Appendix: A Free-Energy Principle for Representation Learning

A. Details of the experimental setup

Datasets. We use the MNIST (LeCun et al., 1998) and CIFAR-10 (Krizhevsky, 2009) datasets for these experiments. The former consists of $28 \times 28$-sized gray-scale images of handwritten digits (60,000 training and 10,000 validation). The latter consists of $32 \times 32$-sized RGB images (50,000 training and 10,000 for validation) spread across 10 classes; 4 of these classes (airplane, automobile, ship, truck) are transportation-based while the others are images of animals and birds.

Architecture and training. All models in our experiments consist of an encoder-decoder pair along with a classifier that takes in the latent representation as input. For experiments on MNIST, both encoder and decoder are multi-layer perceptrons with 2 fully-connected layers, the decoder uses a mean-square error loss, i.e., a Gaussian reconstruction likelihood and the classifier consists of a single fully-connected layer. For experiments on CIFAR-10, we use a residual network (He et al., 2016) with 18 layers as an encoder and a decoder with one fully-connected layer and 4 deconvolutional layers (Noh et al., 2015). The classifier network for CIFAR-10 is a single fully-connected layer. All models use ReLU non-linearities and batch-normalization (Ioffe & Szegedy, 2015). Further details of the architecture are given in Appendix A. We use Adam (Kingma & Ba, 2014) to train all models with cosine learning rate annealing.

The encoder and decoder for MNIST has 784–256–16 neurons on each layer; the encoding $z$ is thus 16-dimensional which is the input to the decoder. The classifier has one hidden layer with 12 neurons and 10 outputs. The encoder for CIFAR-10 is a 18-layer residual neural network (ResNet-18) and the decoder has 4 deconvolutional layers. We used a slightly larger network for the geodesic transfer learning experiment on MNIST. The encoder and decoder have 784–400–64 neurons in each layer with a dropout of probability 0.1 after the hidden layer. The classifier has a single layer that takes the 64-dimensional encoding and predicts 10 classes.

B. Proof of Lemma 3

The second statement directly follows by observing that $F$ is a minimum of affine functions in $(\lambda, \gamma)$. To see the first, evaluate the Hessian of $R$ and $F$

$$\text{Hess}(R) \cdot \text{Hess}(F) = \begin{pmatrix} \frac{\partial^2 R}{\partial \lambda^2} & \frac{\partial^2 R}{\partial \lambda \partial \gamma} \\ \frac{\partial^2 R}{\partial \gamma \partial \lambda} & \frac{\partial^2 R}{\partial \gamma^2} \end{pmatrix} \begin{pmatrix} \frac{\partial^2 F}{\partial \lambda^2} & \frac{\partial^2 F}{\partial \lambda \partial \gamma} \\ \frac{\partial^2 F}{\partial \gamma \partial \lambda} & \frac{\partial^2 F}{\partial \gamma^2} \end{pmatrix}$$

Since we have $F = \min_{\theta}(z|x), d_{\theta}(x|z), m_{\theta}(z) \frac{R + \lambda D + \gamma C}{\partial C} = \frac{\partial R}{\partial D}$, we obtain

$$\lambda = -\frac{\partial R}{\partial D}, \quad \gamma = -\frac{\partial R}{\partial C}, \quad D = \frac{\partial F}{\partial \lambda}, \quad C = \frac{\partial F}{\partial \gamma}.$$

We then have

$$d\lambda = -d\left(\frac{\partial R}{\partial D}\right) = -\frac{\partial^2 R}{\partial D^2} dD - \frac{\partial^2 R}{\partial D \partial C} dC$$

$$= -\frac{\partial^2 R}{\partial D^2} \left(\frac{\partial D}{\partial \lambda} d\lambda + \frac{\partial D}{\partial \gamma} d\gamma\right) - \frac{\partial^2 R}{\partial D \partial C} \left(\frac{\partial C}{\partial \lambda} d\lambda + \frac{\partial C}{\partial \gamma} d\gamma\right)$$

