Evaluating Physically Motivated Loss Functions for Photometric Redshift Estimation

Andrew Engel[1] **Jan Strube**[1,2]

[1]Pacific Northwest National Laboratory
[2]University of Oregon, Institute of Fundamental Physics
{andrew.engel,jan.strube}@pnnl.gov

Abstract

Physical constraints have been suggested to make neural network models more generalizable, act scientifically plausible, and be more data-efficient over unconstrained baselines. In this report, we present preliminary work on evaluating the effects of adding soft physical constraints to computer vision neural networks trained to estimate the conditional density of redshift on input galaxy images for the Sloan Digital Sky Survey. We introduce physically motivated soft constraint terms that are not implemented with differential or integral operators. We frame this work as a simple ablation study where the effect of including soft physical constraints is compared to an unconstrained baseline. We compare networks using standard point estimate metrics for photometric redshift estimation, as well as metrics to evaluate how faithful our conditional density estimate represents the probability over the ensemble of our test dataset. We find no evidence that the implemented soft physical constraints are more effective regularizers than augmentation.

1 Introduction

Empirical photometric redshift algorithms regress a target galaxy redshift from measurements of flux received on a telescope's band-pass filters. Highly scalable algorithms for photometric redshifts are critical for achieving the scientific objectives of many upcoming surveys, including the Vera C. Rubin Legacy Survey of Space and Time (17), the Euclid Wide Survey (21), and the Nancy Roman Space Telescope Wide Area Survey (1). Utilizing legacy survey measurements of spectroscopic redshift as labels, we can cast the problem in a supervised machine learning (ML) framework (4, 8, 11, 13, 25). Recent work that has explored ML for photometric redshifts has focused on how to model and evaluate estimates of the conditional density estimate (CDE) of redshift (5, 10, 18). See Newman & Gruen (23) for a review. The benefit of regressing a CDE instead of a point-estimate is that downstream analyses (e.g., weak lensing studies (22)) make use of the uncertainty and shape encoded in the CDE. This stresses the importance that the CDEs should be well calibrated and be highly expressive.

This brief workshop paper builds on work utilizing deep learning computer vision models to regress photometric redshift conditional density estimates (8, 11, 13, 15, 16) for galaxies with photometry matching the SDSS main galactic sample (34) and with redshift $\hat{z} \leq 0.4$ by incorporating soft physical constraints. These are terms added to a loss function that encode known behavior that solutions should exhibit (20). Our main motivation for including these constraints is to evaluate whether they lead to a more robust model (as suggested in (2, 28)), here measured using generalization performance on a held-out portion of our dataset. A secondary motivation is to analyze a broad set of soft physical constraints for neural network models to explore if they can be incorporated for scientific objectives.

Machine Learning and the Physical Sciences Workshop, NeurIPS 2023.

2 Background

Neural Networks A set of data inputs and targets $\mathcal{D} = \{(\boldsymbol{x}_1, \hat{\boldsymbol{z}}_1), (\boldsymbol{x}_2, \hat{\boldsymbol{z}}_2), \ldots, (\boldsymbol{x}_N, \hat{\boldsymbol{z}}_N)\}$ is sampled from a population $\mathcal{X} \subseteq (\mathbb{R}^{N\times m}, \mathbb{R}^{N\times 1})$, with $N \in \mathbb{Z}$ number of samples and input feature dimension $m \in \mathbb{Z}$. A neural network is a vector valued differentiable function that maps input samples to vector targets, $f(\boldsymbol{x}_i,\ \boldsymbol{\theta}) \to \hat{\boldsymbol{z}}_i$, and is parameterized by a vector of learnable weights $\boldsymbol{\theta}$. These are updated to minimize a scalar objective function $\mathcal{L}$, via first order optimizers. This is known as the standard supervised classification problem.

Photometric Redshift Conditional Density Estimation For this task, we model the probability of redshift given the input features with a neural network $f(\boldsymbol{x}_i,\ \boldsymbol{\theta}) \sim P(\hat{\boldsymbol{z}}|\ \boldsymbol{x}_i)$. We frame the problem as supervised classification (this approach was taken by (25)). The loss, $\mathcal{L}$, is chosen to be the cross entropy loss with a weight decay penalty,

$$\mathcal{L}_{\text{CE}} = \frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} -\log\left(\frac{\exp(f(\boldsymbol{x}_i,\ \boldsymbol{\theta})^c)}{\sum_{j=1}^{C}\exp(f(\boldsymbol{x}_i,\ \boldsymbol{\theta})^j)}\right)\cdot\delta_c^{\hat{\boldsymbol{z}}} + \gamma_0||\boldsymbol{\theta}||^2.$$

Where $f(\boldsymbol{x}_i,\ \boldsymbol{\theta})^j$ is the j-th component of the vector-valued function $f(\boldsymbol{x}_i,\ \boldsymbol{\theta})$, $\delta_c^{\hat{\boldsymbol{z}}}$ is the standard Kronecker delta, and γ_0 is a hyperparameter controlling the relative importance of the weight decay term. We have implicitly binned $\hat{\boldsymbol{z}}$ into discrete bins represented by the vector output of $f(\boldsymbol{x}_i,\ \boldsymbol{\theta})$, such that $\delta_c^{\hat{\boldsymbol{z}}}$ is 1 if $z_c < \hat{\boldsymbol{z}} \leq z_{c+1}$, and 0 otherwise.

Physically Constrained Neural Networks We modify the standard conditional density estimation problem described above by appending additional terms to the cross entropy loss that add additional constraints for the model to obey. In full generality, these additional terms take the form:

$$\mathcal{L}_{\text{phy}} = \sum_{j=1}^{M}\gamma_j\sum_{i=1}^{N}\mathcal{P}_j(f(\boldsymbol{x}_i,\ \boldsymbol{\theta}))$$

where $f(\boldsymbol{x}_i,\ \boldsymbol{\theta})$ is a neural network acting on datapoint $\boldsymbol{x}_i$, M is the number of soft physical constraint terms, $\mathcal{P}_j$ is an additional operator encoding physics, and γ_j is a scalar hyperparameter controlling the relative scale of the constraint. The total loss function can be described as the addition of the ridge regression term and the constraint(s), $\mathcal{L} = \mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{phy}}$

3 Physically Informed Constraints

Probabilistic Spectrophotmetric Flux Calculation The spectrophotometric flux Φ is the flux observed through a photometric filter $R(\lambda)$ by convolving a spectral energy density $F(\lambda)$ with said filter (24). The spectrophotometric flux is given as:

$$\Phi(\lambda) = \frac{\int_{\lambda_a}^{\lambda_b} F(\lambda)R(\lambda)\lambda \mathrm{d}\lambda}{\int_{\lambda_a}^{\lambda_b} c\frac{1}{\lambda}R(\lambda)\mathrm{d}\lambda}$$

let the Φ' represent the spectrophotometric flux that would be measured from the same spectral flux density but at a redshift given by $\boldsymbol{z}_i$. The wavelength of such an observer can be related to the original wavelength by $\lambda' = \frac{1+\boldsymbol{z}_i}{1+\hat{\boldsymbol{z}}_i}\lambda$. We will find it convenient to notate this factor as $\nu := \frac{1+\boldsymbol{z}_i}{1+\hat{\boldsymbol{z}}_i}$. We show in Appendix A that:

$$\Phi'(\nu) = \frac{\int_{\nu\lambda_a}^{\nu\lambda_b} F(\lambda)R(\lambda)\nu^2\lambda \mathrm{d}\lambda}{\int_{\nu\lambda_a}^{\nu\lambda_b} c\frac{1}{\lambda}R(\lambda)\mathrm{d}\lambda}.$$

