SUPSIAM: NON-CONTRASTIVE AUXILIARY LOSS FOR LEARNING FROM MOLECULAR CONFORMERS

Michael Maser
Prescient Design, Genentech
South San Francisco, CA
maserm@gene.com

Ji Won Park
Prescient Design, Genentech
South San Francisco, CA
parj2@gene.com

Joshua Yao-Yu Lin
Prescient Design, Genentech
South San Francisco, CA
liny82@gene.com

Jae Hyeon Lee
Prescient Design, Genentech
South San Francisco, CA
leej226@gene.com

Nathan C. Frey
Prescient Design, Genentech
South San Francisco, CA
freyn6@gene.com

Andrew Watkins
Prescient Design, Genentech
South San Francisco, CA
watkina6@gene.com

ABSTRACT

We investigate Siamese networks for learning related embeddings for augmented samples of molecular conformers. We find that a non-contrastive (positive-pair only) auxiliary task aids in supervised training of Euclidean neural networks (E3NNs) and increases manifold smoothness (MS) around point-cloud geometries. We demonstrate this property for multiple drug-activity prediction tasks while maintaining relevant performance metrics, and propose an extension of MS to probabilistic and regression settings. We provide an analysis of representation collapse, finding substantial effects of task-weighting, latent dimension, and regularization. We expect the presented protocol to aid in the development of reliable E3NNs from molecular conformers, even for small-data drug discovery programs.

1 BACKGROUND & INTRODUCTION

Modeling conformational shape is of critical importance in many molecular machine learning (MolML) tasks (Zheng et al., 2017). This is especially true in the drug discovery (DD) regime, e.g., for predicting the affinity of a ligand-protein binding interaction (Jones et al., 2021). However, many programs in ML-based DD (MLDD) rely on small, noisy datasets ($O\,10^{2-4}$) containing complex structures, making the development of generalizable 3D neural networks (NNs) particularly challenging.

Euclidean NNs (E3NNs) (Geiger & Smidt, 2022) make up the basis of many graph NN (GNN) models with equivariance to SE(3) transformations (Liao & Smidt, 2022; Batatia et al., 2022). Atomic coordinates are used to define radial edges for spatial message passing, increasing GNN expressivity over covalent-only adjacencies (Geiger & Smidt, 2022). Through-space interactions that intuitively influence structure activity relationships (SARs) (Kombo et al., 2013; Sauer & Schwarz, 2003) are thus explicitly modeled.

E3NNs have shown impressive performance for a variety of MolML tasks such as learning neural potentials (Zaidi et al., 2022; Devereux et al., 2020) and predicting electronic properties (Rackers et al., 2022; Thomas et al., 2018). However, their use in drug-activity modeling is comparatively rare, likely owing to the data challenges described above. Furthermore, in this space, little is understood about the generalizability and latent properties of E3NNs. We seek to address this here and to better understand representation learning with molecular conformers.

1.1 MOTIVATION

Given the fundamental dependence of drug-target binding and SARs on 3D structure, we sought an understanding of E3NN behavior around molecular geometries. During supervised training with 3D datasets, we found that models were strikingly sensitive to input coordinates (see Section 4.3).

This was seen as highly problematic, since models are often exposed to structures from varying experimental and/or computational methods in production settings. As such, we sought to develop learning methods to increase the generalizability of E3NNs to geometry perturbations.

Inspired by recent works in self-supervised learning (SSL) for (Mol)ML (Grill et al., 2020; Chen & He, 2020; Wang et al., 2020; Zaidi et al., 2022; Godwin et al., 2021), we devised auxiliary tasks to aid in training smooth, supervised E3NN manifolds. Simple Siamese networks (SimSiam) (Chen & He, 2020) were identified as a promising base method due to the following preliminary guidelines:

1. We desire that networks embed very similar geometries (e.g., augmented pairs) to nearby latent vectors
2. We do not require that distinct conformers of the same graph map to distant latent vectors (i.e., no negative pairs)
3. Likewise, we do not require that conformations of distinct graphs (i.e., unique molecules) map to distant latent vectors

Guideline 1 is intuitively motivated; given the geometry dependence of biophysical interactions that create SARs, we desire models to learn that similar geometries should receive similar predictions. For 2 and 3, our decision to avoid negative pairing is based on two non-assumptions. First, we recognize that, in many cases, it *will* be desirable for most conformers $c \in C$ to be embedded closely to one another. However, for many high-value tasks, we desire that networks precisely discriminate between individual conformers, as they may behave differently in biochemical systems. Second, molecules with unrelated connectivity (2D graph) can adopt very similar 3D shapes and possess similar *functional* properties (Sauer & Schwarz, 2003). Therefore, though often reasonable, we avoid the assumption that is desirable for models to force embeddings of distinct molecules apart.

Given this context, we developed positive-pair-only auxiliary tasks for 3D MolML. Presented herein are detailed investigations of *Sup*ervised *Siam*ese networks (*SupSiam*) for MLDD tasks.

2 Related Work

2.1 Multi-instance learning (MIL) with conformer ensembles (CEs)

Despite the growing literature in equivariant GNNs, relatively little focus has gone toward the effects of 3D conformers themselves (Axelrod & Gomez-Bombarelli, 2020; Ganea et al., 2021a;b; Isert et al., 2022). Existing efforts have even demonstrated a lack of performance gains from multiple-instance learning (MIL) with conformer ensembles (CEs) (Axelrod & Gomez-Bombarelli, 2020). Despite this, we expect that modeling the dynamics of CEs will be critical for many MLDD tasks, particularly for developing oracles that generalize to new 3D structures. To this end, we are unaware of detailed studies of MIL over CEs for activity-related tasks. Prior art in CE-MIL has focused only on electronic or quantum mechanical (QM) property prediction (Axelrod & Gomez-Bombarelli, 2020; Zaidi et al., 2022; Godwin et al., 2021), for which MolML has been well-demonstrated, even with 2D graphs (Wu et al., 2017).

2.2 Contrastive learning (CLR)

Contrastive learning (CLR) is in widespread use as a pre-training method for computer vision and other ML disciplines (Le-Khac et al., 2020). Recently, "MolCLR" was demonstrated to be effective for improving the performance of 2D GNNs in QM property prediction (Wang et al., 2021). Typical augmentation tasks include subgraph masking as well as node and/or edge dropout. Graphs augmented from the same parent molecule are treated as positive pairs, while those derived from different parents are treated as negative pairs. The pre-training objective is to minimize and maximize the embedding distance between positive and negative pairs, respectively.

The rationale for this objective states that similar — but different — 2D graphs should map to similar latent vectors (Le-Khac et al., 2020). We pose that this treatment could be counterproductive given the fundamentals of SARs that we desire models to learn: Changing a molecule's connectivity (input) should change its properties (labels). This is especially concerning in the small-molecule

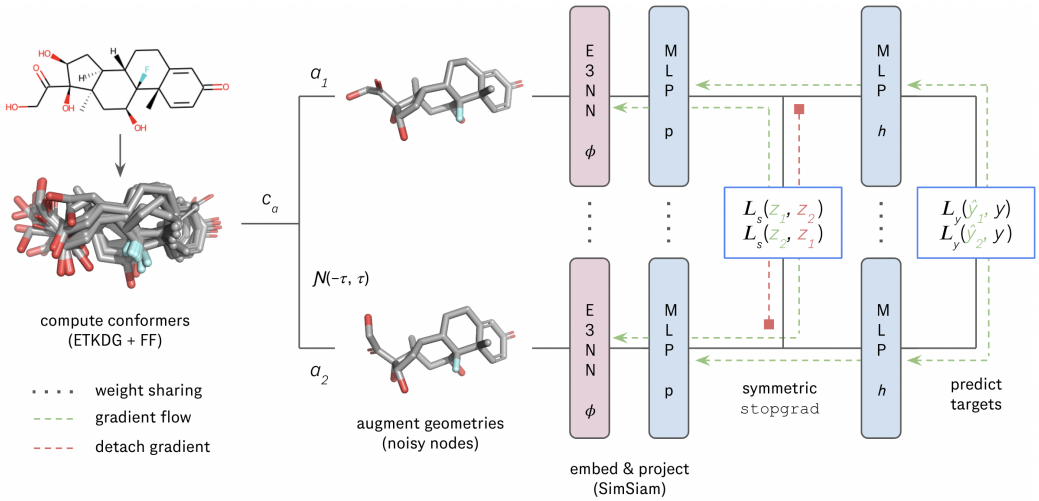


Figure 1: SupSiam pipeline.

setting (roughly ≤ 50 heavy atoms), where "activity cliffs," or drastic label shifts arising from, e.g., single-atom edits, frequently plague DD programs (Stumpfe et al., 2019; Aldeghi et al., 2022).

2.3 Siamese networks (positive-only non-CLR)

Non-CLR has been substantially developed for SSL/pre-training, in part to address the drawbacks of CLR above. SimSiam is an easily implemented non-CLR mechanism that is widely used in CV (Chen & He, 2020). As in CLR, augmented samples derived from parent inputs are treated as positive pairs, and models are trained to minimize the cosine distance between their embeddings. No negative pair constraints are imposed, satisfying our desired setting (see Section 1.1).

A trivial global solution to the non-CLR task is to map all embeddings to a single latent point. To avoid such collapse, in SimSiam loss gradients are backpropagated only for one augmented sample and are detached from the rest. The loss is then symmetrized by multiple backward passes rotating the detached samples. The authors in Chen & He (2020) claim that this `stopgrad` is sufficient to prevent collapse. They demonstrate that the embedding variance along the feature dimension remains roughly stable throughout training, indicating models are not learning to predict identical embeddings for all inputs. However, Li et al. (2022) recently showed that *partial dimensional collapse* (PDC) can occur despite stable overall variance (see Section 3).

2.4 Pre-training by structure denoising

Node denoising ("noisy nodes") has recently been demonstrated as an effective pre-training task for modeling 3D graphs (Godwin et al., 2021; Zaidi et al., 2022). Input coordinates are augmented with Gaussian noise, and models are trained to predict this noise, i.e., to recover the ground-truth structure.

While promising, this approach has two limiting pre-conditions: 1. Access to ground-truth (QM) conformations to denoise to and from; and 2. A suitably large corpus of QM structures of a relevant chemical space for pre-training. These conditions are quite difficult to satisfy for active DD programs, where structure optimization, e.g., by density functional theory (DFT) (Kohn & Sham, 1965), is prohibitively expensive, even for small labeled datasets.

That said, including denoising as an auxiliary objective during supervised fine-tuning was shown to be beneficial, from which we take inspiration. At time of writing, we are unaware of the use of non-CLR tasks for CE-MolML, whether un-, self-, or label-supervised.

3 APPROACH AND METHODS

A schematic of our approach is found in figure 1, and its components are described in detail in the following sections. All data and code are made available at [link to be activated on acceptance for publication].

3.1 CONFORMER PREPARATION AND AUGMENTATION

Following Axelrod & Gomez-Bombarelli (2020), we prepare CEs of modest size ($C_m \leq 10$) for each molecule m in a dataset with the inexpensive Experimental Torsion Knowledge Distance Geometry (ETKDG) method (Wang et al., 2020).

We note that conformer selection for MolML remains an outstanding challenge (Axelrod & Gomez-Bombarelli, 2020; Zaidi et al., 2022; Ganea et al., 2021b). In attempt to isolate the effects of the non-CLR mechanism from a dependence on starting conformers, we randomly sample a single conformer $c \in C_m$ for each molecule at each pass through models. This has the added benefit of computational efficiency over modeling a full CE in each pass. We report aggregated results over repeated training runs to marginalize over the effects of conformer selection.

Following Godwin et al. (2021); Zaidi et al. (2022), we augment conformers c by sampling Gaussian noise $\mathcal{N}(0,1) \in \mathbb{R}^{n\times 3}$ around normalized node positions $v_i^c \in V_c$ to give c_a. The noise scale is controlled by a temperature hyperparameter τ (i.e., noise is sampled from $\mathcal{N}(0,\tau)$). A cutoff radius of 4.0 Å was used for constructing radial graphs, to which self-loops were added.

3.2 SIAMESE E3NNS

E3NNs (Thomas et al., 2018) were utilized as a base architecture to demonstrate our approach, though it is architecture-agnostic. The overall loss for optimization combines a target-prediction term ($\mathcal{L}_y$), a positive-only cosine embedding term ($\mathcal{L}_s$), and an l_2-regularization penalty ($\mathcal{L}_r$) as follows:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{C_m}\sum_{c=1}^{C_m}\left[\frac{1}{A}\sum_{a=1}^{A}\left(\lambda_y\mathcal{L}_y(\hat{y}_a^c, y_i) + \lambda_r\mathcal{L}_r(z_a^c)\right) + \frac{1}{A-1}\sum_{a=2}^{A}\lambda_s\mathcal{L}_s(z_1^c, z_a^c)\right]\right), \quad (1)$$

$$\mathcal{L}_s(z_1^c, z_{a\neq 1}^c) = -\frac{1}{2}\left(\left[\frac{z_1^c}{\|z_1^c\|_2}\cdot\frac{\xi(z_a^c)}{\|z_a^c\|_2}\right] + \left[\frac{z_a^c}{\|z_a^c\|_2}\cdot\frac{\xi(z_1^c)}{\|z_1^c\|_2}\right]\right);\ \mathcal{L}_r(z_a^c) = \|z_a^c\|_2\ , \quad (2)$$

where N is the dataset size, C_m is the number of conformers of molecule m modeled in each pass (1 herein), $\lambda_{y,r,s}$ are sub-task weights, A is the number of augmented samples of each conformer (including parent, 2 herein), z_1^c and z_a^c are the learned embeddings for the parent and augmented conformers, respectively, y_i and $\hat{y}_a^c$ are the ground-truth and predicted labels for molecule i and augmented conformer c_a, respectively, and $\xi(\cdot)$ represents the `stopgrad` operation.

We utilize a moderate-capacity, feed-forward E3NN, the trunk of which consists of 4 convolutional interaction blocks as defined in Geiger & Smidt (2022). This is followed by global mean pooling over node features and a readout multi-layer perceptron (MLP) of length 1. `layernorm` is applied to each interaction block. The penultimate parent and augmented representations $\hat{z}_1^c$ and $\hat{z}_a^c$ are projected by another MLP h to give z_1^c and z_a^c of dimension d.

The Siamese task $\mathcal{L}_s$ in Equation 2 translates to predicting z_a^c from $\hat{z}_1^c$ with h, and vice versa. In this mechanism, each backward pass only propagates through one sample (z_a^c or z_1^c), with gradients detached from the other. This is symmetrized such that each augmentation $c_a \in A$ receives a backward pass.

For the target task $\mathcal{L}_y$ in Equation 1, probabilistic inference was utilized to account for aleatoric uncertainty in the datasets (Kendall & Gal, 2017). Models thus output parameterized distributions over logits, from which we sample before appropriate activation and loss calculation.

Each experiment assesses E3NNs over 5 random weight instantiations, and experiments are run in duplicate (10 total runs). For test-set evaluation, weights of each ensemble member were loaded from the epoch of their best validation-set performance for the applicable metric (*vide infra*).

3.3 DATASETS

It was hypothesized that our approach may be most useful for shape-based MLDD tasks such as binding/affinity prediction. For initial studies, one dataset both in and out of this task type were selected from the Therapeutic Data Commons (TDC) (Huang et al., 2022). These included one classification (3.3.1) and one regression (3.3.2) dataset, for which ablation studies over τ, d, and $\lambda_{s,r}$ were performed.

Dataset splits were used as given in the TDC API (link). For consistency, each dataset was restricted to the set of molecules containing only atoms of types `[C, N, O, F, S, Cl, Br, I]`. Compounds that failed conformer generation were also removed. Initial conformer diversity was imposed by RMSD ≥ 0.1 Å. All conformers were optimized with the Universal Force Field (UFF) using default parameters in RDKit (Landrum, 2022).[1]

Brief dataset details are given below, and we direct the reader to the TDC web page (link) for full descriptions and original references.

3.3.1 PGP_BROCATELLI (**PGP**)

This dataset comprises 1,212 molecules with affinity labels for binding to P-glycoprotein receptors. The task is binary classification; $\mathcal{L}_y$ is binary cross entropy (BCE), and the evaluated metric is ROCAUC.

3.3.2 CLEARANCE_HEPATOCYTE_AZ (**CLEAR**)

This dataset contains 1,020 molecules with continuous labels for degree of hepatocyte clearance. The task is regression; $\mathcal{L}_y$ is mean squared error (MSE), and the evaluated metric is Spearman ρ.

3.4 ANALYSIS AND METRICS

We hypothesized that the auxiliary task in 1 could result in more generalizable E3NNs in small-data regimes. This was analyzed by quantification of *local manifold smoothness* (MS, η_f) (Ng et al., 2022) as a proxy for model f's robustness to conformer noise in unseen data (see Section 1.1).

In Ng et al. (2022), $\eta(f, c)$ is defined as the total percentage of augmented samples c_a of input c that are assigned the mode predicted label in the set (for binary tasks). We generalize this to the probabilistic and regression settings by computing the KL divergence between predicted posterior distributions $(\hat{\mu}_1^c, \hat{\sigma}_1^c)$ and $(\hat{\mu}_a^c, \hat{\sigma}_a^c)$ for parent and augmented samples, respectively:

$$\eta_f = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{C_m(A-1)}\sum_{i=1}^{C_m}\sum_{a=2}^{A} 1 - \left[\log(\hat{\sigma}_a^c) - \log(\hat{\sigma}_1^c) + \frac{(\hat{\sigma}_1^c)^2 + (\hat{\mu}_a^c - \hat{\mu}_1^c)^2}{2(\hat{\sigma}_a^c)^2} - 0.5\right]. \quad (3)$$

While difficult to assess on absolute scale, we utilize η_f to compare between models with different weightings of the subtasks and noise hyperparameters.

Following Chen & He (2020), we detect trivial collapse by quantifying the variance in embeddings (σ_z^2) along the feature axis (see section 2.3). Finally, we follow Li et al. (2022) to detect *partial dimensional collapse*. We quantify the cumulative explained variance (CEV, Γ) of the singular values γ computed via principal component analysis of embedding sets. The CEV up to rank-sorted γ_j (Γ_j) and the area under the full CEV curve ($\mathbf{\Gamma}$) are defined as:

$$\Gamma_j = \frac{\sum_{i=1}^{j}\gamma_i}{\sum_{k=1}^{d}\gamma_k}\,;\ \mathbf{\Gamma} = \frac{1}{d}\sum_{j=1}^{d}\Gamma_j\,, \quad (4)$$

[1] It is possible that initial conformers converge to a small number of locally optimal geometries. We allow this under the assumption that this final set may be most reflective of that observed in a biological setting. I.e., since a random conformer is sampled in each epoch, models are roughly exposed to a Boltzmann-weighted distribution of conformations. Further study into *learned* conformer sampling will be presented in future work.

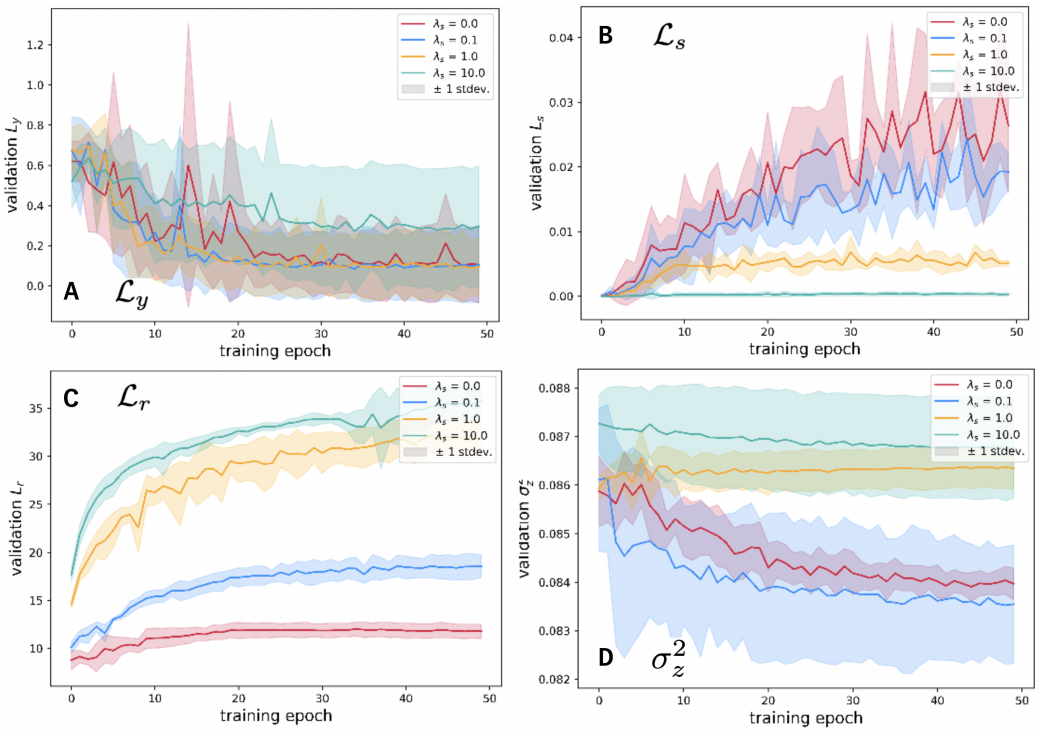


Figure 2: E3NN Pgp-binding classification training profiles at varying λ_s. A) target loss (BCE); B) Siamese loss (cos); C) regularization loss (l_2-norm); D) embedding feature variance. All curves show validation set results with $A = 1$, $d = 128$, $\tau = 0.1$, $\lambda_r = 0.0$.

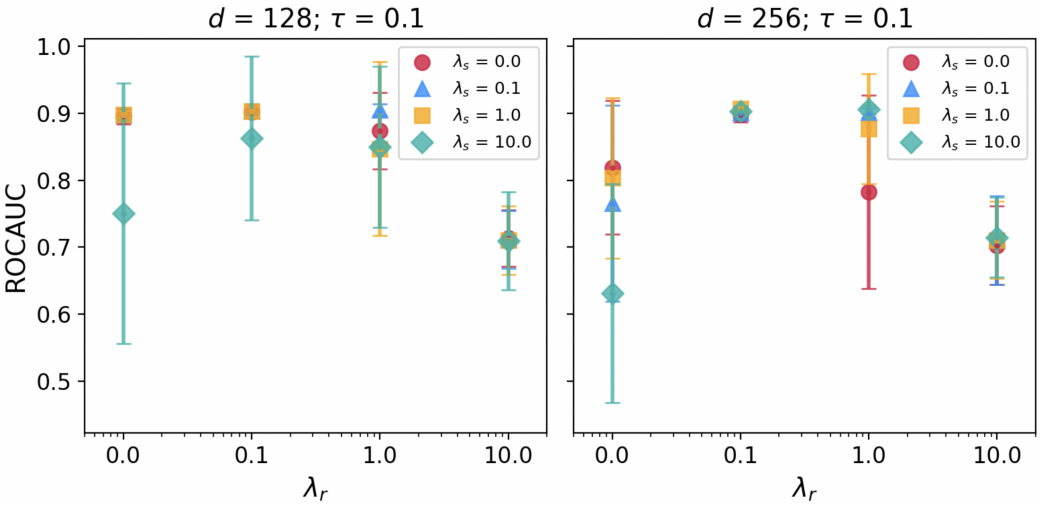


Figure 3: E3NN Pgp-binding classification test set ROCAUC across SupSiam parameters.

where d is the full embedding size. $\mathbf{\Gamma}$ ranges between [0.5, 1.0]; larger values correspond to more rapid coverage of the total CEV over fewer singular values, and thus indicate a larger degree of collapse. $\mathbf{\Gamma} = 0.5$ corresponds to zero PDC.

