Appendix for "JointGAN: Multi-Domain Joint Distribution Learning with Generative Adversarial Nets"

Yunchen Pu [1] **Shuyang Dai** [2] **Zhe Gan** [3] **Weiyao Wang** [2] **Guoyin Wang** [2] **Yizhe Zhang** [3] **Ricardo Henao** [2] **Lawrence Carin** [2]

A. Proof of Proposition 1

We first consider a general optimization problem as the following

$$\min \mathcal{L}(f_1, \ldots, f_K) = \min \sum_{k=1}^{K} p_k(\boldsymbol{x}, \boldsymbol{y}) \log \frac{f_k}{\sum_{i=1}^{K} f_i} . \tag{1}$$

For any f_k, if we fix all other variables, the local optimal f_k^* is given by:

$$\frac{\partial \mathcal{L}}{\partial f_k} = \frac{p_k(\boldsymbol{x}, \boldsymbol{y})}{f_k} - \sum_{j=1}^{K} \frac{p_j(\boldsymbol{x}, \boldsymbol{y})}{\sum_{i=1}^{K} f_i} = 0 \tag{2}$$

$$\Leftrightarrow f_k = \Big(\sum_{i=1, i \neq k}^{K} f_i \Big) \frac{p_k(\boldsymbol{x}, \boldsymbol{y})}{\sum_{j=1, j\neq k}^{K} p_j(\boldsymbol{x}, \boldsymbol{y})} . \tag{3}$$

The K equations below

$$f_k = \Big(\sum_{i=1, i \neq k}^{K} f_i \Big) \frac{p_k(\boldsymbol{x}, \boldsymbol{y})}{\sum_{j=1, j\neq k}^{K} p_j(\boldsymbol{x}, \boldsymbol{y})}, \quad \textit{for } k = 1, \ldots, K , \tag{4}$$

have a global solution with

$$f_k = C p_k(\boldsymbol{x}, \boldsymbol{y}), \quad \textit{for } k = 1, \ldots, K , \tag{5}$$

where $C \neq 0$ is a constant. Let $\hat{f}_k = \frac{f_k}{\sum_{i=1}^{K} f_i}$, the global optimal of (1) is achieved at $\hat{f}_k = \frac{p_k(\boldsymbol{x}, \boldsymbol{y})}{\sum_{j=1}^{K} p_j(\boldsymbol{x}, \boldsymbol{y})}$.

Let $K = 5$ and $\hat{f}_k = g_{\boldsymbol{\omega}}(\boldsymbol{x}, \boldsymbol{y})[k]$. This indicates that with fixed $(\boldsymbol{\theta}, \boldsymbol{\phi})$, the optimal critic $g_{\boldsymbol{\omega}}$ in (10) in the main paper is achieved at

$$g_{\boldsymbol{\omega}^*}(\boldsymbol{x}, \boldsymbol{y})[k] = \frac{p_k(\boldsymbol{x}, \boldsymbol{y})}{\sum_{j=1}^{K} p_j(\boldsymbol{x}, \boldsymbol{y})} . \tag{6}$$

With optimal $g_{\boldsymbol{\omega}^*}$, the objective (10) in the main paper can be expressed as

$$\mathcal{L} = \sum_{k=1}^{5} \mathbb{E}_{p_k(\boldsymbol{x}, \boldsymbol{y})} \log \frac{p_k(\boldsymbol{x}, \boldsymbol{y})}{\sum_{j=1}^{5} p_j(\boldsymbol{x}, \boldsymbol{y})} \tag{7}$$

$$= -5 \log 5 + \sum_{k=1}^{5} \mathrm{KL}\Big(p_k(\boldsymbol{x}, \boldsymbol{y}) \Big|\Big| \frac{\sum_{j=1}^{5} p_j(\boldsymbol{x}, \boldsymbol{y})}{5} \Big) . \tag{8}$$

The global minimum of (8) is achieved at $p_1(\boldsymbol{x}, \boldsymbol{y}) = p_2(\boldsymbol{x}, \boldsymbol{y}) = p_3(\boldsymbol{x}, \boldsymbol{y}) = p_4(\boldsymbol{x}, \boldsymbol{y}) = p_5(\boldsymbol{x}, \boldsymbol{y})$.

B. Calculating the relevance score

We use relevance socre to evaluate the quality and relevance of two generated samples. The relevance score is calculated by the cosine similarity between random variables $\boldsymbol{x}$ and $\boldsymbol{y}$:

$$R(\boldsymbol{x}, \boldsymbol{y}) = \frac{f(\boldsymbol{x})^\top g(\boldsymbol{y})}{||f(\boldsymbol{x})|| \cdot ||g(\boldsymbol{y})||}\,, \tag{9}$$

where $f(\cdot)$ and $g(\cdot)$ are two feature extractors (typically implemented as CNNs) that embed the two high-dimensional data $\boldsymbol{x}$ and $\boldsymbol{y}$ into a shared low-dimensional latent space. After the two feature extractors are trained on the paired dataset $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1,N}$, we can use the cosine similarity to evaluate the relevance of two synthesized joint samples $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{y}}$.

