, q, f) \leq (d_\alpha(q||p)\mathcal{L}(h, p, f))^{\frac{\alpha-1}{\alpha}} M^{\frac{1}{\alpha}} \tag{47}$$

Hence, for $q = p_T$, $p = p_\lambda$ and $h = h_w^\eta$ we will find that:

$$\mathcal{L}(h_w^\eta, p_T, f) \leq \left(d_\alpha(p_T||p_\lambda)\mathcal{L}(h_w^\eta, p_\lambda, f)\right)^{\frac{\alpha-1}{\alpha}} M^{\frac{1}{\alpha}} \tag{48}$$

By Theorem 6, given $\delta > 0$, there exist $\eta > 0$ and $w \in \Delta$ such that $\mathcal{L}(h_w^\eta, p_\lambda, f) \leq \epsilon + \delta$ for any mixture parameter λ. Therefore:

$$\mathcal{L}(h_w^\eta, p_T, f) \leq (d_\alpha(p_T||p_\lambda)(\epsilon + \delta))^{\frac{\alpha-1}{\alpha}} M^{\frac{1}{\alpha}} \tag{49}$$

□

Corollary 3. *Let p_T be an arbitrary target distribution. For any $\delta > 0$, there exists $\eta > 0$ and $w \in \Delta$, such that the following inequality holds for any $\alpha > 1$ and any mixture parameter $\lambda \in \Delta$:*

$$\mathcal{L}(\hat{h}_w^\eta, p_T, f) \leq ((\epsilon^* + \delta) d_\alpha(p_T||\hat{p}_\lambda))^{\frac{\alpha-1}{\alpha}} M^{\frac{1}{\alpha}} \tag{50}$$

where $\hat{p}_\lambda = \sum_{i=1}^k \lambda_i \hat{p}_i(x)$ and $\hat{h}_w^\eta$ is our good hypothesis from Definition 9 but calculated with the estimated probabilities $\hat{p}_i$.

Proof of Corollary 3. By Corollary 1, $\forall i \in \{1, ..., k\}$ and for any $\alpha > 1$: $\mathcal{L}(h_i, \hat{p}_i, f) \leq (d_\alpha(\hat{p}_i||p_i)\epsilon)^{\frac{\alpha-1}{\alpha}} M^{\frac{1}{\alpha}}$. Let us set ϵ^* such that: $\epsilon^* = \max_{i=1}^k \{(d_\alpha(\hat{p}_i||p_i)\epsilon)^{\frac{\alpha-1}{\alpha}} M^{\frac{1}{\alpha}}\}$. Overall, we obtained the following:

- For every $i \in \{1, ..., k\}$: $\mathcal{L}(h_i, \hat{p}_i, f) \leq \epsilon^*$.
- $\hat{h}_w^\eta(x, y) = \sum_{i=1}^k \frac{w_i \hat{p}_i(x) + \eta \frac{U(x)}{k}}{\sum_{j=1}^k (w_j \hat{p}_j(x)) + \eta U(x)} h_i(x, y)$.

We can repeat the proof of Theorem 9 with ϵ^* instead of ϵ, $\hat{p}_i$ instead of p_i and $\hat{h}_w^\eta$ instead of h_w^η. □

In summary, we demonstrated that it is possible to use approximate distributions to calculate a good distribution-weighted combining rule. We have established that the error introduced by using estimated distributions is bounded. Thus, we can address the Multi-Source Adaptation (MSA) problem in real-world applications.

4.4. MSA Algorithm

Alongside the unknown probabilities, another crucial aspect is determining an appropriate vector of weights, denoted as w, to fully establish the distribution-weighted combining rule. The paper by Hoffman, Mohri, and Zhang [9] presents a new algorithm for determining the distribution-weighted combination solution for cross-entropy loss and other losses, based on Difference of Convex (DC) programming.

Lemma 2. *For any target function $f \in \mathcal{F}$ and any $\eta, \eta' \geq 0$, there exists $w \in \Delta$ with $w_i \neq 0$ for all $i \in \{1, ..., k\}$, such that the following holds:*

$$\forall i \in \{1, ..., k\} \quad \mathcal{L}(h_w^\eta, p_i, f) \leq \gamma + \eta' \tag{51}$$

where:$\gamma = \sum_{j=1}^{k} w_j \mathcal{L}(h_w^\eta, p_j, f)$

The proof of Lemma 2 is detailed in [19].

Corollary 4. *For any target function* $f \in \mathcal{F}$ *and any* $\eta' \geq 0$*, there exists* $w \in \Delta$ *with* $w_i \neq 0$ *for all* $i \in \{1, ..., k\}$*, such that the following holds:*

$$\mathcal{L}(h_w^\eta, p_i, f) \leq \mathcal{L}(h_w^\eta, p_w, f) + \eta' \quad \forall i \in \{1, ..., k\} \tag{52}$$

Proof of Corollary 4. By Lemma 2, we obtain:

$$\begin{aligned} \forall i \in \{1, ..., k\} \;\; \mathcal{L}(h_w^\eta, p_i, f) &\leq \gamma + \eta' \\ &= \sum_{j=1}^{k} w_j \mathcal{L}(h_w^\eta, p_j, f) + \eta' \\ &= \mathcal{L}(h_w^\eta, p_w, f) + \eta' \end{aligned}$$

□

Corollary 4 provides a single upper bound for the loss with respect to every p_i. Thus, our problem consists of finding a parameter w verifying this property. This, in turn, can be formulated as the following optimization problem:

$$\min_{w \in \Delta, \rho \in \mathbb{R}} \;\; \rho \;\; s.t. \;\; \mathcal{L}(h_w^\eta, p_i, f) - \mathcal{L}(h_w^\eta, p_w, f) \leq \rho \;\; \forall i \in \{1, ..., k\} \tag{53}$$