4 RESULTS AND DISCUSSION

4.1 TRAINING PROFILES

Figure 2 shows E3NN training profiles on the **Pgp** task at varying λ_s values without regularization ($\lambda_r = 0.0$). Target loss curves are largely consistent across λ_s values, (Figure 2A), though this is not the case in all settings (see Section A.3.1 for full results). It is worth noting that with standard supervision ($\lambda_s = \lambda_r = 0$), training curves are often highly erratic. Further, cosine embedding distance for augmented pairs actually *diverges* over the course of training for purely supervised models (Figure 2B, red). As anticipated, this divergence is mitigated using SupSiam, with an intuitive trend at increasing λ_s. At higher levels of λ_s, smooth training toward $\mathcal{L}_s = 0$ is observed.

Interestingly, the opposite trend is observed for $\mathcal{L}_r$ (Figure 2C). However, this only holds without regularization ($\lambda_r = 0$); with $\lambda_r > 0$, $\mathcal{L}_r$ smoothly converges in all cases (see Section A.3.1).

Surprisingly, embedding-feature variance decreases monotonically over training at low values of λ_s, and is largely flat at higher λ_s (Figure 2D). This effect is dependent on regularization. With $\lambda_r = 0$ as in Figure 2, the maximum $\lambda_s = 10$ actually maintains the highest feature variance throughout training. With $\lambda_r > 0$, however, σ_z^2 behaves as observed in Chen & He (2020) — σ_z^2 sharply decreases in early epochs before recovering to inital levels and plateauing (see Section A.3.1).

4.2 PERFORMANCE METRICS

Final test set performances are shown in Figure 3, with error bars representing $\pm$ 1 stdev. over repeat runs. In line with the observations above, test set ROCAUC is highest at intermediate levels of λ_s and λ_r, which holds across settings of d. At small τ, reasonable performance can be maintained even at maximum λ_s. At maximum λ_r, however, the target task is only minimally learned, regardless of all other settings (see Section A.3.1).

It is important to note that for this task, the maximum ROCAUC obtained (~ 0.91) is slightly below the literature benchmark ($\sim$ 0.95) (Huang et al., 2022). However, benchmark methods largely utilize tree-based learning with cheminformatic representations; modeling this dataset with E3NNs could be expected to be quite challenging. This is much in line with our motivation (see Sections 1, 1.1), and exceeding benchmark metrics is neither a goal nor expectation. Altogether, we find the combination of performance with stable, physically reasonable training profiles obtained with SupSiam (Section 4.1) to be compelling for use in production settings.

The ablation of λ_r and λ_s provides insight into their effects on latent properties. We expect that $\mathcal{L}_s$ is not simply serving to compactify latent space (like $\mathcal{L}_r$) due to the following observations:

1. Maximizing λ_r inhibits training of $\mathcal{L}_y$, regardless of λ_s (Figure 3). Conversely, maximizing λ_s results in viable training profiles at several settings of λ_r (Figures 2A, 3, Section A.3.1);
2. Ablating λ_s is uniformly deleterious to $\mathcal{L}_s$ across λ_r values. Conversely, λ_r has little to no effect on $\mathcal{L}_s$, regardless of λ_s (Figure 2B, Section A.3.1);
3. As expected, at fixed λ_s, $\mathcal{L}_r$ decreases with increasing λ_r (Section A.3.1). Conversely, at fixed λ_r, $\mathcal{L}_r$ actually tends to increase with increasing λ_s (Figure 2C).

These observations point to differing behaviors of Siamese learning and latent regularization. It appears that while $\mathcal{L}_r$ does compactify latent space (lower l_2), this does not necessarily push *related* embeddings to closer cosine distances. Conversely, while $\mathcal{L}_s$ does push related embeddings to closer cosine distances, it does so with an increased expansion of latent space (higher l_2). We thus expect a task-specific balance of λ_s and λ_r may lead to the most desirable model properties.

4.3 MANIFOLD SMOOTHNESS AND PARTIAL DIMENSIONAL COLLAPSE

Figure 4 (top) shows KDEs of per-molecule MS (Equation 3) across λ_s for the models discussed above. In many cases, SupSiam caused drastic reduction in posterior KL divergence between augmented samples, often by several log units. We note that the correlation of μ_f and λ_s is often obscured when $\mathcal{L}_r$ is heavily weighted (see Section A.3.4). In any case, we find MS analysis very informative.

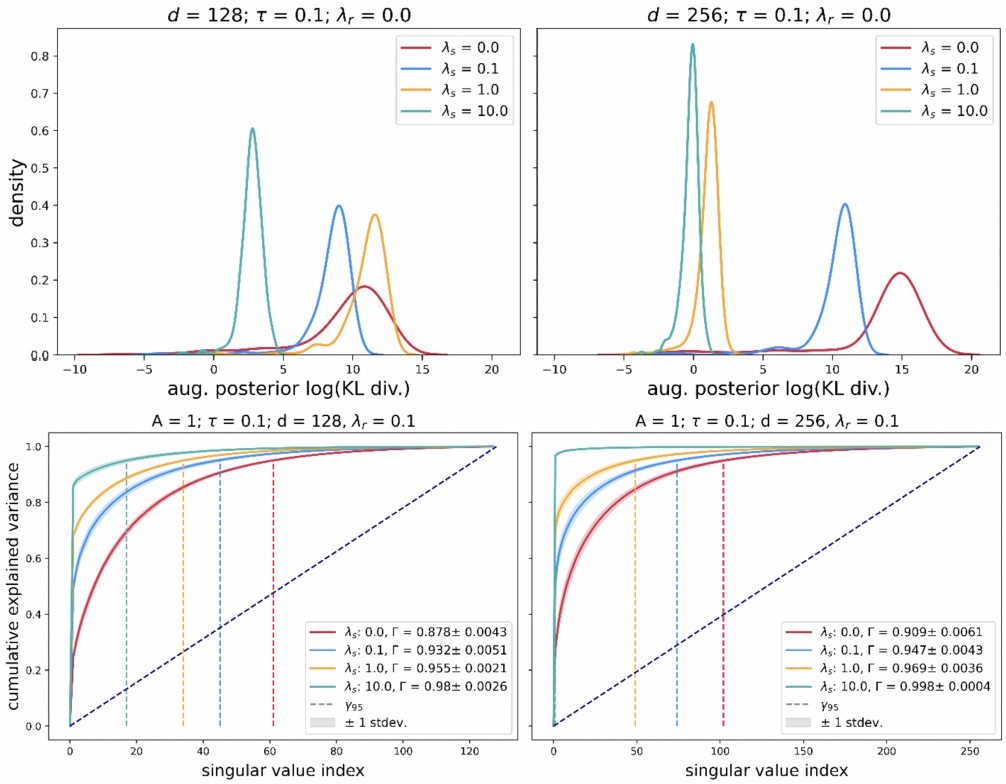


Figure 4: (Top) Pgp E3NN manifold smoothness (η_f) at varying λ_s. All models trained with $\tau = 0.1$, $A = 1$. η_f evaluated with $\tau = 0.1$, $A = 10$. (Bottom) Pgp E3NN partial dimensional collapse at varying λ_s. Vertical lines indicate singular value at which $\Gamma_j = 0.95$ (γ_j^{95}).

Figure 4 (bottom) shows CEV curves across λ_s. We do observe a positive (albeit intuitive) correlation between $\mathbf{\Gamma}$ and λ_s. That said, in many cases only minor increases in PDC were observed at low levels of λ_s vs $\lambda_s = 0.0$. It is striking to note the degree of PDC in the pure supervision setting ($\lambda_s = 0$). This may indicate that supervised E3NNs fit to relatively few input structure features, at least in small-data regimes. We find this alarming, but note that in most models (at $\lambda_s \leq 1$) 95% CEV (γ_j^{95}) is not reached until roughly the $1/3$ or $1/2$ embedding index (Figure 4, bottom).

Contrary to Li et al. (2022), we also observe a positive correlation between $\mathbf{\Gamma}$ and d. In Li et al. (2022), increasing model capacity (via d) was used explicitly to *reduce* PDC. This was rationalized as insufficient capacity resulting in information loss. It is possible that, at least in our data setting ($N \sim 10^{2-4}$), the opposite may be true, where an encoded-information limit is reached at relatively small d, and thus increasing capacity actually results in higher $\mathbf{\Gamma}$s. Increasing λ_r decreased the dependence on λ_s, and in some cases actually reduced PDC (Section A.3.4).

Li et al. (2022) propose continual learning as a mechanism to reduce PDC in Siamese networks. We leave such studies in our setting for future works. For now, we assess that low λ_s settings lead to acceptable $\Delta\mathbf{\Gamma}$ from pure supervision, where collapse appears to be an outstanding issue. We expect that PDC may be a useful tool to analyze learned information in MolML more broadly.

5 Conclusion and Outlook

*Sup*ervised *Siam*ese networks (*SupSiam*) were presented as an approach to train E3NN models on molecular geometries. We observe that the Siamese auxiliary task results in desirable latent properties while maintaining good performance in target prediction. Additionally, in many cases SupSiam was shown to increase manifold smoothness (i.e., decrease model sensitivity) to noise in input structures, a critical challenge in MolML. Lastly, embedding collapse was observed to increase only slightly

over pure supervision. That said, the partial dimensional collapse of E3NNs was observed to be severe in most cases, which we find alarming. Future works will seek to tackle this issue.

Other topics for future work include expansion to other equivariant architectures (Batatia et al., 2022) and augmentation mechanisms, e.g., noising bond distances and angles instead of atomic positions. Further, we are actively exploring non-CLR pre-training mechanisms, which will be reported separately. Finally, optimal task weighting and architecture settings ($\lambda_{y,s,r}, d, \tau$) are likely to be task-specific. We expect an exciting direction for future work will be in, e.g., Bayesian optimization of these parameters.

Overall, we anticipate the findings herein to aid in the training of more robust 3D GNNs on molecular conformers. We expect the approach to be applicable in many 3D modeling tasks for small and large molecules, e.g., proteins. SupSiam may find particular utility in settings where sensitivity to minor 2D structure changes, but insensitivity to minor 3D structure changes, are desirable. We are hopeful it will aid in overcoming challenges such as activity cliffs (Stumpfe et al., 2019) and rough SAR landscapes (Aldeghi et al., 2022), and will lead to more reliable modeling in MLDD.

REFERENCES

Matteo Aldeghi, David E. Graff, Nathan Frey, Joseph A. Morrone, Edward O. Pyzer-Knapp, Kirk E. Jordan, and Connor W. Coley. Roughness of Molecular Property Landscapes and Its Impact on Modellability. *Journal of Chemical Information and Modeling*, 62(19):4660–4671, 2022. ISSN 1549-9596. doi: 10.1021/acs.jcim.2c00903.

Simon Axelrod and Rafael Gomez-Bombarelli. Molecular machine learning with conformer ensembles. *arXiv*, 2020. doi: 10.48550/arxiv.2012.08452.

Ilyes Batatia, Dávid Péter Kovács, Gregor N C Simm, Christoph Ortner, and Gábor Csányi. MACE: Higher Order Equivariant Message Passing Neural Networks for Fast and Accurate Force Fields. *arXiv*, 2022. doi: 10.48550/arxiv.2206.07697.

Xinlei Chen and Kaiming He. Exploring Simple Siamese Representation Learning. *arXiv*, 2020. doi: 10.48550/arxiv.2011.10566.

Christian Devereux, Justin S. Smith, Kate K. Davis, Kipton Barros, Roman Zubatyuk, Olexandr Isayev, and Adrian E. Roitberg. Extending the Applicability of the ANI Deep Learning Molecular Potential to Sulfur and Halogens. *Journal of Chemical Theory and Computation*, 16(7):4192–4202, 2020. ISSN 1549-9618. doi: 10.1021/acs.jctc.0c00121.

Octavian-Eugen Ganea, Xinyuan Huang, Charlotte Bunne, Yatao Bian, Regina Barzilay, Tommi Jaakkola, and Andreas Krause. Independent SE(3)-Equivariant Models for End-to-End Rigid Protein Docking. *arXiv*, 2021a. doi: 10.48550/arxiv.2111.07786.

Octavian-Eugen Ganea, Lagnajit Pattanaik, Connor W Coley, Regina Barzilay, Klavs F Jensen, William H Green, and Tommi S Jaakkola. GeoMol: Torsional Geometric Generation of Molecular 3D Conformer Ensembles. *arXiv*, 2021b. doi: 10.48550/arxiv.2106.07802.

Mario Geiger and Tess Smidt. e3nn: Euclidean Neural Networks. *arXiv*, 2022. doi: 10.48550/arxiv.2207.09453.

Jonathan Godwin, Michael Schaarschmidt, Alexander Gaunt, Alvaro Sanchez-Gonzalez, Yulia Rubanova, Petar Veličković, James Kirkpatrick, and Peter Battaglia. Simple GNN Regularisation for 3D Molecular Property Prediction & Beyond. *arXiv*, 2021. doi: 10.48550/arxiv.2106.07971.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised Learning. *arXiv*, 2020. doi: 10.48550/arxiv.2006.07733.

Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor W. Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Artificial intelligence foundation for therapeutic science. *Nature Chemical Biology*, 18(10):1033–1036, 2022. ISSN 1552-4450. doi: 10.1038/s41589-022-01131-2.

Clemens Isert, Kenneth Atz, José Jiménez-Luna, and Gisbert Schneider. QMugs, quantum mechanical properties of drug-like molecules. *Scientific Data*, 9(1):273, 2022. doi: 10.1038/s41597-022-01390-7.

Derek Jones, Hyojin Kim, Xiaohua Zhang, Adam Zemla, Garrett Stevenson, W. F. Drew Bennett, Daniel Kirshner, Sergio E. Wong, Felice C. Lightstone, and Jonathan E. Allen. Improved Protein–Ligand Binding Affinity Prediction with Structure-Based Deep Fusion Inference. *Journal of Chemical Information and Modeling*, 61(4):1583–1592, 2021. ISSN 1549-9596. doi: 10.1021/acs.jcim.0c01306.

Alex Kendall and Yarin Gal. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? *arXiv*, 2017. doi: 10.48550/arxiv.1703.04977.

W. Kohn and L. J. Sham. Self-Consistent Equations Including Exchange and Correlation Effects. *Physical Review*, 140(4A):A1133–A1138, 1965. ISSN 0031-899X. doi: 10.1103/physrev.140.a1133.

David C. Kombo, Kartik Tallapragada, Rachit Jain, Joseph Chewning, Anatoly A. Mazurov, Jason D. Speake, Terry A. Hauser, and Steve Toler. 3D Molecular Descriptors Important for Clinical Success. *Journal of Chemical Information and Modeling*, 53(2):327–342, 2013. ISSN 1549-9596. doi: 10.1021/ci300445e.

Greg Landrum. RDKit: Open-source cheminformatics., 3 2022. URL `https://www.rdkit.org`.

Phuc H. Le-Khac, Graham Healy, and Alan F. Smeaton. Contrastive Representation Learning: A Framework and Review. *IEEE Access*, 8:193907–193934, 2020. ISSN 2169-3536. doi: 10.1109/access.2020.3031549.

Alexander C Li, Alexei A Efros, and Deepak Pathak. Understanding Collapse in Non-Contrastive Siamese Representation Learning. *arXiv*, 2022. doi: 10.48550/arxiv.2209.15007.

Yi-Lun Liao and Tess Smidt. Equiformer: Equivariant Graph Attention Transformer for 3D Atomistic Graphs. *arXiv*, 2022. doi: 10.48550/arxiv.2206.11990.

Nathan Ng, Neha Hulkund, Kyunghyun Cho, and Marzyeh Ghassemi. Predicting Out-of-Domain Generalization with Local Manifold Smoothness. *arXiv*, 2022. doi: 10.48550/arxiv.2207.02093.

Joshua A Rackers, Lucas Tecot, Mario Geiger, and Tess E Smidt. Cracking the Quantum Scaling Limit with Machine Learned Electron Densities. *arXiv*, 2022. doi: 10.48550/arxiv.2201.03726.

Wolfgang H. B. Sauer and Matthias K. Schwarz. Molecular Shape Diversity of Combinatorial Libraries: A Prerequisite for Broad Bioactivity †. *Journal of Chemical Information and Computer Sciences*, 43(3):987–1003, 2003. ISSN 0095-2338. doi: 10.1021/ci025599w.

Dagmar Stumpfe, Huabin Hu, and Juergen Bajorath. Evolving Concept of Activity Cliffs. *ACS Omega*, 4(11):14360–14368, 2019. ISSN 2470-1343. doi: 10.1021/acsomega.9b02221.

Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation- and translation-equivariant neural networks for 3D point clouds. *arXiv*, 2018. doi: 10.48550/arxiv.1802.08219.

Shuzhe Wang, Jagna Witek, Gregory A. Landrum, and Sereina Riniker. Improving Conformer Generation for Small Rings and Macrocycles Based on Distance Geometry and Experimental Torsional-Angle Preferences. *Journal of Chemical Information and Modeling*, 60(4):2044–2058, 2020. ISSN 1549-9596. doi: 10.1021/acs.jcim.0c00025.

Yuyang Wang, Jianren Wang, Zhonglin Cao, and Amir Barati Farimani. Molecular Contrastive Learning of Representations via Graph Neural Networks. *arXiv*, 2021. doi: 10.48550/arxiv.2102.10056.

Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. MoleculeNet: a benchmark for molecular machine learning. *Chemical Science*, 9(2):513–530, 2017. ISSN 2041-6520. doi: 10.1039/c7sc02664a.

Sheheryar Zaidi, Michael Schaarschmidt, James Martens, Hyunjik Kim, Yee Whye Teh, Alvaro Sanchez-Gonzalez, Peter Battaglia, Razvan Pascanu, and Jonathan Godwin. Pre-training via Denoising for Molecular Property Prediction. *arXiv*, 2022. doi: 10.48550/arxiv.2206.00133.

Yajun Zheng, Colin M. Tice, and Suresh B. Singh. Conformational control in structure-based drug design. *Bioorganic & Medicinal Chemistry Letters*, 27(13):2825–2837, 2017. ISSN 0960-894X. doi: 10.1016/j.bmcl.2017.04.079.

A Appendix

A.1 Data processing

A.1.1 Target distributions

See Section 3.3 for high-level data-processing details. For the Clear task, targets were scaled by log_10 to give a roughly Gaussian prior. For all datasets, oversampling was utilized, where model batches were sampled from the training set weighted proportionally to the inverse of their histogram-bin density as follows:

$$w_i = 1 - p_y(y_i), \tag{5}$$

where p_y represents the prior density distribution.

A.1.2 Conformer ensemble generation

All processing functions are made available in the associated code repository at [link to be activated on acceptance for publication]. The conformer generation pipeline follows that in Axelrod & Gomez-Bombarelli (2020) closely. A model pipeline is as follows:

Algorithm 1 Conformer ensemble generation

Input: Dataset of N observations $\mathcal{D}_n$, with input space $\mathcal{X}$ as molecular SMILES strings, conformer ensemble size c, boolean optimize o, boolean align a, `rdkit.Chem.AllChem` package, `rdkit.Chem.rdMolAlign` package
Output: Dataset of N observations $\mathcal{D}_n$, with input space atomic positions $\mathcal{X} \in \mathbb{R}^3$.
for $\{n = 1, \ldots, N\}$ **do**
    Construct `AllChem.Mol` object $\mathcal{M}_n$
    $\mathcal{M}_n \leftarrow$ `AllChem.AddHs(AllChem.MolFromSmiles(`$\mathcal{X}_n$`))`
    Embed ensemble of 3D conformers with ETKDG
    $\mathcal{M}_n^C \leftarrow$ `AllChem.EmbedMultipleConfs(`$\mathcal{M}_n$`, numConfs=`c`)`
    if o **then**
        Optimize molecular conformers with force field
        $\mathcal{M}_n^C \leftarrow$ `AllChem.UFFOptimizeMoleculeConfs(`$\mathcal{M}_n^C$`, numConfs=`c`)`
    end if
    Remove hydrogen atoms
    $\mathcal{M}_n \leftarrow$ `AllChem.RemoveHs(`$\mathcal{M}_n$`)`
    if a **then**
        Align molecular conformers
        $\mathcal{M}_n^C \leftarrow$ `rdMolAlign.AlignMolConformers(`$\mathcal{M}_n^C$`)`
    end if
end for

A.2 Model architecture

A.2.1 E3NNs

See Section 3.2 for high-level architecture details. Full hyperparameters are included below for a network with hidden dimension $d = 128$:

Table 1: E3NN hyperparameter settings

Name	Setting	Description
irreps_in	128x0e	input-layer irreducible representations
irreps_hidden	128x0e+128x1o+x2e	hidden-layer irreducible representations
irreps_out	128x0e	output-layer irreducible representations
l_max_sh	1	maximum geometric tensor spherical-harmonic level
num_hidden_layers	4	number of E3NN convolution layers
rc	4.0Å	neighborhood-edge radial cutoff distance
irreps_edge	128x0e	edge-layer irreducible representations
radial_num_basis	16	number of basis functions for radial NN
radial_num_hidden	16	radial NN hidden dimension
radial_num_layers	2	radial NN depth
add_self_loops	True	include self-edges in radial graphs

Intermediate layers are treated with ShiftedSoftplus activation and `LayerNorm`.

E3NNs output feature vectors for each node, comprising their flattened and concatenated geometric tensors. These embeddings are pooled by `global_mean_pool` to give single vector representations for each molecule, to which a final linear readout layer is applied.

A.2.2 Siamese projection MLP

For the Siamese task, E3NN readout representations of dimension d are input to a multilayer perceptron (MLP) of depth 2 and dimensions $\{d \times 2, d\}$. Both layers are followed by ShiftedSoftplus activation, and the intermediate representations are treated with `LayerNorm` and dropout of probability 0.2.

A.2.3 Probabilistic MLP

The probabilistic predictive model is a split-head MLP with two modules, each of depth 3 and output dimensions $\{d \times 2, d, 1\}$. Intermediate layers are followed by ShiftedSoftplus activation, `LayerNorm`, and dropout with probability 0.2. The mean (μ) module is unactivated, outputting raw logit values. The variance (σ) module outputs are activated by Softplus to give predicted posterior distributions (μ, σ). Following Kendall & Gal (2017), m samples are drawn from these distributions, and the resulting logits are activated by `sigmoid` for classification tasks and `Tanhshrink` for regression tasks. In the case of regression, the resulting predictions are scaled using the prior parameters (μ_t, σ_t) computed on the training set. An aggregate loss is computed over the sampled predictions, binary cross entropy (BCE) for classification and mean squared error (MSE) for regression.