$$= -\left(\frac{\partial^2 R}{\partial D^2} \frac{\partial^2 F}{\partial \lambda^2} + \frac{\partial^2 R}{\partial D \partial C} \frac{\partial^2 F}{\partial \gamma \partial \lambda}\right) d\lambda - \left(\frac{\partial^2 R}{\partial D \partial C} \frac{\partial^2 F}{\partial \lambda \partial \gamma} + \frac{\partial^2 R}{\partial D^2} \frac{\partial^2 F}{\partial \gamma^2}\right) d\gamma;$$
Then with some effort of computation, we get
\[
\begin{align*}
\frac{\partial^2 R}{\partial \xi^2} \frac{\partial^2 F}{\partial \gamma^2} + \frac{\partial^2 R}{\partial \gamma^2} \frac{\partial^2 F}{\partial \xi^2} &= \frac{\partial^2 R}{\partial \xi \partial \gamma} \frac{\partial^2 F}{\partial \gamma \partial \xi} + \frac{\partial^2 R}{\partial \gamma \partial \xi} \frac{\partial^2 F}{\partial \xi \partial \gamma} = 0; \\
\frac{\partial^2 R}{\partial \xi^2} \frac{\partial^2 F}{\partial \gamma^2} + \frac{\partial^2 R}{\partial \gamma^2} \frac{\partial^2 F}{\partial \xi^2} &= \frac{\partial^2 R}{\partial \xi \partial \gamma} \frac{\partial^2 F}{\partial \gamma \partial \xi} + \frac{\partial^2 R}{\partial \gamma \partial \xi} \frac{\partial^2 F}{\partial \xi \partial \gamma} = -1;
\end{align*}
\]
therefore
\[
\text{Hess}(R) \text{ Hess}(F) = -I.
\]

Since \( \triangleright \) Hess(\( F \)), we have that Hess(\( R \)) \( \triangleright \) 0, then the constraint surface \( f(R, D, C) = 0 \) is convex.

**C. Proof of Lemma 5**

Recall the definition of the objective function (14), first we compute the gradient of the objective function as following:

\[
\begin{align*}
\nabla_\theta J(\theta, \lambda, \gamma) &= -\mathbb{E}_{x \sim p(x)} \nabla_\theta \log Z_{\theta, x} \\
&= -\mathbb{E}_{x \sim p(x)} \frac{1}{Z_{\theta, x}} \nabla_\theta Z_{\theta, x} \\
&= -\mathbb{E}_{x \sim p(x)} \frac{1}{Z_{\theta, x}} \int (-\nabla_\theta H) \exp(-H) \, dz \\
&= -\mathbb{E}_{x \sim p(x)} \langle \nabla_\theta H \rangle
\end{align*}
\]

Then with some effort of computation, we get

\[
A = \nabla_\theta^2 J(\theta, \lambda, \gamma) = \nabla_\theta \mathbb{E}_{x \sim p(x)} \left[ \frac{1}{Z_{\theta, x}} \int \nabla_\theta H \exp(-H) \, dz \right]
\]

\[
= \mathbb{E}_{x \sim p(x)} \left[ \frac{1}{Z_{\theta, x}} \int \nabla_\theta H \exp(-H) \, dz \right]
\]

\[
= \mathbb{E}_{x \sim p(x)} \left[ \left\langle \nabla_\theta^2 H \right\rangle + \left\langle \nabla_\theta H \right\rangle \left\langle \nabla_\theta H \right\rangle^\top - \left\langle \nabla_\theta H \nabla_\theta H \right\rangle \right]
\]

\[
b_\lambda = -\frac{\partial}{\partial \lambda} \nabla_\theta J = -\frac{\partial}{\partial \lambda} \mathbb{E}_{x \sim p(x)} \left[ \frac{1}{Z_{\theta, x}} \int \nabla_\theta H \exp(-H) \, dz \right]
\]

\[
= \mathbb{E}_{x \sim p(x)} \left[ \left\langle \frac{\partial \nabla_\theta H}{\partial \lambda} \right\rangle - \left\langle \frac{\partial H}{\partial \lambda} \nabla_\theta H \right\rangle + \left\langle \frac{\partial H}{\partial \lambda} \right\rangle \langle \nabla_\theta H \rangle \right]
\]

\[
b_\gamma = -\frac{\partial}{\partial \gamma} \nabla_\theta J = -\frac{\partial}{\partial \gamma} \mathbb{E}_{x \sim p(x)} \left[ \frac{1}{Z_{\theta, x}} \int \nabla_\theta H \exp(-H) \, dz \right]
\]

\[
= \mathbb{E}_{x \sim p(x)} \left[ \left\langle \frac{\partial \nabla_\theta H}{\partial \gamma} \right\rangle - \left\langle \frac{\partial H}{\partial \gamma} \nabla_\theta H \right\rangle + \left\langle \frac{\partial H}{\partial \gamma} \right\rangle \langle \nabla_\theta H \rangle \right]
\]
According to the quasi-static constraints (16), we have

$$A\dot{\theta} - \dot{\lambda} b_\lambda - \dot{\gamma} b_\gamma = 0,$$

that implies

$$\dot{\theta} = A^{-1} b_\lambda \dot{\lambda} + A^{-1} b_\gamma \dot{\gamma} = \theta_\lambda \dot{\lambda} + \theta_\gamma \dot{\gamma}. \quad (32)$$