Definition 10. *DC Function [23]: Let* C *be a convex subset of* $\mathbb{R}^n$*. A real-valued function* $f : C \to \mathbb{R}$ *is called DC on* C*, if there exist two convex functions* $g, h : C \to \mathbb{R}$ *such that* f *can be expressed in the form:*

$$f(x) = g(x) - h(x) \tag{54}$$

DC programming problems are programming problems dealing with DC functions. An important class of DC problems is the following:

$$w^* = \inf\{g(x) - h(x) \;\; : x \in X\} \tag{55}$$

where $g$and h are two convex functions in $\mathbb{R}^n$, and X is a closed convex subset of $\mathbb{R}^n$.

Proposition 3. *Assume that the problem* w^* *is solvable. Then, a point* $x^* \in X$ *is an optimal solution to* w^* *if and only if there is* $t^* \in \mathbb{R}$*, such that:*

$$0 = \inf\{-h(x) + t \;\; : x \in X \;\; , t \in \mathbb{R} \;\; , g(x) - t \leq g(x^*) - t^*\} \tag{56}$$

Horst and Thoai [23] developed an algorithm for solving DC programming problems such as w^* based on the above optimality condition. The assumptions in Proposition 3 apply to the MSA problem, since we know there is an optimal solution. The key lies in identifying two convex functions whose difference coincides with the solution of the MSA problem. Let us define the following functions:

$$\begin{aligned} J_w(x, y) &= \sum_{i=1}^{k} w_i p_i(x) h_i(x, y) + \frac{\eta}{k} U(x) h_i(x, y) \\ K_w(x) &= p_w(x) + \eta U(x) \end{aligned} \tag{57}$$

Note that: $h_w^\eta(x, y) = \frac{J_w(x,y)}{K_w(x)}$.

Let us define the following convex functions:

$$u_i(w,f) = -\sum_x [p_i(x) + \eta U(x)] \log(J_w(x, f(x))) \tag{58}$$

$$v_i(w,f) = \sum_x K_w(x) \log(\frac{K_w(x)}{J_w(x, f(x))}) - [p_i(x) + \eta U(x)] \log(K_w(x)) \tag{59}$$

$u_i(w,f)$ is convex since $-\log(J_w)$ is convex as a composition of the convex function $-\log$ with an affine function J_w. Similarly, $-\log(K_w)$ is convex, which shows that the second term in the expression of $v_i(w,f)$ is a convex function. The first term can be written in terms of the unnormalized relative entropy (the unnormalized relative entropy of P and Q is defined by: $B(p||q) = \sum_x p(x) \log\left(\frac{p(x)}{q(x)}\right) + \sum_x (q(x) - p(x))$).It can be shown that the relative entropy is jointly convex using the so-called log-sum inequality (based on the explanation in [9]).

Let us be reminded of our regression loss function:

$$\mathcal{L}(h,p,f) := E_{x\sim p}[L(h,f)] = \sum_{x\in X} L(h,f)p(x) \tag{60}$$

$$L(h,f) := -\log h(x, f(x)) \tag{61}$$

Proposition 4. *Let L be the cross-entropy loss. Then, for* $i \in \{1, ..., k\}$

$$\mathcal{L}(h_w^\eta, p_i, f) - \mathcal{L}(h_w^\eta, p_w, f) = u_i(w,f) - v_i(w,f) \tag{62}$$

Proof of Proposition 4.

$$\begin{aligned}
&\mathcal{L}(h_w^\eta, p_i, f) - \mathcal{L}(h_w^\eta, p_w, f) \\
&= \sum_{x\in X} L(h_w^\eta, f)p_i(x) - \sum_{x\in X} L(h_w^\eta, f)p_w(x) \\
&= \sum_{x\in X} (p_i(x) - p_w(x))L(h_w^\eta, f) \\
&= \sum_{x\in X} (p_i(x) - p_w(x))\Big(-\log(h_w^\eta(x, f(x)))\Big) && \textit{L is the cross entropy loss.} \\
&= \sum_{x\in X} (p_i(x) - p_w(x))\left(-\log(\frac{J_w(x, f(x))}{K_w(x)})\right) && h_w^\eta(x,y) = \frac{J_w(x,y)}{K_w(x)} \\
&= \sum_{x\in X} (p_i(x) - K_w(x) + \eta U(x))\left(-\log(\frac{J_w(x, f(x))}{K_w(x)})\right) && K_w(x) = p_w(x) + \eta U(x) \\
&= \sum_{x\in X} K_w(x)\left(\log(\frac{J_w(x, f(x))}{K_w(x)})\right) \\
&\quad - \sum_{x\in X} (p_i(x) + \eta U(x))\left(\log(\frac{J_w(x, f(x))}{K_w(x)})\right) \\
&= -\sum_{x\in X} K_w(x)\left(\log(\frac{K_w(x)}{J_w(x, f(x))})\right) \\
&\quad - \sum_{x\in X} (p_i(x) + \eta U(x))(\log(J_w(x, f(x))) \\
&\quad + \sum_{x\in X} (p_i(x) + \eta U(x))(\log(K_w(x))) \\
&= u_i(w,f) - v_i(w,f) && \textit{By Equation (58)}
\end{aligned}$$

