4th Symposium on Advances in Approximate Bayesian Inference, 2021 1–33

Learning Consistent Deep Generative Models from Sparsely Labeled Data

Anonymous Authors
Anonymous Institution

1. Introduction

We develop broadly applicable methods for learning flexible models of high-dimensional data, like images, that are paired with (discrete or continuous) labels. We are particularly interested in *semi-supervised learning* (SSL) (Zhu, 2005; Oliver et al., 2018) from data that is sparsely labeled, a common situation in practice due to the cost or privacy concerns associated with data annotation. Given a large and sparsely labeled dataset, we seek a single probabilistic model that *simultaneously* makes good predictions of labels and provides a high-quality generative model of the high-dimensional input data.

Prior approaches for semi-supervised learning of deep generative models include *variational autoencoders* (VAEs) (Kingma et al. (2014)), *generative adversarial networks* (GANs) (Dumoulin et al., 2017; Kumar et al., 2017), *normalizing flows* (Nalisnick et al., 2019; Izmailov et al., 2020) and hybrids of these (Larsen et al., 2016; de Bem et al., 2018; Zhang et al., 2019). We favor VAEs due to their ability to learn low-dimensional representations with high predictive accuracy and their ability to evaluate a learned probability density function.

We have three key contributions. First, we expose the conceptual and practical deficiencies of SSL VAEs built on the M2 approach of Kingma et al. (2014). Second, to address these limitations we provide a new downstream SSL framework – *prediction constrained variational autoencoders* (PC-VAEs) – which learns high-quality generative models while simultaneously enforcing accurate predictions. Third, we show that the generative model structure leads to a natural *consistency* constraint vital for effective semi-supervised learning from very sparse labels. Our experiments demonstrate that *consistent prediction-constrained* (CPC) VAE training leads to accuracy competitive with state-of-the-art SSL methods and integrates well with generative modelling improvements such as "very-deep" VAEs (Child, 2021).

2. Background: Semi-supervised VAEs

We now describe VAEs as deep generative models and review previous methods for SSL of VAEs. SSL tasks provide two training datasets: an unsupervised (unlabeled) dataset $\mathcal{D}^U$ of N feature vectors x and a supervised (labeled) dataset $\mathcal{D}^S$ containing M pairs (x, y) of features x and label $y \in \mathcal{Y}$. Labels are often very sparse ($M \ll N$).

The variational autoencoder (Kingma and Welling, 2014) is an unsupervised learning framework with two components: a generative model and an inference model. The generative model defines for each example a joint distribution $p_\theta(x, z)$ over "features" (observed vector

© A. Authors.

$x \in \mathbb{R}^D$) and "encodings" (latent vector $z \in \mathbb{R}^C$). The VAE "inference model" defines an approximate posterior $q_\phi(z \mid x)$, which is trained to be close to the true posterior ($q_\phi(z \mid x) \approx p_\theta(z \mid x)$) but easier to evaluate:

$$p_\theta(x, z) = \mathcal{N}(z \mid 0, I_C) \cdot \mathcal{F}(x \mid \mu_\theta(z), \sigma_\theta(z)), \qquad q_\phi(z \mid x) \quad = \mathcal{N}(z \mid \mu_\phi(x), \sigma_\phi(x)). \tag{1}$$

The likelihood $\mathcal{F}$ is often multivariate normal, where (deterministic) functions μ_θ and σ_θ define the mean and covariance via trainable parameters θ. Given x, the posterior of z is approximated as normal with mean μ_ϕ and (diagonal) covariance σ_ϕ. Here, ϕ may parameterize *multi-layer perceptrons* (MLPs), *convolutional neural networks* (CNNs), or other (deep) neural nets.

We would ideally learn generative parameters θ by maximizing the marginal likelihood $p_\theta(x)$, integrating over z. As this is intractable, we instead maximize a variational bound: $\max_{\theta,\phi} \quad \sum_{x \in \mathcal{D}} \mathcal{L}^{\text{VAE}}(x; \theta, \phi)$, where,

$$\mathcal{L}^{\text{VAE}}(x; \theta, \phi) = \mathbb{E}_{q_\phi(z|x)} \left[\log \frac{p_\theta(x,z)}{q_\phi(z|x)} \right] \leq \log p_\theta(x). \tag{2}$$

This expectation can be evaluated via Monte Carlo samples from the inference model $q_\phi(z|x)$. Gradients with respect to θ, ϕ can be similarly estimated by the reparameterization "trick" of representing $q_\phi(z \mid x)$ as a linear transformation of standard normal variables (Kingma and Welling, 2014; Rezende et al., 2014).

2.1. Two-Stage SSL: VAE then Predict

VAEs may be used for SSL via a *two-stage* "VAE + GLM". First, train a VAE to maximize the unsupervised likelihood (2) of *all* features x (both labeled $\mathcal{D}^S$ and unlabeled $\mathcal{D}^U$). Second, learn a label-from-code *predictor* $\hat{y}_w(z)$ that maps latent codes z to prediction scores. Our experiments use a *generalized linear model* (GLM) with weights w trained to minimize the cross-entropy loss $\sum_{x,y \in \mathcal{D}^S} \mathbb{E}_{q_\phi(z|x)} \left[\ell_S(y, \hat{y}_w(z)) \right]$. This second stage fixes ϕ and uses only the small labeled dataset.

2.2. SSL of VAEs via Joint Likelihoods

Motivated by limitations of two-stage SSL, Kingma et al. (2014) proposed a VAE-inspired model called "M2" for *joint* generative modeling of labels y and data x. The M2 generative model first samples labels y with frequencies π, and then features x: $p_\theta(x, y, z) = \mathcal{N}(z \mid 0, I_C) \cdot \text{Cat}(y \mid \pi) \cdot \mathcal{F}(x \mid \mu_\theta(y, z), \sigma_\theta(y, z))$. M2 inference sets $q_\phi(y, z \mid x) = q_{\phi^{y|x}}(y \mid x) q_{\phi^{z|x,y}}(z \mid x, y)$.

To train M2, Kingma et al. (2014) maximize the likelihood of all observations:

$$\max_{\theta, \phi^{y|x}, \phi^{z|x,y}} \quad \sum_{x,y \in \mathcal{D}^S} \mathcal{L}^S(x, y; \theta, \phi^{z|x,y}) + \sum_{x \in \mathcal{D}^U} \mathcal{L}^U(x; \theta, \phi^{y|x}, \phi^{z|x,y}). \tag{3}$$

Like unsupervised VAEs, Eq. (3) and its gradients may be approximated via samples from the variational posterior. The first, "supervised" term bounds the feature-and-label *joint* likelihood: $\log p_\theta(x, y) \geq \mathcal{L}^S$,

$$\mathcal{L}^S(x, y; \theta, \phi^{z|x,y}) = \mathbb{E}_{q_{\phi^{z|x,y}}(z|x,y)} \left[\log \frac{p_\theta(x,y,z)}{q_{\phi^{z|x,y}}(z|x,y)} \right].$$

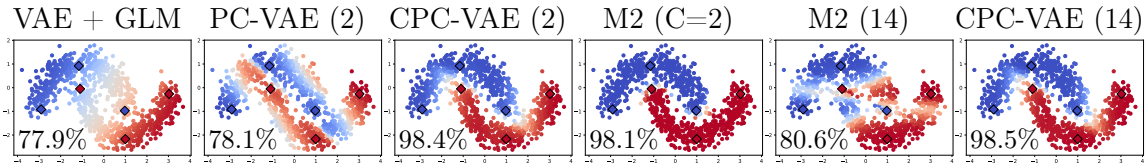


Figure 1: Predictions on half-moon classification (accuracy in corner) for semi-supervised VAE learning from 6 labeled examples (diamonds) and 994 unlabeled examples. Dots are 2-dim. feature vectors colored by predicted probability of mostly likely label. Titles indicate encoding size $C = 2$ or $C = 14$. M2 (Kingma et al., 2014) accuracy *deterioriates* when capacity increases from $C = 2$ to 14 (drop from 98.1% to 80.6% accuracy). Our CPC-VAE is reliable at any capacity.

The second, "unsupervised" term is a variational bound for the features-only likelihood $\log p_\theta(x) \geq \mathcal{L}^U$, where $\mathcal{L}^U = \mathbb{E}_{q_\phi(y,z|x)}\left[\log \frac{p_\theta(x,y,z)}{q_\phi(y,z|x)}\right]$ in terms of $\mathcal{L}^S$ is:

$$\mathcal{L}^U = \sum_{y \in \mathcal{Y}} q_{\phi^{y|x}}(y \mid x) \left(\mathcal{L}^S - \log q_{\phi^{y|x}}(y \mid x)\right). \tag{4}$$

M2's prediction dilemma and heuristic fix. After training parameters θ, ϕ, we need to predict labels y given test data x. M2's structure assumes we make predictions via the inference model's discriminator $q_{\phi^{y|x}}(y \mid x)$. However, the discriminator's parameter $\phi^{y|x}$ is only informed by the *unlabeled* data via the objective $\mathcal{L}^U$ of (4); it is not used to compute $\mathcal{L}^S$. We cannot expect good predictions from a parameter that does not touch labels.

To partially overcome this issue, Kingma et al. (2014) use a weighted objective:

$$\max_{\theta,\phi} \sum_{x,y \in \mathcal{D}^S} \left(\alpha \log q_{\phi^{y|x}}(y \mid x) + \lambda \mathcal{L}^S(x, y; \theta, \phi^{z|x,y})\right) + \sum_{x \in \mathcal{D}^U} \mathcal{L}^U(x; \theta, \phi^{y|x}, \phi^{z|x,y}). \tag{5}$$

This objective pushes the inference model's discriminator $q_{\phi^{y|x}}(y|x)$ to do well on the labeled set via an extra loss term (weighted by hyperparameter $\alpha > 0$). Weight $\lambda > 0$ further balances supervised and unsupervised terms.

3. CPC Variational Autoencoders

3.1. Prediction Constrained VAEs

We develop a framework for *jointly* learning a generative model of features x and making label-given-feature predictions $\hat{y}(x)$ of uncompromised quality, by requiring predictions to meet a user-specified quality threshold. Our *prediction constrained* training objective enables end-to-end estimation of all parameters while incorporating the task-specific costs relevant in evaluation ("test") scenarios, via user-chosen loss functions.

Generative model. Our generative model does not include labels y, only features x and encodings z. The joint distribution $p_\theta(x, z)$ and inference model $q_\phi(z \mid x)$ factorize as in the unsupervised VAE of Eq. (1). While M2 included labels in its generative model (Kingma et al., 2014), our goals are different: we seek to predict labels from features, but do not need other conditionals.

Label-from-feature prediction. To predict labels y from features x, we sample encoding $z \sim q_\phi(z|x)$ from the inference model and then predict a label $\hat{y}_w(z)$ as in the two-stage method of Sec. 2.1. By *sharing* latent code z, the generative model is involved in

predictions. Li et al. (2019) promote the robustness properties of this downstream model structure, which corresponds to their "GBZ" architecture, but do not explore SSL.