**D. Computation of Iso-classification constraint**

We start with computing the gradient of classification loss, clear that $C = \mathbb{E}_{x \sim p(x)} [-\int dz \ c(z|x) \log c(y|z)] = -\mathbb{E}_{x \sim p(x)} (\ell)$, where $\ell = \log c_0(y_x|z)$ is the logarithm of the classification loss, then

$$\nabla_\theta C = -\nabla_\theta \mathbb{E}_{x \sim p(x)} \left[ \frac{1}{Z_{\theta,x}} \int \ell \ \exp(-H) \ dz \right]$$

$$= - \mathbb{E}_{x \sim p(x)} \left[ \frac{1}{Z_{\theta,x}} \left( \int (-\nabla_\theta H) \ \exp(-H) \ dz \right) \left( \int \ell \ \exp(-H) \ dz \right) \right] + \frac{1}{Z_{\theta,x}} \int \nabla_\theta \ell \ \exp(-H) \ dz - \frac{1}{Z_{\theta,x}} \int \ell \ \exp(-H) \ dz$$

$$= - \mathbb{E}_{x \sim p(x)} \left[ \langle \nabla_\theta \ell \rangle + \langle \nabla_\theta H \rangle \langle \ell \rangle - \langle \ell \ \nabla_\theta H \rangle \right];$$

$$\frac{\partial}{\partial \lambda} C = - \mathbb{E}_{x \sim p(x)} \left[ \frac{1}{Z_{\theta,x}} \int \ell \ \exp(-H) \ dz \right]$$

$$= - \mathbb{E}_{x \sim p(x)} \left[ \left( \int -\frac{\partial H}{\partial \lambda} \ \exp(-H) \ dz \right) \left( \int \ell \ \exp(-H) \ dz \right) \right] - \frac{1}{Z_{\theta,x}} \int \ell \ \frac{\partial H}{\partial \lambda} \ \exp(-H) \ dz$$

$$= - \mathbb{E}_{x \sim p(x)} \left[ \langle \frac{\partial H}{\partial \lambda} \rangle \langle \ell \rangle - \langle \ell \ \frac{\partial H}{\partial \lambda} \rangle \right];$$

$$\frac{\partial}{\partial \gamma} C = - \mathbb{E}_{x \sim p(x)} \left[ \frac{1}{Z_{\theta,x}} \int \ell \ \exp(-H) \ dz \right]$$

$$= - \mathbb{E}_{x \sim p(x)} \left[ \left( \int -\frac{\partial H}{\partial \gamma} \ \exp(-H) \ dz \right) \left( \int \ell \ \exp(-H) \ dz \right) \right] - \frac{1}{Z_{\theta,x}} \int \ell \ \frac{\partial H}{\partial \gamma} \ \exp(-H) \ dz$$

$$= - \mathbb{E}_{x \sim p(x)} \left[ \langle \frac{\partial H}{\partial \gamma} \rangle \langle \ell \rangle - \langle \ell \ \frac{\partial H}{\partial \gamma} \rangle \right].$$

The iso-classification loss constrains together with quasi-static constrains imply that:

$$0 = \frac{d}{dt} C$$

$$= \dot{\theta} \nabla_\theta C + \dot{\lambda} \frac{\partial C}{\partial \lambda} + \dot{\gamma} \frac{\partial C}{\partial \gamma}$$

$$= \lambda \left( \theta_\lambda \nabla_\theta C + \frac{\partial C}{\partial \lambda} \right) + \gamma \left( \theta_\gamma \nabla_\theta C + \frac{\partial C}{\partial \gamma} \right)$$