□

Using the proof above, our optimization problem

$$\min_{w\in\Delta,\rho\in\mathbb{R}} \ \rho \ \ s.t. \ \ \mathcal{L}(h_w^\eta, p_i, f) - \mathcal{L}(h_w^\eta, p_w, f) \leq \rho \ \ \forall i \in \{1, ..., k\}$$

is a DC programming problem, since it is the difference between two convex functions. In light of all of the above, our optimization problem can be cast as the following variational form of a DC-programming problem: let us set (w_t) to be the sequence defined by repeatedly solving the following convex optimization problem:

- Target function: $min\ \rho$.
- Constraints:
  1. $u_i(w, f) - v_i(w, f) - (w - w_t)\nabla v_i(w_t, f) \leq \rho$
  2. $\sum_{i=1}^{k} w_i - 1 = 0$
  3. $-w_i \leq 0 \quad \forall i \in \{1, ..., k\}$

where $w_0 \in \Delta$ is an arbitrary starting value. Then, (w_t) is guaranteed to converge to a *local minimum* of the optimization problem [9].

Given the fact that an optimal hypothesis h_w^η exists, we converted the MSA problem into an optimization problem and cast it to a DC programming form in order to find a local optimum. This way, we are able to find the parameter w which is used in the distribution-weighted combination rule.

5. Empirical Results

In this section we present two sets of experiments. The first set is designed to evaluate the accuracy of approximating distributions using the VRLU and VRS methods, and the second set demonstrates the application of these estimates for the MSA problem.

5.1. VRLU and VRS Experiments

We present a series of experiments conducted to evaluate the performance of **VRLU**α and **VRS**α_+, α_- and compare them to the performance of existing methods such as the Evidence Lower Bound (ELBO) and Rényi upper and lower bounds. The goal of these experiments is to assess the effectiveness of the proposed methods and to determine their advantages and limitations. The methods we will examine in this section are detailed below:

- **VAE**—minimizing KL divergence—maximizing the ELBO.
- **VR**—minimizing Rényi divergence using variational Rényi upper / lower bound with MC, for different values of α.
- **VRLU**—minimizing Rényi divergence using our variational Rényi log upper bound with MC for different values of negative α.
- **VRS**—minimizing Rényi divergence using the (sandwich) upper-lower bound with MC for different values of negative and positive α.

All of our experiments were conducted using PyTorch. Throughout the experiments, we used K = 50 samples for Monte Carlo (MC) approximation; trained the VAE models using the ADAM optimizer [24]; and set the learning rate to 0.001 and the batch size to 128 for the training set, and 32 for the test set. Our VAE model includes a total of 6 linear layers. The first 3 are the encoder layers, and the last 3 are the decoder layers. The dimension of the latent space is 50. We suggest two perspectives to evaluate and compare performances:

- *Quality of the decoded signal*—Reconstruction error, measured by Mean Square Error (MSE) and Cross-Entropy (CE).
- *Quality of the evidence approximation*—Maximizing the evidence log-likelihood, $\log p(x)$; the higher the better.

5.1.1. Digits Experiment

In the following experiment, we used the 'MNIST', 'USPS', and 'SVHN' datasets, all of which contain digit images (See Figure 6). They all share 10 classes of digits. The 'USPS' dataset consists of 7291 training images and 2007 test images of size 16×16. The 'MNIST' dataset consists of 60,000 training images and 10,000 test images of size 28×28. 'SVHN'

is obtained from house numbers in Google Street View images. It has 73,257 training images and 26,032 test images of size 32×32. If we look at Figure 6, we can see that the graphical representation of digits in 'USPS', 'SVHN', and 'MNIST' is very diverse; hence, each domain has a very different distribution.

Figure 6. Digits datasets visualization.

We compared the learning curves of $\mathbf{VRS}_{\alpha_+,\alpha_-}$ with $\alpha_- \in \{-0.5, -2\}$ and $\alpha_+ \in \{0.5, 2\}$ and $\mathbf{VR}_\alpha$ with $\alpha \in \{0.5, 2, 5\}$ over the 'MNIST' dataset. Figure 7 demonstrates that $\mathbf{VRS}_{\alpha_+,\alpha_-}$ converged faster than $\mathbf{VR}_\alpha$ and the resulting loss value is smaller for both α values. Also, we can see that $\mathbf{VR}_{0.5}$ performs better than $\mathbf{VR}_2$, and $\mathbf{VR}_2$ performs better than $\mathbf{VR}_5$. This observation is in sync with the results reported in [3].

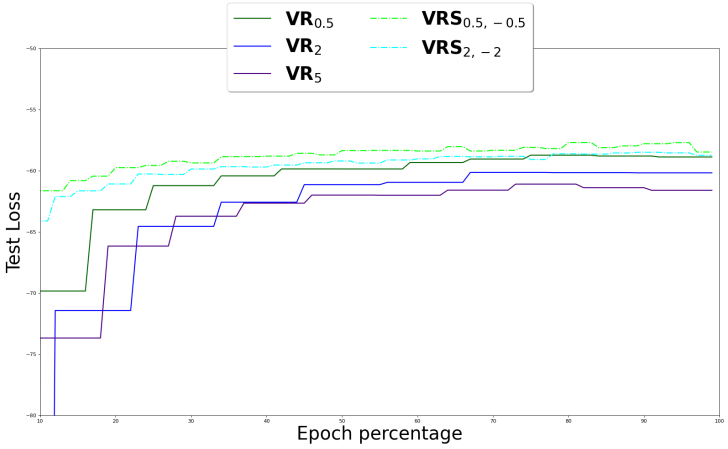

Figure 7. Comparison between $\mathbf{VR}_\alpha$ and $\mathbf{VRS}_{\alpha_+,\alpha_-}$ learning curves over 'MNIST' dataset. Training with different values of α. The y axis detailed the values of the VR and VRS bounds, which is the approximation of the log evidence (the higher the better).