Our neural network $f(\boldsymbol{x}_i,\ \boldsymbol{\theta})$ is trained to approximate the conditional density of redshift given the input data, $p(\boldsymbol{z}_i|\boldsymbol{x}_i)$. We can incorporate this CDE by calculating the expectation of Φ'. Interpreting the c-th output of $f(\boldsymbol{x}_i,\ \boldsymbol{\theta})$ to represent the probability of redshift $p(\boldsymbol{z}_i)$, we have $\mathbb{E}[\Phi'](f(\boldsymbol{x}_i,\ \boldsymbol{\theta})) = \int \Phi(\nu)f(\boldsymbol{x}_i,\ \boldsymbol{\theta})\mathrm{d}\boldsymbol{z}$. From this our first physical constraint can be stated as:

$$\mathcal{P}_{\text{Flux}}(f(\boldsymbol{x}_i,\ \boldsymbol{\theta})) = \frac{|\mathbb{E}[\Phi'](f(\boldsymbol{x}_i,\ \boldsymbol{\theta})) - \Phi'(1)|}{\Phi'(1)}.$$

Invariance to Rotations "Invariances" are constraints that can be interpreted as perturbations over which the model output should be unchanged (the term was coined in the context of generative model discriminators in Shah et al. (32)). The cosmological redshift of a galaxy is independent of the orientation of the galaxy in the sky. We define the operator $\Theta(\boldsymbol{x})$ to be the "rotation" operator, which takes as input image $\boldsymbol{x}$ and randomly rotates and flips $\boldsymbol{x}$ about the vertical or horizontal axis, then returns the resulting image. Our network output is constrained to be invariant under these rotations by incorporating the physical constraint

$$\mathcal{P}_{\text{rotation}}(f(\boldsymbol{x}_i,\ \boldsymbol{\theta})) = |f(\boldsymbol{x}_i,\ \boldsymbol{\theta}) - f(\Theta(\boldsymbol{x}_i),\ \boldsymbol{\theta})|^2.$$

Invariance to Background Pixels The redshift of a galaxy is independent of the pixel values that do not record flux from photons emitted from that galaxy. Let $\mathcal{B}(\boldsymbol{x})$ be an operator that resamples the noise of the background sky in input image $\boldsymbol{x}$ and returns the resulting image. We describe the background operator in more detail in appendix B.3. We can write the constraint of invariance to background as

$$\mathcal{P}_{\text{background}}(f(\boldsymbol{x}_i,\ \boldsymbol{\theta})) = |f(\boldsymbol{x}_i,\ \boldsymbol{\theta}) - f(\mathcal{B}(\boldsymbol{x}_i),\ \boldsymbol{\theta})|^2.$$

Conditional Density Estimate Loss To interpret the output of $f(\boldsymbol{x}_i,\ \boldsymbol{\theta})$ as a conditional density estimate certain properties must be met. One such property is that we would expect the set of many observations of the random variable $\mathcal{Z}$ from a given galaxy to follow the distribution $f(\boldsymbol{x}_i,\ \boldsymbol{\theta})$. Because there is only one observed redshift per galaxy, we can not evaluate this property exactly; however, we can evaluate a related value which is equal to the mean squared differences between the estimate of CDE and the true CDE up to a constant of integration (18).

$$\mathcal{P}_{\text{CDE}}(f(\boldsymbol{x}_i,\ \boldsymbol{\theta})) = \int f(\boldsymbol{x}_i,\ \boldsymbol{\theta})^2 \mathrm{d}z - 2(f(\boldsymbol{x}_i,\ \boldsymbol{\theta})^{c=\hat{z}})^2$$

This term is referred to as the CDE loss within the photometric redshift literature (6, 10, 18).

4 Methodology

Compute We train all models on a DGX-2 A100 server on a single Nvidia A100 GPU.

Datasets We use the benchmark SDSS galactic redshift dataset first described in Pasquet et al. (25) and made available online in Dey et al. (9)*. We summarize the preparation here, but would refer the reader to Pasquet et al. (25) for details. The dataset is created from SDSS DR12 by selecting all spectroscopically confirmed galaxies that satisfy petroMag_r < 17.77, which is the same cut-off for target selection in the SDSS main galactic sample (34). 64 x 64 pixel cutouts are created in each of the five SDSS photometric bands (SDSS-u, SDSS-g, SDSS-r, SDSS-i, and SDSS-z) centered on each galaxy. We place 121,543 (20%) samples randomly into a held-out "test" dataset to measure performance at the end of our study. There are 486,169 samples in our training dataset, from which we randomly select 20,000 galaxies for a "validation set" that we can use to monitor performance throughout each training run. In addition, extinction due to dust is queried using the galaxies astrometric coordinates and incorporated into the network as an additional input at the fully connected head (30). This technique is now common practice (25).

Models Neural networks are constructed as a series of non-linear learnable functions that are commonly abstracted into "layers", with the entire configuration of layers called an "architecture". The choice of architecture represents a significant hyperparameter. Previous photometric redshift works evaluated the Inception architecture (15, 25, 35), the Capsule Network (8, 29), and ResNet50 (13, 14). For simplicity, we chose to evaluate our model on the ResNet50 architecture provided in the repository of (13), due to preference for the PyTorch framework (26), and for the fact that ResNet50 architecture has the best reported performance on the scatter-based metric. We provide additional implementation details such as choice of hyperparameters in Appendix B.4.

Augmentation The invariance to shifts and background pixels would more classically be used as part of a data augmentation pipeline (13, 25). To separate the regularizing effect of augmentation from the specific inclusion of the invariance to the loss function, we will train our baseline model with

*Data available online at: https://biprateep.de/encapZulate-1/data.html

and without rotation and background re-sampling augmentations. In practice, we note that because we already include the rotation and sampling as part of our pipeline, we actually combine these into one term:

$$\mathcal{P}_{\text{invariances}}(f(\boldsymbol{x}_i,\ \boldsymbol{\theta})) = |f(\boldsymbol{x}_i,\ \boldsymbol{\theta}) - f(\Theta(\mathcal{B}(\boldsymbol{x}_i)),\ \boldsymbol{\theta})|^2.$$

4.1 Metrics

We will track three point estimate metrics consistent with evaluations from prior work. As a measure of spread, we will report the **median absolute deviation MAD** $= 1.4826 \times \text{median}(|\frac{f(\boldsymbol{x}_i,\ \boldsymbol{\theta}) - \hat{\boldsymbol{z}}}{1+\hat{\boldsymbol{z}}}|)$.[†] We track the **bias** of the residuals as $\text{bias} = \mathbb{E}[\frac{f(\boldsymbol{x}_i,\ \boldsymbol{\theta}) - \hat{\boldsymbol{z}}}{1+\hat{\boldsymbol{z}}}]$. Finally, we report the **catastrophic outlier rate**, $\mathcal{O}$ as the fraction of predictions with scaled residual greater than 0.05. These are the same metrics as reported in (8). Taking the number of datapoints in the test dataset to be N_{test}, and the set of integers up to and including N as $\{1, 2, \ldots, N\} = [N]$ then $\mathcal{O} = \frac{100}{N_{\text{test}}} \times \left|\{i \in [N_{\text{test}}]|\ \|\frac{f(\boldsymbol{x}_i,\ \boldsymbol{\theta}) - \hat{\boldsymbol{z}}}{1+\hat{\boldsymbol{z}}}\| > 0.05\}\right|$.

