1 Derivation of VP loss

Given two probability distributions $p(\mathbf{y})$ and $q(\hat{\mathbf{y}})$, the Kantorovich's optimal transport (OT) problem is given by [7]

$$\mathcal{L}(p,q) = \inf_{\Gamma(\mathbf{y},\hat{\mathbf{y}})\in\Pi(p,q)} \int c(\mathbf{y},\hat{\mathbf{y}})\, d\Gamma(\mathbf{y},\hat{\mathbf{y}}), \tag{1}$$

where $c(\cdot,\cdot)$ is a nonnegative, measurable function, and $\Pi(p,q)$ is the set of all couplings such that their marginal probability distributions are p and q. If $c(\mathbf{y},\hat{\mathbf{y}})$ is also a lower semi-countinuous function, then by Kantorovich's duality, the OT problem in (1) is equivalent to the following

$$\mathcal{L}(p,q) = \sup_{\phi,\psi\in L^1} \int \phi(\mathbf{y})dp(\mathbf{y}) + \int \psi(\hat{\mathbf{y}})dq(\hat{\mathbf{y}}) \quad \text{s.t.} \quad \phi(\mathbf{y})+\psi(\hat{\mathbf{y}}) \le c(\mathbf{y},\hat{\mathbf{y}}), \tag{2}$$

where L^1 denotes the set of functions such that their absolute value is Lebesgue integrable, and the constraint is satisfied almost everywhere (according to $p(\mathbf{y})$ and $q(\hat{\mathbf{y}})$). From the constraint, we have an upper bound for $\psi(\cdot)$,

$$\psi(\hat{\mathbf{y}}) \le \inf_{y\in S(p)} \{c(\mathbf{y},\hat{\mathbf{y}}) - \phi(\mathbf{y})\} = -\sup_{y\in S(p)} \{\phi(\mathbf{y}) - c(\mathbf{y},\hat{\mathbf{y}})\} \tag{3}$$

where $\mathcal{S}(p)$ denotes the support of p. Note that if the right hand side of (3) is in L^1, then taking $\psi(\cdot)$ as its upper bound maximizes (2). Let $p(\mathbf{y})$ be a probability distribution supported only on the vertices of the simplex, with mass probabilities indicated by the entries of $Y \in \Delta^{K-1}$, that is, $p(\mathbf{y}) = \sum_{k=1}^{K} Y^k \cdot \delta(\mathbf{y}-\mathbf{e}_k)$. Replacing this expression in (3),

$$\psi(\hat{\mathbf{y}}) \le -\max_{k\in\{1,\dots,K\}} \{\phi(\mathbf{e}_k) - c(\mathbf{e}_k,\hat{\mathbf{y}})\}. \tag{4}$$

If $c(\cdot,\cdot)$ is a metric, then it is non-negative and bounded in a bounded domain ($\Delta^{K-1}\times\Delta^{K-1}$), hence Lebesgue integrable. Assuming $\phi(\mathbf{e}_k) < \infty$ for $k = [1,\dots,K]$, each of the K functions within the maximum operator in (4) is Lebesgue integrable, and thus the maximum itself is also L^1.Hence, we can maximize (2) by replacing $\psi(\cdot)$ by its (feasible) upper bound as follows,

$$\mathcal{L}(p,q) = \sup_{\Phi\in\mathbb{R}^K} \langle \Phi, Y\rangle - \mathbb{E}_{\hat{\mathbf{y}}\sim q}\left\{\max_k \{\Phi_k - c(\mathbf{e}_k,\hat{\mathbf{y}})\}\right\}. \tag{5}$$

where we have defined $\Phi_k = \phi(\mathbf{e}_k)$ for $k = 1,\dots,K$, $\langle\cdot,\cdot\rangle$ denotes dot product, and $\mathbb{E}_{\hat{\mathbf{y}}\sim q}\{\cdot\}$ is the expectation operator. Now, since we don't have access to the actual distribution $q(\cdot)$, we approximate the analytical expectation by the empirical expectation from N samples, $\hat{\mathbf{y}}_i$ for $i = 1,..,N$,