$$= - \lambda \mathbb{E}_{x \sim p(x)} \left[ \langle \frac{\partial H}{\partial \lambda} \rangle \langle \ell \rangle - \langle \ell \frac{\partial H}{\partial \lambda} \rangle \right] + \langle \theta_\lambda \nabla_\theta H \rangle \langle \ell \rangle - \langle \ell \theta_\lambda \nabla_\theta H \rangle + \langle \theta_\gamma \nabla_\theta H \rangle \langle \ell \rangle - \langle \ell \theta_\gamma \nabla_\theta H \rangle + \langle \theta_\gamma \nabla_\theta \ell \rangle \langle \ell \rangle - \langle \ell \theta_\gamma \nabla_\theta \ell \rangle$$

$$= C_\lambda \dot{\lambda} + C_\gamma \dot{\gamma},$$

where the third equation is followed by the equilibrium dynamics (17) for parameters $\theta$. So far we developed the constrained dynamics for iso-classification process:

$$0 = C_\lambda \dot{\lambda} + C_\gamma \dot{\gamma}$$

$$\dot{\theta} = \theta_\lambda \dot{\lambda} + \theta_\gamma \dot{\gamma}. \quad (33)$$
E. Iso-classification equations for changing data distribution

In this section we analyze the dynamics for iso-classification loss process when the data distribution evolves with time. \( \frac{\partial p(x)}{\partial t} \) will lead to additional terms that represent the partial derivatives with respect to \( t \) on both the quasi-static and iso-classification constraints. More precisely, the new terms are

\[
b_t = -\frac{\partial}{\partial t} \nabla_\theta J = -\int \frac{\partial p(x)}{\partial t} \langle \nabla_\theta H \rangle \, dx;
\]

\[
\frac{\partial}{\partial t} C = -\int \frac{\partial p(x)}{\partial t} \langle \ell \rangle \, dx,
\]

then the quasi-static and iso-classification constraints are ready to be modified as

\[
0 \equiv \frac{d}{dt} \nabla_\theta J(\theta, \lambda, \gamma) \iff 0 = \nabla_\theta^2 F \dot{\theta} + \lambda \frac{\partial \nabla_\theta F}{\partial \lambda} + \gamma \frac{\partial \nabla_\theta F}{\partial \gamma} + \frac{\partial \nabla_\theta^2 F}{\partial \gamma} \frac{\partial \nabla_\theta C}{\partial \gamma} \frac{\partial \nabla_\theta C}{\partial \gamma} \frac{\partial \nabla_\theta C}{\partial \gamma}
\]

\[
\iff \dot{\theta} = \lambda A^{-1} b_\lambda + \gamma A^{-1} b_\gamma + A^{-1} b_t
\]

\[
0 \equiv \frac{d}{dt} C \iff 0 = \theta^\top \nabla_\theta C + \lambda \frac{\partial C}{\partial \lambda} + \gamma \frac{\partial C}{\partial \gamma} + \frac{\partial C}{\partial \gamma}
\]

\[
\iff 0 = \lambda \left( \theta^\top \nabla_\theta C + \frac{\partial C}{\partial \lambda} \right) + \gamma \left( \theta^\top \nabla_\theta C + \frac{\partial C}{\partial \gamma} \right) + \left( \theta^\top \nabla_\theta C + \frac{\partial C}{\partial \gamma} \right)
\]

\[
\iff 0 = \lambda C_\lambda + \gamma C_\gamma + C_t,
\]

where \( A, b_\lambda, b_\gamma, C_\lambda \) and \( C_\gamma \) where \( C_\lambda \) and \( C_\gamma \) are as given in lemma 5 and (21) with the only change being that the outer expectation is taken with respect to \( x \sim p(x, t) \). The new terms that depends on time \( t \) are

\[
C_t = -\int \frac{\partial p(x, t)}{\partial t} \langle \ell \rangle \, dx - \mathbb{E}_{x \sim p(x, t)} \left[ \langle \theta^\top_i \nabla_\theta H \rangle \ell - \langle \theta^\top_i \nabla_\theta H \ell \rangle + \langle \theta^\top_i \nabla_\theta \ell \rangle \right]
\]

with \( \ell = \log c_\theta(y_{x_t} | z) \). We can combine modified quasi-static and iso-classification constraints to get

\[
\dot{\theta} = \left( \theta_\lambda - \frac{C_\lambda}{C_\gamma} \theta_\gamma \right) \lambda + \left( \theta_t - \frac{C_t}{C_\gamma} \theta_\gamma \right)
\]

\[
= \dot{\theta}_\lambda \lambda + \dot{\theta}_t
\]

This indicates that \( \theta = \theta(\lambda, t) \) is a surface parameterized by \( \lambda \) and \( t \), equipped with a basis of tangent plane \( (\dot{\theta}_\lambda, \dot{\theta}_t) \).