In addition, we also evaluate the performance of our network in modeling the conditional density estimate. We report the value of the mean conditional density estimate loss evaluated over the held-out test dataset and provide visualizations of the probability integral transform (PIT) (7, 27). We introduce and present PIT visualizations in appendix C.

5 Results

A series of ResNet50 neural networks were trained on a random sample of galaxies cutouts from the SDSS DR12 spectroscopic main galaxy sample catalogue to predict the conditional density estimate of redshift under various types of loss functions. We have three different experimental groups. Our baseline evaluates ResNet50 trained in the manner of (25) with just the cross-entropy loss function. We train a second baseline in the same manner but include both random resampling of background pixels from the sky-distribution and random flips and rotations as augmentations in our data loading pipeline. Our next group uses all of the physical constraints identified in section 3 in tandem with the cross entropy loss term. We present our results in Table 1 alongside contemporary works evaluated on nearly the same dataset.

Table 1: **Ablation Performance Metrics And Comparable Works.** point-performance metrics including MAD as a metric of scatter, Bias as the average residual, and $\mathcal{O}$ as the percentage of scaled residuals greater than 0.15 are reported. We measure minor improvements in MAD and Bias using our physical constraints.

Model	MAD	Bias	$\mathcal{O}$	CDE loss	Train Time [h]
Baseline w/o Aug.	1.17e-2	7.5e-4	**1.48%**	**-8.34e-4**	**7.16**
Baseline w/ Aug.	1.17e-2	7.1e-4	2.22%	-7.69e-4	8.16
Physically Constrained	**1.12e-2**	**-1.5e-4**	1.88%	-8.04e-4	24.52
Beck et al. (3)	1.43e-2	1.6e-3	2.5%	*	*
Dey et al. (8)	8.98e-3	7e-5	0.19%	*	*
Hayat et al. (13)	8.25e-3	1e-4	0.21%	*	*
Pasquet et al. (25)	9.12e-3	1e-4	0.31%	*	*

6 Discussion

This preliminary work seeks to answer whether a set of physical constraints can be added to a network to increase generalization. Our work is interesting because we study soft physical constraints that are not of the typical form from the physically informed literature– the operators take neither a differential or integral form. We show that it is possible to add penalty terms sensitive to known properties that photometric redshift estimates should obey as additional terms to the loss function.

[†]The 1.4826 coefficient scales MAD such that for normally distributed $1 \cdot \text{MAD} = 1 \cdot \text{STD}$

Our experiment shows that the augmentations implemented have a positive effect on the point-wise performance of the network, and decrease the variance in the conditional probability density output over the space of augmented images (see appendix E). We observed including PINN terms did not meaningfully improve over augmentation, having very little effect overall on the point-estimate performance metrics (table 1), how well calibrated the conditional probability estimates are (appendix C), how much variance the network exhibits on the augmentations (appendix E), or on the overall structure of the photometric redshift vs spectroscopic redshift relationship (appendix D). Worse, they dramatically increase training time.

7 Future Work

The most obvious extension of this work would be to search over hyper-parameter γ_i for each loss-term. We would also like multiple runs with different seeds for each experiment to understand the variance of the performance metrics under each experimental group. Our individual pipeline to create the constraints could be improved: for example, through better modeling of which pixels contain flux from the host galaxy. Finally, prior work on understanding the failures of soft-physical constraints from the perspective of the loss function geometry suggest a methodology to visualize the loss landscape and assess whether it is likely additional loss terms help or hurt optimization (12, 20). We would augment our study of physical constraints to visualize the loss. It is believed that high variance loss landscapes contribute to poor performance through their geometry being difficult for optimizers to navigate (20).

8 Limitations

This is preliminary work and all early conclusions that we draw are limited by that fact. Most crucially, we have not performed a hyper-parameter search over values of γ, nor have we studied the variation in results over multiple training runs initialized from different seeds. For example, this could reveal that in expectation one of the experimental groups could be said to perform better than the others. Our methodology is also limited by the fact that we have not found an efficient way to calculate each of the PINN terms, so even if we found that these PINN terms were to help, its unclear whether small research groups without access to large compute clusters could benefit.

9 Acknowledgements

This work made use of data products from the Sloan Digital Sky Survey data release 12. Funding for SDSS-III has been provided by the Alfred P. Sloan Foundation, the Participating Institutions, the National Science Foundation, and the U.S. Department of Energy Office of Science. The SDSS-III web site is http://www.sdss3.org/.

SDSS-III is managed by the Astrophysical Research Consortium for the Participating Institutions of the SDSS-III Collaboration including the University of Arizona, the Brazilian Participation Group, Brookhaven National Laboratory, Carnegie Mellon University, University of Florida, the French Participation Group, the German Participation Group, Harvard University, the Instituto de Astrofisica de Canarias, the Michigan State/Notre Dame/JINA Participation Group, Johns Hopkins University, Lawrence Berkeley National Laboratory, Max Planck Institute for Astrophysics, Max Planck Institute for Extraterrestrial Physics, New Mexico State University, New York University, Ohio State University, Pennsylvania State University, University of Portsmouth, Princeton University, the Spanish Participation Group, University of Tokyo, University of Utah, Vanderbilt University, University of Virginia, University of Washington, and Yale University.

We thank Nell Byler and Gautham Narayan for help in preparing this manuscript. A.E. and J.S. were partly supported by the Open Call Initiative, under the Laboratory Directed Research and Development (LDRD) Program at Pacific Northwest National Laboratory (PNNL). PNNL is a multi-program national laboratory operated for the U.S. Department of Energy (DOE) by Battelle Memorial Institute under Contract No. DE-AC05-76RLO 1830.