$$\mathcal{L}(p,q) \approx \max_{\Phi\in\mathbb{R}^K} \left[\langle \Phi, Y\rangle - \frac{1}{N}\|\Phi\mathbf{1}_N^T - C\|_{\infty,1}\right], \tag{6}$$

where we have written the problem in matrix form, with $C \in \mathbb{R}^{K\times N}$ a cost matrix such that $C(j,i) = c(\mathbf{e}_j,\hat{\mathbf{y}}_i)$, and $\|A\|_{\infty,1}$ denotes the sum of the infinity norm (maximum) of the columns of the matrix A. One can write (6) as a linear program (LP) by defining a new variable $\mathbf{s}\in\mathbb{R}^N$ which carries out the max operation as

$$LP(Y,\{\hat{\mathbf{y}}_i\}_{i=1}^N) = \max_{\Phi\in\mathbb{R}^K,\mathbf{s}\in\mathbb{R}^N} \langle\Phi,Y\rangle - \frac{1}{N}\langle\mathbf{s},\mathbf{1}_N\rangle \qquad \text{s.t.} \qquad \Phi\mathbf{1}_N^T - \mathbf{1}_K\mathbf{s}^T \le C, \tag{7}$$

hence we have arrived to the dual form of discrete optimal transport, which can be solved by standard LP solvers.

2 Architectural details

2.1 Deep Sparse Detector

	Layer	Kernel Size	Output channels
Encoder (E_{Θ_E})	Complex conv. + ReLU	1	8
	Complex conv. + ReLU	3	16
	Complex conv. + ReLU	5	32
	Complex conv. + ReLU	5	32
	Local spatial softmax ($\lambda = 0.1$)	17	32
	Non-maximum suppression	17	32
Classification head (C_{Θ_C})	Concatenate [real, imag]	–	64
	Linear Layer + ReLU	–	128
	Linear Layer + ReLU	–	256
	Linear Layer + ReLU	–	512
	Linear Layer + ReLU	–	128
	Linear Layer + ReLU	–	8
	Linear Layer + Softmax	–	3
Reconstruction head (R_{Θ_R})	Complex conv. + ReLU	5	32
	Complex conv. + ReLU	5	16
	Complex conv. + ReLU	3	1

Table 1: Architectural parameters of each component of DSD.

2.2 Architectures utilized as baselines for proportion prediction

Tables 2, 3, and 4 show the models utilized for direct proportion prediction, heatmap estimation, and classification approaches, respectively. The oracle detections for the classifier correspond to those originally learned by our model.

	Components	Kernel Size	Output channels
ResNet-152 [8]	Conv2D	3	3
	Adaptive Avg Pool 2D (224×224)	–	3
	ResNet-152	–	2048
	Linear Layer + ReLU	–	3
	Softmax	–	3
Ours	Complex conv. + ReLU	1	8
	Complex conv. + ReLU	3	16
	Complex conv. + ReLU	5	32
	Complex conv. + ReLU	5	32
	Concatenate [real,imag]	–	64
	Global average pooling	512	64
	Linear Layer + ReLU	–	3
	Softmax	–	3

Table 2: Architectures used for direct regression

3 Implementation details

The model was trained in a two-stage process. First, the encoder and reconstruction head were trained with stochastic gradient descent (SGD) with learning rate $1e^{-5}$, batch size 7, for 10 epochs. Data augmentation was randomly performed applying vertical or horizontal flip with probability 0.5 each (independently).