A.3 Full results

A full grid search study was run over the following hyperparameter ranges:

Table 2: SupSiam hyperparameter screen

Name	Values	Description
n	10	number of model runs
e	50	number of training epochs
τ	[0.1, 1]	node noise multiplier
d	[128, 256]	hidden dimension
λ_s	[0.0, 0.1, 1.0, 10.0]	Siamese loss weight
λ_r	[0.0, 0.1, 1.0, 10.0]	l_2 loss weight

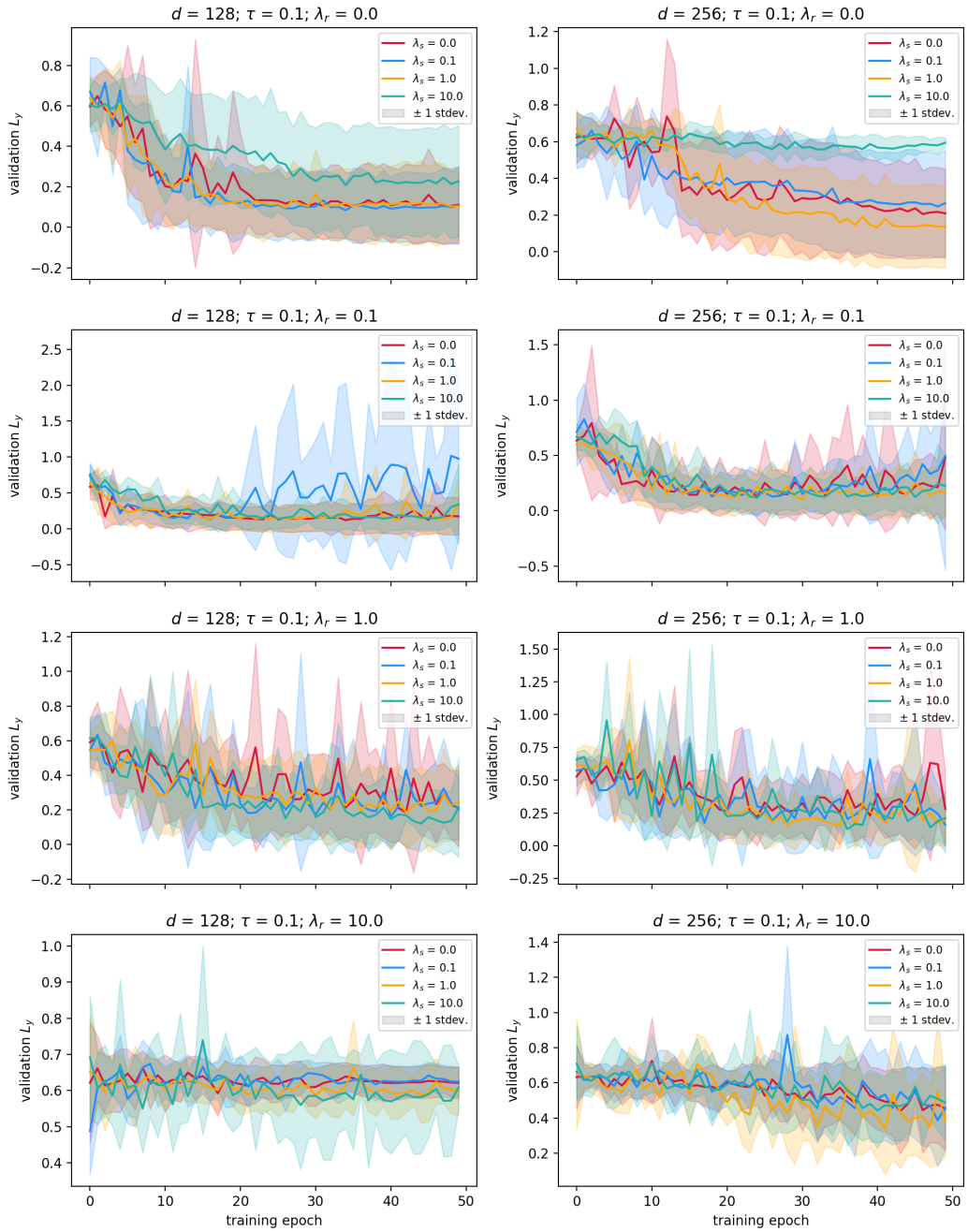
A.3.1 TRAINING PROFILES

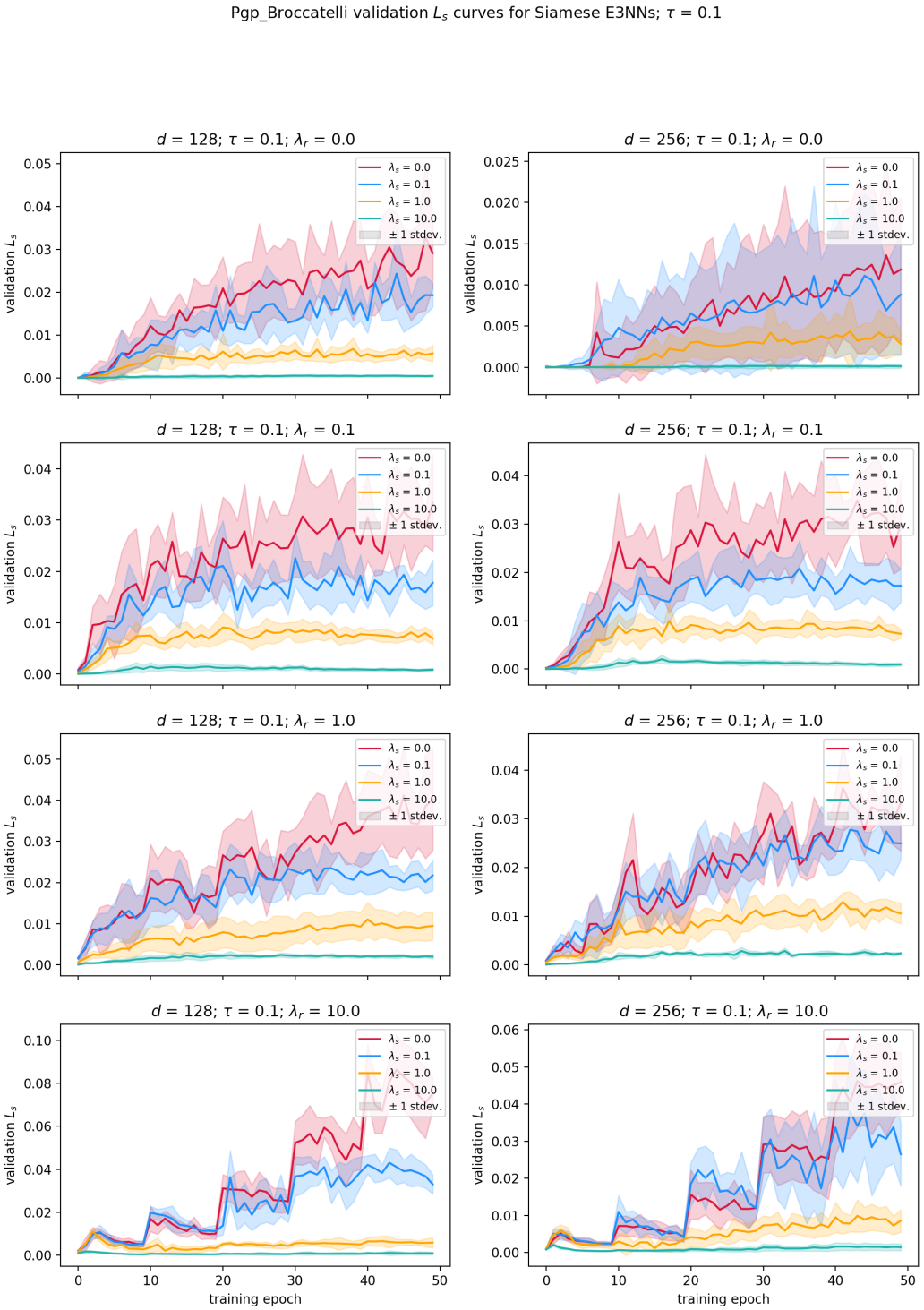
Model performance and properties were tracked throughout training and included $\mathcal{L}_y$, $\mathcal{L}_s$, $\mathcal{L}_r$, and σ_z^2 (see Equations 1, 2). `roc_auc_score`s were additionally tracked for classification tasks, and Spearman ρ for regression.

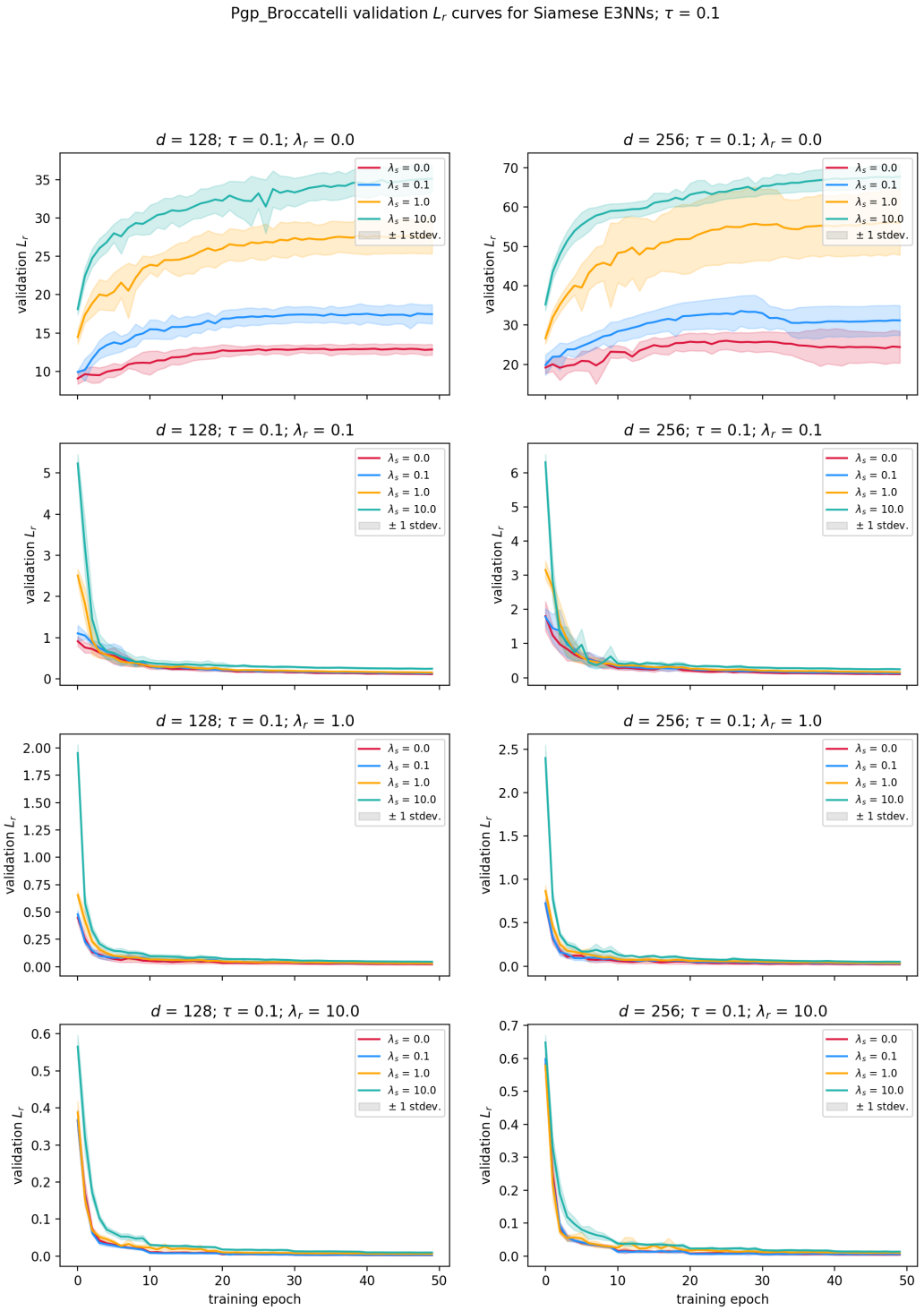
Full results are provided below. Each figure contains 1 metric above, from 1 task, and at 1 value of τ over the course of training. Plots are arranged with increasing d along the horizontal axis, and increasing λ_r along the vertical axis. Scatter plots like Figure 3 are also included, where all models were loaded from the epoch of their best validation set metric performance (`roc_auc_score` or Spearman ρ). These are organized with increasing d along the horizontal, increasing τ along the vertical, and increasing λ_r on the subplot x-axis.

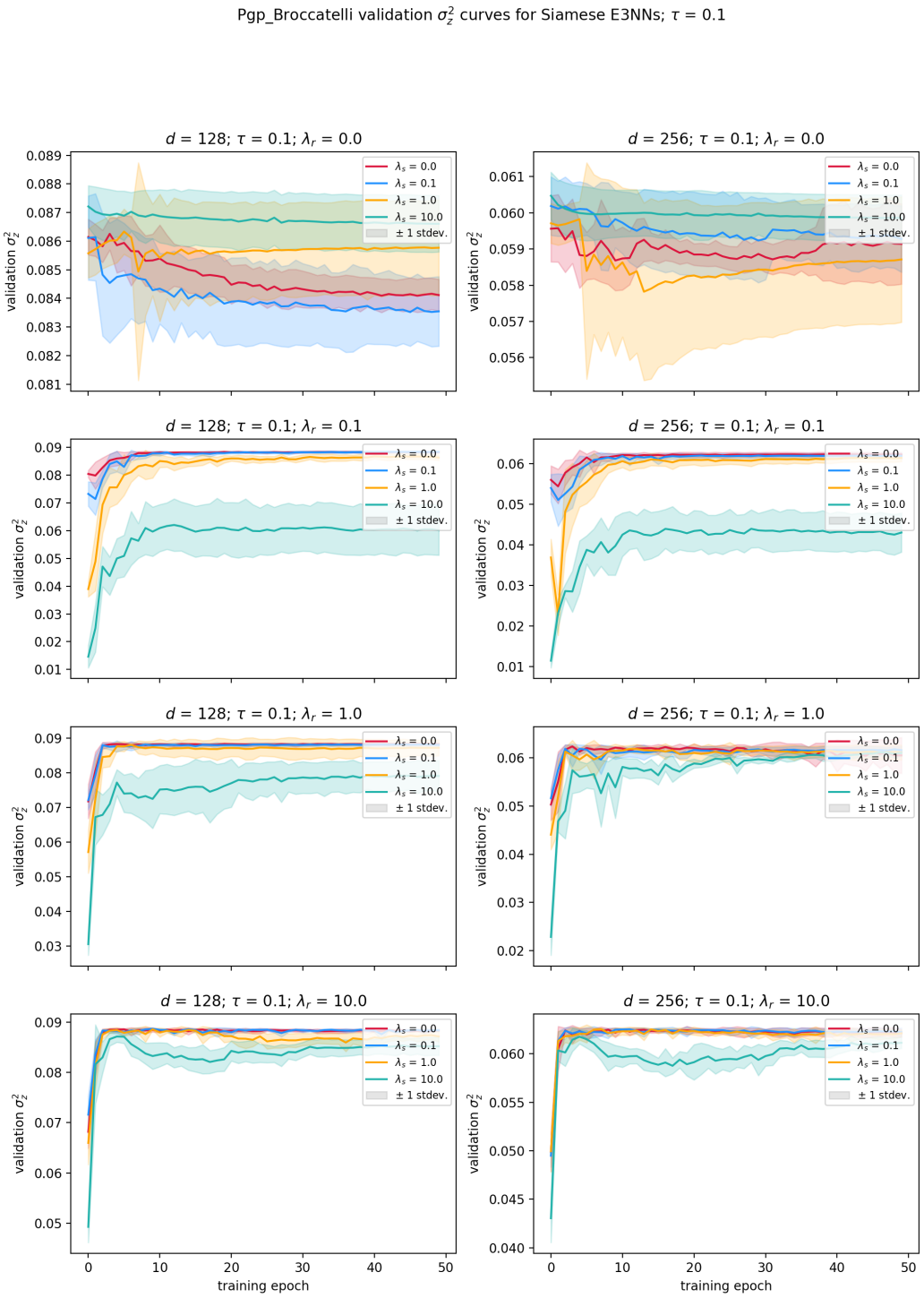
Note that for the **Pgp** task, `roc_auc_score`s in training figures were calculated with predictions binarized at $\hat{y} \geq 0.99$. For the scatter plots, however, full ROC curves were plotted over a range of 100 binarization thresholds evenly spaced from [0.0, 1.0]. Areas under these curves were then directly computed for each repeat model, giving the results shown in the plots.

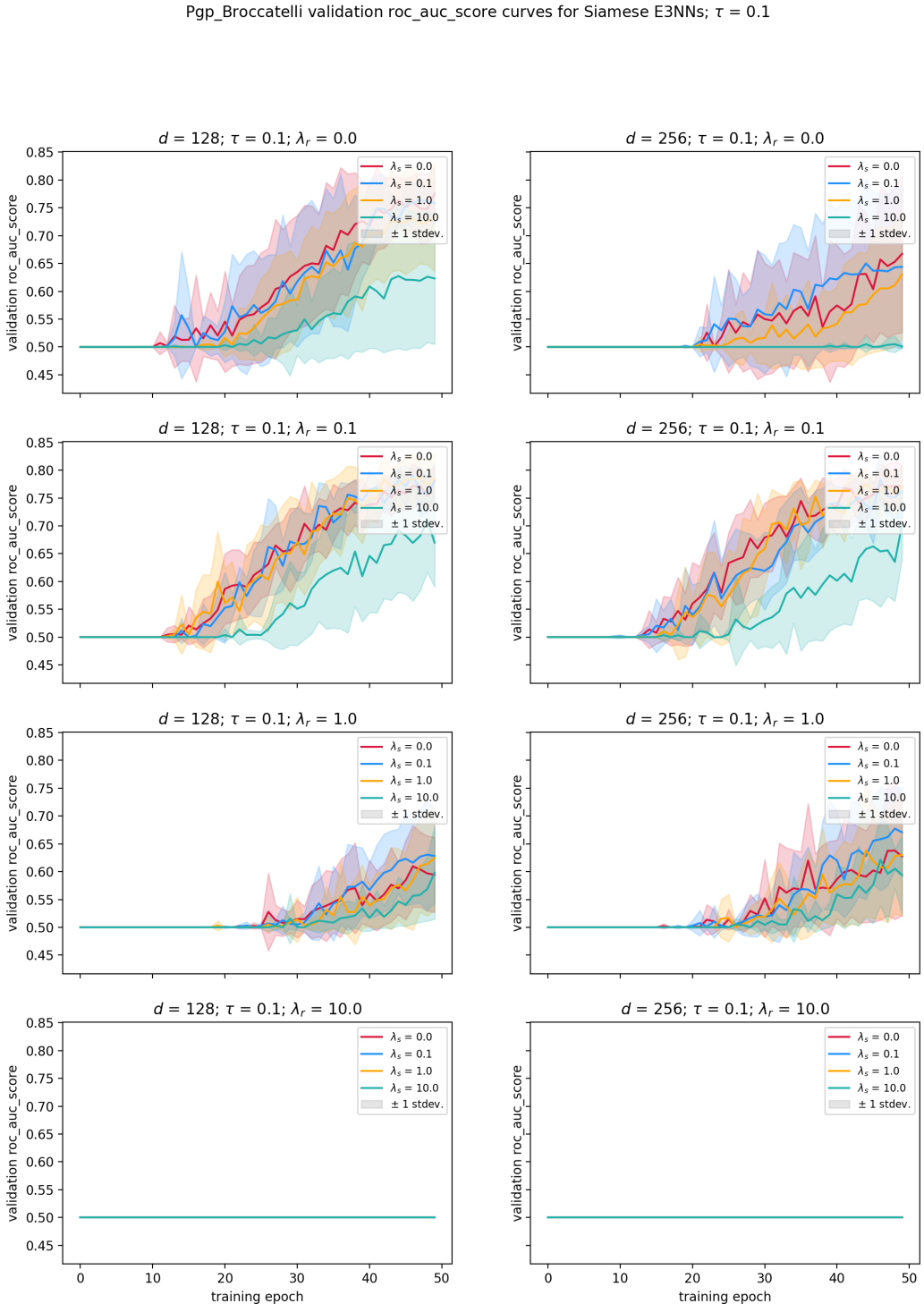
A.3.2 PGP

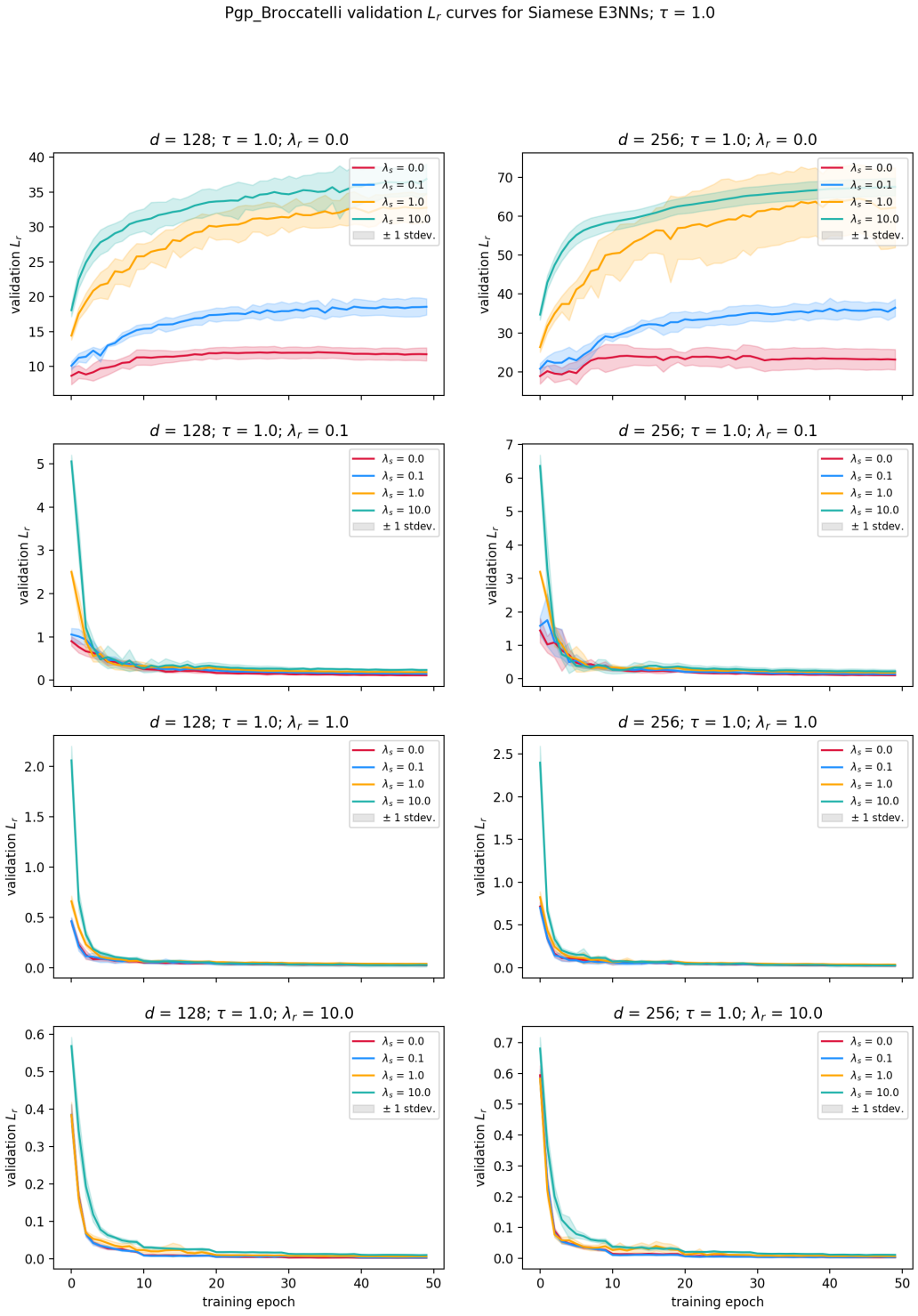

Pgp_Broccatelli validation L_y curves for Siamese E3NNs; τ = 0.1
d = 128; τ = 0.1; λ_r = 0.0
d = 256; τ = 0.1; λ_r = 0.0
d = 128; τ = 0.1; λ_r = 0.1
d = 256; τ = 0.1; λ_r = 0.1
d = 128; τ = 0.1; λ_r = 1.0
d = 256; τ = 0.1; λ_r = 1.0
d = 128; τ = 0.1; λ_r = 10.0
d = 256; τ = 0.1; λ_r = 10.0
λ_s = 0.0
λ_s = 0.1
λ_s = 1.0
λ_s = 10.0
± 1 stdev.
validation L_y
training epoch