References

[1] Akeson, R., Armus, L., Bachelet, E., et al. 2019, arXiv e-prints, arXiv:1902.05569, doi: 10.48550/arXiv.1902.05569

[2] Banerjee, C. K., Nguyen, K., Fookes, C., & Karniadakis, G. E. 2023, ArXiv, abs/2305.18035

[3] Beck, R., Dobos, L., Budav'ari, T., Szalay, A. S., & Csabai, I. 2016, Monthly Notices of the Royal Astronomical Society, 460, 1371

[4] Collister, A., & Lahav, O. 2003, Publications of the Astronomical Society of the Pacific, 116, 345

[5] Dalmasso, N., Pospisil, T., Lee, A. B., et al. 2019, Astron. Comput., 30, 100362

[6] Dalmasso, N., Pospisil, T., Lee, A. B., et al. 2020, Astronomy and Computing, 30, 100362, doi: 10.1016/j.ascom.2019.100362

[7] Dawid, A. P. 1984, J. R. Stat. Soc. A, 2, 278

[8] Dey, B., Andrews, B. H., Newman, J. A., et al. 2021, Monthly Notices of the Royal Astronomical Society

[9] Dey, B., Andrews, B. H., Newman, J. A., et al. 2022, Monthly Notices of the Royal Astronomical Society, 515, 5285, doi: 10.1093/mnras/stac2105

[10] Dey, B., Newman, J. A., Andrews, B. H., et al. 2021, ArXiv, abs/2110.15209

[11] D'Isanto, A., & Polsterer, K. L. 2018, Astronomy and Astrophysics, 609, A111, doi: 10.1051/0004-6361/201731326

[12] Elhamod, M., & Karpatne, A. 2023, arXiv e-prints, arXiv:2309.14601, doi: 10.48550/arXiv.2309.14601

[13] Hayat, M. A., Stein, G., Harrington, P., Lukić, Z., & Mustafa, M. 2021, The Astrophysical Journal Letters, 911, L33, doi: 10.3847/2041-8213/abf2c7

[14] He, K., Zhang, X., Ren, S., & Sun, J. 2015, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770

[15] Henghes, B., Thiyagalingam, J., Pettitt, C., Hey, T., & Lahav, O. 2022, Mon. Not. Roy. Astron. Soc., 512, 1696, doi: 10.1093/mnras/stac480

[16] Hoyle, B. 2015, Astron. Comput., 16, 34

[17] Ivezić, Ž., Kahn, S. M., Tyson, J. A., et al. 2019, The Astrophysical Journal, 873, 111, doi: 10.3847/1538-4357/ab042c

[18] Izbicki, R., & Lee, A. B. 2016, Journal of Computational and Graphical Statistics, 25, 1297

[19] Kingma, D. P., & Ba, J. 2014, CoRR, abs/1412.6980

[20] Krishnapriyan, A., Gholami, A., Zhe, S., Kirby, R., & Mahoney, M. W. 2021, in Advances in Neural Information Processing Systems, ed. A. Beygelzimer, Y. Dauphin, P. Liang, & J. W. Vaughan. https://openreview.net/forum?id=a2Gr9gNFD-J

[21] Laureijs, R., Amiaux, J., Arduini, S., et al. 2011, Euclid Definition Study Report. https://arxiv.org/abs/1110.3193

[22] Mandelbaum, R. 2018, Annual Review of Astronomy and Astrophysics, 56, 393, doi: 10.1146/annurev-astro-081817-051928

[23] Newman, J. A., & Gruen, D. 2022, Annual Review of Astronomy and Astrophysics

[24] Oke, J. B., & Gunn, J. E. 1983, The Astrophysical Journal, 266, 713, doi: 10.1086/160817

[25] Pasquet, J., Bertin, E., Treyer, M., Arnouts, S., & Fouchez, D. 2019, Astronomy and Astrophysics, 621, A26, doi: 10.1051/0004-6361/201833617

[26] Paszke, A., Gross, S., Massa, F., et al. 2019, in Neural Information Processing Systems. https://api.semanticscholar.org/CorpusID:202786778

[27] Polsterer, K. L., D'Isanto, A., & Gieseke, F. 2016, arXiv: Instrumentation and Methods for Astrophysics

[28] Raissi, M., Perdikaris, P., & Karniadakis, G. 2019, Journal of Computational Physics, 378, 686, doi: https://doi.org/10.1016/j.jcp.2018.10.045

[29] Sabour, S., Frosst, N., & Hinton, G. E. 2017, ArXiv, abs/1710.09829

[30] Schlegel, D. J., Finkbeiner, D. P., & Davis, M. 1998, The Astrophysical Journal, 500, 525, doi: 10.1086/305772

[31] Schmidt, S. J., Malz, A. I., Malz, A. I., et al. 2020, Monthly Notices of the Royal Astronomical Society

[32] Shah, V., Joshi, A., Ghosal, S., et al. 2019, ArXiv, abs/1906.01626

[33] Singal, J., Silverman, G., Jones, E., et al. 2021, The Astrophysical Journal, 928

[34] Strauss, M. A., Weinberg, D. H., Lupton, R. H., et al. 2002, The Astronomical Journal, 124, 1810, doi: 10.1086/342343

[35] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. 2015, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2818

A Full Derivation of Spectrophotometric Flux

If we model the spectrophotometric flux as redshifting the spectra then the two are related as $F(\lambda') = F(\lambda)$. The photometric filter $R(\lambda)$ is defined in the observer's frame and does not receive any redshifting. With this in place we can write the spectrophotometric flux that an observer at redshift $f(\boldsymbol{x}_i,\ \boldsymbol{\theta})$ from the same galaxy would measure:

$$\Phi'(\lambda') = \frac{\int_{\lambda'_a}^{\lambda'_b} F(\lambda')R(\lambda)\lambda' \mathrm{d}\lambda'}{\int_{\lambda'_a}^{\lambda'_b} c\frac{1}{\lambda'}R(\lambda)\mathrm{d}\lambda'}$$

Making our substitutions and utilizing a change of variables, we can state the spectrophotmetric flux as an equation of the original wavelength and ν. We will drop the argument λ as we will show that the crucial dependence of Φ' on $f(\boldsymbol{x}_i,\ \boldsymbol{\theta})$ is encapsulated in the term ν.

$$\Phi'(\nu) = \frac{\int_{\nu\lambda_a}^{\nu\lambda_b} F(\lambda)R(\lambda)\nu^2\lambda \mathrm{d}\lambda}{\int_{\nu\lambda_a}^{\nu\lambda_b} c\frac{1}{\lambda}R(\lambda)\mathrm{d}\lambda}$$

Because $f(\boldsymbol{x}_i,\ \boldsymbol{\theta})$ approximates a conditional density estimate of $\boldsymbol{z}_i$, we can incorporate this CDE by calculating the expectation of Φ'. Interpreting the c-th output of $f(\boldsymbol{x}_i,\ \boldsymbol{\theta})$ to represent the probability of redshift $p(\boldsymbol{z}_i)$:

$$\mathbb{E}[\Phi'](f(\boldsymbol{x}_i,\ \boldsymbol{\theta})) = \int \Phi(\nu)f(\boldsymbol{x}_i,\ \boldsymbol{\theta})\mathrm{d}\boldsymbol{z}.$$

From this our first physical constraint can be stated as:

$$\mathcal{P}_{\text{Flux}} = \frac{|\mathbb{E}[\Phi'](f(\boldsymbol{x}_i,\ \boldsymbol{\theta})) - \Phi'(1)|}{\Phi'(1)}.$$