Constrained PC objective. Unlike two-stage models, our approach does not predict post-hoc with a previously learned generative model. Instead, we train the predictor simultaneously with the generative model via a novel *prediction-constrained* (PC) objective:

$$\max_{\theta,\phi^{z|x},w} \quad \sum_{x\in\mathcal{D}^U\cup\mathcal{D}^S} \mathcal{L}^{\text{VAE}}(x;\theta,\phi^{z|x}), \quad \text{subj. to:} \quad \frac{1}{M}\sum_{x,y\in\mathcal{D}^S} \underbrace{\mathbb{E}_{q_\phi(z|x)}[\ell_S(y,\hat{y}_w(z))]}_{\mathcal{P}(x,y;\phi^{z|x},w)} \leq \epsilon. \quad (6)$$

The constraint requires that any feasible solution achieve average prediction loss less than $\epsilon > 0$ on the labeled training set. The loss function ℓ_S may be flexibly specified based on task-specific needs.

Unconstrained PC objective. Using the KKT conditions, we define an equivalent unconstrained objective that maximizes the likelihood but penalizes inaccurate predictions:

$$\max_{\theta,\phi^{z|x},w} \quad \sum_{x\in\mathcal{D}^U\cup\mathcal{D}^S} \mathcal{L}^{\text{VAE}}(x;\theta,\phi^{z|x}) - \lambda \sum_{x,y\in\mathcal{D}^S} \mathcal{P}(x,y;\phi^{z|x},w). \quad (7)$$

Here $\lambda > 0$ is a Lagrange multiplier chosen to ensure that the target prediction constraint is achieved; smaller loss tolerances ϵ require larger values of λ. This PC objective, and gradients for parameters θ, ϕ, w, can be estimated via Monte Carlo samples from $q_\phi(z \mid x)$.

Justification. While the PC objective of Eq. (7) may look superficially similar to Eq. (5), we emphasize two key differences. First, it couples a generative likelihood and a prediction loss via shared variational parameters $\phi^{z|x}$. This makes both generative and discriminative performance depend on the *same* learned encoding z. In contrast, the M2 objective uses a label-given-features predictor that does not share *any* parameters $\phi^{y|x}$ with the supervised likelihood $\mathcal{L}^S$. Our approach may coherently use *any* amount of labels (from very sparse to fully-labeled) to inform the generative model, while fully-labeled M2 decouples the generative and discriminative models. Second, our objective is more affordable: no term requires an expensive marginalization over labels (or lossy approximation to avoid this marginalization), easing applications to big unlabeled datasets and enabling learning from continuous labels.

Hyperparameters. The major hyperparameter influencing PC training is the constraint multiplier $\lambda \geq 0$. Setting $\lambda = 0$ leads to unsupervised maximum likelihood training (or MAP training, given priors on θ) of a classic VAE. Setting $\lambda = 1$ and choosing a probabilistic loss $-\log p(y \mid z)$ produces a "supervised VAE" that maximizes the joint likelihood $p_\theta(x, y)$. In practice we use validation data to select the best of several candidate λ values.

3.2. Improved Predictions via Consistency

While the PC objective is effective given sufficient labeled data (see supplement), it may generalize poorly when labels are very sparse (see Fig. 1). This fundamental problem arises because in the PC objective of Eq. (9), the parameters w of the predictor $\hat{y}_w(z)$ are only informed by the small labeled training dataset.

Let $x \sim p_\theta(\cdot \mid z')$ and $\bar{x} \sim p_\theta(\cdot \mid z')$ be two observations sampled from the *same* latent code z'. Even if the true label y of x is uncertain, we know that for this model to be useful for predictive tasks, $\bar{x}$ must have the *same* label as x. We formalize this (and dramatically

boost performance) via a *consistency constraint* requiring label predictions for common-code data pairs $(x, \bar{x})$ to approximately match .

Given x, we predict labels $\hat{y}_w(z)$ via codes $z \sim q_\phi(z \mid x)$. Alternatively, given x we can *simulate* alternative features $\bar{x}$ with matching code z by sampling from the inference and generative models, and then predict the label for $\bar{x}$. We force the label predictions y for x, and $\bar{y}$ for $\bar{x}$, to be similar via a (cross-entropy) consistency penalty $\ell_C(y, \bar{y})$. We constrain the maximum consistency violations on *unlabeled* and *labeled* data:

$$\mathcal{C}^U(x; \theta, \phi, w) \triangleq \mathbb{E}_{q_\phi(z|x)} \left[\mathbb{E}_{p_\theta(\bar{x}|z)} \left[\mathbb{E}_{q_\phi(\bar{z}|\bar{x})} \left[\ell_C(\hat{y}_w(z), \hat{y}_w(\bar{z})) \right] \right] \right], \tag{8}$$

$$\mathcal{C}^S(x, y; \theta, \phi, w) \triangleq \mathbb{E}_{q_\phi(z|x)} \left[\mathbb{E}_{p_\theta(\bar{x}|z)} \left[\mathbb{E}_{q_\phi(\bar{z}|\bar{x})} \left[\ell_C(y, \hat{y}_w(\bar{z})) \right] \right] \right].$$

Consistent PC: Unconstrained objective. To train parameters, we use multiplier $\gamma > 0$ to enforce consistency constraints for both unlabeled and labeled features , yielding the objective:

$$\max_{\theta,\phi,w} \sum_{x \in \mathcal{D}^U \cup \mathcal{D}^S} \mathcal{L}^{\text{VAE}}(x; \theta, \phi) - \sum_{x \in \mathcal{D}^U} \gamma \mathcal{C}^U(x; \theta, \phi, w) + \sum_{x,y \in \mathcal{D}^S} -\lambda \mathcal{P}(x, y; \phi, w) - \gamma \mathcal{C}^S(x, y; \theta, \phi, w),$$

where $\mathcal{L}$ is the unsupervised likelihood bound, $\mathcal{P}$ is the predictor loss, and $\mathcal{C}^U, \mathcal{C}^S$ are the consistency costs.

Aggregate label consistency. We further regularize predictions with an *aggregate label consistency* constraint, which forces the distribution of label predictions to be close to a target distribution π (typically, the empirical distribution of $\mathcal{D}^S$). This discourages predictions on unlabeled examples from collapsing to a single value. We define the aggregate consistency loss as $\ell_A(\pi, \mathbb{E}_{x \sim \mathcal{D}^U, z \sim q(z|x)}[\hat{y}_w(z)])$, and again use a cross-entropy penalty.

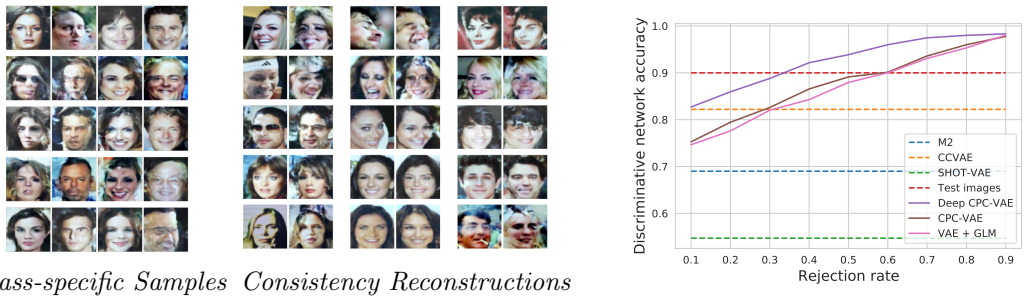
Generative Model Innovations By design, our CPC-VAE may boost SSL performance by incorporating advances in generative models. Our experiments explore three improvements to the basic VAE: noise-robust pixel likelihoods, affine transformations for poorly-aligned data, and "very deep" VAEs with many stochastic layers (Child, 2021). More details about all these innovations are in the supplement.

4. Experiments

We compare our consistent prediction-constrained (CPC) VAE to baselines on two goals: useful generative modeling of images x and classification accuracy of y given x. For our methods, we use encoder/decoder networks based on the WRN-28-2 architecture (Zagoruyko and Komodakis, 2016) and train with Adam (Kingma and Ba, 2014) using balanced minibatches of 50% labeled and 50% unlabeled data. Hyperparameter search used Optuna (Akiba et al., 2019) to maximize validation accuracy. Reproducible details are in the supplement.

Datasets. For the CelebA dataset (Liu et al., 2015) with 1000 labeled and 159,770 unlabeled images, we predict 4 classes that combine gender (woman/man) and facial expression (neutral/smiling). We also test SVHN (Netzer et al., 2011) and MNIST (LeCun et al., 2010)

CPC-VAEs improve SSL classification accuracy. In Tab. 1, CPC achieves the top accuracy (86.22%) among all methods (6 VAEs, 2 discrim.) on CelebA. On SVHN, CPC also tops the SSL VAEs while matching the non-generative VAT. CPC also is in the top 3 of all methods on MNIST, less than 0.4% from the leader.



(a) Generative examples

(b) Classified accuracy

Figure 2: 2a: *Left:* Samples from the learned generative model conditioned on class (by column: *neutral woman, neutral man, smiling woman, smiling man*). Samples are chosen via rejection sampling in the latent space with a threshold of 95% confidence in the target class. *Right:* Reconstructions of test images. Each pair shows an image and sample sharing only the deepest stochastic layer, sampling other layers. 2b: Evaluation of class-conditional generation on CelebA. We assess the ability of generative models (trained with 1000 labels) to produce samples from a target class recognizable by an independent classifier (discriminative WRN) trained on fully labeled Celeb-A data. For the CPC-VAE and VAE + GLM models, the label is not a discrete input to the generative model. We therefore draw class-conditional samples based on label *confidence* using the following process: *(1)* Draw samples from the latent prior $p(z)$, *(2)* divide according to the prediction made by the latent classifier $\hat{y}_w(z)$, *(3)* reject a percentage of the samples with the lowest confidence in the predicted label, *(4)* reconstruct images from $p_\theta(x|z)$ using the accepted latent samples. We show the accuracy of the independent classifier as a function of the rejection rate.

SSL?	Gen?	Source	Method	MNIST (100)	SVHN (1000)	CelebA (1000)
✓	✓	ours	**CPC-VAE**	98.29 (±0.50)	**94.22** (±0.62)	**86.22**
✓	✓	Tab. 1-2 of Kingma et al.	M1 + M2	96.67 (±0.14)	63.98 (±0.10)	*79.28*
✓	✓	Tab. 2 of Maaløe et al.	SDGM	**98.68** (±0.07)	83.39 (±0.24)	-
✓	✓	Tab. 3 of Feng et al.	SHOT-VAE	96.88 (±0.22)	71.18 (±0.49)	*77.1*
✓	✓	Tab. 6 of Joy et al.	CCVAE	92.7 (200 labels)	-	*84.20*
✓		Tab. 3-4 of Miyato et al.	VAT	**98.64** (±0.03)	**94.23** (±0.32)	*81.48*
		ours	Discrim.	73.91 (±1.45)	87.7 (±1.02)	76.10

Table 1: SSL image classification across methods and datasets, reporting mean test set accuracy (+/- std. dev.) across 10 runs on distinct random samples of the labeled set; only 1 run is feasible on large CelebA. Check in first column indicates the method uses both unlabeled and labeled data, not just the labeled set. Check in second column indicates the method is a generative model. Italicized entries indicate our own experimental results using the cited methods on the 4-class CelebA task, matching architectures and preprocessing to our CPC-VAE. For reproducibility details, see supplement.