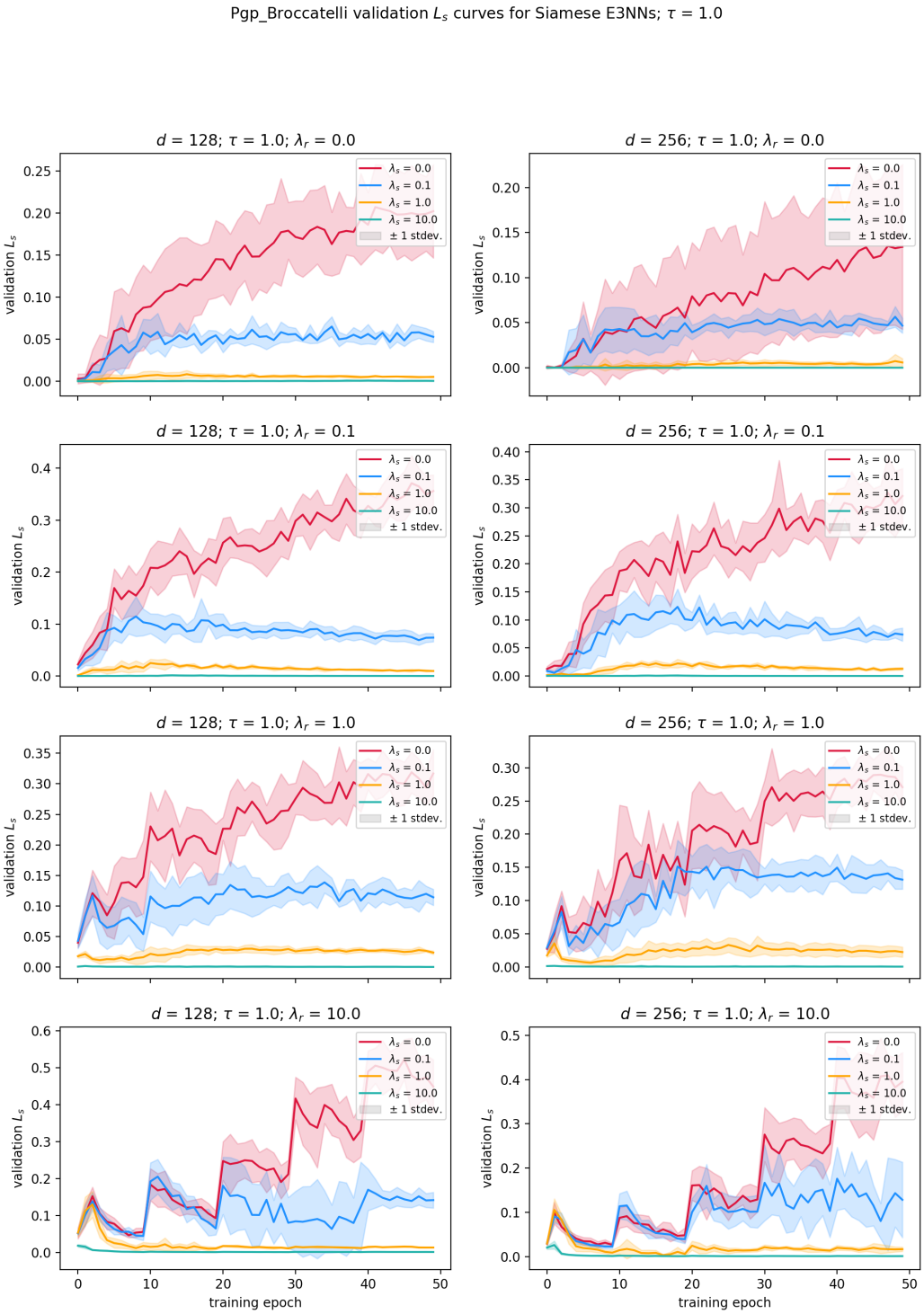

Pgp_Broccatelli validation L_s curves for Siamese E3NNs; τ = 0.1
d = 128; τ = 0.1; λ_r = 0.0
d = 256; τ = 0.1; λ_r = 0.0
d = 128; τ = 0.1; λ_r = 0.1
d = 256; τ = 0.1; λ_r = 0.1
d = 128; τ = 0.1; λ_r = 1.0
d = 256; τ = 0.1; λ_r = 1.0
d = 128; τ = 0.1; λ_r = 10.0
d = 256; τ = 0.1; λ_r = 10.0
validation L_s
training epoch
λ_s = 0.0
λ_s = 0.1
λ_s = 1.0
λ_s = 10.0
± 1 stdev.

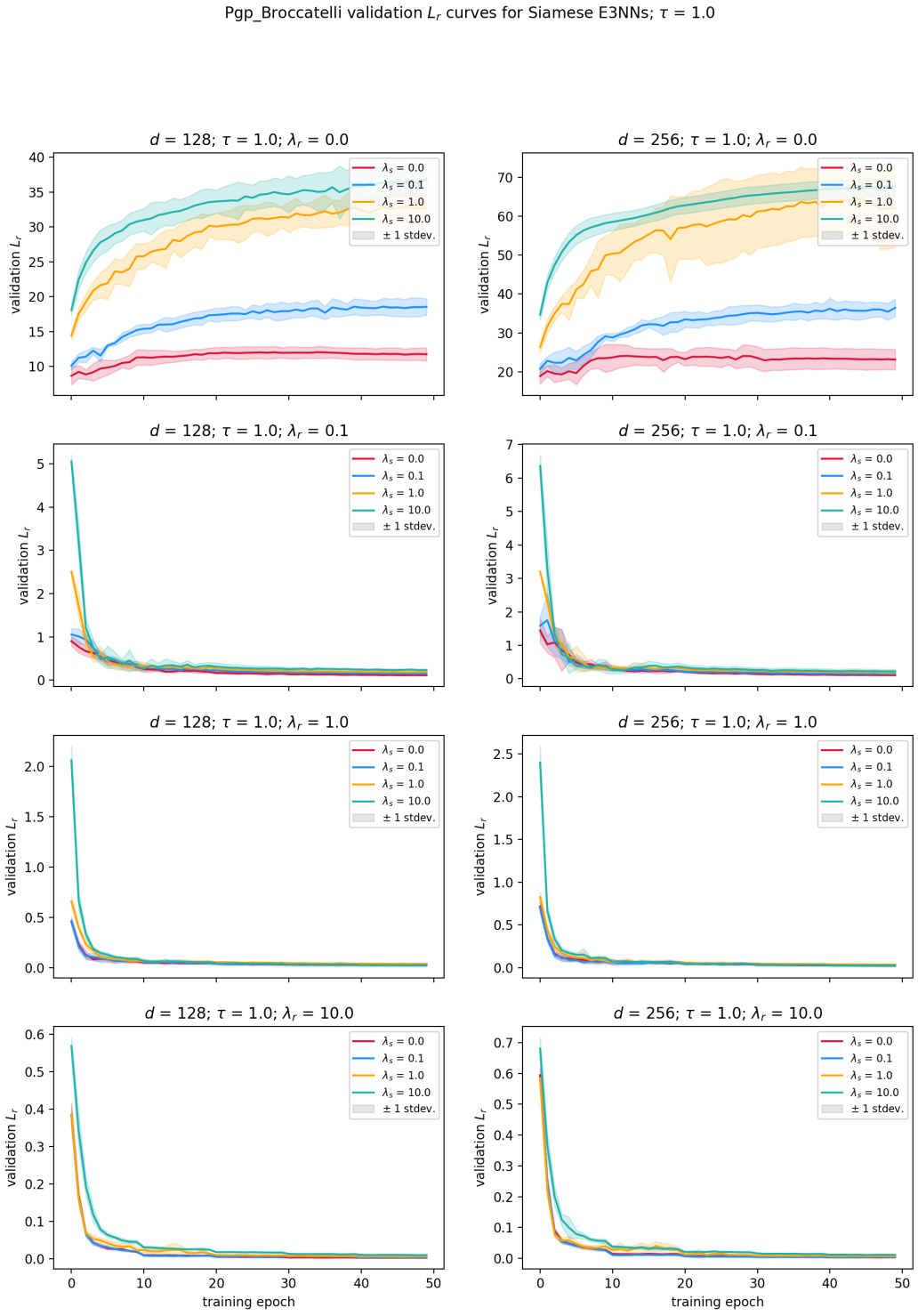

Pgp_Broccatelli validation L_r curves for Siamese E3NNs; τ = 0.1
d = 128; τ = 0.1; λ_r = 0.0
d = 256; τ = 0.1; λ_r = 0.0
d = 128; τ = 0.1; λ_r = 0.1
d = 256; τ = 0.1; λ_r = 0.1
d = 128; τ = 0.1; λ_r = 1.0
d = 256; τ = 0.1; λ_r = 1.0
d = 128; τ = 0.1; λ_r = 10.0
d = 256; τ = 0.1; λ_r = 10.0
validation L_r
training epoch
λ_s = 0.0
λ_s = 0.1
λ_s = 1.0
λ_s = 10.0
± 1 stdev.

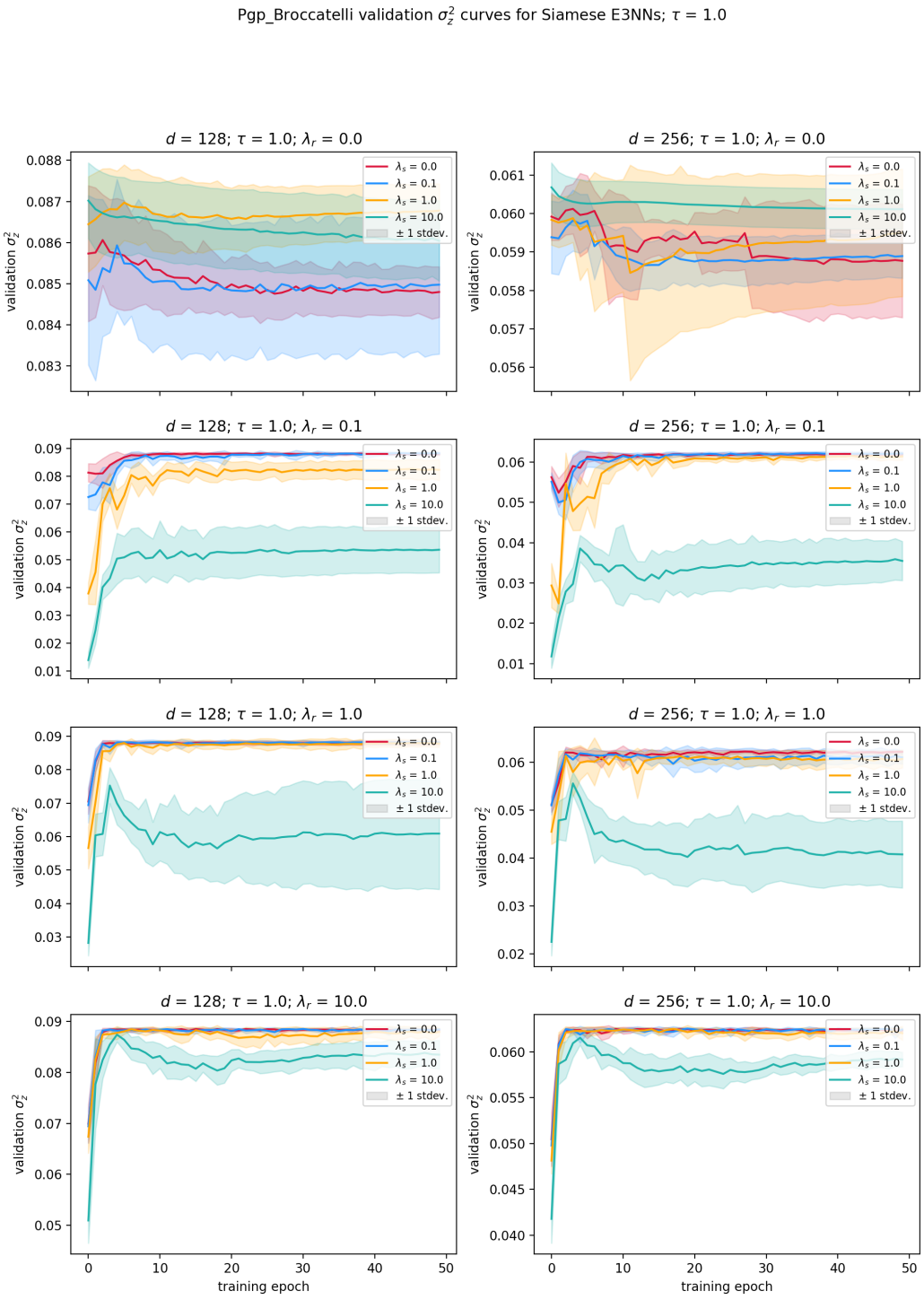

Pgp_Broccatelli validation σ_z^2 curves for Siamese E3NNs; τ = 0.1
d = 128; τ = 0.1; λ_r = 0.0
d = 256; τ = 0.1; λ_r = 0.0
d = 128; τ = 0.1; λ_r = 0.1
d = 256; τ = 0.1; λ_r = 0.1
d = 128; τ = 0.1; λ_r = 1.0
d = 256; τ = 0.1; λ_r = 1.0
d = 128; τ = 0.1; λ_r = 10.0
d = 256; τ = 0.1; λ_r = 10.0
validation σ_z^2
training epoch
λ_s = 0.0
λ_s = 0.1
λ_s = 1.0
λ_s = 10.0
± 1 stdev.

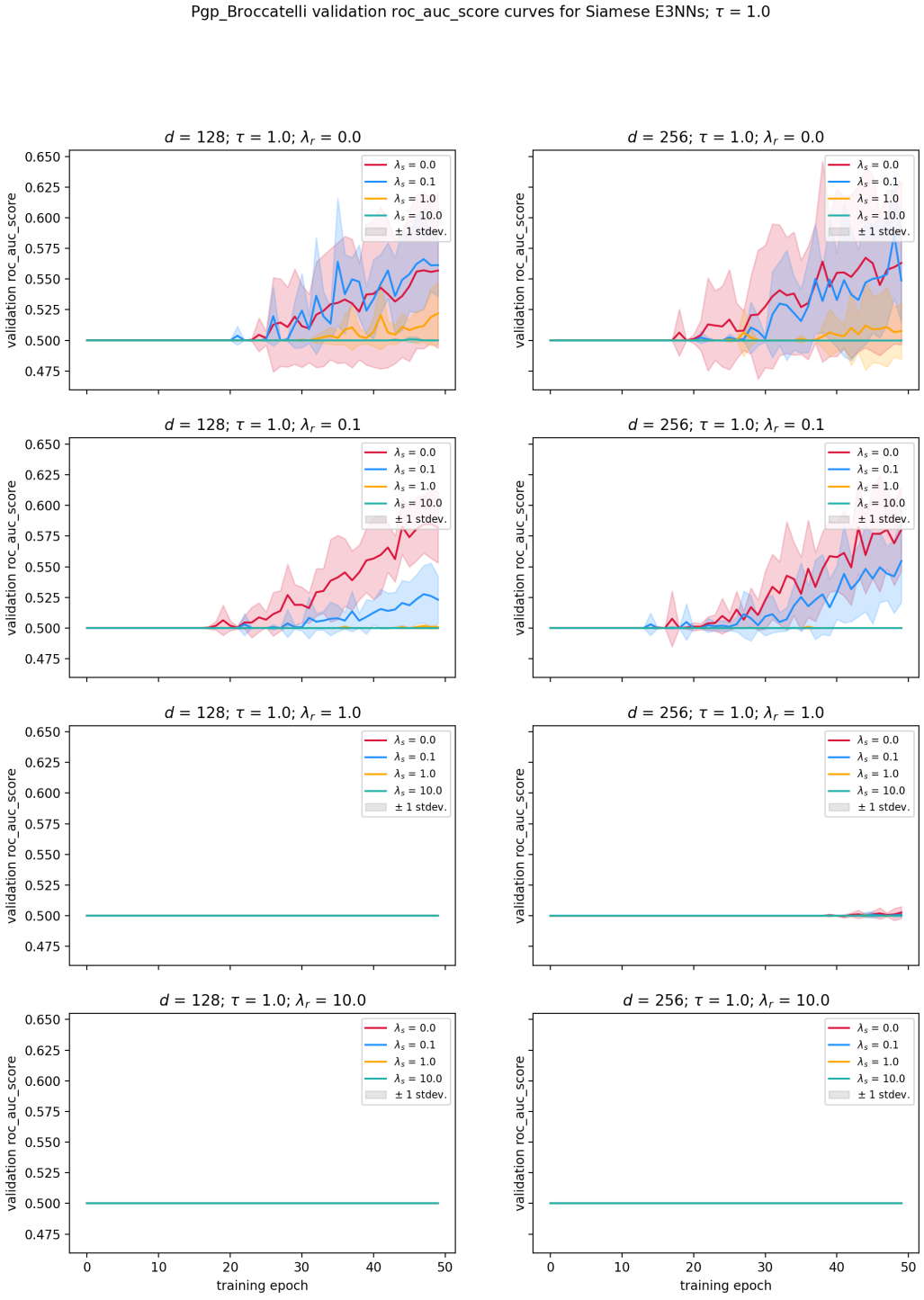

Pgp_Broccatelli validation roc_auc_score curves for Siamese E3NNs; τ = 0.1
d = 128; τ = 0.1; λr = 0.0
d = 256; τ = 0.1; λr = 0.0
d = 128; τ = 0.1; λr = 0.1
d = 256; τ = 0.1; λr = 0.1
d = 128; τ = 0.1; λr = 1.0
d = 256; τ = 0.1; λr = 1.0
d = 128; τ = 0.1; λr = 10.0
d = 256; τ = 0.1; λr = 10.0
validation roc_auc_score
training epoch
λs = 0.0
λs = 0.1
λs = 1.0
λs = 10.0
± 1 stdev.

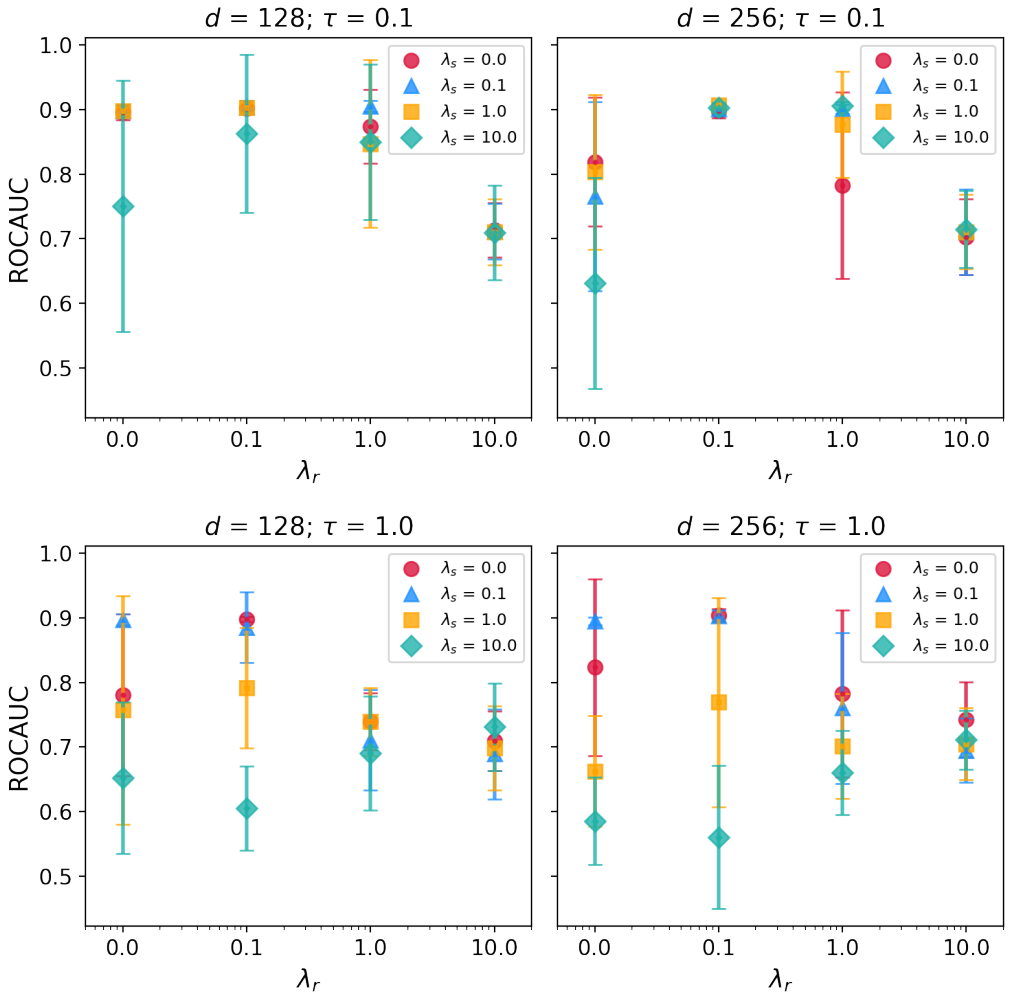

Pgp_Broccatelli validation L_r curves for Siamese E3NNs; τ = 1.0
d = 128; τ = 1.0; λ_r = 0.0
d = 256; τ = 1.0; λ_r = 0.0
d = 128; τ = 1.0; λ_r = 0.1
d = 256; τ = 1.0; λ_r = 0.1
d = 128; τ = 1.0; λ_r = 1.0
d = 256; τ = 1.0; λ_r = 1.0
d = 128; τ = 1.0; λ_r = 10.0
d = 256; τ = 1.0; λ_r = 10.0
validation L_r
training epoch
λ_s = 0.0
λ_s = 0.1
λ_s = 1.0
λ_s = 10.0
± 1 stdev.


Pgp_Broccatelli validation L_s curves for Siamese E3NNs; τ = 1.0
d = 128; τ = 1.0; λ_r = 0.0
d = 256; τ = 1.0; λ_r = 0.0
d = 128; τ = 1.0; λ_r = 0.1
d = 256; τ = 1.0; λ_r = 0.1
d = 128; τ = 1.0; λ_r = 1.0
d = 256; τ = 1.0; λ_r = 1.0
d = 128; τ = 1.0; λ_r = 10.0
d = 256; τ = 1.0; λ_r = 10.0
validation L_s
training epoch
λ_s = 0.0
λ_s = 0.1
λ_s = 1.0
λ_s = 10.0
± 1 stdev.