B Various Implementation Notes

B.1 Spectrophotometric Loss

Finally, a few notes on the implementation of this computation. We approximate the integrals discretely using the trapezoid rule, which requires that all terms be known at the same λ'. We linearly interpolate the photometric filter $R(\lambda)$ to achieve this mapping. The entire computation is done inside the PyTorch computational graph, allowing backpropogation through the calculation. One limitation of this technique is that $F(\lambda)$ is only known on the measurement interval of the original measuring spectropraph. If the bandpass filter intersects with a region where the $F(\lambda)$ is outside the measurement range, we set the value of $F(\lambda)$ to zero. This biases our calculation of $\mathbb{E}[\Phi']$. Finally, we avoid the denominator of Φ' calculation becoming zero by defining Φ' to be piece-wise equal to 0 when $\int_{\nu\lambda_a}^{\nu\lambda_b} c\frac{1}{\lambda}R(\lambda)\mathrm{d}\lambda = 0$. A second limitation is that we ignore the measurement error in the spectral flux density, the photometric bandpass filters, and the measured redshift $\hat{\boldsymbol{z}}$.

B.2 Rotation Loss

As a few notes on our rotational loss implementation, we only evaluate random flips and rotations in increments of $90°$, which will preserve the original pixel values to ignore the effects of aliasing. An interesting alternative implementation would be to query for additional pixels than necessary to create our galaxy cutouts which would enable arbitrary rotations without interpolating unknown pixel values. You could ignore the aliasing effects, then randomly draw a rotation angle ϕ from the interval $[0, 360°)$, and provide this rotation angle to the neural network. Using the rotation angle to rotate the input image as the first layer, it maybe possible to state this constraint as a differential operator $\frac{\mathrm{d}f(\boldsymbol{x}_i, \boldsymbol{\theta})}{\mathrm{d}\phi} = 0$, but we leave this to future/ongoing work.

B.3 Resampling Loss

As a note on the implementation: separating the background pixels from the galaxy pixel in a manner fast enough to be used as part of the pipeline to a neural network is not a solved problem. In this iteration of the work we have chosen to trade-off exact modeling of the source for speed in our deep learning pipeline so we do not rely on explicit modeling of the surface brightness of the galaxy. We take a conservative approach instead by setting a threshold value on a background pixel that is equal to the 1σ clipped average of the pixels in the image. Pixels that have value less than 1σ above the average pixel value are considered background and are randomly drawn from a Gaussian with mean and standard deviation measured from the image. Non-target sources (stars, other nearby galaxies) are not removed by this technique, but it is fast enough to be used in our training pipeline.

B.4 Neural Network Details

The neural network utilized is a standard ResNet50 architecture (14), re-initialized with random weights from the He Normal initialization. Table 2 gives the hyper-parmaeters for each experimental run. The Adam optimizer (19) was used for each experimental run. For hyperparameters not listed in Table 2 default values as provided in PyTorch 2.0.0 release were used. No dropout was used. The final activation had width 1025, where the extinction due to dust was concatenated into the feature space.

Table 2: Hyperparameters

Hyperparameter	**Baseline**	**Baseline w/ Aug.**	**PINN**
γ_{L2}	1e-06	1e-06	1e-06
γ_{Φ}	0	0	1e-1
$\gamma_{\text{Invariance}}$	0	0	1e-1
γ_{CDELoss}	0	0	1e-1
Initial LR	1e-3	1e-3	1e-3
Batch Size	128	128	128
NEpochs	50	50	50
Optimizer	Adam	Adam	Adam
LR Schedule factor	1e-1	1e-1	1e-1
LR schedule patience	2	2	2

C PIT Visual Metric

The Probability Integral Transform (PIT) is a visual metric that relays information about the bias and how overly or underly-dispersed the probability distributions are. The PIT is a common place metric for photometric redshift evaluation (31). It is simply a histogram of occurrences of cumulative probability distribution integrated up to the true value. The cumulative probability distribution up to the true value is defined:

$$CDF_i = \int_{-\infty}^{\hat{\boldsymbol{z}}} f(\boldsymbol{x}_i,\, \boldsymbol{\theta})\mathrm{d}z.$$

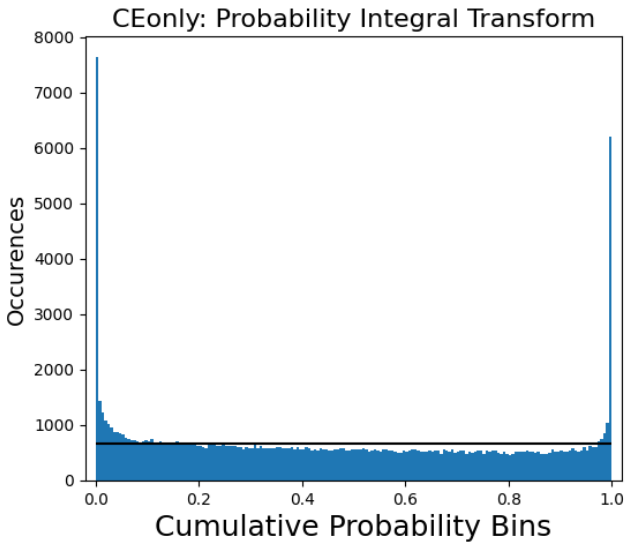

Figure 1: **Probability Integral Transform of Baseline** CE without augmentation is biased towards underestimation and is greatly under-dispersed, especially at the extreme tails.

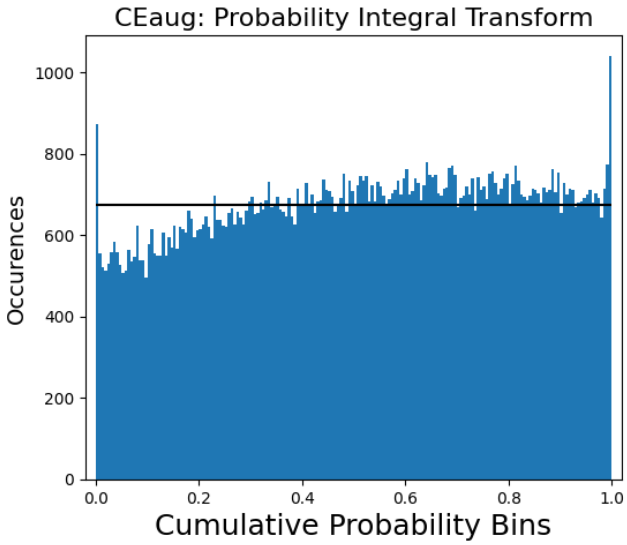

Figure 2: **Probability Integral Transform of Baseline with augmentations** CE with Augmentation appears biased towards overestimation and is under-dispersed at the extreme tails, to a lesser extent than the original CE without augmentation.

In the below, if the rate of occurrences all fall along the horizontal black line then the probability outputs from the model are well-calibrated. See figures 1, 2, and 3.