CPC-VAEs generate better images. Because predictive likelihoods may be an ineffective measure of generative model quality (Theis et al., 2016), we use a *reclassification* paradigm (like Joy et al. (2021)) for assessment of label-informed generative models. Fig. 2shows that the CPC-VAE, and especially the (very) Deep CPC-VAE, generate images that are better recognized (by an independent classifier) as examples of the intended class. Samples and reconstructions from our Deep CPC-VAE are visually plausible (Fig. 2a). See the supplement for visuals for other models and datasets.

References

Martín Abadi, Ashish Agarwal, Paul Barham, et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. MixMatch: A Holistic Approach to Semi-Supervised Learning. In *Advances in Neural Information Processing Systems*, 2019. URL http://arxiv.org/abs/1905.02249.

Ming-Wei Chang, Lev Ratinov, and Dan Roth. Guiding Semi-Supervision with Constraint-Driven Learning. In *Proc. of the Annual Meeting of the Association for Computational Linguistics*, 2007.

Rewon Child. Very deep VAEs generalize autoregressive models and can outperform them on images. *ICLR conference paper, arXiv:2011.10650v2*, 2021.

Robert G Cowell, Philip Dawid, Steffen L Lauritzen, and David J Spiegelhalter. *Probabilistic networks and expert systems: Exact computational methods for Bayesian networks*. Springer Science & Business Media, 2006.

Rodrigo de Bem, Arnab Ghosh, Thalaiyasingam Ajanthan, Ondrej Miksik, N. Siddharth, and Philip Torr. A Semi-supervised Deep Generative Model for Human Body Analysis. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018. URL http://openaccess.thecvf.com/content_eccv_2018_workshops/w11/html/de_A_Semi-supervised_Deep_Generative_Modelfor_Human_Body_Analysis_ECCVW_2018_paper.html.

Terrance DeVries, Ishan Misra, and Changhan Wang. Does Object Recognition Work for Everyone? In *1st Workshop on Computer Vision for Global Challenges*, 2019.

Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially Learned Inference. In *International Conference on Learning Representations (ICLR)*, 2017.

Hao-Zhe Feng, Kezhi Kong, Minghao Chen, Tianye Zhang, Minfeng Zhu, and Wei Chen. SHOT-VAE: Semi-supervised Deep Generative Models With Label-aware ELBO Approximations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, number 8, 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/16909.

Michael Figurnov, Shakir Mohamed, and Andriy Mnih. Implicit reparameterization gradients. In *Advances in Neural Information Processing Systems*, 2018.

Kuzman Ganchev, João Graça, Jennifer Gillenwater, and Ben Taskar. Posterior Regularization for Structured Latent Variable Models. *Journal of Machine Learning Research*, 11:2001–2049, 2010.

Jonathan Gordon and José Miguel Hernández-Lobato. Combining deep generative and discriminative models for Bayesian semi-supervised learning. *Pattern Recognition*, 100, 2020.

Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Proceedings of the 17th International Conference on Neural Information Processing Systems*, NIPS'04, page 529–536, Cambridge, MA, USA, 2004. MIT Press.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations (ICLR)*, 2017.

Michael C. Hughes, Gabriel Hope, Leah Weiner, Thomas H. McCoy, Roy H. Perlis, Erik B. Sudderth, and Finale Doshi-Velez. Semi-Supervised Prediction-Constrained Topic Models. In *Artificial Intelligence and Statistics*, 2018. URL http://proceedings.mlr.press/v84/hughes18a.html.

Maximilian Ilse, Jakub M. Tomczak, Christos Louizos, and Max Welling. DIVA: Domain Invariant Variational Autoencoders. In *Proceedings of the Third Conference on Medical Imaging with Deep Learning*, pages 322–348. PMLR, 2020. URL https://proceedings.mlr.press/v121/ilse20a.html.

Pavel Izmailov, Polina Kirichenko, Marc Finzi, and Andrew Gordon Wilson. Semi-Supervised Learning with Normalizing Flows. In *International Conference on Machine Learning*, pages 4615–4630. PMLR, 2020. URL http://proceedings.mlr.press/v119/izmailov20a.html.

Tommi S Jaakkola, Marina Meila, and Tony Jebara. Maximum entropy discrimination. In *Advances in Neural Information Processing Systems*, 1999.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, and koray kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, 2015. URL http://papers.nips.cc/paper/5854-spatial-transformer-networks.pdf.

Ananya Harsh Jha, Saket Anand, Maneesh Singh, and V. S. R. Veeravasarapu. Disentangling Factors of Variation with Cycle-Consistent Variational Auto-encoders. In *European Conference on Computer Vision (ECCV)*. Springer International Publishing, 2018.

Tom Joy, Sebastian Schmon, Philip Torr, Siddharth N, and Tom Rainforth. Capturing Label Characteristics in VAEs. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=wQRlSUZ5V7B.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, 2014. URL http://arxiv.org/abs/1412.6980.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations*, 2014. URL http://arxiv.org/abs/1312.6114.

Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, 2014. URL https://papers.nips.cc/paper/5352-semi-supervised-learning-with-deep-generative-models.pdf.

Volodymyr Kuleshov and Stefano Ermon. Deep Hybrid Models: Bridging Discriminative and Generative Approaches. In *Uncertainty in Artificial Intelligence*, page 10, 2017.

Abhishek Kumar, Prasanna Sattigeri, and Tom Fletcher. Semi-supervised Learning with GANs: Manifold Invariance with Improved Inference. In *Advances in Neural Information Processing Systems*, 2017. URL https://papers.nips.cc/paper/7137-semi-supervised-learning-with-gans-manifold-invariance-with-improved-inference.pdf.

Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *International Conference on Machine Learning*, pages 1558–1566, 2016. URL http://proceedings.mlr.press/v48/larsen16.html.

Julia A. Lasserre, Christopher M. Bishop, and Thomas P. Minka. Principled Hybrids of Generative and Discriminative Models. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, CVPR '06, pages 87–94, USA, 2006. IEEE Computer Society.

Y LeCun, F. J. Huang, and L. Bottou. Learning Methods for Generic Object Recognition with Invariance to Pose and Lighting. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2004. URL http://yann.lecun.com/exdb/publis/pdf/lecun-04.pdf.

Yann LeCun, Corinna Cortes, and CJ Burges. MNIST handwritten digit database, 2010. URL http://yann.lecun.com/exdb/mnist/.

C. Li, J. Zhu, and B. Zhang. Max-Margin Deep Generative Models for (Semi-)Supervised Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(11): 2762–2775, 2018.

Yingzhen Li, John Bradshaw, and Yash Sharma. Are Generative Classifiers More Robust to Adversarial Attacks? In *Proceedings of the 36th International Conference on Machine Learning*, pages 3804–3814. PMLR, 2019. URL https://proceedings.mlr.press/v97/li19a.html.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. Auxiliary Deep Generative Models. *arXiv:1602.05473 [cs, stat]*, 2016. URL http://arxiv.org/abs/1602.05473.

David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. Fairness through Causal Awareness: Learning Causal Latent-Variable Models for Biased Data. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, pages 349–358, Atlanta, GA, USA, 2019. Association for Computing Machinery.

Gideon S Mann and Andrew McCallum. Generalized expectation criteria for semi-supervised learning with weakly labeled data. *Journal of Machine Learning Research*, 11(Feb):955–984, 2010.

Andrew McCallum, Chris Pal, Greg Druck, and Xuerui Wang. Multi-Conditional Learning: Generative/Discriminative Training for Clustering and Classification. In *AAAI Conference on Artificial Intelligence*, 2006. URL https://www.aaai.org/Papers/AAAI/2006/AAAI06-069.pdf.

Matthew B A McDermott, Tom Yan, Tristan Naumann, Nathan Hunt, Harini Suresh, Peter Szolovits, and Marzyeh Ghassemi. Semi-Supervised Biomedical Translation with Cycle Wasserstein Regression GANs. In *Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, page 8, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/viewFile/16938/15951.

Michele Merler, Nalini Ratha, Rogerio S. Feris, and John R. Smith. Diversity in Faces. *arXiv:1901.10436 [cs]*, 2019. URL http://arxiv.org/abs/1901.10436.

Andrew C Miller, Ziad Obermeyer, John P Cunningham, and Sendhil Mullainathan. Discriminative Regularization for Latent Variable Models with Applications to Electrocardiography. In *International Conference on Machine Learning*, page 10, 2019. URL https://proceedings.mlr.press/v97/miller19a/miller19a.pdf.

Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, Timnit Gebru, et al. Model Cards for Model Reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, pages 220–229, Atlanta, GA, USA, 2019. Association for Computing Machinery.

Shira Mitchell, Eric Potash, Solon Barocas, Alexander D'Amour, and Kristian Lum. Prediction-Based Decisions and Fairness: A Catalogue of Choices, Assumptions, and Definitions. *arXiv:1811.07867 [stat]*, 2018. URL http://arxiv.org/abs/1811.07867.

Takeru Miyato, Shin-Ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1979–1993, 2019. URL https://ieeexplore.ieee.org/document/8417973/.

Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Hybrid Models with Deep and Invertible Features. In *International Conference on Machine Learning*, pages 4723–4732. PMLR, 2019. URL http://proceedings.mlr.press/v97/nalisnick19b.html.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading Digits in Natural Images with Unsupervised Feature Learning. In *NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011. URL http://ufldl.stanford.edu/housenumbers.

Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems*, pages 3235–3246, 2018.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *International Conference on Machine Learning*, pages 1278–1286, 2014. URL http://proceedings.mlr.press/v32/rezende14.pdf.

Oleh Rybkin, Kostas Daniilidis, and Sergey Levine. Simple and effective vae training with calibrated decoders, 2020.

Shreya Shankar, Yoni Halpern, Eric Breck, James Atwood, Jimbo Wilson, and D. Sculley. No Classification without Representation: Assessing Geodiversity Issues in Open Data Sets for the Developing World. In *NeurIPS 2017 Workshop on Machine Learning for the Developing World*, 2017. URL http://arxiv.org/abs/1711.08536.

Marek Smieja, Maciej Wolczyk, Jacek Tabor, and Bernhard C. Geiger. SeGMA: Semi-Supervised Gaussian Mixture Autoencoder. *IEEE transactions on neural networks and learning systems*, 32(9):3930–3941, 2021.

Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder Variational Autoencoders. In *Advances in Neural Information Processing Systems*, 2016. URL https://proceedings.neurips.cc/paper/2016/file/6ae07dcb33ec3b7c814df797cbda0f87-Paper.pdf.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. In *4th International Conference on Learning Representations*, 2016. URL http://arxiv.org/abs/1511.01844.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V. Le. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems*, 2020.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Edwin R. Hancock Richard C. Wilson and William A. P. Smith, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 87.1–87.12. BMVA Press, September 2016. ISBN 1-901725-59-6. doi: 10.5244/C.30.87. URL https://dx.doi.org/10.5244/C.30.87.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Xiang Zhang, Lina Yao, and Feng Yuan. Adversarial Variational Embedding for Robust Semi-supervised Learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 139–147, Anchorage AK USA, 2019. ACM.

Tinghui Zhou, Philipp Krahenbuhl, Mathieu Aubry, Qixing Huang, and Alexei A. Efros. Learning Dense Correspondence via 3D-Guided Cycle Consistency. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 117–126, Las Vegas, NV, USA, 2016. IEEE.

Jun Zhu, Ning Chen, and Eric P Xing. Bayesian inference with posterior regularization and applications to infinite latent SVMs. *Journal of Machine Learning Research*, 15(1): 1799–1847, 2014.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, Venice, 2017. IEEE.

Xiaojin Zhu. Semi-Supervised Learning Literature Survey. Technical Report Technical Report 1530, Department of Computer Science, University of Wisconsin Madison., 2005.

Appendix A. Code Availability

Code for our methods is available for download here:
https://www.dropbox.com/s/b43xcsnhs5kevue/PC-VAE-REVIEW-RELEASE.zip?dl=1.
It is available for browsing here:
https://anonymous.4open.science/repository/0fa4ef53-5e92-40df-ad0a-bcdab28f6df0/pcvae/. Code for our very-deep VAE model is forthcoming.

Appendix B. Methods: Very Deep CPC-VAE

For large-scale experiments on complex datasets such as Celeb-A, we employ a variant of the *very-deep VAE* proposed by (Child, 2021). This model uses a ladder-VAE (Sønderby et al., 2016) structure that operates at progressively finer-scales in the generative process. Please refer to these works for the specifics of the model, here we focus on our modifications to this architecture.

In the original very-deep VAE, the initial "bottom-up" encoder produces parameters for the variational posterior of the topmost stochastic layer $q_\phi(z_0 \mid x)$, while intermediate outputs from this network are used as ladder connections influencing corresponding variational distributions for intermediate stochastic layers. Figure 3(left) illustrates this structure, where $d_K, \ldots, d_1, d_0$ are (deterministic) layer outputs from the bottom up encoder network. We let K equal the number of intermediate stochastic layers (2 for the simplified model in Fig. 3, many more in our experiments).

For our very-deep CPC-VAE we modify this structure by splitting the bottom-up encoder into two separate networks. We retain the ladder-structured bottom up network (parameters ξ) to influence the approximate posteriors for intermediate layers $(q_\xi(z_1|x), q_\xi(z_2|x), \ldots)$. We do not modify the architecture of this network, but we do not use its final output $q_\xi(z_0|x)$.

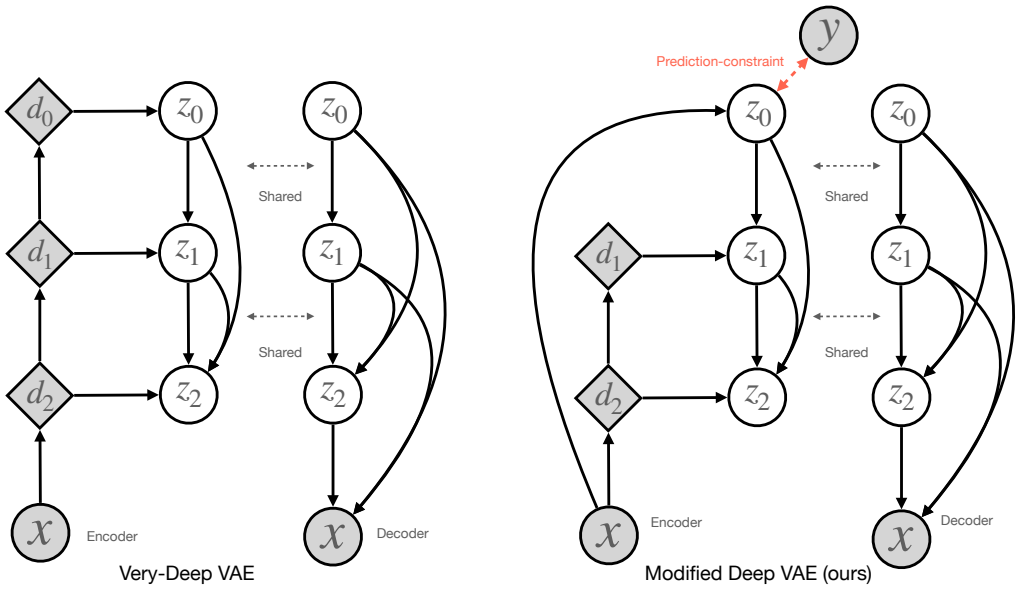


Figure 3: Comparison of the VAE model architecture proposed by (Child, 2021) and the modified version used by our CPC-VAE, which applies prediction and consistency constraints to z_0. Shaded nodes indicate observed variables, while diamond nodes are deterministic intermediate outputs of the encoder network. As in (Child, 2021), portions of the network structure are shared between the encoder and decoder.

Instead we introduce a separate encoder network for $q_\phi(z_0|x)$, with parameters denoted ϕ. For this network we use the same WRN architecture employed in our single stochastic layer VAE experiments. Figure 3 (right) illustrates this modified encoder structure. The generative model remains unchanged, as it is not influenced by ladder connections. We find that this model architecture helps encourage consistency that affects the entire generative hierarchy, leading to higher test accuracy. Note that this change also does not affect the bottom-up factorization of the variational distribution. As in prior work, $p(z)$ and $q(z|x)$ are factorized as follows:

$$p_\theta(z) = p(z_0)p_\theta(z_1|z_0)\cdots p_\theta(z_K|z_{<K})),$$
$$q_{\phi,\xi}(z) = q_\phi(z_0|x)q_\xi(z_1|z_0,x)...q_\xi(z_K|z_{<K},x).$$

PC-VAE Architecture: For the PC-VAE and CPC-VAE the prediction constraint is applied only to a subset of the latent variables, specifically those of the topmost stochastic layer (z_0), so that the constraint only affects a small number of global latent variables. With this structure implicit label information is accessible at every scale in the generative process. Our (constrained) PC-VAE objective becomes:

$$\max_{\theta,\phi^{z|x},w} \sum_{x\in\mathcal{D}^U\cup\mathcal{D}^S} \mathcal{L}^{\text{VAE}}(x;\theta,\phi^{z_0|x},\xi^{z_{>0}|x,z_0}), \quad \text{subj. to:} \quad \frac{1}{M}\sum_{x,y\in\mathcal{D}^S} \underbrace{\mathbb{E}_{q_\phi(z_0|x)}[\ell_S(y,\hat{y}_w(z_0))]}_{\mathcal{P}(x,y;\phi^{z_0|x},w)} \leq \epsilon. \tag{9}$$

This design simplifies the classification structure and limits over-fitting. Due to the multi-scale structure of the very-deep VAE, latent variables lower in the hierarchy are highly localized, making them less suitable as features for predicting global image classes.

CPC-VAE Architecture: When applying our consistency constraint, we follow our assumption that z_0 should fully determine the class of an image and thus we condition our consistency reconstruction only on the topmost stochastic layer, z_0. In practice, this preserves the global representation relevant to the class label, while allowing consistency reconstructions to exhibit significant local variations. The distribution for "neighboring" images $\bar{x}$ assumed to have consistent class labels is defined as:

$$p_\theta(\bar{x}|z_0) \propto p_\theta(\bar{x}|z_{\leq K})p_\theta(z_K|z_{<K})...p_\theta(z_1|z_0), \tag{10}$$

$$q_{\phi,\theta}(\bar{x}|x) = p_\theta(\bar{x}|z_0)q_\phi(z_0|x). \tag{11}$$

Our corresponding supervised and unsupervised consistency losses are then defined as:

$$\mathcal{C}^U(x;\theta,\phi,w) \triangleq \mathbb{E}_{q_\phi(z_0|x)}\left[\mathbb{E}_{p_\theta(\bar{x}|z_0)}\left[\mathbb{E}_{q_\phi(\bar{z}_0|\bar{x})}\left[\ell_C(\hat{y}_w(z),\hat{y}_w(\bar{z}_0))\right]\right]\right], \tag{12}$$

$$\mathcal{C}^S(x,y;\theta,\phi,w) \triangleq \mathbb{E}_{q_\phi(z_0|x)}\left[\mathbb{E}_{p_\theta(\bar{x}|z_0)}\left[\mathbb{E}_{q_\phi(\bar{z}_0|\bar{x})}\left[\ell_C(y,\hat{y}_w(\bar{z}_0))\right]\right]\right]. \tag{13}$$

Appendix C. Methods: Robust Likelihoods with Spatial Transformations

C.1. Noise-Normal Likelihood

As discussed in Sec. 3.4, we use a "Noise-Normal" distribution as the pixel likelihood for many of our experiments. We define this distribution to be a parameterized two-component mixture of a *truncated-normal* distribution and a *uniform* distribution. We will use ρ to denote the mixture probability of the Normal component, and μ and σ to denote the mean and standard deviation of the truncated-normal, respectively. The generative model (or decoder) predicts a distinct outlier probability $(1-\rho)$ for each pixel. We assume that pixel intensities are defined on the domain $[-1,1]$ and rescale our datasets to match. We can write the *probability density function* of the Noise-Normal distribution via the standard normal PDF $\phi(\cdot)$, and standard normal CDF $\Phi(\cdot)$, as follows:

$$f(x \mid \rho,\mu,\sigma) = \rho\left(\frac{\phi\left(\frac{x-\mu}{\sigma}\right)}{\Phi\left(\frac{1-\mu}{\sigma}\right)-\Phi\left(\frac{-1-\mu}{\sigma}\right)}\right) + (1-\rho)\left(\frac{1}{2}\right). \tag{14}$$

We can similarly express the *cumulative distribution function* of the Noise-Normal distribution as:

$$F(x \mid \rho,\mu,\sigma) = \rho\left(\frac{\Phi\left(\frac{x-\mu}{\sigma}\right)-\Phi\left(\frac{-1-\mu}{\sigma}\right)}{\Phi\left(\frac{1-\mu}{\sigma}\right)-\Phi\left(\frac{-1-\mu}{\sigma}\right)}\right) + (1-\rho)\left(\frac{x+1}{2}\right). \tag{15}$$

In order to propagate gradients through the sampling process of the noise-normal distribution, we use the *implicit reparameterization gradients* approach of (Figurnov et al., 2018). Given

a sample x drawn from this distribution, we compute the gradient with respect to the parameters ρ, μ, and σ as:

$$\nabla_{\rho,\mu,\sigma} x = \frac{-\nabla_{\rho,\mu,\sigma} F(x \mid \rho, \mu, \sigma)}{f(x \mid \rho, \mu, \sigma)}. \tag{16}$$

When fitting the parameters of this distribution using gradient descent, we enforce the constraints that $\rho \in [0, 1]$, $\mu \in [-1, 1]$, and $\sigma > 0$. To do this, we optimize unconstrained parameters ρ_*, μ_*, σ_*, and then define $\rho = \text{sigmoid}(\rho_*)$, $\mu = \tanh(\mu_*)$, and $\sigma = \text{softplus}(\sigma_*)$.