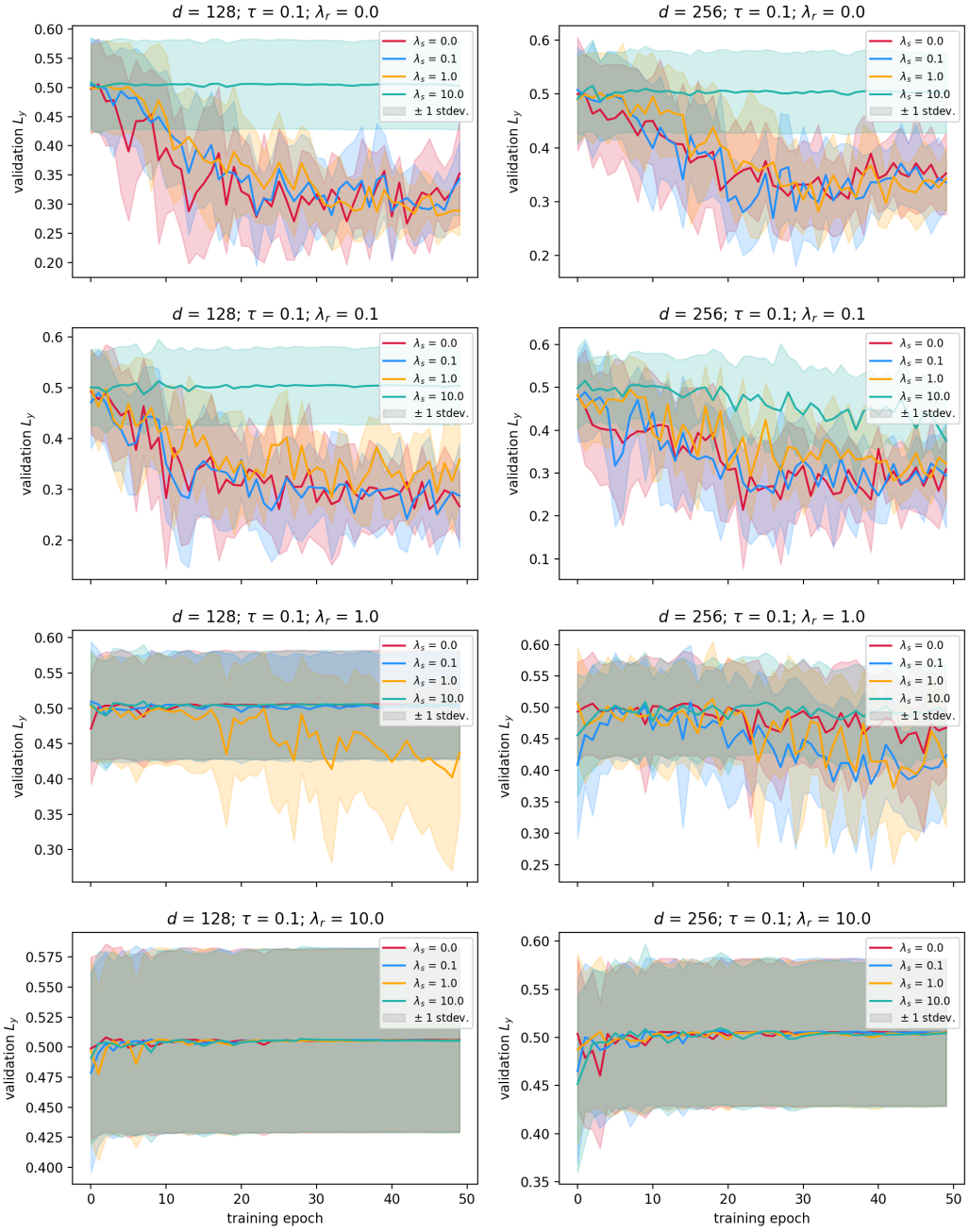

Pgp_Broccatelli validation L_r curves for Siamese E3NNs; τ = 1.0
d = 128; τ = 1.0; λ_r = 0.0
d = 256; τ = 1.0; λ_r = 0.0
d = 128; τ = 1.0; λ_r = 0.1
d = 256; τ = 1.0; λ_r = 0.1
d = 128; τ = 1.0; λ_r = 1.0
d = 256; τ = 1.0; λ_r = 1.0
d = 128; τ = 1.0; λ_r = 10.0
d = 256; τ = 1.0; λ_r = 10.0
validation L_r
training epoch
λ_s = 0.0
λ_s = 0.1
λ_s = 1.0
λ_s = 10.0
± 1 stdev.

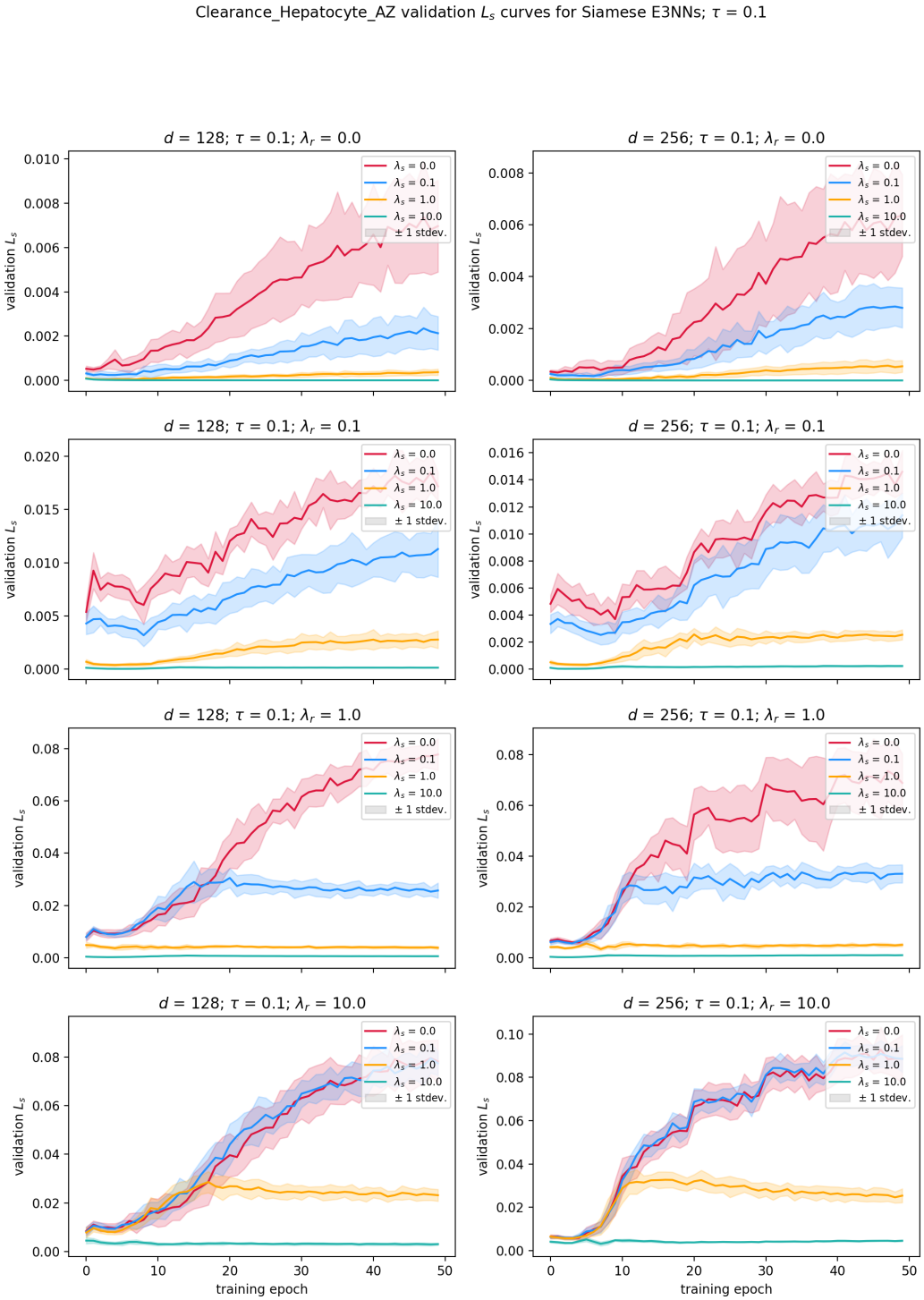

Pgp_Broccatelli validation σ_z^2 curves for Siamese E3NNs; τ = 1.0
d = 128; τ = 1.0; λ_r = 0.0
d = 256; τ = 1.0; λ_r = 0.0
d = 128; τ = 1.0; λ_r = 0.1
d = 256; τ = 1.0; λ_r = 0.1
d = 128; τ = 1.0; λ_r = 1.0
d = 256; τ = 1.0; λ_r = 1.0
d = 128; τ = 1.0; λ_r = 10.0
d = 256; τ = 1.0; λ_r = 10.0
validation σ_z^2
training epoch
λ_s = 0.0
λ_s = 0.1
λ_s = 1.0
λ_s = 10.0
± 1 stdev.

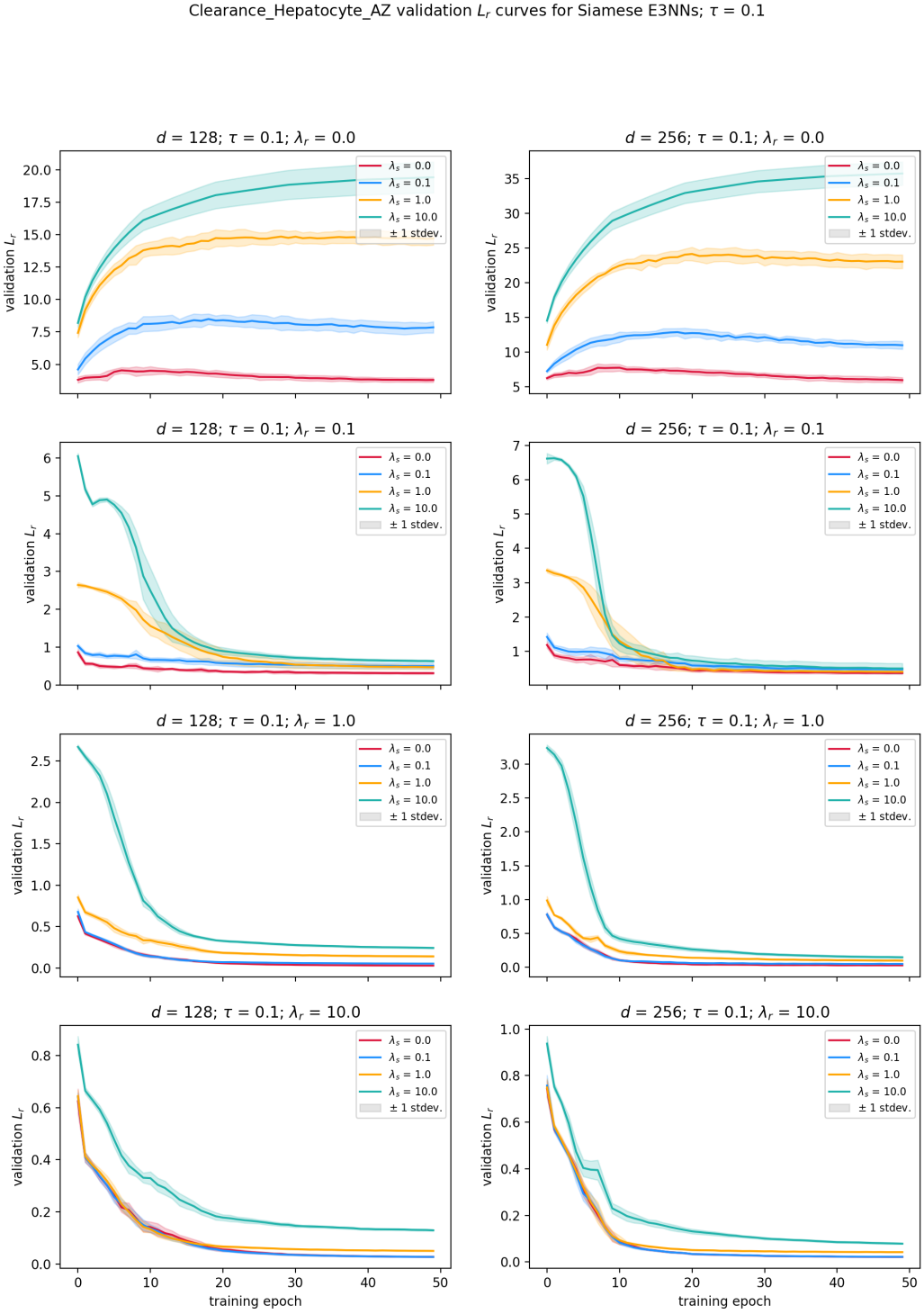

Pgp_Broccatelli validation roc_auc_score curves for Siamese E3NNs; τ = 1.0
d = 128; τ = 1.0; λr = 0.0
d = 256; τ = 1.0; λr = 0.0
d = 128; τ = 1.0; λr = 0.1
d = 256; τ = 1.0; λr = 0.1
d = 128; τ = 1.0; λr = 1.0
d = 256; τ = 1.0; λr = 1.0
d = 128; τ = 1.0; λr = 10.0
d = 256; τ = 1.0; λr = 10.0
validation roc_auc_score
training epoch
λs = 0.0
λs = 0.1
λs = 1.0
λs = 10.0
± 1 stdev.

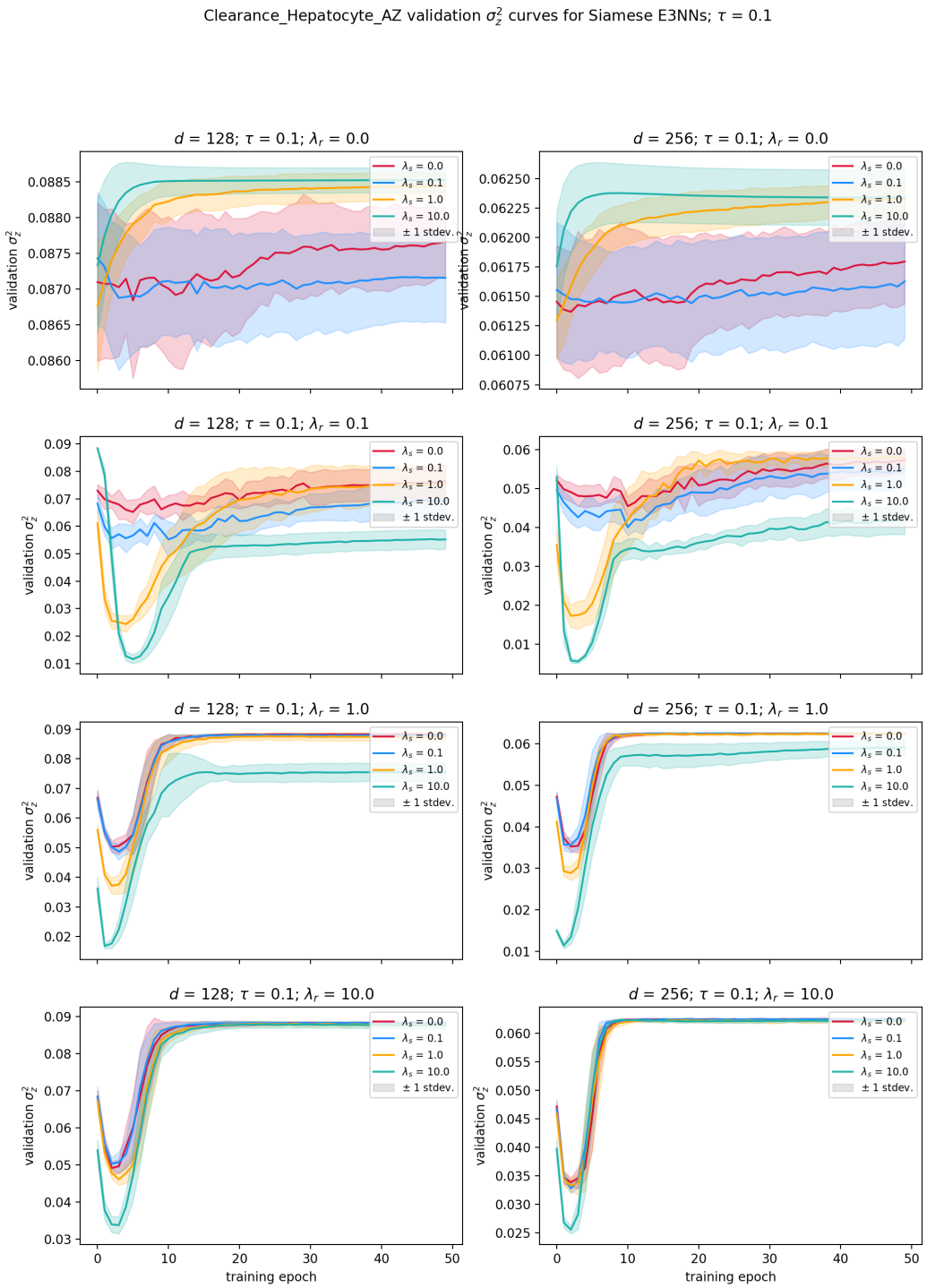

$d = 128; \tau = 0.1$
$d = 256; \tau = 0.1$
$d = 128; \tau = 1.0$
$d = 256; \tau = 1.0$
ROCAUC
λ_r
$\lambda_s = 0.0$
$\lambda_s = 0.1$
$\lambda_s = 1.0$
$\lambda_s = 10.0$
0.0
0.1
1.0
10.0
0.5
0.6
0.7
0.8
0.9
1.0

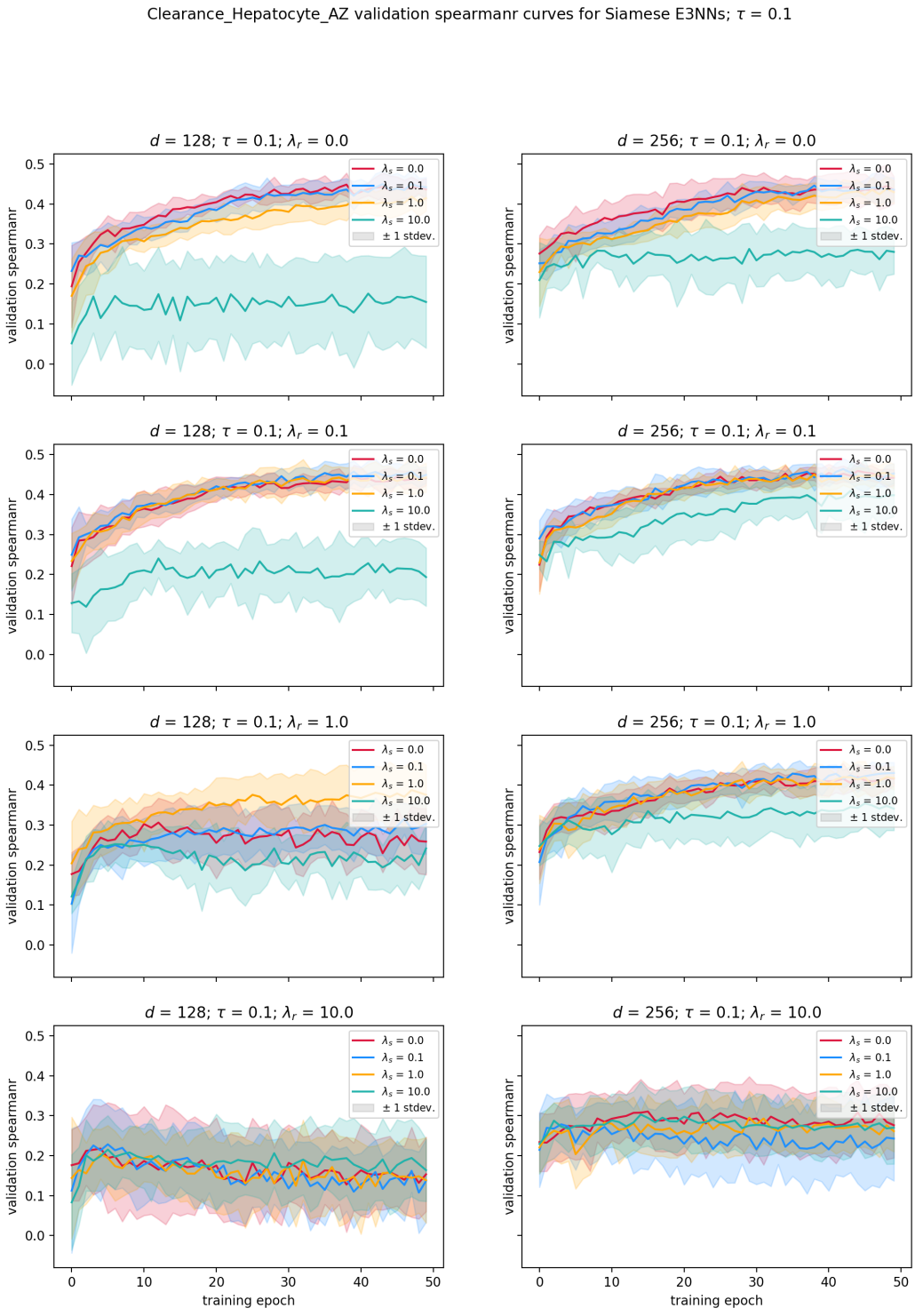
A.3.3 CLEAR

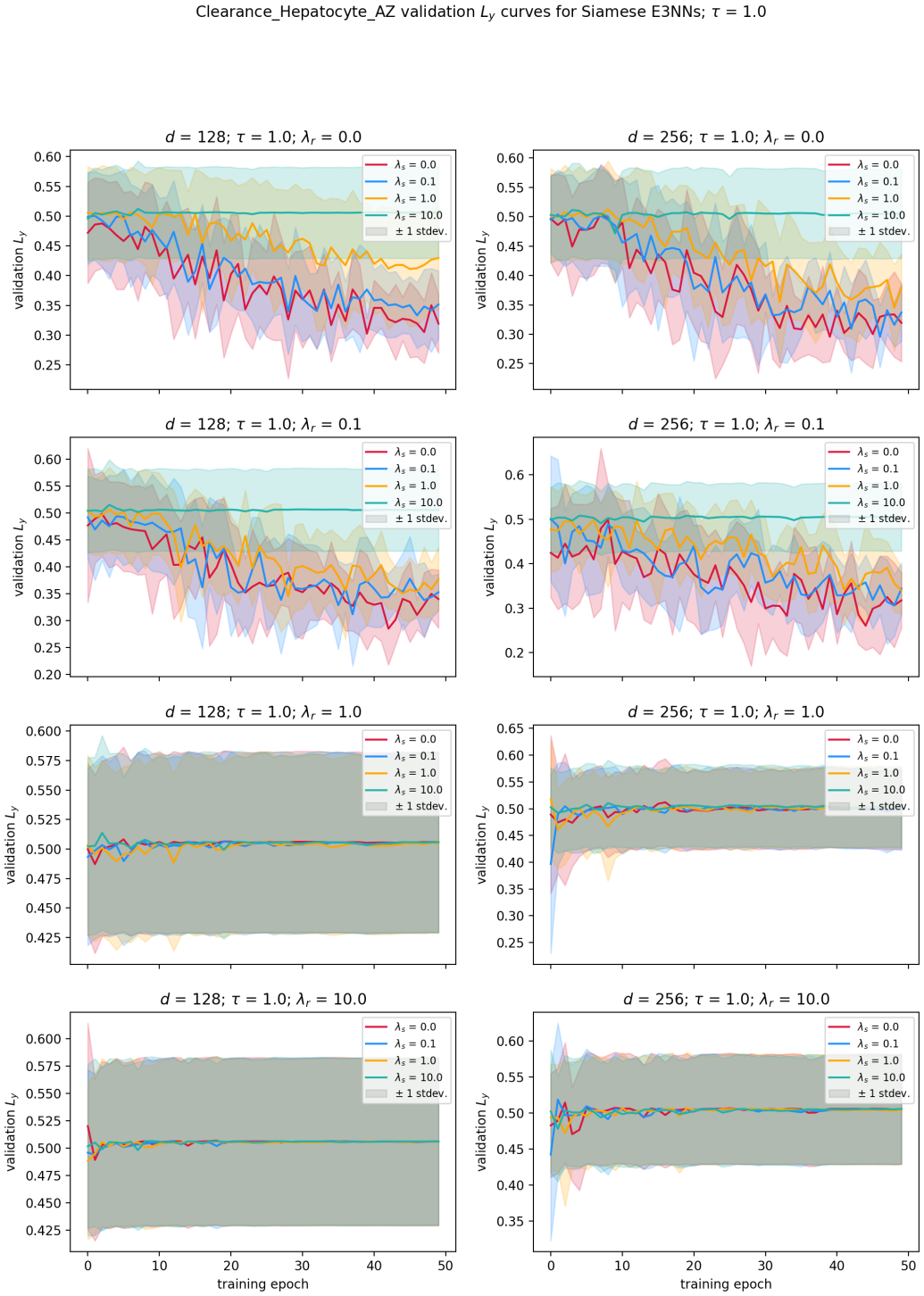

Clearance_Hepatocyte_AZ validation L_y curves for Siamese E3NNs; τ = 0.1
d = 128; τ = 0.1; λ_r = 0.0
d = 256; τ = 0.1; λ_r = 0.0
d = 128; τ = 0.1; λ_r = 0.1
d = 256; τ = 0.1; λ_r = 0.1
d = 128; τ = 0.1; λ_r = 1.0
d = 256; τ = 0.1; λ_r = 1.0
d = 128; τ = 0.1; λ_r = 10.0
d = 256; τ = 0.1; λ_r = 10.0
validation L_y
training epoch
λ_s = 0.0
λ_s = 0.1
λ_s = 1.0
λ_s = 10.0
± 1 stdev.