	Components	Kernel Size	Output channels
C-FCRN [3]	Conv2D + ReLU + Maxpool (2×2)	3	32
	Conv2D + ReLU + Maxpool (2×2)	3	64
	Conv2D + ReLU + Maxpool (2×2)	3	128
	Conv2D + ReLU + Maxpool (2×2)	3	512
	Concatenate residual connection layer 3	–	640
	Upsample (2×2) + Conv2D + ReLU	2	128
	Concatenate residual connection layer 2	–	192
	Upsample (2×2) + Conv2D + ReLU	2	64
	Concatenate residual connection layer 1	–	96
	Upsample (2×2) + Conv2D + ReLU	2	32
	Conv2D + ReLU	3	3
	Global sum	512	3
	Normalization	–	3
Ours	Complex conv. + ReLU	1	8
	Complex conv. + ReLU	3	16
	Complex conv. + ReLU	5	32
	Complex conv. + ReLU	5	32
	Concatenate [real,imag]	–	64
	Conv2D + ReLU	3	4
	Pixel-wise softmax + Global sum	–	4
	Normalization	–	4

Table 3: Architectures used for heatmap regression

	Components	Kernel Size	Output channels
Le Net [1]	Conv2D + ReLU	3	6
	Avg pooling	2	6
	Conv2D + ReLU	3	16
	Linear Layer + ReLU	–	120
	Linear Layer + ReLU	–	84
	Linear Layer + Softmax	–	3
Encoder (E_{Θ_E})	Complex conv. + ReLU	1	8
	Complex conv. + ReLU	3	16
	Complex conv. + ReLU	5	32
	Complex conv. + ReLU	5	32
	Local spatial softmax ($\lambda = 0.1$)	9	32
	Non-maximum suppression	9	32
	Concatenate [real, imag]	–	64
	Linear Layer + ReLU	–	32
	Linear Layer + Softmax	–	3

Table 4: Architectures used for classification using oracle detections

Once the encoder was trained with the reconstruction loss, its parameters were frozen, and the detected objects and their features were used to train the classifier with Adam optimizer (learning rate $1e^{-3}$ and full batch for 1000 epochs).

4 Complement to representation learning methods

To evaluate the complementary nature of our method, we implement two archetypal methods of representation learning-based LLP and combine them with our proposed loss. In particular,

– **LLP-GAN** [4]. While our classifier architecture serves as discriminator, we implement a generator to map samples from a uniform distribution to

inputs of the classifier (feature vectors of size 64). We do so by three linear layers followed by ReLU nonlinearities (dimensions $25 \to 256 \to 256 \to 64$). We train the model adversarially as they proposed, but also evaluate the performance obtained when the KL-div term used to train the discriminator is replaced by our VP loss.

– **Contrastive pretraining [9]**. we followed the idea presented in [9] where after contrastive pretraining, an entropic regularized KL-div (with parameter $\lambda = 0.01$) is utilized for fine-tuning. They also propose to incorporate the FLM approach described earlier in this section, so we present results with and without this pseudo-labeling method. Given that the input to the classifier are not images but features, it is unclear how to compute the contrastive loss defined in [9] (e.g. what type of augmentations are appropriate). Thus, we pretrain the model with the domain agnostic contrastive loss introduced in [5] instead, with SGD, and learning rate $1e^{-4}$ for 1000 epochs.

Table 5 shows the results in synthetic and real data. In both cases the VP loss (both variants) leads to improvements with respect to the original case, thus **confirming the complementary nature of the proposed VP loss and representation learning methods**.

	Loss	Accuracy (Syn.)	Error (Real)
LLP-GAN [4]	**KL-div**	33.33	8.13
	VP-L2	35.94	6.38
	VP-CE	**47.54**	**6.23**
Contrastive pretraining [9,5]	**KL-div + entropy**	18.11	52.78
	KL-div + entropy + FLM	33.33	25.30
	VP-L2	59.01	**5.44**
	VP-CE	**61.02**	6.15

Table 5: Accuracy and mean absolute error in synthetic (Syn.) and real (Real) data, respectively.