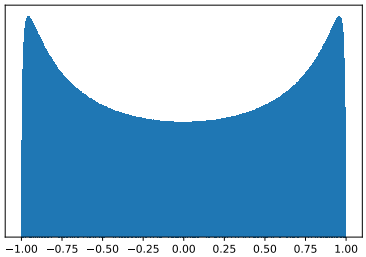


Figure 4: Prior distribution for latent parameters $\bar{z}_t^{(i)} = \tanh(z_t^{(i)})$ used to represent affine transformations.

C.2. Spatial Transformer VAE

Our spatial transformer VAE retains the structure of a standard VAE, but reinterprets the latent code z as two components. As described in Sec. 4.3, the first 6 latent dimensions z_t are associated with affine transformation parameters capturing image translation, rotation, scaling, and shear:

- $z_t^{(1)} \to$ *horizontal translation*,
- $z_t^{(2)} \to$ *vertical translation*,
- $z_t^{(3)} \to$ *rotation*,
- $z_t^{(4)} \to$ *shear*,
- $z_t^{(5)} \to$ *horizontal scale*,
- $z_t^{(6)} \to$ *vertical scale*.

The remainder of the latent code, z_*, generates parameters for independent per-pixel likelihoods.

To constrain our transformations to a fixed range of plausible values, we construct M_t using parameters $\bar{z}_t^{(i)} = \tanh(z_t^{(i)})$ that are first mapped to the interval $[-1, +1]$, and then linearly rescaled to an appropriate range via hyperparameters $\alpha^{(1)}, \ldots, \alpha^{(6)}$. Figure 4 illustrates that the induced prior for $\bar{z}_t^{(i)}$ is heaviest for extreme values, encouraging aggressive augmentation when sampling from the prior. The mapping function could be changed to modify this distribution for other applications.

Given these latent transformation parameters, we define an affine transformation matrix M_t:

$$M_t = \begin{bmatrix} 1 & 0 & \alpha^{(1)}\bar{z}_t^{(1)} \\ 0 & 1 & \alpha^{(2)}\bar{z}_t^{(2)} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos(\alpha^{(3)}\bar{z}_t^{(3)}) & -\sin(\alpha^{(3)}\bar{z}_t^{(3)}\alpha^{(4)}\bar{z}_t^{(4)}) & 0 \\ \sin(\alpha^{(3)}\bar{z}_t^{(3)}) & \cos(\alpha^{(3)}\bar{z}_t^{(3)}\alpha^{(4)}\bar{z}_t^{(4)}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} (\alpha^{(5)})^{\bar{z}_t^{(5)}} & 0 & 0 \\ 0 & (\alpha^{(6)})^{\bar{z}_t^{(6)}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (17)$$

To determine the parameters of the likelihood function for the pixel at coordinate (i, j), we use the generative model (or decoder) output at the pixel (i', j') for which

$$\begin{bmatrix} i \\ j \\ 1 \end{bmatrix} = M_t \begin{bmatrix} i' \\ j' \\ 1 \end{bmatrix}. \tag{18}$$

This corresponds to applying horizontal and vertical scaling, followed by rotation and shear, followed by translation. As (i', j') may not correspond to integer coordinates, a *spatial transformer layer* (Jaderberg et al., 2015) uses bilinear interpolation of the non-transformed likelihood parameters, appropriately padding the boundaries of the decoder output. For the Noise-Normal distribution we independently interpolate the ρ, μ, and σ^2 parameters.

Appendix D. Related work

D.1. Related work on SSL of VAEs since M2

Gordon and Hernández-Lobato (2020) explore SSL for VAEs by indirectly coupling discriminative and generative parameters via a joint prior. Such "parameter coupling" (Lasserre et al., 2006) still requires expensive sums over labels when computing likelihoods. We show that directly integrating generative parameters in predictions leads to superior performance (see Sec. 4).

Joy et al. (2021) propose the *Characteristic-Capturing VAE* (CCVAE) which, like M2, optimizes a lower bound on the joint probability of x and y via an *upstream* generative structure: $p(x, y, z) = p(y)p(z|y)p(x|z)$. The CCVAE has improved class-specific generative performance, but Joy et al. acknowledge that it does not significantly improve SSL accuracy.

Feng et al. (2021) introduce SHOT-VAE, a variant of M2 that seeks a better justification for the questionable $\log q(y|x)$ term. They apply label smoothing (Szegedy et al., 2016) to M2, producing a "smoothed" variational objective that contains a KL-divergence term incorporating $\log q(y|x)$. The experiments of Feng et al. suggest that the SSL accuracy gains of SHOT-VAE are partially due to their additional inclusion of a variant of Mixup data augmentation (Zhang et al., 2017).

The SeGMA model (Smieja et al., 2021) maps classes to components of a Gaussian mixture in the latent code space, and adapts Wasserstein autoencoders to avoid explicit representation of discrete mixture assignments. SeGMA may be used for classification, but their results emphasize its ability to interpolate observations.

D.2. Related Work on Constrained Learning

Imposing constraints to supervise probabilistic models is a widespread idea (Jaakkola et al., 1999; Chang et al., 2007; Mann and McCallum, 2010). Our PC objective can be seen as an instance of the broad family of *posterior regularization* (PR) techniques (Ganchev et al., 2010) for latent variable models. Zhu et al. (2014) present a regularized Bayesian formulation for *fully-supervised* nonparametric latent variable models. We emphasize that PR is an extremely broad family of approaches; our novel constraints are crucial to the success of the CPC-VAE, and are not similar to prior PR methods.

Li et al. (2018) propose an objective related to Eq. (9) for training VAEs from *fully-supervised* data with a max-margin loss. They also claim that an M2-like model is more effective in *semi-supervised* tasks when labels are rare, yet lack empirical evidence for this claim (no direct comparisons). In contrast, we demonstrate that with consistency constraints, Eq. (9) is in fact a superior *semi-supervised* objective for VAEs in terms of accuracy, reliability, and training speed. Hughes et al. (2018) also suggest a constrained objective like Eq. (9), but specialized to semi-supervised *topic models*.

Our unconstrained PC objective (7) has connections to the multi-conditional objective of McCallum et al. (2006), which was extended to deep generative models (both explicit VAEs and implicit GANs) by Kuleshov and Ermon (2017). This prior work does *not use consistency*, does not present our constraint-based view of SSL, and exclusively uses *implicit GANs* in SSL experiments. The DIVA model (Ilse et al., 2020) uses a similar prediction penalty for domain adaptation, but does not impose consistency constraints.

Related work on consistency. Recent non-generative image classifiers have used loss functions that encourage both accuracy and a notion of consistency on unlabeled data, such as consistency under adversarial perturbations (Miyato et al., 2019) or feature interpolations (Berthelot et al., 2019). *Unsupervised Data Augmentation* (UDA) (Xie et al., 2020) achieves state-of-the-art vision and text SSL classification by enforcing label consistency on augmented perturbations of unlabeled features, but requires highly-engineered augmentation routines (e.g., image processing libraries). In contrast, we learn a generative model that samples features whose label predictions need to be consistent. Our CPC-VAE applies to new domains where augmentation routines are unavailable; the learned generator provides augmentations.

More broadly, "cycle-consistency" has improved generative adversarial learning for images (Zhu et al., 2017; Zhou et al., 2016) and biomedical data (McDermott et al., 2018). Others have developed cycle-consistent objectives for VAEs (Jha et al., 2018) that make the encodings z consistent. Miller et al. (2019) consider feature-to-label prediction in VAEs and enforce consistency with reconstructed predictions on fully-labeled data. In contrast, our work focuses on SSL and enforces consistency in code-to-label prediction.

Appendix E. Ablation study

We compare variants of CPC, M2, and MMVA (Li et al., 2018) for SSL training on MNIST using a common architecture. We test variants of CPC without consistency, without spatial transformations, and without aggregate label loss. Ablations of M2 and MMVA try different model capacities (number of layers) and α penalties.

Method	MNIST (100)	Method	MNIST (100)	Method	MNIST (100)
CPC (2 Layer)	**96.68** ($\pm$0.54)	M2[§] (1 L, α=0.1, B)	88.03 ($\pm$1.71)	MMVA (1 L, α=$*$)	80.50 ($\pm$2.56)
CPC (2 L, w/o A)	94.27 ($\pm$3.78)	M2 (2 L, α=0.1, B)	83.32 ($\pm$5.22)	MMVA (2 L, α=$*$)	83.50 ($\pm$2.51)
CPC (2 L, w/o ST)	91.93 ($\pm$1.65)	M2 (4 L, α=0.1, B)	47.05 ($\pm$8.13)	MMVA (2 L, α=.1)	58.27 ($\pm$5.82)
CPC (4 L, w/o ST)	93.78 ($\pm$2.25)	M2 (4 L, α=$*$, B)	68.15 ($\pm$3.43)		
PC (2 L)	80.49 ($\pm$3.31)	M2 (1 L, α=0.1, N)	73.93 ($\pm$8.12)	VAE + GLM (2 L)	72.90 ($\pm$1.98)

Table 2: Ablation study on MNIST comparing our SSL VAEs to M2 (Kingma et al., 2014) and MMVA (Li et al., 2018). We use our own implementation, except for entry marked [§] from Kingma et al. (2014). We use a common MLP architecture with $C = 50$ and 1000 units per hidden layer. We indicate the likelihood: Noise-Normal (N, used by all CPC runs) or Bernoulli (B, used by M2 and MMVA). *Left:* Our innovations (consistency, spatial transforms (ST), and aggregate loss (A)) improve accuracy. *Center:* M2's accuracy deteriorates with larger networks, even after tuning α (α=$*$) instead of Kingma et al.'s default (α=0.1). CPC results are stable as size increases. *Right:* MMVA results are *worse* than CPC's.