To learn the feature extractors, a ranking model (Huang et al., 2013) is trained as below. First, we consider using the samples of $\boldsymbol{x}$ to query samples of $\boldsymbol{y}$. Given that we have the relevance scores between the query $\boldsymbol{x}$ and each of the target sample $\boldsymbol{y}_j$: $R(\boldsymbol{x}, \boldsymbol{y}_j)$, we define the posterior probability of the correct candidate given the query by the following softmax function

$$P(\boldsymbol{y}^+|\boldsymbol{x}) = \frac{\exp(\gamma R(\boldsymbol{x}, \boldsymbol{y}^+))}{\sum_{\boldsymbol{y}' \in \mathcal{Y}} \exp(\gamma R(\boldsymbol{x}, \boldsymbol{y}'))}\,, \tag{10}$$

where $\boldsymbol{y}^+$ denotes the correct target sample (the positive sign denotes that it is a positive sample), γ is a tuning hyper-parameter in the softmax function (to be tuned empirically on a validation set). $\mathcal{Y}$ denotes the set of candidate samples to be ranked, which includes the positive sample $\boldsymbol{y}^+$ and J randomly selected incorrect (negative) candidates $\{\boldsymbol{y}_j^-; j = 1, \ldots, J\}$.

Similarly, we also consider using samples of $\boldsymbol{y}$ to query samples of $\boldsymbol{x}$. A corresponding posterior is defined as follow:

$$P(\boldsymbol{x}^+|\boldsymbol{y}) = \frac{\exp(\gamma R(\boldsymbol{y}, \boldsymbol{x}^+))}{\sum_{\boldsymbol{x}' \in \mathcal{X}} \exp(\gamma R(\boldsymbol{y}, \boldsymbol{x}'))}\,. \tag{11}$$

The model parameters are learned to maximize the likelihood of the correct candidates given the queries across the training set, in both the above directions. That is, we minimize the following loss function

$$L(\boldsymbol{\theta}) = -\log \prod_{(\boldsymbol{x}, \boldsymbol{y}^+)} P(\boldsymbol{y}^+|\boldsymbol{x}) - \log \prod_{(\boldsymbol{y}, \boldsymbol{x}^+)} P(\boldsymbol{x}^+|\boldsymbol{y})\,,$$

where the product is over all training samples, and $\boldsymbol{\theta}$ denotes the parameters (to be learned), which includes all the model parameters in the deep feature extractors. The above cost function is minimized by backpropagation and (mini-batch) stochastic gradient descent.

C. More Results

C.1. Results for the two-step baseline

Figure 1. Generated paired samples trained with WGAN-GP+Pix2pix on paired data.

Figure 2. Generated paired samples trained with WGAN-GP+CycleGAN on unpaired data.