F. Optimally transporting the data distribution

We first give a brief description of the theory of optimal transportation. The optimal transport map between the source task and the target task will be used to define a dynamical process for the task. We only compute the transport for the inputs \( x \) between the source and target distributions and use a heuristic to obtain the transport for the labels \( y \). This choice is made only to simplify the exposition; it is straightforward to handle the case of transport on the joint distribution \( p(x, y) \).

If i.i.d samples from the source task are denoted by \( \{x^*_1, \ldots, x^*_n_s\} \) and those of the target distribution are \( \{x^*_1, \ldots, x^*_n_t\} \) the empirical source and target distributions can be written as

\[
p^s(x) = \frac{1}{n_s} \sum_{i=1}^{n_s} \delta_{x^*}, \text{and } p^t(x) = \frac{1}{n_t} \sum_{i=1}^{n_t} \delta_{x^*}
\]

respectively; here \( \delta_{x^*} \) is a Dirac delta distribution at \( x^* \). Since the empirical data distribution is a sum of a finite number of Dirac measures, this is a discrete optimal transport problem and easy to solve. We can use the Kantorovich relaxation to denote by \( B \) the set of probabilistic couplings between the two distributions:

\[
B = \{ \Gamma \in \mathbb{R}^{n_s \times n_t} : \Gamma 1_{n_s} = p, \Gamma^\top 1_{n_t} = q \}.
\]
where $1_n$ is an $n$-dimensional vector of ones. The Kantorovich formulation solves for

$$
\Gamma^* = \arg\min_{\Gamma \in \mathcal{B}} \sum_{i=1}^{n_s} \sum_{t=1}^{n_t} \Gamma_{ij} \kappa_{ij}
$$

(36)

where $\kappa \in \mathbb{R}^{n_s \times n_t}$ is a cost function that models transporting the datum $x^s_i$ to $x^t_j$. This is the metric of the underlying data domain and one may choose any reasonable metric for $\kappa = ||x^s_i - x^t_j||^2_2$. The problem in (36) is a convex optimization problem and can be solved easily; in practice we use the Sinkhorn’s algorithm (Cuturi, 2013) which adds an entropic regularizer $-h(\Gamma) = \sum_{ij} \Gamma_{ij} \log \Gamma_{ij}$ to the objective in (36).

**F.1. Changing the data distribution**

Given the optimal probabilistic coupling $\Gamma^*$ between the source and the target data distributions, we can interpolate between them at any $t \in [0, 1]$ by following the geodesics of the Wasserstein metric

$$
p(x, t) = \arg\min_p (1 - t)W_2^2(p^s, p) + tW_2^2(p, p^t).
$$

For discrete optimal transport problems, as shown in Villani (2008), the interpolated distribution $p_t$ for the metric $\kappa_{ij} = ||x^s_i - x^t_j||^2_2$ is given by

$$
p(x, t) = \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} \Gamma_{ij}^* \delta_{x-(1-t)x^s_i + tx^t_j}.
$$

(37)

Observe that the interpolated data distribution equals the source and target distribution at $t = 0$ and $t = 1$ respectively and it consists of linear interpolations of the data in between.

**Remark 11 (Interpolating the labels).** The interpolation in (37) gives the marginal on the input space interpolated between the source and target tasks. To evaluate the functionals in Section 3 for the classification setting, we would also like to interpolate the labels. We do so by setting the true label of the interpolated datum $x = (1 - t)x^s_i + tx^t_j$ to be linear interpolation between the source label and the target label.

$$
y(x, t) = (1 - t)\delta_{y - y^s_i} + t\delta_{y - y^t_j}
$$

for all $i, j$. Notice that the interpolated distribution $p(x, t)$ is a sum of Dirac delta distributions weighted by the optimal coupling. We therefore only need to evaluate the labels at all the interpolated data.

**Remark 12 (Linear interpolation of data).** Our formulation of optimal transportation leads to a linear interpolation of the data in (23). This may not work well for image-based data where the square metric $\kappa_{ij} = ||x^s_i - x^t_j||^2_2$ may not be the appropriate metric. We note that this interpolation of data is an artifact of our choice of $\kappa_{ij}$, other choices for the metric also fit into the formulation and should be viable alternatives if they result in efficient computation.