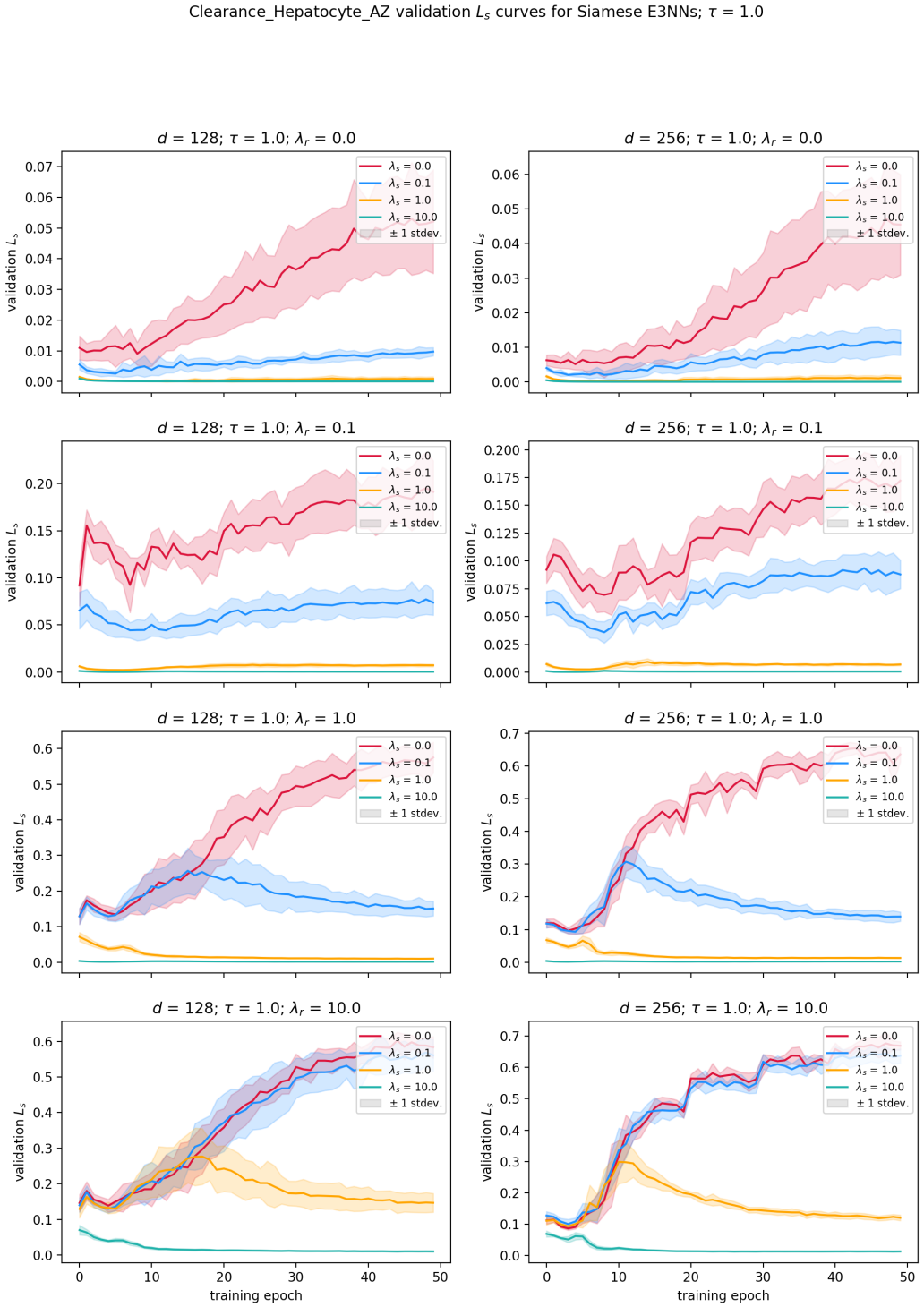

Clearance_Hepatocyte_AZ validation L_s curves for Siamese E3NNs; τ = 0.1
d = 128; τ = 0.1; λ_r = 0.0
d = 256; τ = 0.1; λ_r = 0.0
d = 128; τ = 0.1; λ_r = 0.1
d = 256; τ = 0.1; λ_r = 0.1
d = 128; τ = 0.1; λ_r = 1.0
d = 256; τ = 0.1; λ_r = 1.0
d = 128; τ = 0.1; λ_r = 10.0
d = 256; τ = 0.1; λ_r = 10.0
validation L_s
training epoch
λ_s = 0.0
λ_s = 0.1
λ_s = 1.0
λ_s = 10.0
± 1 stdev.

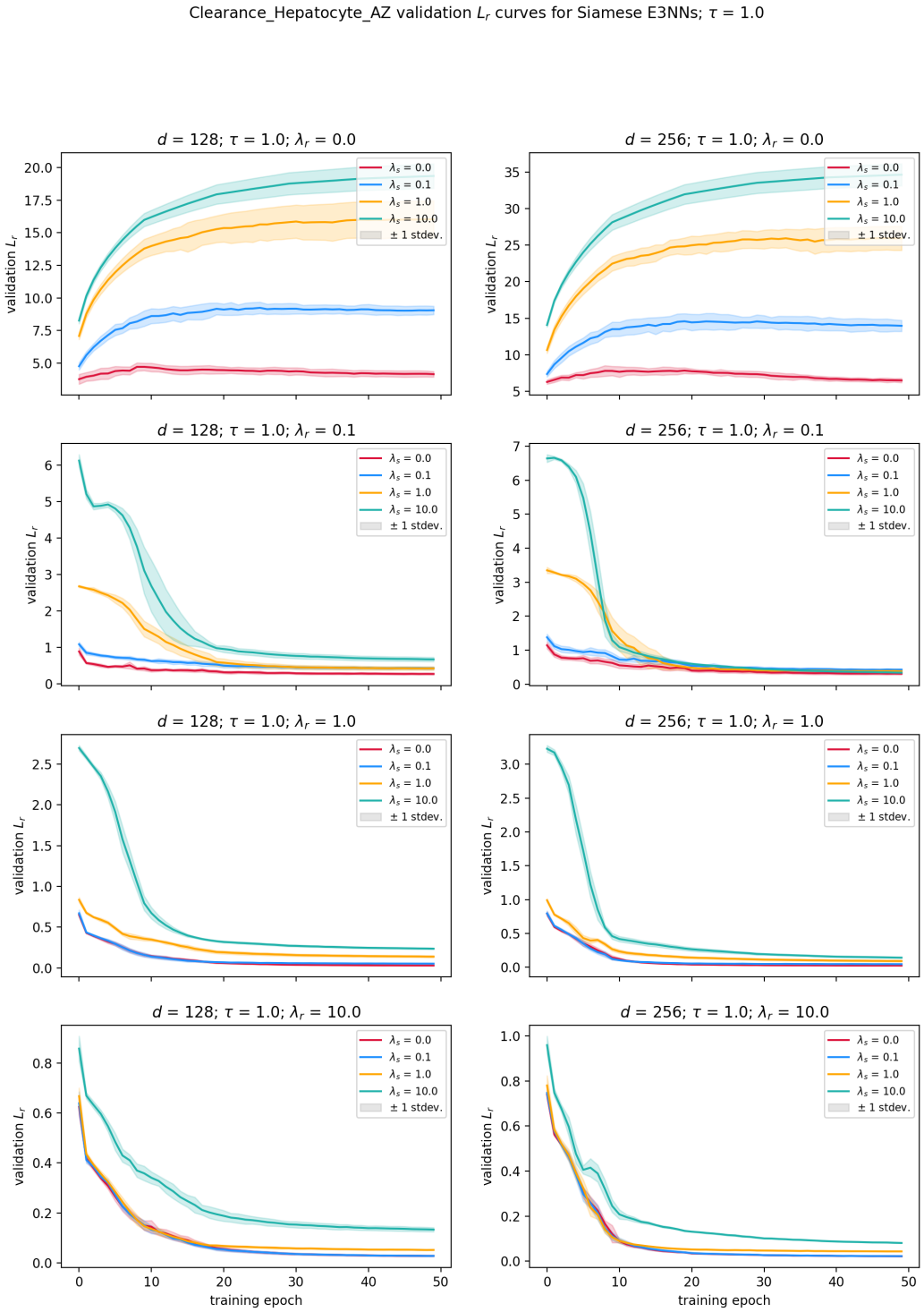

Clearance_Hepatocyte_AZ validation L_r curves for Siamese E3NNs; τ = 0.1
d = 128; τ = 0.1; λ_r = 0.0
d = 256; τ = 0.1; λ_r = 0.0
d = 128; τ = 0.1; λ_r = 0.1
d = 256; τ = 0.1; λ_r = 0.1
d = 128; τ = 0.1; λ_r = 1.0
d = 256; τ = 0.1; λ_r = 1.0
d = 128; τ = 0.1; λ_r = 10.0
d = 256; τ = 0.1; λ_r = 10.0
validation L_r
training epoch
λ_s = 0.0
λ_s = 0.1
λ_s = 1.0
λ_s = 10.0
± 1 stdev.

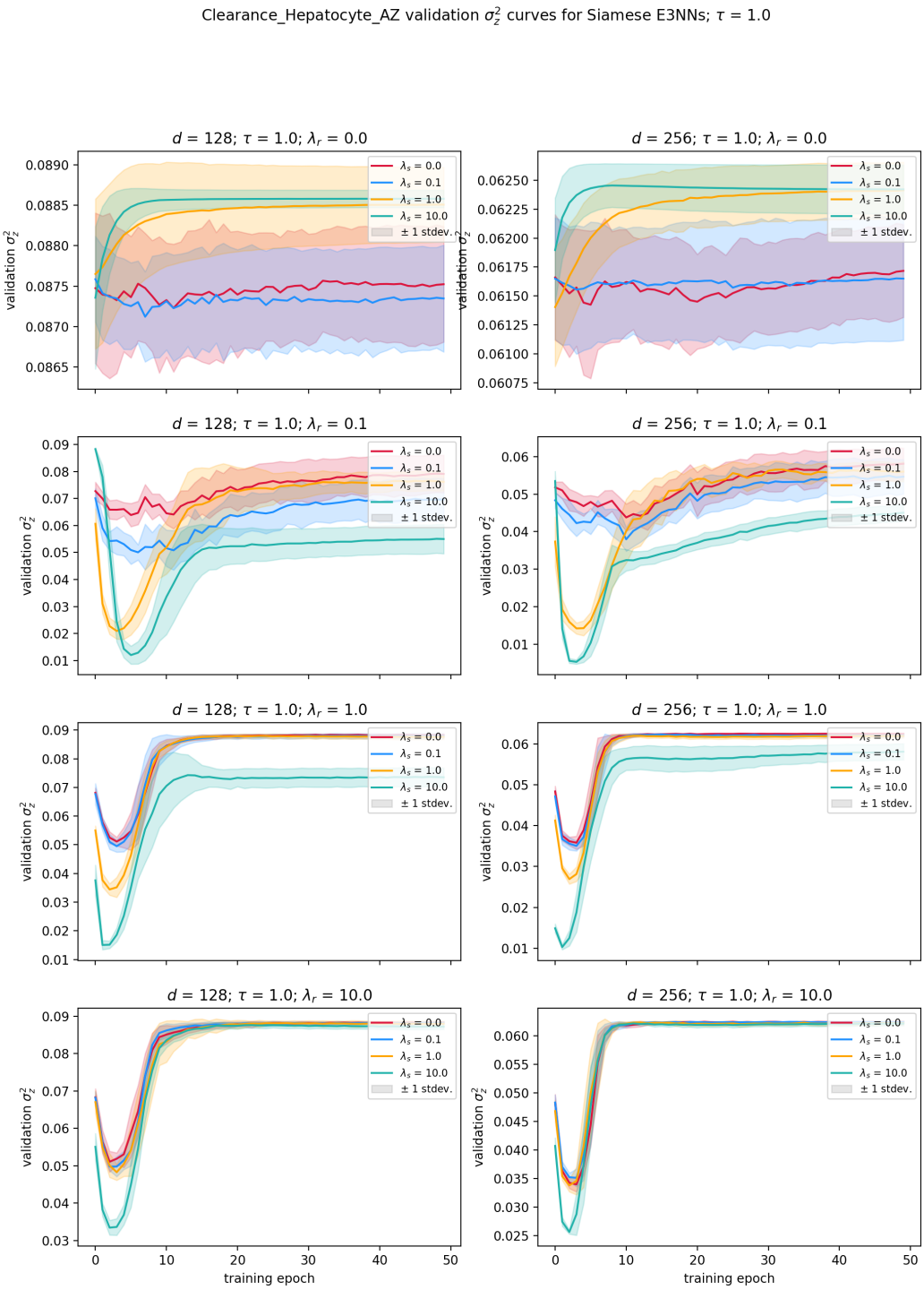

Clearance_Hepatocyte_AZ validation σ_z^2 curves for Siamese E3NNs; τ = 0.1
d = 128; τ = 0.1; λ_r = 0.0
d = 256; τ = 0.1; λ_r = 0.0
d = 128; τ = 0.1; λ_r = 0.1
d = 256; τ = 0.1; λ_r = 0.1
d = 128; τ = 0.1; λ_r = 1.0
d = 256; τ = 0.1; λ_r = 1.0
d = 128; τ = 0.1; λ_r = 10.0
d = 256; τ = 0.1; λ_r = 10.0
validation σ_z^2
training epoch
λ_s = 0.0
λ_s = 0.1
λ_s = 1.0
λ_s = 10.0
± 1 stdev.

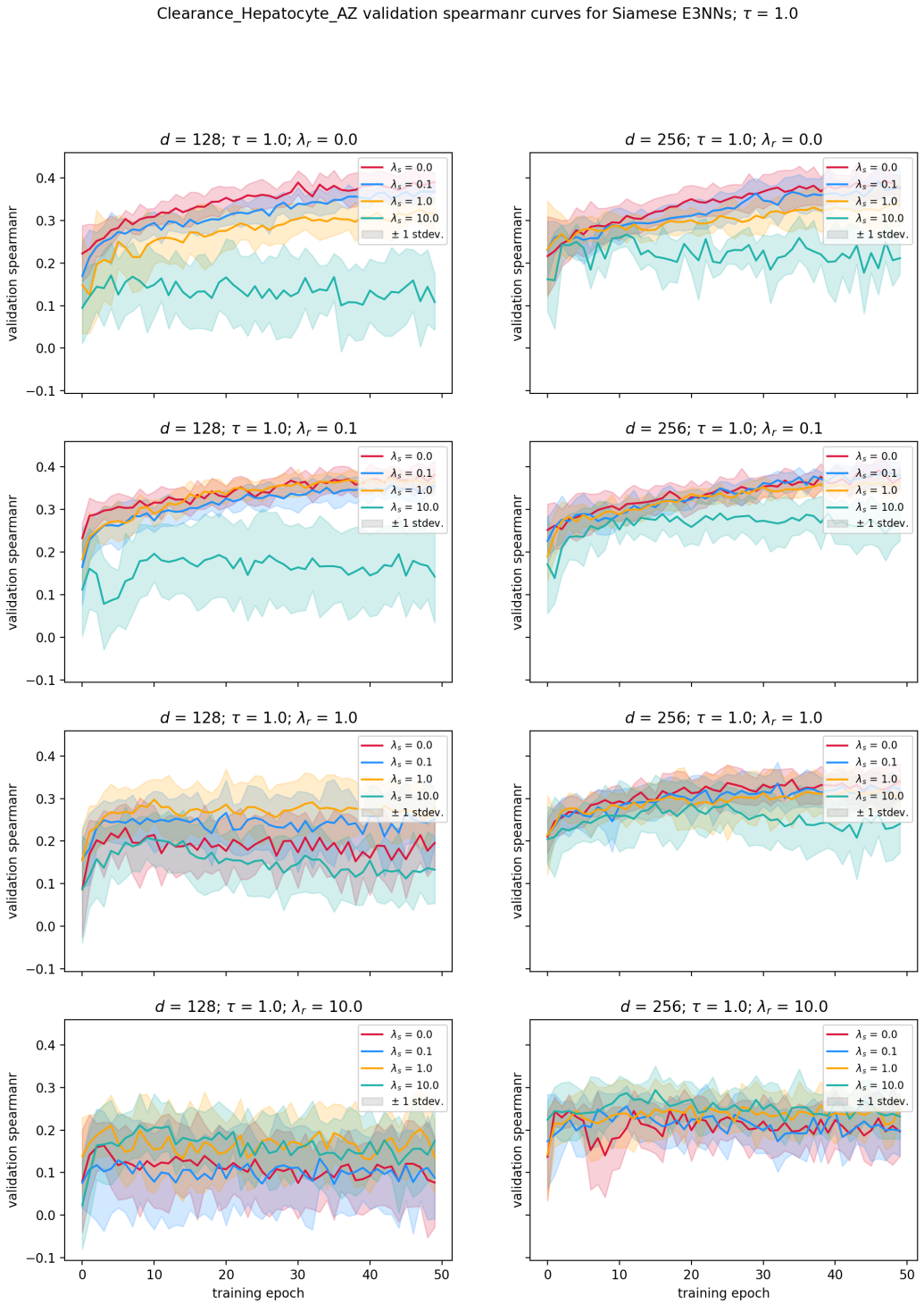

Clearance_Hepatocyte_AZ validation spearmanr curves for Siamese E3NNs; τ = 0.1
d = 128; τ = 0.1; λr = 0.0
d = 256; τ = 0.1; λr = 0.0
d = 128; τ = 0.1; λr = 0.1
d = 256; τ = 0.1; λr = 0.1
d = 128; τ = 0.1; λr = 1.0
d = 256; τ = 0.1; λr = 1.0
d = 128; τ = 0.1; λr = 10.0
d = 256; τ = 0.1; λr = 10.0
validation spearmanr
training epoch
λs = 0.0
λs = 0.1
λs = 1.0
λs = 10.0
± 1 stdev.

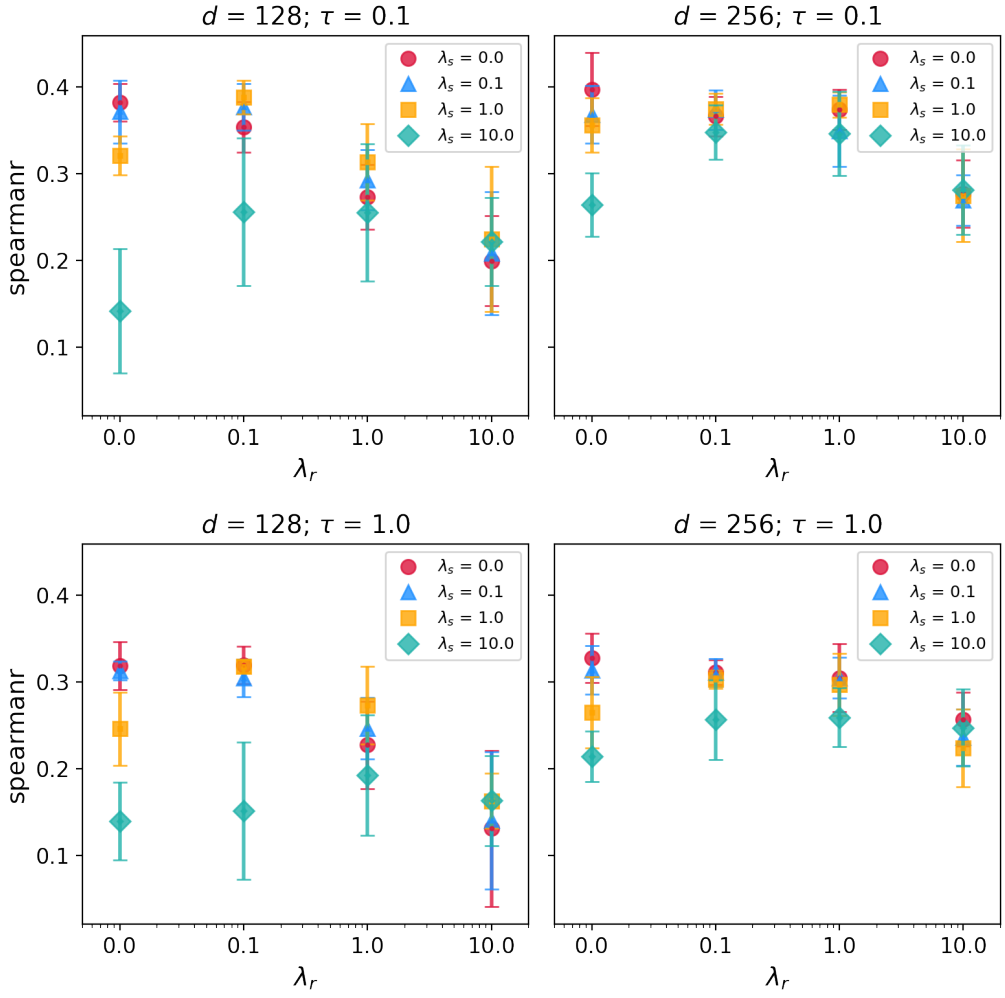

Clearance_Hepatocyte_AZ validation L_y curves for Siamese E3NNs; τ = 1.0
d = 128; τ = 1.0; λ_r = 0.0
d = 256; τ = 1.0; λ_r = 0.0
d = 128; τ = 1.0; λ_r = 0.1
d = 256; τ = 1.0; λ_r = 0.1
d = 128; τ = 1.0; λ_r = 1.0
d = 256; τ = 1.0; λ_r = 1.0
d = 128; τ = 1.0; λ_r = 10.0
d = 256; τ = 1.0; λ_r = 10.0
validation L_y
training epoch
λ_s = 0.0
λ_s = 0.1
λ_s = 1.0
λ_s = 10.0
± 1 stdev.