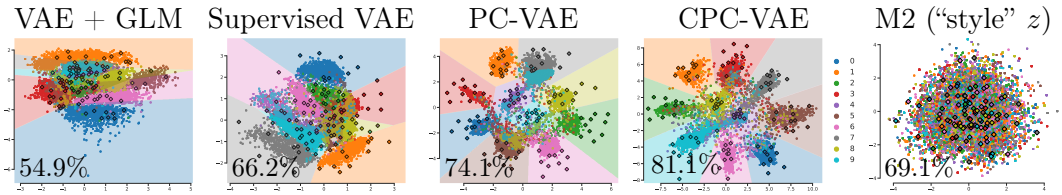


Figure 5: Semi-supervised VAE learning with 2-dim. encodings of MNIST (accuracy in corner). All methods use 49,900 unlabeled examples and 100 labeled (emphasized). We show each image's most likely encoding z, colored by true label y, and predicted decision boundaries if possible. *Baselines (from left):* 2-stage unsupervised VAE + GLM (Sec. 2.1) and a "supervised" VAE maximizing joint likelihood $\log p(x, y)$ (a special case of our PC method with $\lambda = 1$, Sec. 3.1). *Our methods:* Prediction constrained VAE (PC-VAE with $\lambda = 25$, Sec. 3.1) and consistent prediction constrained VAE (CPC-VAE, Sec. 3.2). *Competitor:* M2 (Kingma et al., 2014) intentionally decouples label y from "style" encoding z.

Appendix F. Visualization of latent space

To provide intuition for differences among model variants, Fig. 5 shows encodings for VAE models of MNIST digits with latent dimension $C = 2$, given only 10 labeled examples per class.

Appendix G. CPC-VAEs as a decision network

Fig. 5 provides further intution for the CPC-VAE method by formalizing it as a decision network.

Appendix H. Results: Visualizations of Sampled Images from Generative Models

Spatial Transformations and Consistency Constraints

To illustrate the simulated data that plays a key role in our consistency constraints, Figures 7 and 8 show images sampled from CPC-VAE models trained on MNIST and SVHN. Given some real image x, we first sample $z \sim q_\phi(\cdot \mid x)$, and then generate $\bar{x} \sim p_\theta(\cdot \mid z)$. We further show how images change as affine parameters for our spatial transformer VAE are varied. These parameters are *not* fixed when applying consistency constraints, encouraging models to learn how to align images.

Class-conditional sampling

A standard VAE generates data by sampling $z \sim \mathcal{N}(0, I)$, and then sampling $x \sim \mathcal{N}(\mu_\theta(z), \sigma_\theta(z))$, or an alternative like the Noise-Normal likelihood. For the PC-VAE or CPC-VAE, we can further sample images conditioned on a particular class label. As labels are not explicitly part of the generative model, we accomplish this by sampling images that would be confidently predicted as the target class. We use a rejection sampler, repeatedly sampling $z \sim \mathcal{N}(0, I)$ until a sample meets the criteria: $p_w(y \mid z) > 1 - \epsilon$, for some target threshold ϵ. We typically use $\epsilon = 0.05$ in our experiments. Alternatively, as shown in figure 5, we can reject a fraction of the samples of each class.

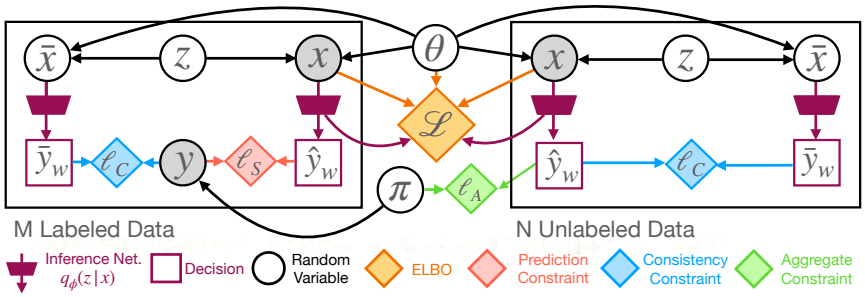


Figure 6: Formalization of our CPC-VAE as a decision network (Cowell et al., 2006). Circular nodes are random variables, including latent codes z and observed features x. Shaded nodes are observed, including the class labels y for some data (left). Square decision nodes indicate label predictions $\hat{y}_w$ via the inference network $q_\phi(z \mid x)$. Diamonds indicate four losses ((a)-(d) below) that influence the prediction of labels and latent variables. *(a) Generative likelihood:* Like standard VAEs, our methods favor generative parameters θ and variational posteriors q_ϕ that maximize the variational bound $\mathcal{L}$ (*orange*). *(b) Prediction accuracy:* Unlike previous semi-supervised VAEs, we do not model the probability of labels y given z or x. Instead, we treat label prediction as a decision problem, with task-motivated loss ℓ_S (*red*), that constrains the encoding posterior $q_\phi(z|x)$ (and therefore the generative model). *(c) Prediction consistency:* For unlabeled data (right), we know that if two observations x and $\bar{x}$ are generated from the same latent code z, they should have identical labels; otherwise, the model cannot have high accuracy. The loss ℓ_C (*blue*) enforces this *consistency*. *(d) Aggregate consistency:* The predicted label frequencies for unlabeled data should be close to the empirical frequencies π of labeled data. The loss ℓ_A (*green*) enforces this, penalizing degenerate solutions to ℓ_C that use the same label $\hat{y}_w$ for most unlabeled data.

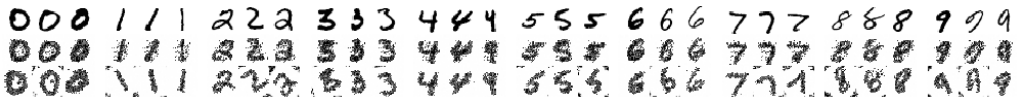


Figure 7: Sampled reconstructions used to compute the consistency loss during training. *Top:* Original image. *Middle:* Sampled reconstructions using a "Noise-Normal" likelihood. *Bottom:* Sampled reconstructions with spatial affine transformations sampled from the prior.

Figure 8: Visualization of spatial transform CPC-VAE reconstructions (trained with full labels). Each triplet shows *left:* the original image, *center:* the reconstructed image, and *right:* the "aligned" reconstruction obtained by setting the affine transform dimensions of the latent code to the prior mean. We see that the model learns a canonical orientation for each digit.

Fig. 2 in the main text shows 2-dimensional latent space encodings of the MNIST dataset using several different models. We provide a complementary visualization of generative models in Fig. 12, where we compare class-conditional samples for three of these models. The unsupervised VAE's encodings of some classes (e.g., 2's and 4's and 8's and 9's) are not separated, and samples thus frequently appear to be the wrong class. Model M2 (Kingma et al., 2014) explicitly encodes the class label as a latent variable, but nevertheless many sampled images do not visually match the conditioned class. In contrast, for our CPC-VAE model almost all samples are easily recognized as the target class.

We illustrate class-conditional samples for our CPC-VAE model of SVHN in Fig. 13, and for our CPC-VAE models of Celeb-A in Fig. 16. The SVHN samples show rich variability while clearly retaining the digit identity. For both the standard and very-deep CPC-VAE models of Celeb-A, the corresponding class can be easily determined from the sampled images. We also see the clearly superior image detail and realism that the state-of-the-art very-deep CPC-VAE architecture provides, which leads to substantially improved reclassification accuracy in Fig. 5.

Appendix I. Results: Training time comparison

Figure 17 below provides an empirical comparison of the average training time cost per step using the MNIST models summarized in our main paper's Table 2. Our CPC-VAE implementation runs both the encoder and decoder networks twice to compute the objective (once for the standard VAE loss and an additional time to compute the consistency reconstruction and prediction), thus the runtime is approximately twice that of the PC-VAE: the PC-VAE requires 38 milliseconds per training step, while the CPC-VAE requires 80.7 milliseconds.

Furthermore, our empirical findings show that training M2 is more expensive than our proposed CPC-VAE in practice, which we expect given the runtime analysis in Sec. 2. The M2 model must run the encoder and decoder networks once *per class* in order to compute the loss, due to the marginalization of the labels required for the unsupervised loss in Eq. (4). This increases the runtime by a factor equivalent to the number of classes. In our empirical test, we see that the training time per step is 6.7x that of the PC-VAE model, close to the 10x slowdown we would expect for the 10 digit classes of MNIST. In our experiments, we did not find substantial differences in the size of networks or number of training steps needed to train each of these models effectively.

Figure 9: Unsupervised VAE

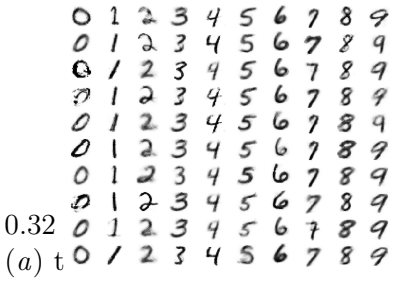


Figure 10: CPC-VAE

Figure 11: M2

Figure 12: Class-conditional samples of the 10 possible digit classes in the MNIST dataset. Each column shows multiple samples from one specific digit class. From left to right, each panel shows samples from a standard unsupervised VAE, our CPC-VAE, and model M2 (Kingma et al., 2014). All models use a 2-dimensional latent code, and are trained on the MNIST dataset with 100 labeled examples (10 per class).

Appendix J. Results: Additional classification results

Figure 18 shows extended results on the toy half-moons dataset, including results using 100 labels. These extended results demonstrate that the PC-VAE model works well when the fraction of labeled data is large enough, significantly outperforming the 2-stage VAE+GLM model. When the labeled fraction becomes extremely small, as in the 6-label case, the consistency constraint becomes necessary for good performance.

We also performed experiments on the NORB dataset (LeCun et al., 2004), to compare to results in prior semi-supervised VAE works. Results are shown in table 3. These results

Figure 13: Class-conditional samples of the 10 possible digit classes in the SVHN dataset. The generative model was trained on the fully labeled SVHN dataset with prediction and consistency constraints. Samples were chosen via rejection sampling in the latent space with a threshold of 95% confidence in the target class.

Method	NORB (1000)
CPC-VAE	**92.0** ($\pm$1.21)
SDGM	90.6 ($\pm$0.04)
Discrim.	86.7 ($\pm$1.32)

Table 3: Results on the NORB dataset

further reinforce our claims that the CPC-VAE outperforms previous SSL VAEs on image tasks.

Appendix K. Results: Sensitivity to constraint multiplier hyperparameters

We compare the test accuracy for our consistency-constrained model for MNIST over a range of values for both λ (the prediction constraint multiplier) and γ (the consistency constraint multiplier) in Figure 21. All runs used our best consistency-constrained model for MNIST using dense networks. We kept all hyperparameters identical to the previous results (see section L), changing only the value of interest for each run.

We see that the resulting test accuracy smoothly varies across several orders of magnitude, with the optimal result being at or near the values we chose for our experiments. Performance is superior to the M2 baseline model for a wide range of hyperparameter values.