Figure 8 depicts the mean squared error (MSE) for the different learning methods. We can see that the MSE reconstruction error of all Variational Rényi methods, and specifically $\mathbf{VRS}_{0.5,-0.5}$, are better than VAE reconstruction error in all of the datasets.

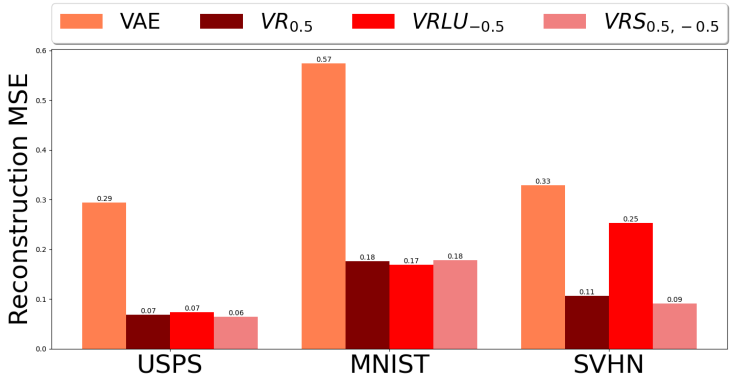

Figure 8. Comparison of the MSE values of **VAE**, $\mathbf{VR}_{0.5}$, $\mathbf{VRLU}_{-0.5}$ and $\mathbf{VRS}_{0.5,-0.5}$ over Digits datasets.

5.1.2. Faces Experiment

We performed a similar experiment on a dataset of facial expressions known as PIE. The PIE dataset consists of a few parts, each corresponding to a different posture. Specifically, we choose PIE05 (left pose), PIE07 (top pose), and PIE09 (bottom pose). In each subset (pose), all face images were taken under different lighting, illumination, and expression conditions (see Figure 9).

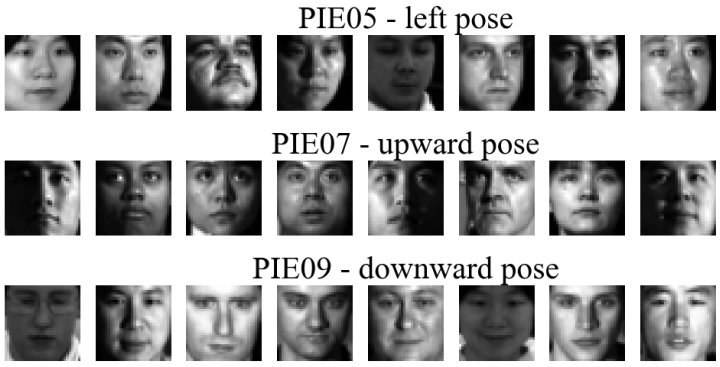

Figure 9. PIE datasets visualization.

We divided each dataset into training and testing sets, in a ratio of 2:1. We created **VAE**, $\mathbf{VR}_\alpha$ and $\mathbf{VRS}_{\alpha_+,\alpha_-}$ models for each 'PIE' domain (left pose, up position, and down position). Each model was trained on its corresponding training set. We calculated the log-likelihood estimations for each domain and compared them. The results are presented in Figure 10. We can see that the $\mathbf{VRS}_{\alpha_+,\alpha_-}$ model achieved the best results. In addition, for $\alpha_+ = 0.5$, we obtained slightly better results than for $\alpha_+ = 2$, which is compatible with all previous results.

To summarize, we demonstrated the performance of the $\mathbf{VRS}_{\alpha_+,\alpha_-}$ algorithm on the digits datasets ('MNIST', 'USPS', 'SVHN') and 'PIE' datasets (left pose, up position, and down position), and compared them against the (KL divergence-based) **VAE**, the Variational Rényi $\mathbf{VR}_\alpha$ upper and lower bounds, and the $\mathbf{VRLU}_\alpha$ upper bound minimization. In all cases, the $\mathbf{VRS}_{\alpha_+,\alpha_-}$ algorithm presented good results, many of which are the best performances compared to the other methods.

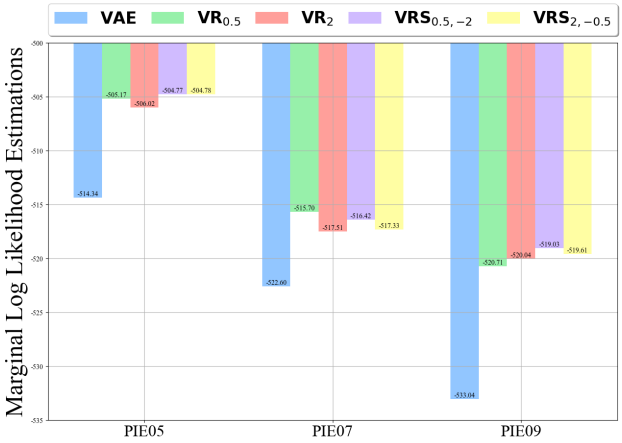

Figure 10. Comparison between the log-likelihood estimates, calculated using the models **VAE**, $\mathbf{VR}_\alpha$ and $\mathbf{VRS}_{\alpha_+,\alpha_-}$ with different values of α. Each model was trained on a specific domain of 'PIE'.