C.2. Additional results for JointGAN on modeling multi-domain images

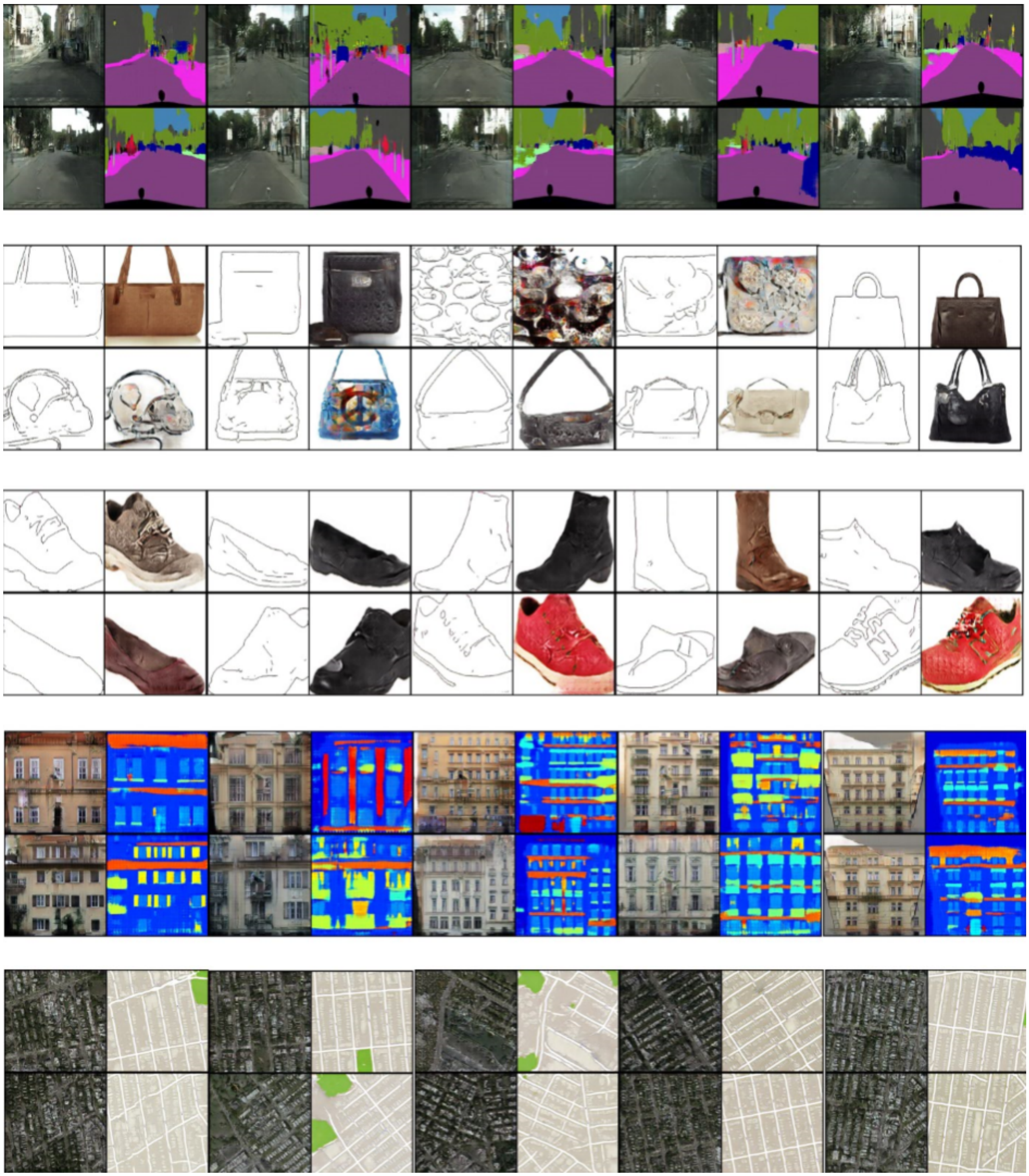

Figure 3. Generated paired samples trained with paired data.

Figure 4. Generated paired samples trained with unpaired data.