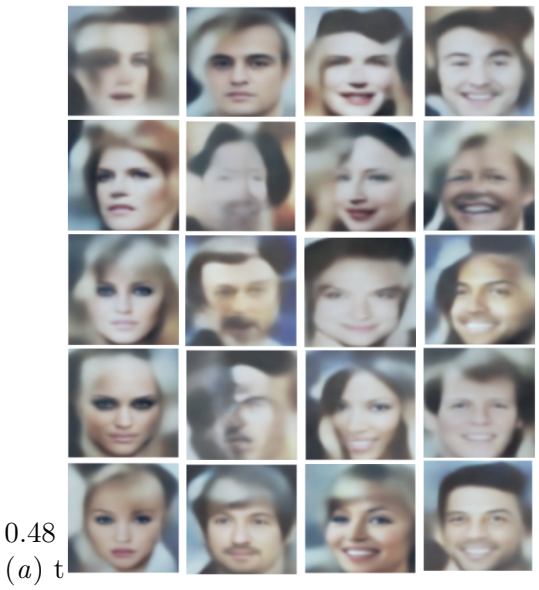


Figure 14: Single stochastic-layer VAE

Figure 15: "Very-deep" VAE

Figure 16: Comparison of class-conditional samples of Celeb-A from a standard VAE and the deep VAE. From left to right, classes are *woman-neutral, man-neutral, woman-smiling, man-smiling*. Both models were trained as a semi-supervised CPC-VAE with 1000 labels. Samples from the single-layer VAE show the mean output for each pixel, samples from the very-deep model are fully sampled including at the pixel level.

Appendix L. Experimental Protocol

Here we provide details about models and experiments which did not fit into the primary paper.

L.1. Hyperparameter optimization

The hyperparameter search for all models, including the CPC-VAE and various baselines, used Optuna (Akiba et al., 2019) to achieve the best accuracy on a labeled validation set. For our 2-layer and 4-layer M2 experiments, we used our own implementation (available in our code release) and followed the hyperparameters used by the original authors. For the 4-layer variant, we tested 10 different settings of α, ranging from 0.05 to 50, reporting both the result using the original suggested value of $\alpha = 0.1$ and the best value from our search ($\alpha = 10$). For M2, we also dynamically reduced the learning rate when the validation loss plateaued.

L.2. Network architectures

For our PC-VAE and CPC-VAE models of the MNIST data, we use fully-connected encoder and decoder networks with two hidden layers, 1000 hidden units per layer, and softplus activation functions. Like the M2 model (Kingma et al., 2014), we use a 50-dimensional latent space. The original M2 experiments used networks with a single hidden layer of 500 units. We compare this to replications with networks matching ours, as well as 4-layer networks.

For the SVHN and NORB datsets, we adapt the wide-residual network architecture (WRN-28-2) (Zagoruyko and Komodakis, 2016) that was proposed as a standard for semi-supervised deep learning research in (Oliver et al., 2018). In particular, we use this architecture for our encoder with two notable changes: We replace the final global average pooling layer with a final dense layer that outputs means and variances for the latent space. We find that this structure provides the capacity needed for accomplishing both generative and discriminative tasks with a single network. For the decoder network we use a "mirrored" version of this architecture, reversing the order of layer sizes used, replacing convolutions with transposed convolutions, and removing pooling layers. We maintain the residual structure of

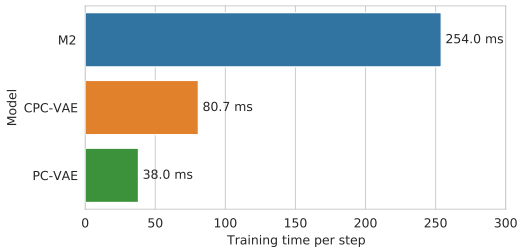


Figure 17: Comparison of training time per update step of stochastic gradient descent. Each model was trained on the semi-supervised MNIST with 100 labels using hyperparameter settings identical to those used in Table 2. Experiments were run on an RTX Titan GPU, using a common codebase built on top of Tensorflow (Abadi et al., 2015) that implements all methods. Each time reported is the average training step time over the second epoch.

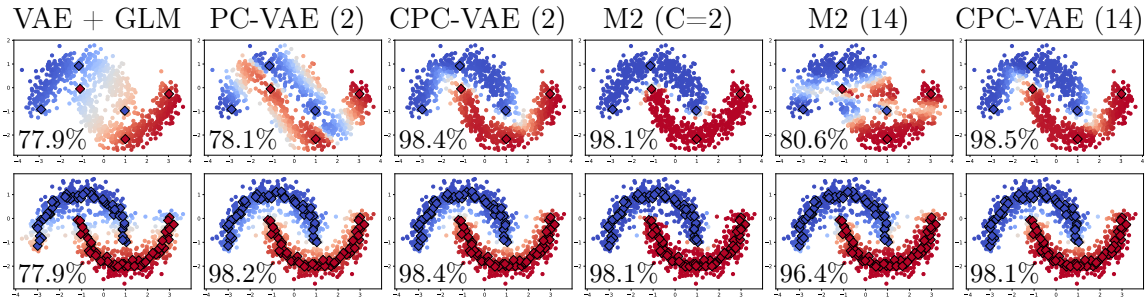
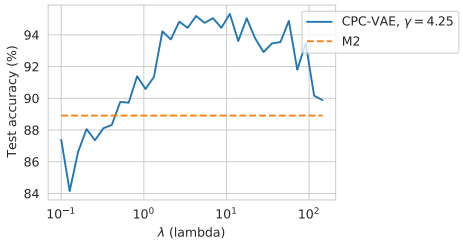
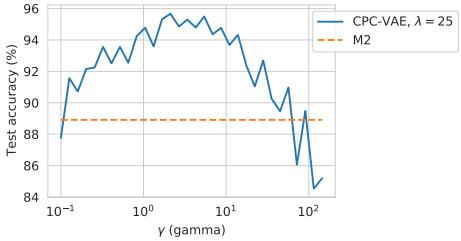


Figure 18: Predictions on half-moon classification (accuracy in corner) for semi-supervised VAE learning. Dots are 2-dim. feature vectors colored by predicted probability of mostly likely label, labeled examples are shown as larger diamonds. Titles indicate encoding size $C = 2$ or $C = 14$. M2 (Kingma et al., 2014) accuracy *deterioriates* when capacity increases from $C = 2$ to 14 (drop from 98.1% to 80.6% accuracy). Our CPC-VAE is reliable at any capacity via constraints that ensure prediction quality. *Top row:* Learning from 6 labeled examples (diamonds) and 994 unlabeled examples. *Bottom row:* Learning from 100 labeled examples and 900 unlabeled examples.



0.45
(a) t

Figure 19: Sensitivity to λ



0.45
(a) t

Figure 20: Sensitivity to γ

Figure 21: Sensitivity of test accuracy to the constraint (Lagrange multiplier) hyperparameters λ and γ.

the network. Our best classification results with this architecture were achieved with a latent space dimension of 200 for SVHN and NORB, and a dimension of 50 for MNIST and CelebA.

For CelebA we used a similar WRN structure, but with 2 fewer residual blocks per level (WRN-20-2). This allowed for faster training on the higher resolution images, while providing good validation accuracy.

For the "very-deep" VAE model we used the larger (WRN-28-2) wide resnet for the z_0 encoder. For the ladder encoder and decoder we retained the bottleneck block structure from (Child, 2021), but used only 30 stochastic layers. Each latent representation used 16 channels. Each bottleneck block used an input/output width of 128 channels with a bottleneck width of 64 channels. All other details were retained from (Child, 2021).

L.3. Additional regularization

In developing and evaluating our CPC model we explored several common regularizers taken from deep semi-supervised learning and VAE literature. The SVHN results in Table 1 in the main text included these terms as part of the training loss (with the weighting specified in the hyperparameter summary below). Our experiments on CelebA omitted these terms, and more recent (unpublished) testing on SVHN suggests that these additions are unnecessary for achieving good semi-supervised performance. However, we lacked the time and resources to repeat our full set of replicated experiments without these regularizers, so we discuss them here for completeness.

Beta-VAE. As an additional form of regularization for our model, we allow our hyperparameter optimization to adjust a weight on the KL-divergence term in the variational lower bound, which we call $\boldsymbol{\beta}$ as in previous work (Higgins et al., 2017):

$$\mathcal{L}_{\beta}^{\text{VAE}}(x;\theta,\phi) = \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x \mid z)\right] + \boldsymbol{\beta} \cdot \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p(z)}{q_\phi(z \mid x)}\right] \tag{19}$$

This allows us to encourage $q_\phi(z \mid x)$ to more closely conform to the prior, which may be necessary to balance the scale of the objective, depending on the likelihoods used and the dimensionality of the dataset.

Prediction model regularization. We add two standard regularization terms to the prediction model used in our constraint, $\hat{y}_w(y \mid z)$. The first is an ℓ_2 regularizer on the regression weights, $||w||_2^2$, to help reduce overfitting. The second is an entropy penalty. As $\hat{y}_w(z)$ defines a categorical distribution over labels, we compute this as: $-\mathbb{E}_{\hat{y}_w(y|z)}[\log \hat{y}_w(y \mid z)]$, which has been shown to be helpful for semi-supervised learning in (Grandvalet and Bengio, 2004) and was used as part of the standardized training framework of (Oliver et al., 2018). We allowed our hyperparameter optimization approach to select appropriate weights for both terms.

L.4. Image pre-processing

For all of our image datasets, we rescale the inputs to the range [-1, 1]. For our NORB classification results, we downsample each image to 48x48 pixels. For our SVHN classification results, we convert images to greyscale to reduce the representational load on our generative model. Before the grayscale conversion, we apply contrast normalization to better disambiguate the colors within each image.

For the SVHN and NORB results, we follow the recommendation of a recent survey of semi-supervised learning methods (Oliver et al., 2018) and apply a single data augmentation technique: random translations by up to 2-pixels in each direction. For generative results, we retained the original color images and trained with full labels.

For the CelebA dataset we use the aligned images, first cropping images to be square, then rescaling to 96x96 and finally taking a center crop of 64x64 pixels to remove most of the background. We retained color, but applied contrast normalization as with SVHN. No data augmentation was used for this dataset.

L.5. Likelihoods

For all experiments on the half-moon toy data, we used a normal likelihood. For all of our image datasets except CelebA, we use the Noise-Normal likelihood for our CPC methods.

For our implementation of M2 for extensive experiments on MNIST we retained the Bernoulli likelihood used by the original authors (Kingma et al., 2014). That is, we rescaling each pixel's numerical intensity value to the unit interval [0,1], and then sampled binary values from a Bernoulli with probability equal to the intensity.

For our experiments on CelebA, we found that the Noise-Normal likelihood was unnecessary. Instead we employed a Normal likelihood, using the formulation suggested by (Rybkin et al., 2020), which trains a single, global variance parameter, σ^*_θ, that is shared across all images/pixels: $x \sim \mathcal{N}(\mu_\theta(z), \sigma^*_\theta)$. This likelihood was used for both the standard and very-deep VAEs trained on Celeb-A.