5.2. MSA Experiments

In this section, we review a series of experiments designed to tackle the MSA problem. In all of the experiments, we used the DC-programming algorithm, presented in [9], to provide a solution. We used real-world datasets: the digit dataset and Office31 dataset. For all of the datasets, the probability distributions p_i are not readily available to the learner. Thus, we used the **VAE**, $\mathbf{VR}_\alpha$ and $\mathbf{VRS}_{\alpha_+,\alpha_-}$ models to approximate the probabilities $\hat{p}_i$. More concretely, given an MSA scenario, where we have k source domains and one target domain, we train a variational inference model for each source domain i. We then use the estimated distributions as input to the DC programming algorithm, which, in turn, finds the optimal vector w used to construct the distribution-weighted combination hypothesis h_w^η (Definition 9) for the target domain. We term the technique described above as **VRS-MSA**. Finally, we compared our performances to the results presented by Cortes et al. in [20].

5.2.1. Digits Experiment

In the following experiment, we used the digits datasets, SVHN, MNIST, and USPS, as our source domains. For each domain, we trained a convolutional neural network (CNN) of the same architecture as in [25], and used the output from the softmax score layer as our base predictors h_i. We also trained the **VAE**, $\mathbf{VR}_\alpha$ and $\mathbf{VRS}_{\alpha_+,\alpha_-}$ models for each domain using the respective training sets. We used these trained models to approximate the domains' distributions $\hat{p}_i$.

For the DC-programming algorithm, we used 1000 image–label pairs from each domain, thus being a total of 3000 labeled pairs, to learn the parameter w. We compared our **VRS-MSA** algorithm against the results presented in [20], and report performances on each of the three test datasets, on combinations of two test datasets, and on all test datasets combined.

Table 2 details the accuracy scores obtained by running our **VRS-MSA** model and the following models:

- CNN-s, CNN-m, and CNN-u: each trained on the single source domain SVHN, MNIST, and USPS, respectively.
- CNN-unif: a classifier trained on a uniform combination of the source domains' data.
- CNN-joint: a global classifier trained on all of the source domains' data combined.
- The GMSA model: a generative MSA model using the DC programming algorithm. To obtain the data distribution, GMSA used the last layer before softmax from each of the domains' classifiers.
- The DMSA model: this is based on a discriminative technique using an estimate of the conditional probabilities (the probability that point x belongs to source i).

Table 2. Digit Dataset Accuracy (s—SVHN, m—MNIST and u—USPS). Previous results were taken from [20]. Bold labels signify the top score within the respective column.

Models	**Test Datasets**							
	s	**m**	**u**	**mu**	**su**	**sm**	**smu**	**Mean**
CNN-s	92.3	66.9	65.6	66.7	90.4	85.2	84.2	78.8
CNN-m	15.7	99.2	79.7	96.0	20.3	38.9	41.0	55.8
CNN-u	16.7	62.3	96.6	68.1	22.5	29.4	32.9	46.9
CNN-unif	75.7	91.3	92.2	91.4	76.9	80.0	80.7	84.0
CNN-joint	90.9	99.1	96.0	98.6	91.3	93.2	93.3	94.6
GMSA	91.4	98.8	95.6	98.3	91.7	93.5	93.6	94.7
DMSA	92.3	**99.2**	**96.6**	**98.8**	92.6	94.2	94.3	95.4
VAE-MSA	72.1	97.7	94.6	96.0	92.3	95.7	95.7	92.0
VR_2-MSA	72.4	99.1	94.9	96.5	89.3	96.1	95.6	92.0
$VR_{0.5}$-MSA	70.0	99.1	95.1	96.5	89.2	96.1	95.7	91.7
$VRS_{2,-2}$-MSA	74.2	99.1	94.7	96.5	89.3	96.1	95.6	92.2
$VRS_{2,-0.5}$-MSA	71.5	98.9	95.7	96.5	87.5	95.9	95.6	91.6
$VRS_{0.5,-2}$-MSA	72.5	99.1	94.7	96.5	90.1	**96.1**	**95.7**	92.1
$VRS_{0.5,-0.5}$-MSA	76.0	99.1	94.6	96.5	89.4	95.8	95.4	92.4
VAE-SGD	93.8	99.0	94.6	98.3	93.8	95.2	95.2	95.7
VR_2-SGD	93.9	98.5	94.8	97.9	94.0	95.2	95.2	95.6
$VR_{0.5}$-SGD	93.7	99.0	94.8	98.3	93.8	95.2	95.2	95.7
$VRS_{2,-2}$-SGD	93.7	99.0	94.7	98.3	93.8	95.2	95.2	**95.7**
$VRS_{2,-0.5}$-SGD	93.9	98.4	95.0	97.8	94.0	95.2	95.1	95.6
$VRS_{0.5,-2}$-SGD	93.9	98.5	94.9	97.9	93.4	95.2	95.1	95.6
$VRS_{0.5,-0.5}$-SGD	**93.9**	98.4	94.9	97.8	**94.0**	95.2	95.2	95.6

Our **VRS-MSA** model demonstrates competitive performance, with particularly strong results on the union of the SVHN and MNIST test sets and the union of the SVHN, MNIST, and USPS test sets. Moreover, among VI models, $\mathbf{VRS}_{0.5,-0.5}$ achieved the best average score. This result is consistent with our previous results, which state that the closer α is to zero, the better the approximation of the log evidence.

However, the performance on the SVHN domain is lower in comparison to the other classifiers. Taking a closer look at the parameter $w = (w_{_{MNIST}} : 0.73, w_{_{USPS}} : 0.19, w_{_{SVHN}} : 0.08)$ reveals that the value assigned to the SVHN domain, denoted as $w_{_{SVHN}}$, is relatively low at 0.08. Since the distribution weighted combining rule is a weighted combination of all source hypotheses with weights assignment w, this indicates that the SVHN domain has a minimal impact on the calculation of h^{η}_{w}. Additionally, the log probability obtained for the SVHN domain using the VI models is quite low compared to the other domains. These low values result in very small probabilities when taking the exponent, which can be difficult to work with in practice.