Clearance_Hepatocyte_AZ validation L_s curves for Siamese E3NNs; τ = 1.0
d = 128; τ = 1.0; λ_r = 0.0
d = 256; τ = 1.0; λ_r = 0.0
d = 128; τ = 1.0; λ_r = 0.1
d = 256; τ = 1.0; λ_r = 0.1
d = 128; τ = 1.0; λ_r = 1.0
d = 256; τ = 1.0; λ_r = 1.0
d = 128; τ = 1.0; λ_r = 10.0
d = 256; τ = 1.0; λ_r = 10.0
validation L_s
training epoch
λ_s = 0.0
λ_s = 0.1
λ_s = 1.0
λ_s = 10.0
± 1 stdev.


Clearance_Hepatocyte_AZ validation L_r curves for Siamese E3NNs; τ = 1.0
d = 128; τ = 1.0; λ_r = 0.0
d = 256; τ = 1.0; λ_r = 0.0
d = 128; τ = 1.0; λ_r = 0.1
d = 256; τ = 1.0; λ_r = 0.1
d = 128; τ = 1.0; λ_r = 1.0
d = 256; τ = 1.0; λ_r = 1.0
d = 128; τ = 1.0; λ_r = 10.0
d = 256; τ = 1.0; λ_r = 10.0
validation L_r
training epoch
λ_s = 0.0
λ_s = 0.1
λ_s = 1.0
λ_s = 10.0
± 1 stdev.


Clearance_Hepatocyte_AZ validation σ_z^2 curves for Siamese E3NNs; τ = 1.0
d = 128; τ = 1.0; λ_r = 0.0
d = 256; τ = 1.0; λ_r = 0.0
d = 128; τ = 1.0; λ_r = 0.1
d = 256; τ = 1.0; λ_r = 0.1
d = 128; τ = 1.0; λ_r = 1.0
d = 256; τ = 1.0; λ_r = 1.0
d = 128; τ = 1.0; λ_r = 10.0
d = 256; τ = 1.0; λ_r = 10.0
validation σ_z^2
training epoch
λ_s = 0.0
λ_s = 0.1
λ_s = 1.0
λ_s = 10.0
± 1 stdev.

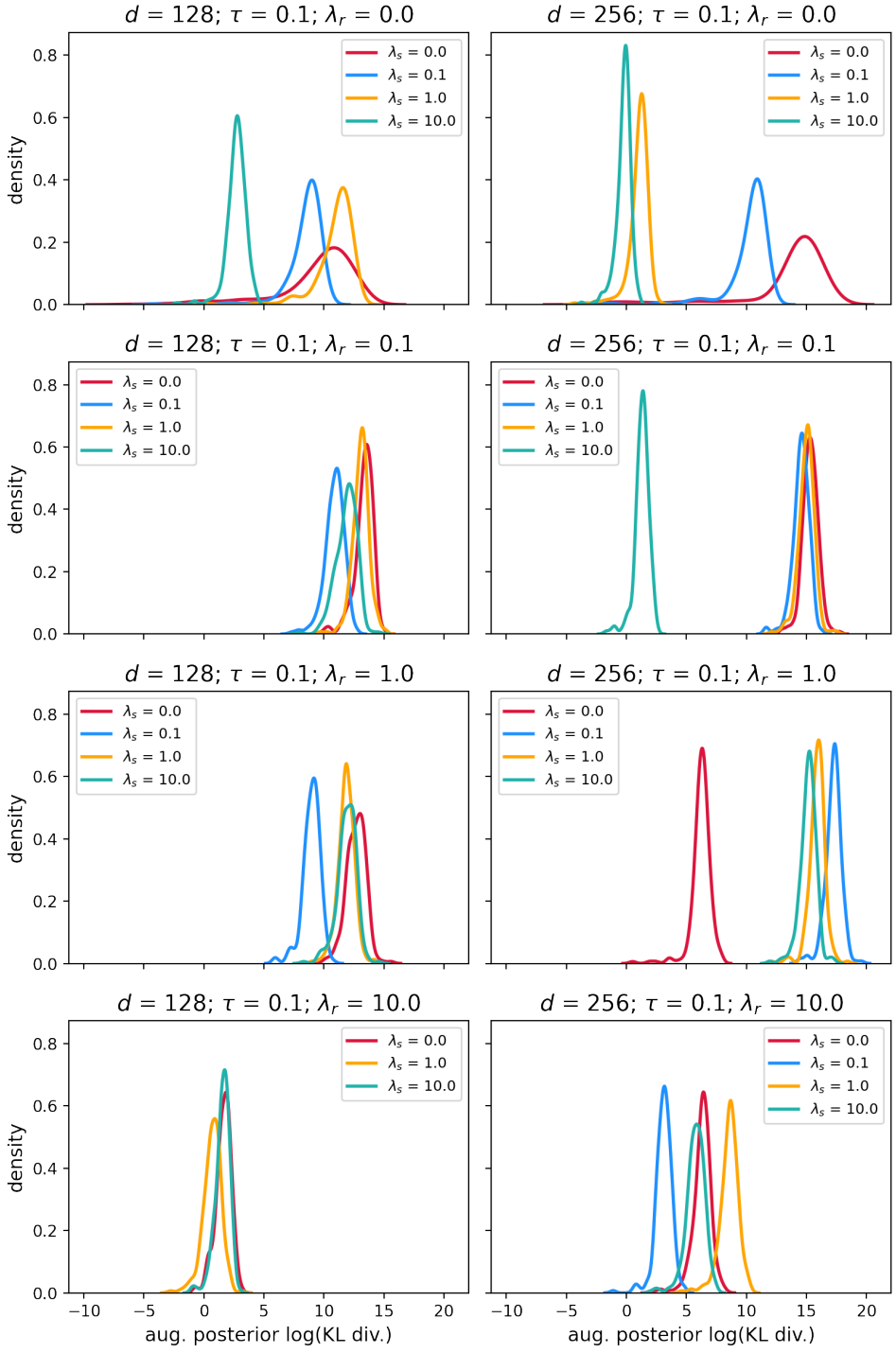

Clearance_Hepatocyte_AZ validation spearmanr curves for Siamese E3NNs; τ = 1.0
d = 128; τ = 1.0; λr = 0.0
d = 256; τ = 1.0; λr = 0.0
d = 128; τ = 1.0; λr = 0.1
d = 256; τ = 1.0; λr = 0.1
d = 128; τ = 1.0; λr = 1.0
d = 256; τ = 1.0; λr = 1.0
d = 128; τ = 1.0; λr = 10.0
d = 256; τ = 1.0; λr = 10.0
validation spearmanr
training epoch
λs = 0.0
λs = 0.1
λs = 1.0
λs = 10.0
± 1 stdev.

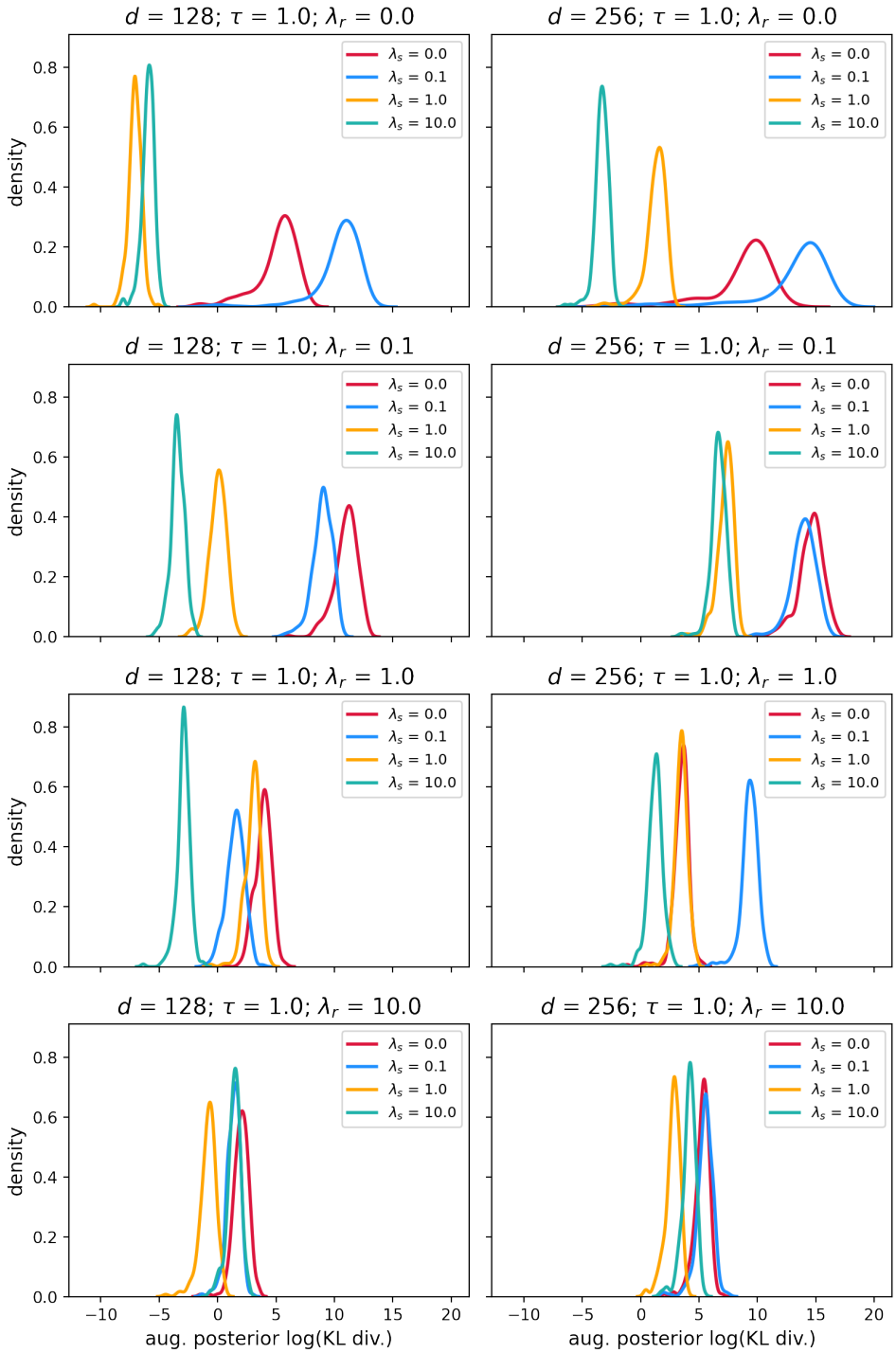

d = 128; τ = 0.1
d = 256; τ = 0.1
d = 128; τ = 1.0
d = 256; τ = 1.0
spearmanr
λ_r
λ_s = 0.0
λ_s = 0.1
λ_s = 1.0
λ_s = 10.0
0.1
0.2
0.3
0.4
0.0
0.1
1.0
10.0

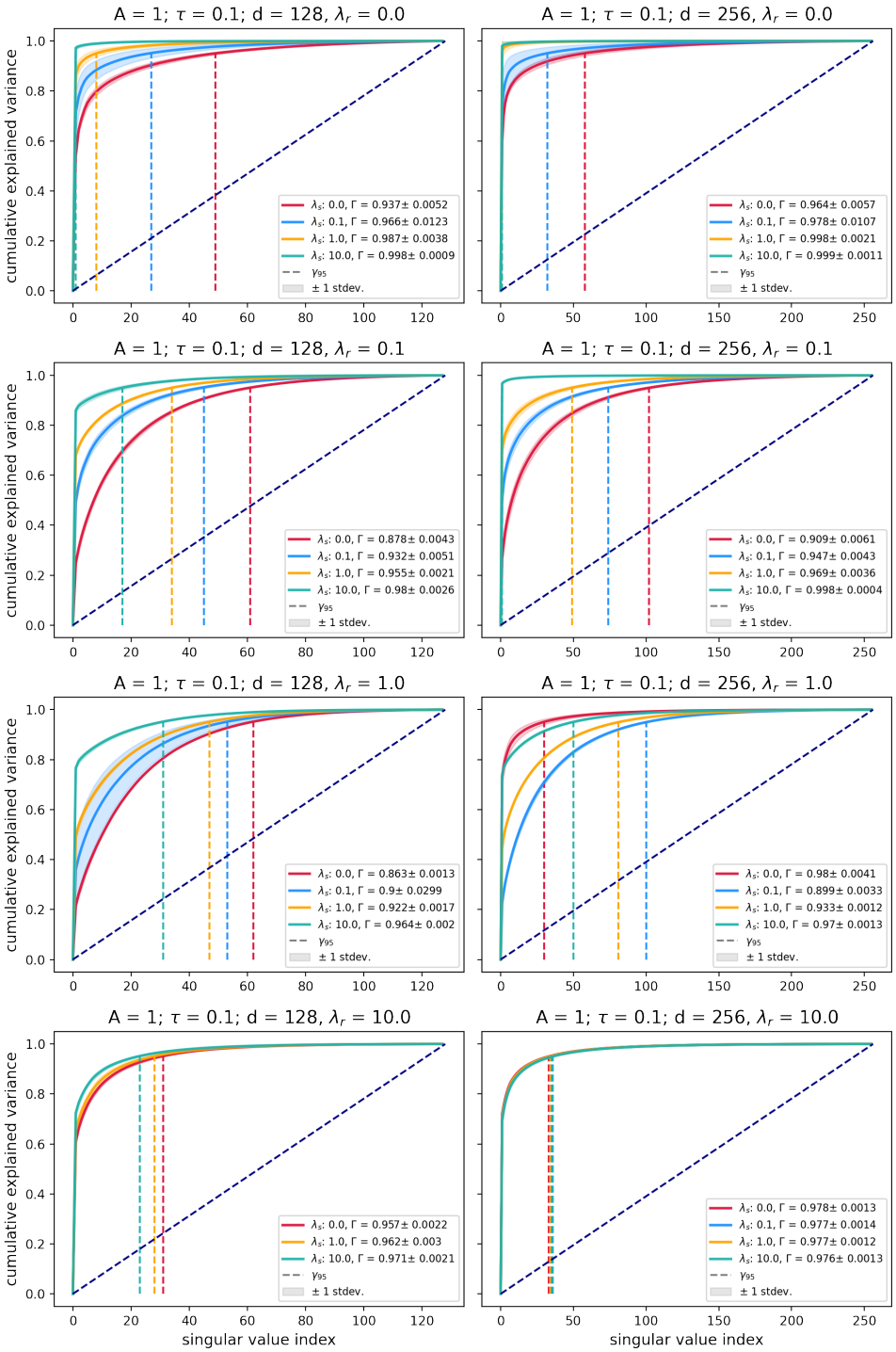
A.3.4 MANIFOLD SMOOTHNESS AND PARTIAL DIMENSIONAL COLLAPSE

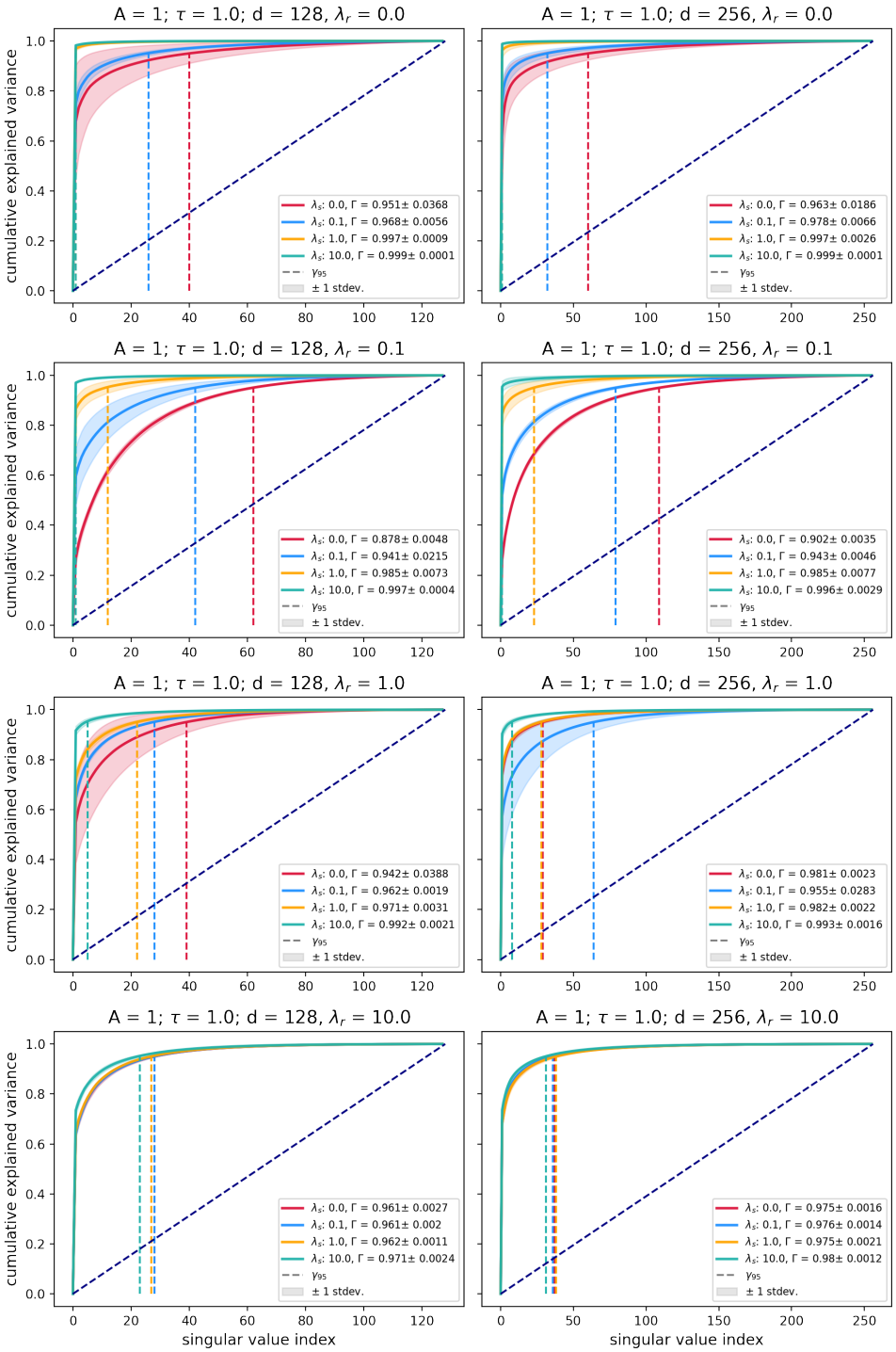
See Sections 3.4 and 4.3 for details on the analysis of manifold properties. Expanded results from Figure 4 are included below, organized in the same manner as those in Section A.3.1. All results shown were computed for test set samples only, which were never seen in training or validation.