D Visualizing Point-Performance

In this section we plot the point estimates of photometric redshift from the expectation of the conditional density estimate output from our model against the true spectroscopic redshift using a Kernel Density Estimate (KDE) to visualize the density. We plot all catastrophic outliers as points to visualize their distribution as well. See figures 4, 5, 6

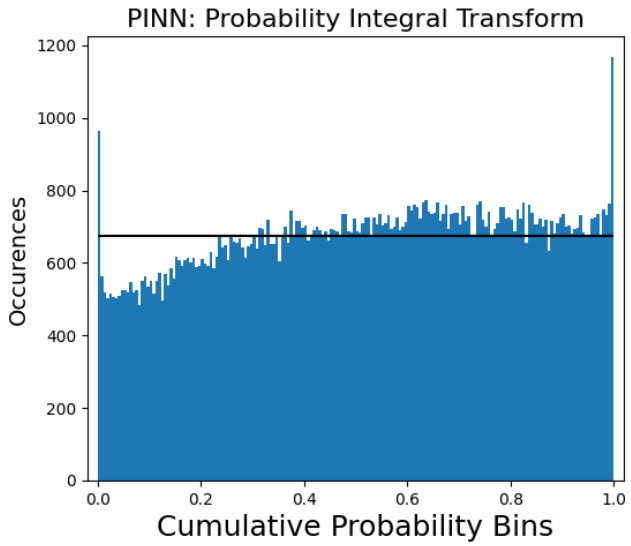

Figure 3: **Probability Integral Transform of Physically Constrained network.** Our Physically constrained network appears biased towards overestimation and is under dispersed at the extreme tails. The PIT plot appears very similar to CE with augmentation.

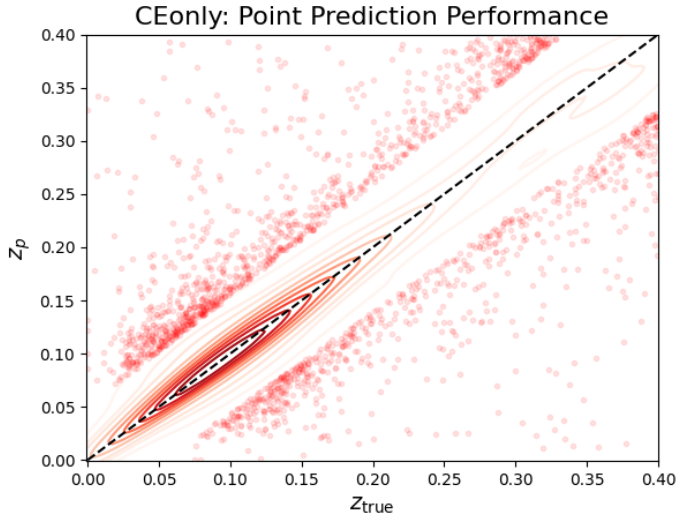

Figure 4: **KDE of point-estimate performance for Baseline.** We plot the Kernel Density Estimate and overlay all outliers defined as galaxies whose scaled residual point-estimation of redshift are greater than 0.15. While the density of outliers fall near the conic swept by the definition's limit, we see especially catastrophic outliers throughout the space.

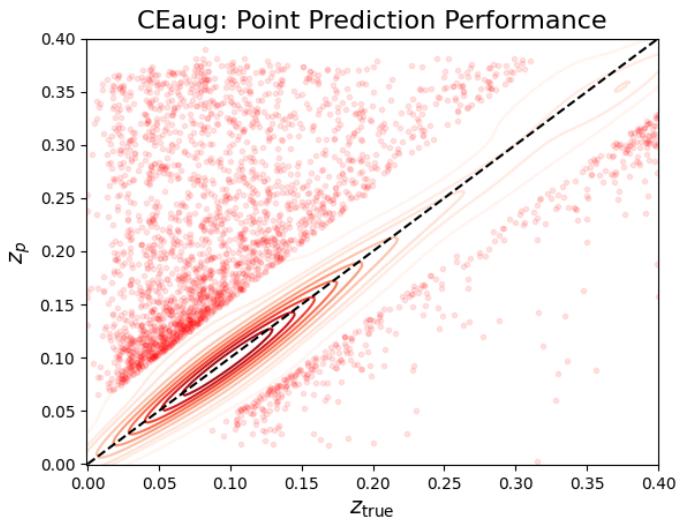

Figure 5: **KDE of point-estimate performance for Baseline with augmentations.** We plot the Kernel Density Estimate and overlay all outliers defined as galaxies whose scaled residual point-estimation of redshift are greater than 0.15. In comparison to the baseline model, we see the especially catastrophic outliers are much more likely to be over estimates of a redshift.

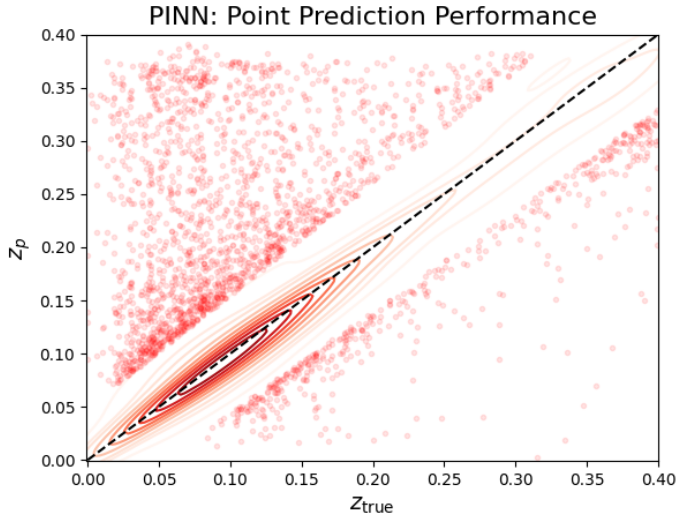

Figure 6: **KDE of point-estimate performance for PINN experiment.** We plot the Kernel Density Estimate and overlay all outliers defined as galaxies whose scaled residual point-estimation of redshift are greater than 0.15. In comparison to the baseline and with augmentation models, we see similarity between our PINN experiment and augmentation.

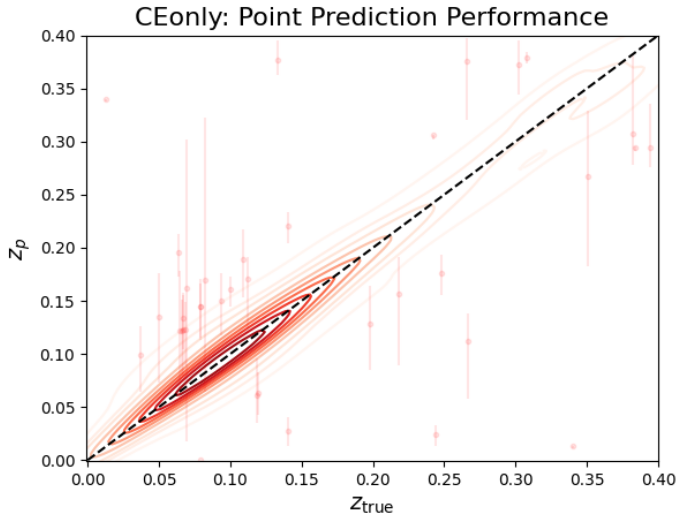

Figure 7: **KDE of point-estimate performance for Baseline, with random selection of errors** We plot the same KDE as fig 4, but for a small selection of outlier residuals we also plot the error bars inferred from the probabilistic output of the model.