Furthermore, we devised a method that uses Stochastic Gradient Descent (SGD), rather than DC programming, to get a good classifier for the target domain. For each image x, every possible label $y_1, ..., y_c$, and every source domain, we created the following input:

$$(p_1(x, y_1), ..., p_1(x, y_c), ..., p_k(x, y_1), ..., p_k(x, y_c), h_1(x, y_1), ..., h_1(x, y_c), ..., h_k(x, y_1), ..., h_k(x, y_c))$$

Given image x, the SGD model learns a matching between the input vector above and the true label of x. This method is termed **VRS-SGD**. Similarly to VRS-MSA, we used 1000 images from each domain to train the SGD model. The results of the VRS-SGD are reported at the last section of Table 2.

The SGD score for the SVHN test set stands out as the highest, leading to an improvement in the combined test set that includes both SVHN and USPS. One advantage of the VRS-SGD method is its ability to overcome the issue of misalignment among different VRS models by adjusting its learned weights to match the input scale. This makes the VRS-SGD

method particularly valuable when working with source domains where the probabilities are smaller compared to other domains.

5.2.2. Office Experiment

In the following experiment, we used the Office31 dataset, which is used mainly in domain adaptation scenarios. The Office31 dataset contains 31 object categories in three domains: Amazon, DSLR, and Webcam (see Figure 11). The 31 categories in the dataset consist of objects commonly encountered in office settings, such as keyboards, file cabinets, and laptops. The Amazon domain contains on average 90 images per class and 2817 images in total. As these images were captured from a website of online merchants, they are captured against a clean background and at a unified scale. The DSLR domain contains 498 low-noise high-resolution images (4288 $\times$ 2848). There are 5 objects per category. Each object was captured from different viewpoints on average 3 times. For Webcam, the 795 images of low resolution (640 $\times$ 480) exhibit significant noise and color as well as white balance artifacts.

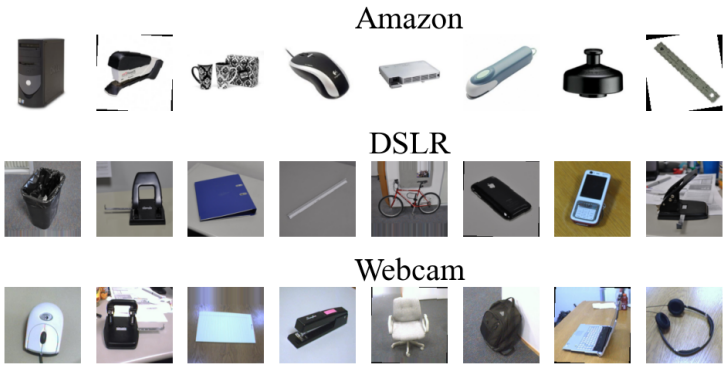

Figure 11. Office datasets visualization.

We carried out the **VRS-MSA** experiment on Office31 dataset. We divided the dataset into two splits following [26]. For the training data, we used 20 samples per category for Amazon and 7 for both DSLR and Webcam. We used the rest of the samples as test data. For each domain, we used ResNet50 architecture pre-trained on ImageNet, and trained it over the domain's training set. We extracted the penultimate layer output from ResNet50 architecture and trained our variational inference models **VAE**, $\mathbf{VR}_{\alpha}$ and $\mathbf{VRS}_{\alpha_+,\alpha_-}$ on this pre-trained feature. The VI models were used to approximate the distributions p_i. For our predictors h_i, we extracted the output from the ResNet50 architecture and used softmax layer to calculate the probabilities. We used a batch size of 32 in the training set and 16 in the test set.

We measured the performance of these baselines on each of the three test sets, on combinations of two test sets, and all test sets combined. We compared our **VRS-MSA** model against previous results presented by Cortes et al. [20]. While Cortes et al. only provided results for individual test sets, we additionally presented results for various combinations of test sets, providing a more comprehensive comparison of the performance of VI models. Among the models tested, our $\mathbf{VRS}_{0.5,-0.5}$ model achieved the highest results in most test set combinations and had the best overall score, which supports our previous findings that a value of α close to zero leads to a better approximation of the log evidence.

We compared our results to the DMSA algorithm, each source predictor (CNN for Amazon, DSLR and Webcam), the uniform combination, CNN-unif, a network jointly trained on all source data combined, CNN-joint, and GMSA with kernel density estimation [9]. The results are reported in Table 3.

Table 3. Office Dataset Accuracy (a—Amazon, w—Webcam, d—DSLR). Previous results were taken from [20]. Bold labels signify the top score within the respective column.