C.3. Additional results for JointGAN on modeling caption features and images

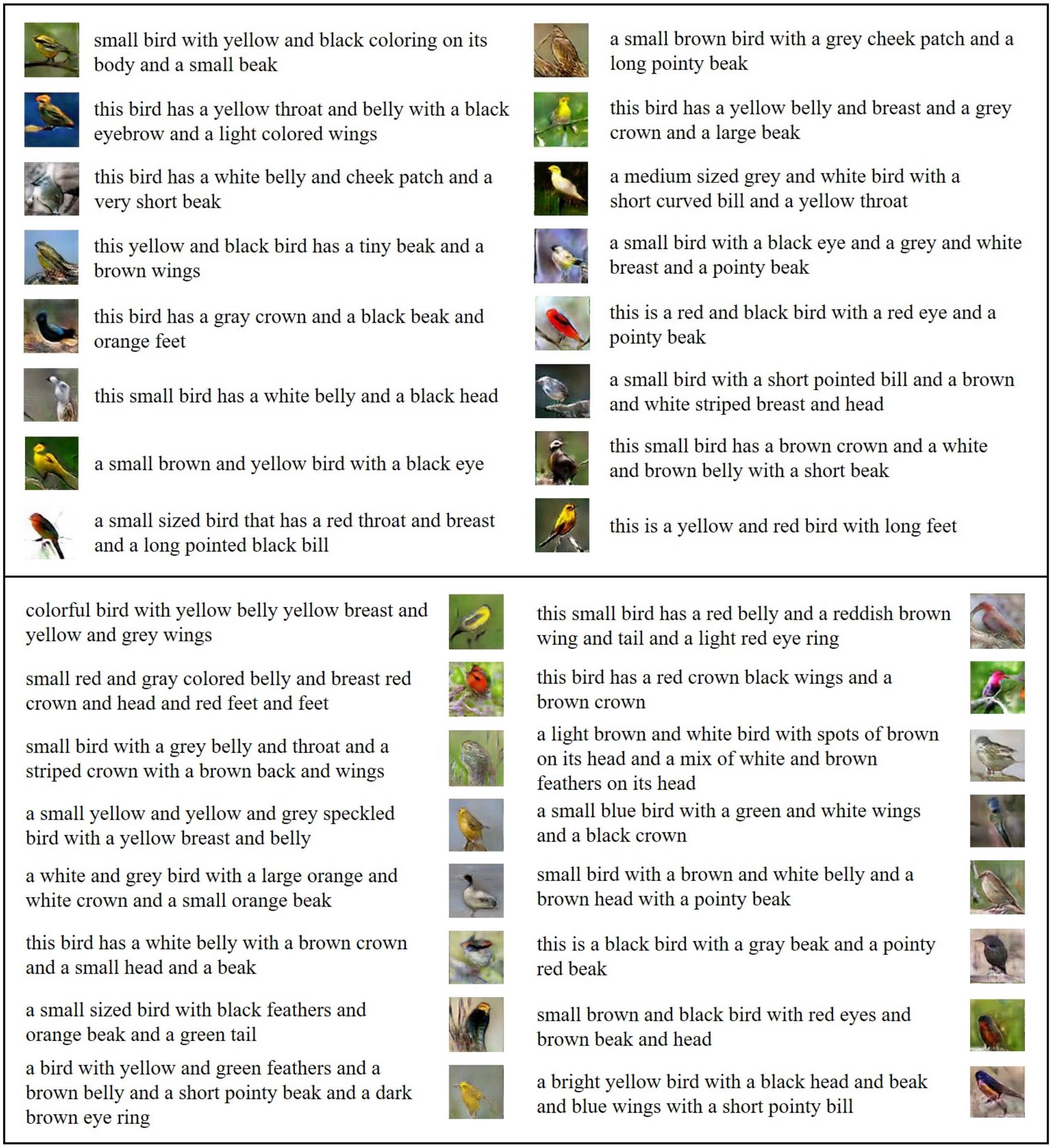

Figure 5. Generated paired samples of caption features and images. Top block: from generated images to caption features. Bottom block: from generated caption features to images.

C.4. Model architecture for JointGAN on modeling caption features and images

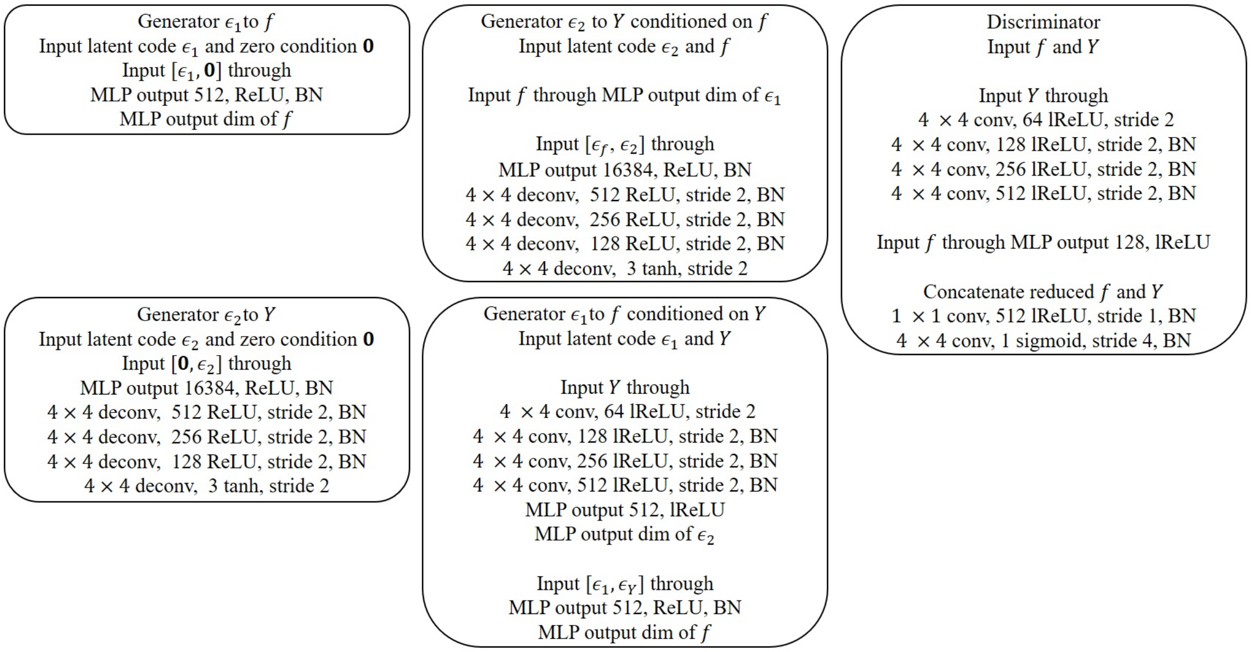

Figure 6. Model architecture for JointGAN on modeling caption features and images.

References

Huang, P.-S., He, X., Gao, J., Deng, L., Acero, A., and Heck, L. Learning deep structured semantic models for web search using clickthrough data. In *CIKM*, 2013.