L.6. Optimization with Minibatches that Balance Labeled and Unlabeled Sets

All models were trained using minibatch stochastic gradient descent via the ADAM (Kingma and Ba, 2014) optimizer. Batch sizes were 100 for MNIST, 64 for SVHN/NORB and 32 for CelebA. We decayed the learning rate according the *cosine decay* (Loshchilov and Hutter, 2016) strategy (without restarts).

Importantly, each minibatch was engineered to contain exactly 50% labeled and 50% unlabeled samples. Several prior works in semi-supervised deep learning (Kingma et al., 2014; Oliver et al., 2018) have employed a balanced stochastic gradient optimization approach where training batches are selected to have equal numbers of labeled and unlabeled examples. This prevents instances where batches have no labeled examples and reduces the variance of stochastic training. We can implement a similar scheme for the consistency-constrained VAE without changing the expectation of the objective.

Recall the (unconstrained) CPC-VAE objective from section 3:

$$\sum_{x \in \mathcal{D}^U \cup \mathcal{D}^S} \mathcal{L}^{\text{VAE}}(x; \theta, \phi) - \sum_{x \in \mathcal{D}^U} \gamma \mathcal{C}^U(x; \theta, \phi, w) + \sum_{x,y \in \mathcal{D}^S} -\lambda \mathcal{P}(x, y; \phi, w) - \gamma \mathcal{C}^S(x, y; \theta, \phi, w),$$

We can rewrite an equivalent objective, separating out the supervised and unsupervised terms:

$$\begin{aligned} \frac{1}{|\mathcal{D}^S|} \sum_{x \in \mathcal{D}^S} |\mathcal{D}^S| \mathcal{L}^{\text{VAE}}(x; \theta, \phi) - \frac{1}{|\mathcal{D}^S|} \sum_{x,y \in \mathcal{D}^S} \lambda |\mathcal{D}^S| \mathcal{P}(x, y; \phi, w) - \frac{1}{|\mathcal{D}^S|} \sum_{x,y \in \mathcal{D}^S} \gamma |\mathcal{D}^S| \mathcal{C}^S(x, y; \theta, \phi, w) \\ + \frac{1}{|\mathcal{D}^U|} \sum_{x \in \mathcal{D}^U} |\mathcal{D}^U| \mathcal{L}^{\text{VAE}}(x; \theta, \phi) - \frac{1}{|\mathcal{D}^U|} \sum_{x \in \mathcal{D}^U} \gamma |\mathcal{D}^U| \mathcal{C}^U(x; \theta, \phi, w) \end{aligned} \tag{20}$$

Rewriting in terms of expectations over the labeled and unlabeled datasets, leads to a natural approach for optimizing via balanaced batches:

$$|\mathcal{D}^S|\mathbb{E}_{x,y\in\mathcal{D}^S}\left[\mathcal{L}^{\text{VAE}}(x;\theta,\phi)-\lambda\mathcal{P}(x,y;\phi,w)-\gamma\mathcal{C}^S(x,y;\theta,\phi,w)\right] \tag{21}$$
$$+|\mathcal{D}^U|\mathbb{E}_{x\in\mathcal{D}^U}\left[\mathcal{L}^{\text{VAE}}(x;\theta,\phi)-\gamma\mathcal{C}^U(x;\theta,\phi,w)\right]$$

We see that in our stochastic optimization, we can get an unbiased estimate of the objective and its gradient by sampling labeled and unlabeled batches separately. With batch size B for both labeled and unlabeled batches our estimated objective becomes:

$$\frac{|\mathcal{D}^S|}{B}\sum_{x,y\in B^S}\left[\mathcal{L}^{\text{VAE}}(x;\theta,\phi)-\lambda\mathcal{P}(x,y;\phi,w)-\gamma\mathcal{C}^S(x,y;\theta,\phi,w)\right] \tag{22}$$

$$+\frac{|\mathcal{D}^U|}{B}\sum_{x\in B^U}\left[\mathcal{L}^{\text{VAE}}(x;\theta,\phi)-\gamma\mathcal{C}^U(x;\theta,\phi,w)\right] \tag{23}$$

$$B^S\sim\mathcal{D}^S,\ B^U\sim\mathcal{D}^U$$

We may also normalize by an additional scale factor of $\frac{1}{|\mathcal{D}^S|+|\mathcal{D}^U|}$ to approximately remove dependence on the dataset size.

In our experiments we remove the terms $|\mathcal{D}^S|$ and $|\mathcal{D}^U|$, essentially treating the labeled and unlabeled datasets as being of equal size. We found this approach simpler, though we have not performed a thorough set of experiments to evaluate the practical differences.

L.7. Hardware

Experiments were run on a variety of hardware systems. Models for SVHN and Norb were trained primarily on an Nvidia DGX-2 system with V100 GPUs. Experiments on MNIST and Celeb-A were run on a workstation using a Titan RTX and 2080TI GPUs. Additional MNIST and Celeb-A experiments were run using 2080TI GPUs rented through vast.ai.

L.8. Summary of hyperparameter settings for final results

Table 1 provides all hyperparameter settings used in our experiments.

Appendix M. Dataset Details

For each dataset considered in our paper, we provide a more detailed overview of its contents and properties. We comply with the stated terms of use for all listed datasets and assert that our work is for non-commercial research purposes.

M.1. MNIST

Overview. We consider a 10-way exclusive categorization task for MNIST digits.

We use 28-by-28 pixel grayscale images.

Public availability. We will make code to extract our version available after publication.

Data statistics. Statistics for MNIST are shown in Table 5.

Hyperparameter	**MNIST (100)**	**SVHN (1000)**	**NORB (1000)**	**CelebA (1000)**
Encoder/decoder	2 FC layers	WRN-28-2	WRN-28-2	WRN-20-2 (very-deep: see L.2)
Dense layer size	1000 units	-	-	-
Network activations	Softplus	Leaky ReLU	Leaky ReLU	Leaky ReLU / GeLU (very-deep)
Latent dimension	50	200	200	50 / 16 (very-deep)
Pixel likelihood	Noise-Normal	Noise-Normal	Noise-Normal	(σ) Normal (Rybkin et al., 2020)
Prediction multiplier λ	25	140	80	150 / 25 (very-deep)
Consistency multiplier γ	4.25λ	1.25λ	4λ	2λ
Aggregate consistency	0.1λ	0.2λ	0.2λ	0.5λ
β-VAE weight	1	1.3	2	1
Predictor reg. ($\|\|w\|\|_2^2$)	1	1	1	0
Entropy reg.	0.5λ	0.5λ	0.5λ	0
($\mathbb{E}_{p_w(y\|z)}[\log p_w(y\|z)]$)	-			
Translation range	$0.2 \times$ W	$0.2 \times$ W	$0.2 \times$ W	-
($\alpha^{(1)} = \alpha^{(2)}$)				
Rotation range ($\alpha^{(3)}$)	0.4 rad	0.5 rad	0.4 rad	-
Shear range ($\alpha^{(4)}$)	0.2 rad	0.2 rad	0.2 rad	-
Scale range	1.5	1.5	1.5	-
($\alpha^{(5)} = \alpha^{(6)}$)				
Optimizer	ADAM	ADAM	ADAM	ADAM
Learning rate	3×10^{-4}	3×10^{-4}	3×10^{-4}	3×10^{-4}

Table 4: Hyperparameter settings for semi-supervised learning experiments with our CPC-VAE. For translation range, the symbol W denotes the image width (in pixels). Relevant difference between the standard VAE and deep VAE trained on Celeb-A are noted.

split	num. examples	label distribution
labeled train	100	[0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1]
unlabeled train	49900	[0.1 0.11 0.1 0.1 0.1 0.09 0.1 0.1 0.1 0.1]
labeled valid	10000	[0.1 0.11 0.1 0.1 0.1 0.09 0.1 0.1 0.1 0.1]
labeled test	10000	[0.1 0.11 0.1 0.1 0.1 0.09 0.1 0.1 0.1 0.1]

Table 5: MNIST dataset.

M.2. SVHN

Overview. We consider a 10-way exclusive categorization task for SVHN digits.

We use 32x32 pixel grayscale images.

Public availability. We will make code to extract our version available after publication.

Data statistics. Statistics for SVHN are shown in Table 6.

split	num. examples	label distribution
labeled train	1000	[0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10]
unlabeled train	62257	[0.07 0.19 0.15 0.12 0.10 0.09 0.08 0.08 0.07 0.06]
labeled valid	10000	[0.07 0.19 0.14 0.12 0.10 0.09 0.08 0.08 0.07 0.06]
labeled test	26032	[0.07 0.20 0.16 0.11 0.10 0.09 0.08 0.08 0.06 0.06]

Table 6: SVHN dataset.

M.3. NORB

Overview.

We use 48x48 pixel grayscale images.

Public availability. We will make code to extract our version available after publication.

Data statistics. Statistics for NORB are shown in Table 7.

split	num. examples	label distribution
labeled train	1000	[0.2 0.2 0.2 0.2 0.2]
unlabeled train	21300	[0.2 0.2 0.2 0.2 0.2]
labeled valid	2000	[0.2 0.2 0.2 0.2 0.2]
labeled test	24300	[0.2 0.2 0.2 0.2 0.2]

Table 7: NORB dataset.

M.4. CelebA

Overview.

Labels for our verision of the CelebA dataset were generated from the provided attributes. Our dataset used 4 classes: woman/neutral face, man/neutral face, woman/smiling, man/smiling.

Public availability. We will make code to extract our version available after publication.

Data statistics. Statistics for CelebA are shown in Table 8.

Appendix N. Broader Impacts

Like many models used to make predictions, ours has the potential for both beneficial and not-so-beneficial impact on human society. For any work that uses our models to make real-world predictions, we recommend that users give thought about how even simple natural image datasets may be biased towards certain regions and cultures (DeVries et al., 2019;

split	num. examples	label distribution
labeled train	1000	[0.25 0.25 0.34 0.16]
unlabeled train	21300	[0.25 0.25 0.34 0.16]
labeled valid	2000	[0.25 0.25 0.34 0.16]
labeled test	24300	[0.27 0.23 0.35 0.15]

Table 8: CelebA dataset.

Merler et al., 2019; Shankar et al., 2017). We further suggest application developers should invest in developing diagnostics to measure and report how predictions impact different subpopulations (Mitchell et al., 2019). Future work could integrate constraints that enforce various definitions of group-level fairness (Mitchell et al., 2018) into our approach, as have been developed for other deep latent variable models (Madras et al., 2019).

Specifically for the CelebA dataset of celebrity faces, we acknowledge that some of its recorded attributes may be used problematically (e.g. "big nose" or identification of specific racial groups). We deliberately did not focus on such attributes but instead focused on two attributes thought to be more benign: facial expression (neutral vs. smiling) and gender (male vs. female). We acknowledge and affirm that human gender is not binary; our classifiers learn the stereotypical patterns of celebrity photos from a specific human culture and time period.