5 Detection results

We compare the detection performance of the proposed DSD to (1) Cellpose [6]; (2) a baseline approach (which applies local softmax and NMS directly on the absolute value of the input image); and (3) CSC priors [10], previously proposed for cell detection in lensless imaging. Detection metrics are reported in Table 6. For each method, a grid search over the hyperparameter space is performed using the validation set, and the best models are reported. Cellpose is presented as a generalist method that does not require retraining [6], yet in our case it obtains the lowest performance across all detection metrics, which emphasizes that *lensless imaging can benefit from specialized models*. DSD outperforms the baseline and achieves the best precision. The CSC priors method obtains the best recall and f1-score, however *its performance comes at the expense of additional data labeling and computational complexity* (it uses holographic reconstruction $SPR_{\mathcal{T}(\cdot)}$ as preprocessing [2], and retrieves one cell per iteration). In real data we

utilize WBC concentration as a proxy to evaluate the detection performance of our method. The last row of 6 shows a significant correlation ($\rho = 0.93$) between the number of cells detected by our model and the GT cell concentrations, and same conclusions obtained from synthetic data hold for real data.

Method	Input	Synthetic data			Real data
		Precision	Recall	F1-score	Corr. coeff.
Cellpose [6]	$\lvert H * \mathcal{T} \rvert$	0.89	0.64	0.74	$\rho = 0.35$
Baseline	$\lvert H * \mathcal{T} \rvert$	0.92	0.88	0.90	$\rho = 0.86$
DSD (ours)	$H * \mathcal{T}$	**0.96**	0.90	0.93	$\rho = 0.93$
CSC priors [10]	$SPR_{\mathcal{T}}(H)$	0.94	**0.95**	**0.94**	$\rho =$ **0.95**

Table 6: Detection results. Mean detection metrics in synthetic data. Corr. coeff. between GT concentration and predicted counts in real data.

References

1. Habibzadeh, M., Krzyżak, A., Fevens, T.: White blood cell differential counts using convolutional neural networks for low resolution images. In: Artificial Intelligence and Soft Computing: 12th International Conference, ICAISC 2013, Zakopane, Poland, June 9-13, 2013, Proceedings, Part II 12. pp. 263–274. Springer (2013) 3
2. Haeffele, B.D., Stahl, R., Vanmeerbeeck, G., Vidal, R.: Efficient reconstruction of holographic lens-free images by sparse phase recovery. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 109–117. Springer (2017) 4
3. He, S., Minn, K.T., Solnica-Krezel, L., Anastasio, M.A., Li, H.: Deeply-supervised density regression for automatic cell counting in microscopy images. Medical Image Analysis **68**, 101892 (2021) 3
4. Liu, J., Wang, B., Qi, Z., Tian, Y., Shi, Y.: Learning from label proportions with generative adversarial networks. Advances in neural information processing systems **32** (2019) 3, 4
5. Nandy, J., Saket, R., Jain, P., Chauhan, J., Ravindran, B., Raghuveer, A.: Domain-agnostic contrastive representations for learning from label proportions. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management. pp. 1542–1551 (2022) 4
6. Stringer, C., Wang, T., Michaelos, M., Pachitariu, M.: Cellpose: a generalist algorithm for cellular segmentation. Nature methods **18**(1), 100–106 (2021) 4, 5
7. Villani, C.: Topics in optimal transportation, vol. 58. American Mathematical Soc. (2021) 1
8. Xue, Y., Ray, N., Hugh, J., Bigras, G.: Cell counting by regression using convolutional neural network. In: Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part I 14. pp. 274–290. Springer (2016) 2
9. Yang, H., Zhang, W., Lam, W.: A two-stage training framework with feature-label matching mechanism for learning from label proportions. In: Asian Conference on Machine Learning. pp. 1461–1476. PMLR (2021) 4
10. Yellin, F., Haeffele, B.D., Roth, S., Vidal, R.: Multi-cell detection and classification using a generative convolutional model. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8953–8961 (2018) 4, 5