Models	Test Datasets							
	a	**w**	**d**	**aw**	**ad**	**wd**	**awd**	**Mean**
Resnet-a	82.2	75.8	77.6	-	-	-	-	-
Resnet-w	63.3	95.7	95.7	-	-	-	-	-
Resnet-d	64.6	94.0	95.8	-	-	-	-	-
Resnet-unif	79.3	96.7	97.2	-	-	-	-	-
GMSA	82.1	96.8	96.7	-	-	-	-	-
DMSA	82.2	**97.2**	97.4	-	-	-	-	-
VAE-MSA	76.6	93.4	98.6	81.0	79.8	95.0	82.7	86.7
VR_2-MSA	76.0	94.1	98.2	80.5	79.0	95.2	82.4	86.5
$VR_{0.5}$-MSA	77.3	93.1	98.6	81.5	80.5	94.8	83.5	87.0
$VRS_{0.5,-2}$-MSA	69.0	93.0	**99.0**	74.6	72.6	94.9	77.0	82.9
$VRS_{2,-0.5}$-MSA	78.0	93.2	98.6	82.0	80.7	94.8	83.7	87.3
$VRS_{2,-2}$-MSA	81.6	92.2	98.6	84.5	84.0	94.3	86.0	88.7
$VRS_{0.5,-0.5}$-MSA	81.7	92.4	98.6	84.6	84.2	94.5	86.1	88.9
VAE-SGD	92.2	95.0	96.8	92.8	92.7	95.6	93.2	94.0
VR_2-SGD	92.2	95.0	96.8	92.8	92.7	95.6	93.1	94.0
$VR_{0.5}$-SGD	92.2	95.0	96.8	92.8	92.7	95.6	93.2	94.0
$VRS_{2,-2}$-SGD	92.2	95.0	96.8	92.7	92.7	95.6	93.1	94.0
$VRS_{2,-0.5}$-SGD	92.2	94.8	97.2	92.7	92.7	95.6	93.2	**94.1**
$VRS_{0.5,-2}$-SGD	92.2	95.0	96.8	92.8	92.7	95.6	93.1	94.0
$VRS_{0.5,-0.5}$-SGD	**92.2**	95.0	96.8	**92.8**	**92.7**	**95.6**	**93.2**	94.0

Our **VRS-MSA** model demonstrates competitive achievements, with particularly strong results on the test set DSLR. We note that the DSLR's high score comes at the expense of Amazon's and Webcam's high scores. This is because the vector $w = (w_{Amazon} : 0.25, w_{DSLR} : 0.71, w_{Webcam} : 0.04)$ learned in the DC programming algorithm determined the weight of each domain. When the DSLR domain receives more weight, it comes at the expense of the weight given to the other domains.

Likewise, the VRS-SGD method achieved competitive scores compared to the models using the DC algorithm. We can see that the VRS-SGD score for the Amazon test set is the highest and, as a result, the scores on test sets that include Amazon were also improved.

6. Summary

In this study, we reviewed and analyzed the methods to estimate data probabilities where traditional computation methods have failed. Specifically, we examined variational inference (VI) models, such as Variational Autoencoder (VAE) [27], which we aimed to improve using different divergence methods. We examined the properties of the Kullback–Leibler divergence, the Rényi divergence (which is essentially a family of divergences parameterized by $\alpha \in \mathbb{R}$), and the χ divergence. We derived the ELBO, the VR, and the CUBO bounds for the log evidence, and presented a new upper bound, termed VRLU, for which its MC approximation remains an upper. We used VRLU to devise a new (sandwiched) upper–lower bound variational inference method (VRS). The VRS loss function combines the VR lower bound (with positive α) and the new VRLU upper bound (with negative α), thus providing a tighter estimate for the log evidence.

We performed several experiments designed to test the performance of the new VRS model. We compared VAE, VR, VRLU, and VRS models over the digits datasets and PIE datasets, using different values of positive and negative α. In all cases, the VRS algorithm presented good results, many of which are the best performances compared to the other methods. We note, in passing, that the selection of the α value may depend on the data, an observation that was indicated in previous studies, as well [3,14].

In addition, we demonstrated the usage of VRS in MSA applications. We combined the DC-programming algorithm (suggested in [9]) with our VRS model, to obtain more accurate density estimates and improve the accuracy of the hypothesis for the target domain. We performed experiments to compare the accuracy of the resulting hypothesis in two MSA datasets: the digits and Office31 datasets. We compared our new model using VAE, VR, and VRS to the previous models, GMSA and DMSA, presented in [20].

Our empirical evaluation revealed that the proposed VRS-MSA model demonstrated competitive performance, and in certain instances even surpassed the performance of models reported in previous studies. Additionally, among the VI models tested, the VRS model achieved the highest overall score, which supports the conclusion that accurate probability estimates are necessary for the success of the weighted combination hypothesis h_w^η.

Nonetheless, it is important to note that the VRS-MSA model achieved lower scores in certain individual test sets, where the weight parameter w was assigned a low value for that particular domain. When the weight parameter is low, it is important to take into account both the probability $p_i(x)$ and the domain-specific hypothesis h_i. For example, if the image x is from the SVHN domain, the probabilities $p_{mnist}(x)$ and $p_{usps}(x)$ should be relatively low in comparison to $p_{svhn}(x)$, such that the value of h_{svhn} is the most prominent in the weighted combination hypothesis. Our VRS-MSA model operates by training a VRS model for each domain, which learns its latent space vectors based on a Gaussian distribution, and outputs the probability in relation to these latent vectors $p_\theta(x|z)$. Consequently, for each domain, the Gaussian distribution may have slight variations in variance, which can influence the log evidence value output from the VRS model. Therefore, the DC programming model, which takes into account the probabilities from all domains simultaneously, may be affected by the different scales of the probability measurements across the domains.

Looking forward, further work is required to disentangle the complexities of the aforementioned VRS-MSA. Specifically, in this work, we have not formed a connection between the latent variables of each VRS model of the different domains. It will be interesting to see how such a connection (of normalization, scaling of the probability measurements, or latent space alignment) will affect the compatibility of the probabilities. In addition, some researchers suggest even using a common latent feature space in the autoencoder models [28]. Building such a network using our VRS loss might improve the results of the VRS-MSA model. However, it is worth noting that such a common model would lack the separation and privacy of domains that we have achieved using distinct VRS models.