A.3.5 PGP

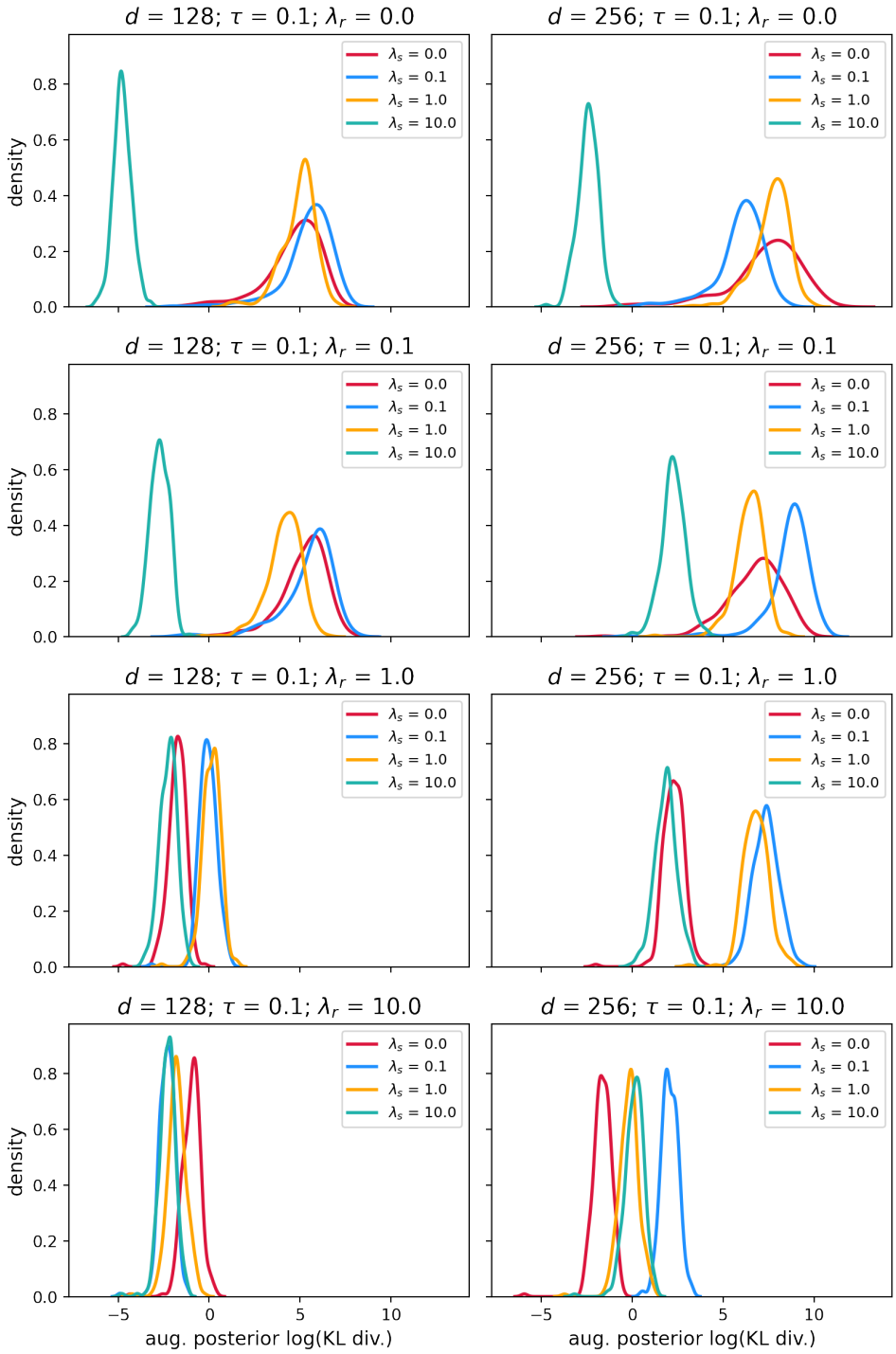


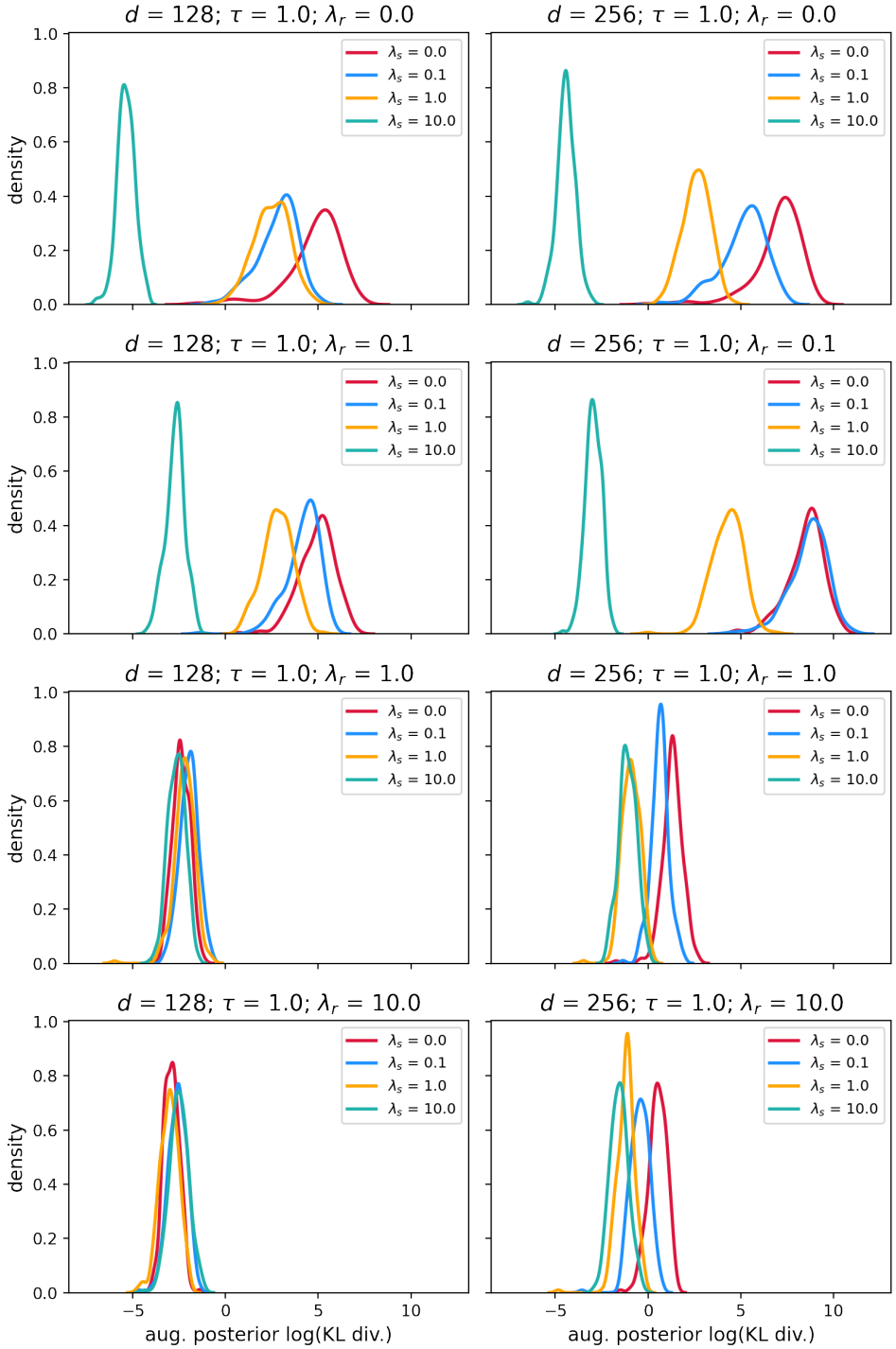

$d = 128;\ \tau = 1.0;\ \lambda_r = 0.0$
$d = 256;\ \tau = 1.0;\ \lambda_r = 0.0$
$d = 128;\ \tau = 1.0;\ \lambda_r = 0.1$
$d = 256;\ \tau = 1.0;\ \lambda_r = 0.1$
$d = 128;\ \tau = 1.0;\ \lambda_r = 1.0$
$d = 256;\ \tau = 1.0;\ \lambda_r = 1.0$
$d = 128;\ \tau = 1.0;\ \lambda_r = 10.0$
$d = 256;\ \tau = 1.0;\ \lambda_r = 10.0$
$\lambda_s = 0.0$
$\lambda_s = 0.1$
$\lambda_s = 1.0$
$\lambda_s = 10.0$
density
0.0
0.2
0.4
0.6
0.8
−10
−5
0
5
10
15
20
aug. posterior log(KL div.)

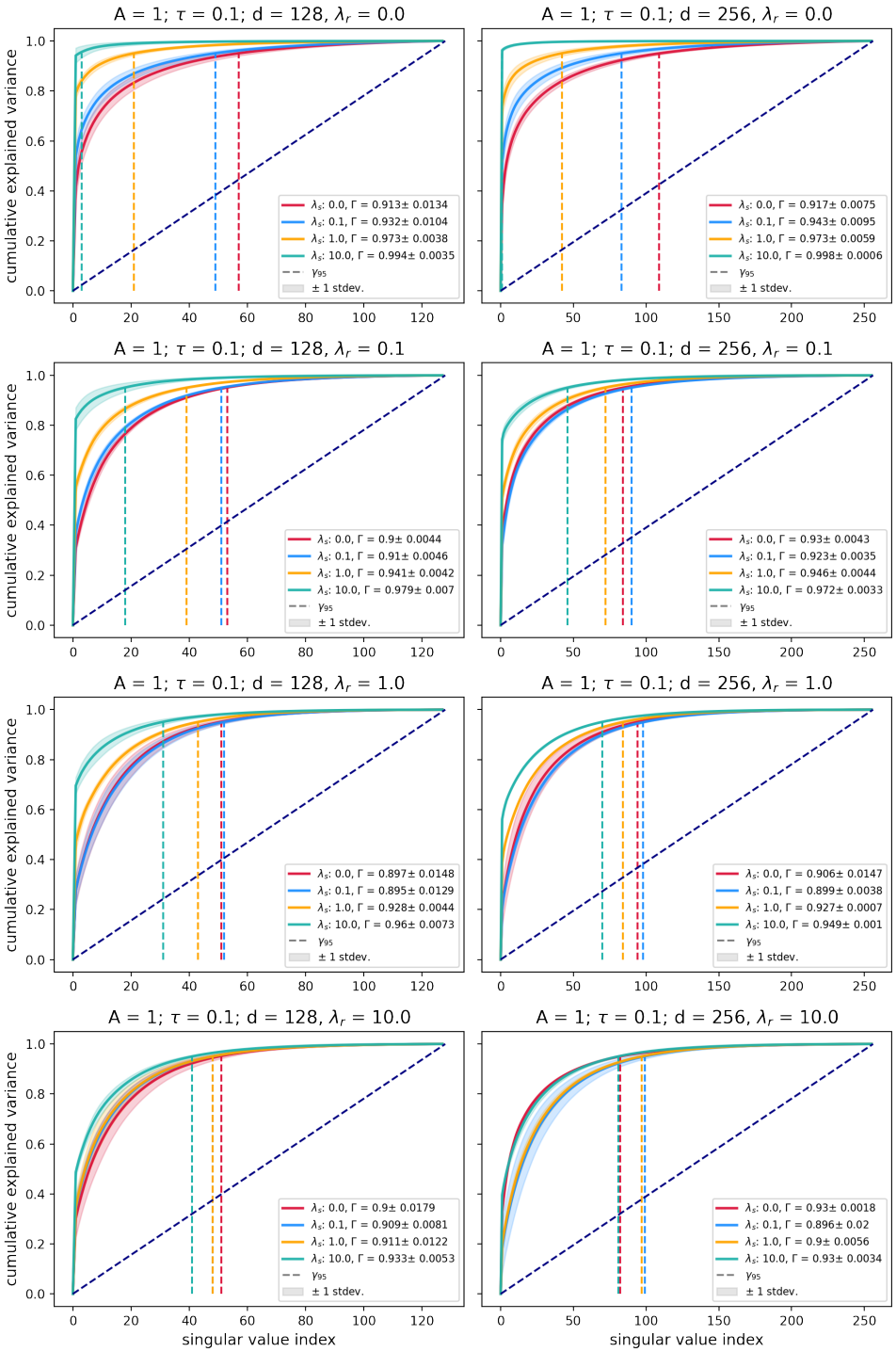

A = 1; τ = 0.1; d = 128, λr = 0.0
A = 1; τ = 0.1; d = 256, λr = 0.0
λs: 0.0, Γ = 0.937± 0.0052
λs: 0.1, Γ = 0.966± 0.0123
λs: 1.0, Γ = 0.987± 0.0038
λs: 10.0, Γ = 0.998± 0.0009
λs: 0.0, Γ = 0.964± 0.0057
λs: 0.1, Γ = 0.978± 0.0107
λs: 1.0, Γ = 0.998± 0.0021
λs: 10.0, Γ = 0.999± 0.0011
A = 1; τ = 0.1; d = 128, λr = 0.1
A = 1; τ = 0.1; d = 256, λr = 0.1
λs: 0.0, Γ = 0.878± 0.0043
λs: 0.1, Γ = 0.932± 0.0051
λs: 1.0, Γ = 0.955± 0.0021
λs: 10.0, Γ = 0.98± 0.0026
λs: 0.0, Γ = 0.909± 0.0061
λs: 0.1, Γ = 0.947± 0.0043
λs: 1.0, Γ = 0.969± 0.0036
λs: 10.0, Γ = 0.998± 0.0004
A = 1; τ = 0.1; d = 128, λr = 1.0
A = 1; τ = 0.1; d = 256, λr = 1.0
λs: 0.0, Γ = 0.863± 0.0013
λs: 0.1, Γ = 0.9± 0.0299
λs: 1.0, Γ = 0.922± 0.0017
λs: 10.0, Γ = 0.964± 0.002
λs: 0.0, Γ = 0.98± 0.0041
λs: 0.1, Γ = 0.899± 0.0033
λs: 1.0, Γ = 0.933± 0.0012
λs: 10.0, Γ = 0.97± 0.0013
A = 1; τ = 0.1; d = 128, λr = 10.0
A = 1; τ = 0.1; d = 256, λr = 10.0
λs: 0.0, Γ = 0.957± 0.0022
λs: 1.0, Γ = 0.962± 0.003
λs: 10.0, Γ = 0.971± 0.0021
λs: 0.0, Γ = 0.978± 0.0013
λs: 0.1, Γ = 0.977± 0.0014
λs: 1.0, Γ = 0.977± 0.0012
λs: 10.0, Γ = 0.976± 0.0013
γ95
± 1 stdev.
cumulative explained variance
singular value index

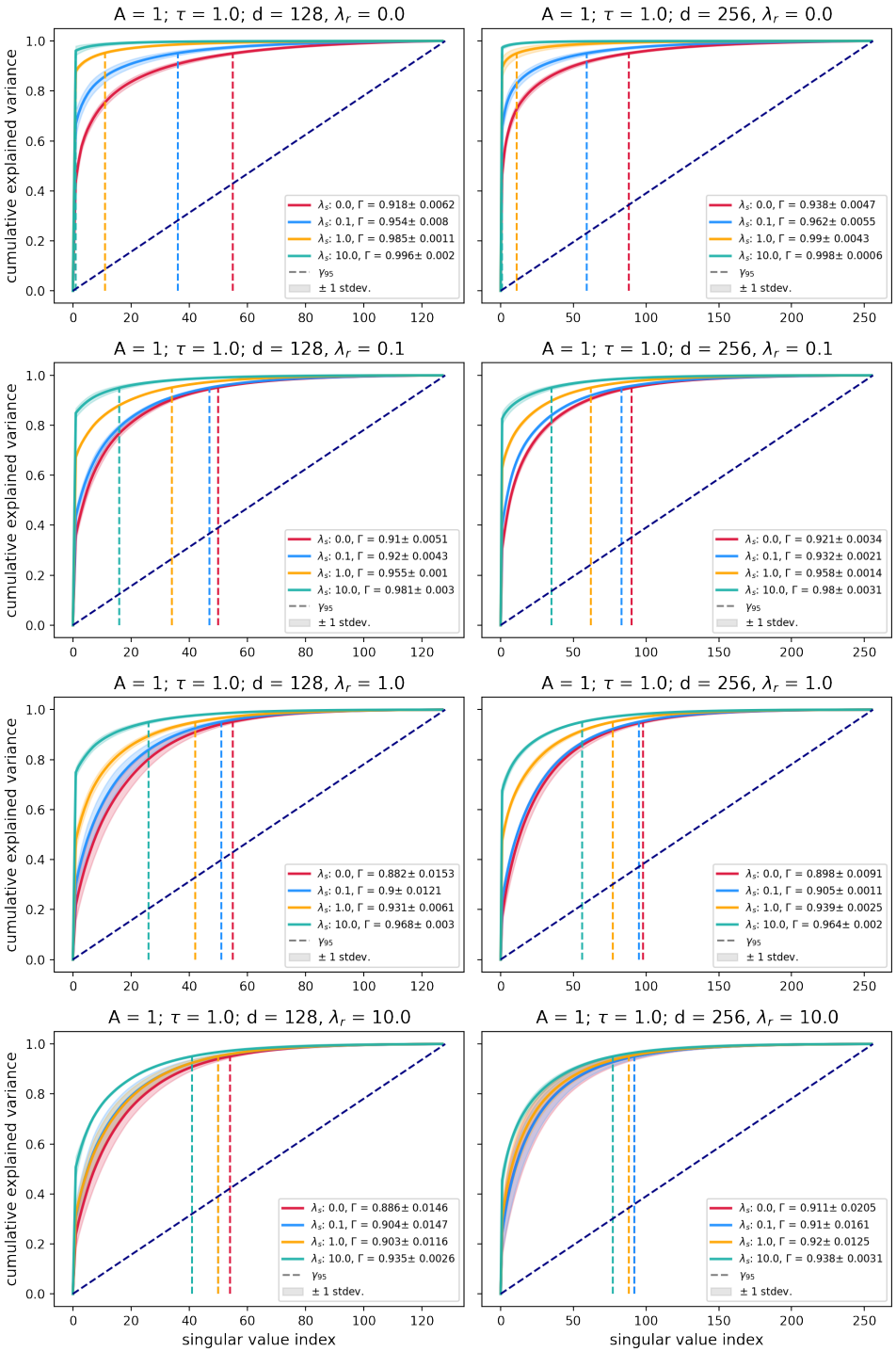

A = 1; τ = 1.0; d = 128, λr = 0.0
A = 1; τ = 1.0; d = 256, λr = 0.0
A = 1; τ = 1.0; d = 128, λr = 0.1
A = 1; τ = 1.0; d = 256, λr = 0.1
A = 1; τ = 1.0; d = 128, λr = 1.0
A = 1; τ = 1.0; d = 256, λr = 1.0
A = 1; τ = 1.0; d = 128, λr = 10.0
A = 1; τ = 1.0; d = 256, λr = 10.0
cumulative explained variance
singular value index
λs: 0.0, Γ = 0.951± 0.0368
λs: 0.1, Γ = 0.968± 0.0056
λs: 1.0, Γ = 0.997± 0.0009
λs: 10.0, Γ = 0.999± 0.0001
γ95
± 1 stdev.
λs: 0.0, Γ = 0.963± 0.0186
λs: 0.1, Γ = 0.978± 0.0066
λs: 1.0, Γ = 0.997± 0.0026
λs: 10.0, Γ = 0.999± 0.0001
λs: 0.0, Γ = 0.878± 0.0048
λs: 0.1, Γ = 0.941± 0.0215
λs: 1.0, Γ = 0.985± 0.0073
λs: 10.0, Γ = 0.997± 0.0004
λs: 0.0, Γ = 0.902± 0.0035
λs: 0.1, Γ = 0.943± 0.0046
λs: 1.0, Γ = 0.985± 0.0077
λs: 10.0, Γ = 0.996± 0.0029
λs: 0.0, Γ = 0.942± 0.0388
λs: 0.1, Γ = 0.962± 0.0019
λs: 1.0, Γ = 0.971± 0.0031
λs: 10.0, Γ = 0.992± 0.0021
λs: 0.0, Γ = 0.981± 0.0023
λs: 0.1, Γ = 0.955± 0.0283
λs: 1.0, Γ = 0.982± 0.0022
λs: 10.0, Γ = 0.993± 0.0016
λs: 0.0, Γ = 0.961± 0.0027
λs: 0.1, Γ = 0.961± 0.002
λs: 1.0, Γ = 0.962± 0.0011
λs: 10.0, Γ = 0.971± 0.0024
λs: 0.0, Γ = 0.975± 0.0016
λs: 0.1, Γ = 0.976± 0.0014
λs: 1.0, Γ = 0.975± 0.0021
λs: 10.0, Γ = 0.98± 0.0012

A.3.6 CLEAR


d = 128; τ = 0.1; λr = 0.0
d = 256; τ = 0.1; λr = 0.0
d = 128; τ = 0.1; λr = 0.1
d = 256; τ = 0.1; λr = 0.1
d = 128; τ = 0.1; λr = 1.0
d = 256; τ = 0.1; λr = 1.0
d = 128; τ = 0.1; λr = 10.0
d = 256; τ = 0.1; λr = 10.0
λs = 0.0
λs = 0.1
λs = 1.0
λs = 10.0
density
0.0
0.2
0.4
0.6
0.8
−5
0
5
10
aug. posterior log(KL div.)


$d = 128$; $\tau = 1.0$; $\lambda_r = 0.0$
$d = 256$; $\tau = 1.0$; $\lambda_r = 0.0$
$d = 128$; $\tau = 1.0$; $\lambda_r = 0.1$
$d = 256$; $\tau = 1.0$; $\lambda_r = 0.1$
$d = 128$; $\tau = 1.0$; $\lambda_r = 1.0$
$d = 256$; $\tau = 1.0$; $\lambda_r = 1.0$
$d = 128$; $\tau = 1.0$; $\lambda_r = 10.0$
$d = 256$; $\tau = 1.0$; $\lambda_r = 10.0$
$\lambda_s = 0.0$
$\lambda_s = 0.1$
$\lambda_s = 1.0$
$\lambda_s = 10.0$
density
0.0
0.2
0.4
0.6
0.8
1.0
−5
0
5
10
aug. posterior log(KL div.)


A = 1; τ = 0.1; d = 128, λr = 0.0
A = 1; τ = 0.1; d = 256, λr = 0.0
A = 1; τ = 0.1; d = 128, λr = 0.1
A = 1; τ = 0.1; d = 256, λr = 0.1
A = 1; τ = 0.1; d = 128, λr = 1.0
A = 1; τ = 0.1; d = 256, λr = 1.0
A = 1; τ = 0.1; d = 128, λr = 10.0
A = 1; τ = 0.1; d = 256, λr = 10.0
cumulative explained variance
singular value index
λs: 0.0, Γ = 0.913± 0.0134
λs: 0.1, Γ = 0.932± 0.0104
λs: 1.0, Γ = 0.973± 0.0038
λs: 10.0, Γ = 0.994± 0.0035
λs: 0.0, Γ = 0.917± 0.0075
λs: 0.1, Γ = 0.943± 0.0095
λs: 1.0, Γ = 0.973± 0.0059
λs: 10.0, Γ = 0.998± 0.0006
λs: 0.0, Γ = 0.9± 0.0044
λs: 0.1, Γ = 0.91± 0.0046
λs: 1.0, Γ = 0.941± 0.0042
λs: 10.0, Γ = 0.979± 0.007
λs: 0.0, Γ = 0.93± 0.0043
λs: 0.1, Γ = 0.923± 0.0035
λs: 1.0, Γ = 0.946± 0.0044
λs: 10.0, Γ = 0.972± 0.0033
λs: 0.0, Γ = 0.897± 0.0148
λs: 0.1, Γ = 0.895± 0.0129
λs: 1.0, Γ = 0.928± 0.0044
λs: 10.0, Γ = 0.96± 0.0073
λs: 0.0, Γ = 0.906± 0.0147
λs: 0.1, Γ = 0.899± 0.0038
λs: 1.0, Γ = 0.927± 0.0007
λs: 10.0, Γ = 0.949± 0.001
λs: 0.0, Γ = 0.9± 0.0179
λs: 0.1, Γ = 0.909± 0.0081
λs: 1.0, Γ = 0.911± 0.0122
λs: 10.0, Γ = 0.933± 0.0053
λs: 0.0, Γ = 0.93± 0.0018
λs: 0.1, Γ = 0.896± 0.02
λs: 1.0, Γ = 0.9± 0.0056
λs: 10.0, Γ = 0.93± 0.0034
γ95
± 1 stdev.


A = 1; τ = 1.0; d = 128, λr = 0.0
A = 1; τ = 1.0; d = 256, λr = 0.0
A = 1; τ = 1.0; d = 128, λr = 0.1
A = 1; τ = 1.0; d = 256, λr = 0.1
A = 1; τ = 1.0; d = 128, λr = 1.0
A = 1; τ = 1.0; d = 256, λr = 1.0
A = 1; τ = 1.0; d = 128, λr = 10.0
A = 1; τ = 1.0; d = 256, λr = 10.0
cumulative explained variance
singular value index
λs: 0.0, Γ = 0.918± 0.0062
λs: 0.1, Γ = 0.954± 0.008
λs: 1.0, Γ = 0.985± 0.0011
λs: 10.0, Γ = 0.996± 0.002
γ95
± 1 stdev.
λs: 0.0, Γ = 0.938± 0.0047
λs: 0.1, Γ = 0.962± 0.0055
λs: 1.0, Γ = 0.99± 0.0043
λs: 10.0, Γ = 0.998± 0.0006
λs: 0.0, Γ = 0.91± 0.0051
λs: 0.1, Γ = 0.92± 0.0043
λs: 1.0, Γ = 0.955± 0.001
λs: 10.0, Γ = 0.981± 0.003
λs: 0.0, Γ = 0.921± 0.0034
λs: 0.1, Γ = 0.932± 0.0021
λs: 1.0, Γ = 0.958± 0.0014
λs: 10.0, Γ = 0.98± 0.0031
λs: 0.0, Γ = 0.882± 0.0153
λs: 0.1, Γ = 0.9± 0.0121
λs: 1.0, Γ = 0.931± 0.0061
λs: 10.0, Γ = 0.968± 0.003
λs: 0.0, Γ = 0.898± 0.0091
λs: 0.1, Γ = 0.905± 0.0011
λs: 1.0, Γ = 0.939± 0.0025
λs: 10.0, Γ = 0.964± 0.002
λs: 0.0, Γ = 0.886± 0.0146
λs: 0.1, Γ = 0.904± 0.0147
λs: 1.0, Γ = 0.903± 0.0116
λs: 10.0, Γ = 0.935± 0.0026
λs: 0.0, Γ = 0.911± 0.0205
λs: 0.1, Γ = 0.91± 0.0161
λs: 1.0, Γ = 0.92± 0.0125
λs: 10.0, Γ = 0.938± 0.0031