D.1 Visualizing Outliers in Point-Performance

In this section we visualize the performance on our test-dataset just as above, but use the error-bar plotting functionality to randomly select a few catastrophic errors and plot their 0.05-0.95 confidence regions are vertical error bars. We compute the 0.05-0.95 confidence regions from the output of our model. We see evidence that many of these catastrophic errors may actually be within the expected tolerance given the output estimate of the conditional density. Future work could investigate the use of these error estimates to flag likely catastrophic errors from the test dataset for downstream cosmological analysis. Previous works have utilized photometric band alone to identify likely catastrophic outliers (33). See figures 7, 8, and 9.

E Visualizing Predictions and Variability due to Augmentations

In this section we randomly select 12 galaxies from the test set and show the distribution of outputs from each model on those same 12 galaxies. We use our augmentation pipeline to randomly add flips and rotations, and resample the background, to produce 20 different estimates of the same galaxy's conditional density of Redshift. The main observation is that the baseline model without augmentation produced more collapsed probability estimates with more variation to these rotations, which we should not expect the network to be sensitive to. See figures 10, 11, and 12.

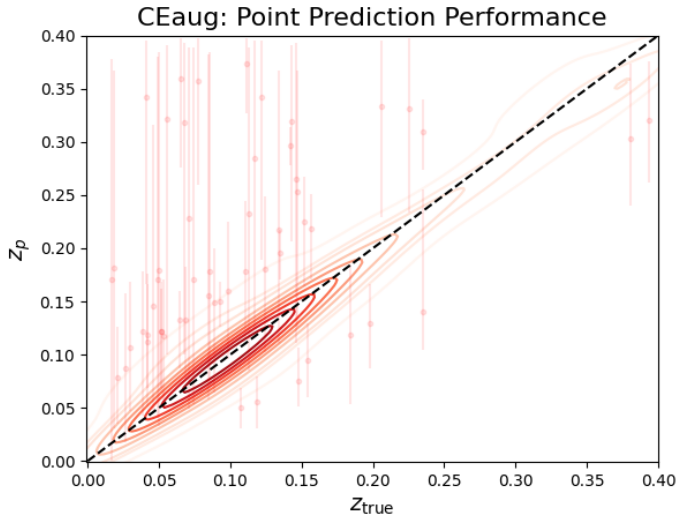

Figure 8: **KDE of point-estimate performance for Baseline with augmentations, with random selection of errors** We plot the same KDE as fig 5, but for a small selection of outlier residuals we also plot the error bars inferred from the probabilistic output of the model. In comparison to baseline, the error bars of these catastrophic outliers more frequently reach the correct value, implying the model is more accurately conveying its uncertainty about these points.

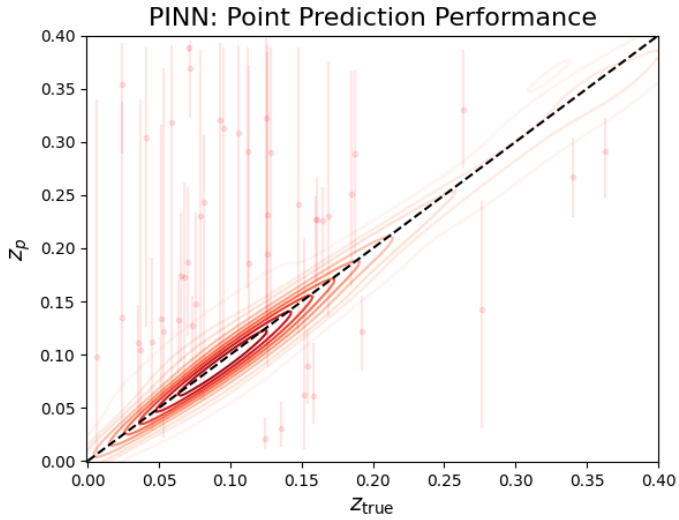

Figure 9: **KDE of point-estimate performance for PINN experiment, with random selection of errors** We plot the same KDE as fig 6, but for a small selection of outlier residuals we also plot the error bars inferred from the probabilistic output of the model. In comparison to the baseline and with augmentation figures, we see the constrained results as similar to the augmentation model

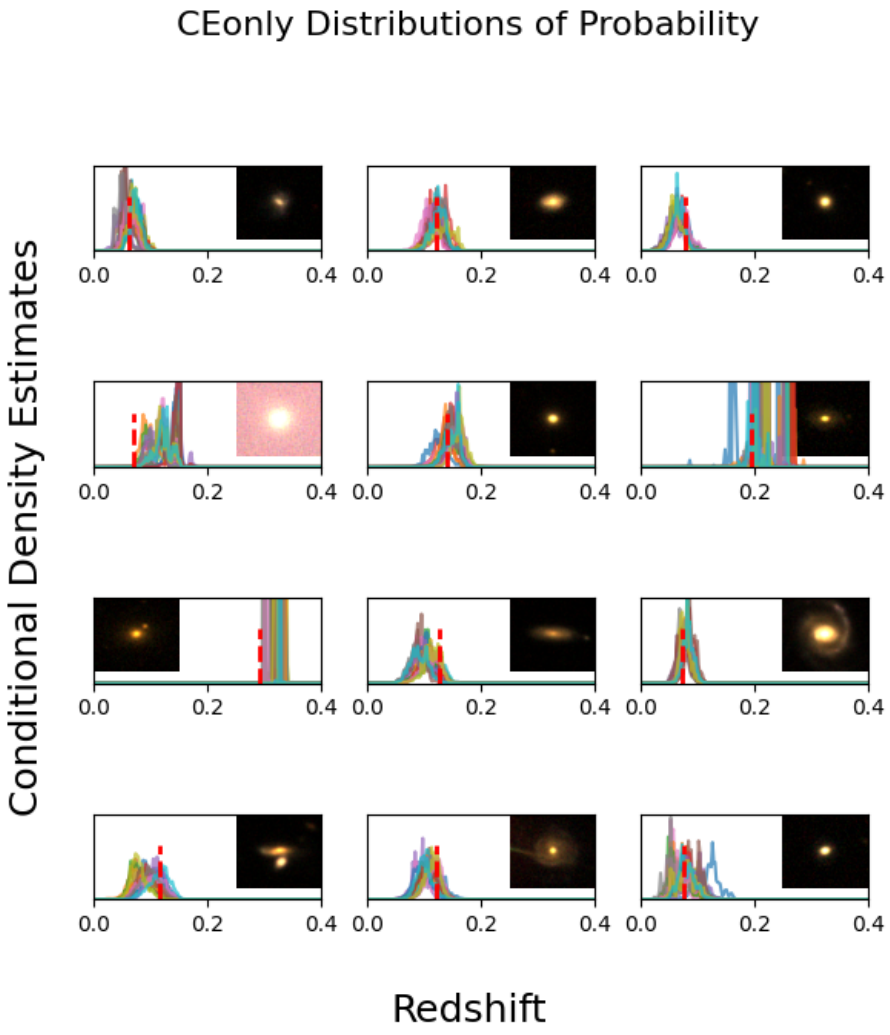

Figure 10: **A selection of Baseline model output conditional density estimates plot over 20 random rotations and background re-samplings of the shown galaxy.** The red dashed line is the value of true redshift. Each colored distribution is 1 of 20 re-sampled CDE from out network.