We would also like to extend our experiments on the VRS model: First, it will be interesting to examine the different values of negative and positive α values and search for the best combination of α_- and α_+. Second, since α may be data-dependent, it will be interesting to explore the possibility to make α a trainable parameter. It can also be used to adjust the degree of relative risk aversion. These directions are left for future research efforts.

Author Contributions: Methodology, D.Z.; Software, D.Z.; Investigation, D.Z.; Supervision, S.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: https://github.com/DanaOshri/Multiple-Source-Adaptation-using-Variational-R-nyi-Bound-Optimization (accessed on 26 September 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jordan, M.I.; Ghahramani, Z.; Jaakkola, T.S.; Saul, L.K. An introduction to variational methods for graphical models. *Mach. Learn.* **1999**, *37*, 183–233. [CrossRef]
2. Kingma, D.P.; Welling, M. Auto-Encoding Variational Bayes. *Statistics* **2014**, *1050*, 1.
3. Li, Y.; Turner, R.E. Rényi divergence variational inference. *Adv. Neural Inf. Process. Syst.* **2016**, arXiv:1602.02311 .
4. Zhang, C.; Bütepage, J.; Kjellström, H.; Mandt, S. Advances in variational inference. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 2008–2026. [CrossRef]
5. Hoffman, M.D.; Blei, D.M.; Wang, C.; Paisley, J. Stochastic variational inference. *J. Mach. Learn. Res.* **2013**, *14*, 1303–1347.
6. Wan, N.; Li, D.; Hovakimyan, N. F-divergence variational inference. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 17370–17379.
7. Akrami, H.; Joshi, A.A.; Li, J.; Aydore, S.; Leahy, R.M. Robust Variational Autoencoder. *arXiv* **2019**, arXiv:1905.09961.
8. Hernandez-Lobato, J.; Li, Y.; Rowland, M.; Bui, T.; Hernández-Lobato, D.; Turner, R. Black-box alpha divergence minimization. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 1511–1520.
9. Hoffman, J.; Mohri, M.; Zhang, N. Algorithms and theory for multiple-source adaptation. *Adv. Neural Inf. Process. Syst.* **2018**, arXiv:1805.08727.
10. Csiszár, I. I-divergence geometry of probability distributions and minimization problems. *Ann. Probab.* **1975**, *3*, 146–158. [CrossRef]
11. MacKay, D.J. *Information Theory, Inference and Learning Algorithms*; Cambridge University Press: Cambridge, UK, 2003.
12. Shekhovtsov, A.; Schlesinger, D.; Flach, B. VAE Approximation Error: ELBO and Exponential Families. In Proceedings of the International Conference on Learning Representations (ICLR), Vienna, Austria, 4 May 2021.
13. Van Erven, T.; Harremos, P. Rényi divergence and Kullback-Leibler divergence. *IEEE Trans. Inf. Theory* **2014**, *60*, 3797–3820. [CrossRef]
14. Dieng, A.B.; Tran, D.; Ranganath, R.; Paisley, J.; Blei, D. Variational Inference via χ Upper Bound Minimization. *Adv. Neural Inf. Process. Syst.* **2017**, arXiv:1611.00328v4.
15. Kroese, D.P.; Brereton, T.; Taimre, T.; Botev, Z.I. Why the Monte Carlo method is so important today. *Wiley Interdiscip. Rev. Comput. Stat.* **2014**, *6*, 386–392. [CrossRef]
16. Poole, B.; Ozair, S.; Van Den Oord, A.; Alemi, A.; Tucker, G. On variational bounds of mutual information. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 5171–5180.
17. Ji, C.; Shen, H. Stochastic variational inference via upper bound. *arXiv* **2019**, arXiv:1912.00650.
18. Pradier, M.F.; Hughes, M.C.; Doshi-Velez, F. Challenges in Computing and Optimizing Upper Bounds of Marginal Likelihood Based on Chi-Square Divergences. In *Second Symposium on Advances in Approximate Bayesian Inference*. 2019. Available online: https://openreview.net/forum?id=BJxk51h4FS (accessed on 10 September 2023).
19. Mansour, Y.; Mohri, M.; Rostamizadeh, A. Domain adaptation with multiple sources. *Adv. Neural Inf. Process. Syst.* **2008**, *21*, 1041–1048.
20. Cortes, C.; Mohri, M.; Suresh, A.T.; Zhang, N. A discriminative technique for multiple-source adaptation. In Proceedings of the International Conference on Machine Learning, Virtual Event, 18–24 July 2021; pp. 2132–2143.
21. Mansour, Y.; Mohri, M.; Rostamizadeh, A. Multiple source adaptation and the Rényi divergence. In Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, 18–21 June 2009; pp. 367–374
22. Cheung, W. Generalizations of Hölders inequality. *Sciences* **2001**, *26*, 7–10.
23. Horst, R.; Thoai, N.V. DC programming: Overview. *J. Optim. Theory Appl.* **1999**, *103*, 1–43. [CrossRef]
24. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
25. French, G.; Mackiewicz, M.; Fisher, M. Self-ensembling for visual domain adaptation. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
26. Saenko, K.; Kulis, B.; Fritz, M.; Darrell, T. Adapting visual category models to new domains. In Proceedings of the European conference on computer vision, Heraklion, Greece, 5–11 September 2010; pp. 213–226.
27. Kingma, D.P.; Welling, M. An introduction to variational autoencoders. *Found. Trends Mach. Learn.* **2019**, *12*, 307–392. [CrossRef]
28. Wu, F.; Zhuang, X. Unsupervised domain adaptation with variational approximation for cardiac segmentation. *IEEE Trans. Med Imaging* **2021**, *40*, 3555–3567. [CrossRef] [PubMed]