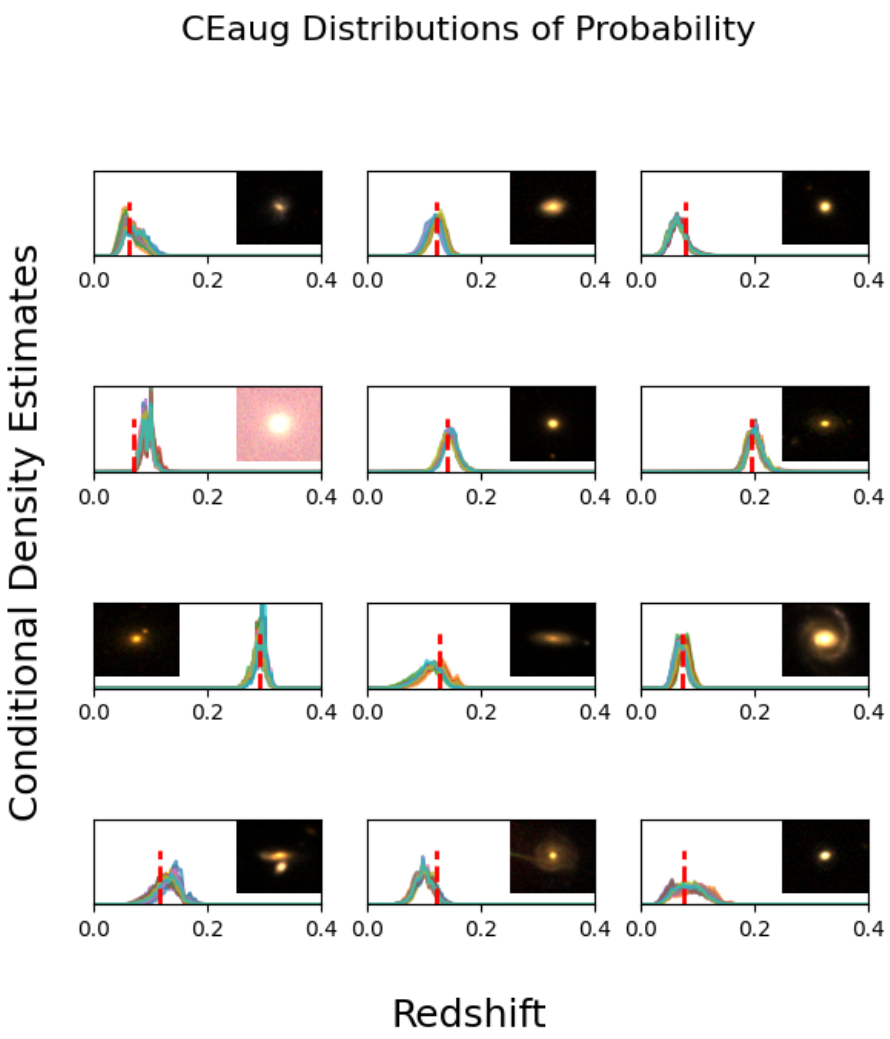

Figure 11: **A selection of Baseline w/ Augmentation model output conditional density estimates plot over 20 random rotations and background re-samplings of the shown galaxy.** In comparison the baseline model, the distributions are markedly less variant. This is the precise effect our invariance to rotation and background samples seeks to achieve, so it would appear these terms are more-or-less satisfied through augmentation alone.

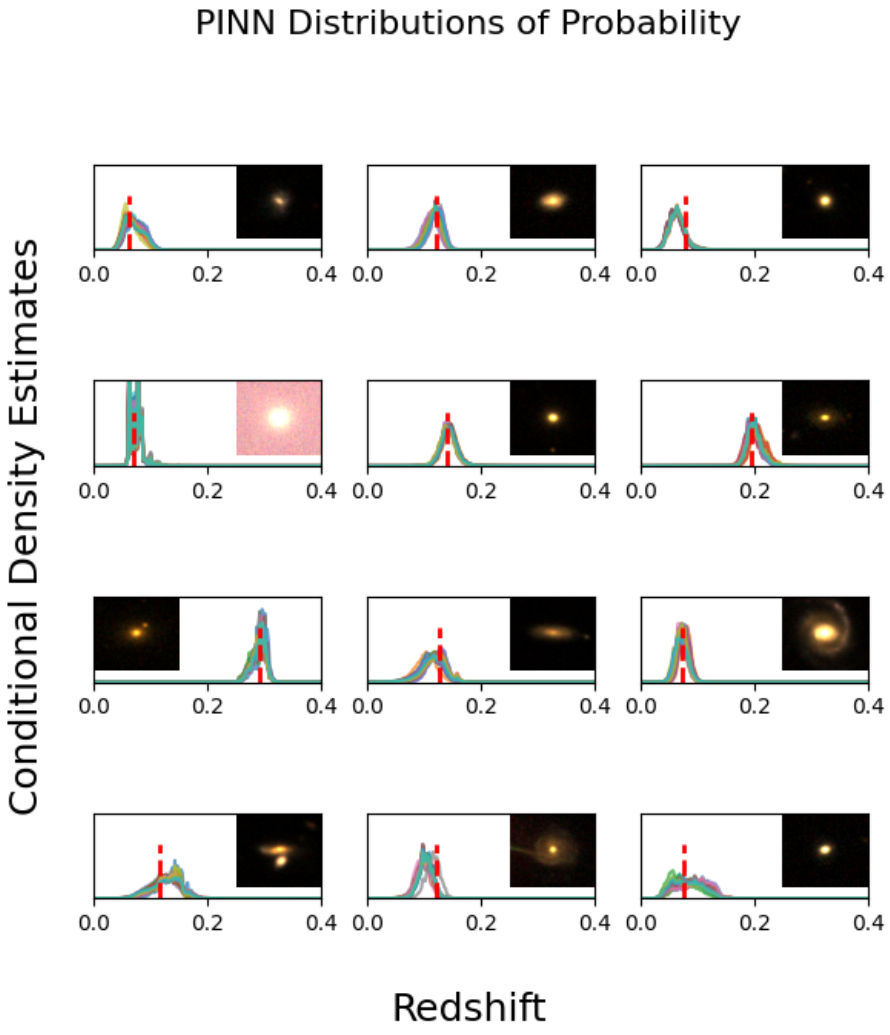

Figure 12: **A selection of PINN model output conditional density estimates plot over 20 random rotations and background re-samplings of the shown galaxy.** In comparison to the baseline model, out PINN model is less variant, but it is very similar to the augmentation model, which is much cheaper to train.