# It depends: Incorporating correlations for joint aleatoric and epistemic uncertainties of high-dimensional output spaces

**Anonymous authors**
**Paper under double-blind review**

## Abstract

Uncertainty Quantification (UQ) plays a vital role in enhancing the reliability of deep learning model predictions, especially in scenarios with high-dimensional output spaces. This paper addresses the dual nature of uncertainty — aleatoric and epistemic — focusing on their joint integration in high-dimensional regression tasks. For example, in applications like medical image segmentation or restoration, aleatoric uncertainty captures inherent data noise, while epistemic uncertainty quantifies the model's confidence in unfamiliar conditions. Modeling both jointly enables more reliable predictions by reflecting both unavoidable variability and knowledge gaps, whereas modeling only one limits transparency and robustness. We propose a novel approach that approximates the resulting joint uncertainty using a low-rank plus diagonal covariance structure, capturing essential output correlations while avoiding the computational burdens of full covariance matrices. Unlike prior work, our method explicitly combines aleatoric and epistemic uncertainties into a unified second-order distribution that supports robust downstream analyses like sampling and log-likelihood evaluation. We further introduce stabilization strategies for efficient training and inference, achieving superior UQ in the tasks of image inpainting, colorization, and optical flow estimation. See Appendix A for notation used throughout.

## 1 Introduction

In high-risk settings such as AI-supported decision-making, UQ is an essential requirement to support a viable level of reliability and trustworthiness. For instance, AI tools in medicine and healthcare may benefit from a sound UQ (Hüllermeier & Waegeman, 2021; Tran et al., 2022; Band et al., 2022; Gruber et al., 2023; Lopez et al., 2023). Current Bayesian UQ methods neglect output correlations in high-dimensional settings, limiting reliability in tasks like image inpainting and optical flow (Kendall & Gal, 2017). A common strategy in this context is to distinguish between two types of uncertainty: aleatoric and epistemic. Aleatoric uncertainty is modeled as part of the head of a model, often using distributions like Gaussian for regression. It is generally considered to be irreducible by collecting more information, like increasing the size of the dataset, and therefore can be seen as inherent data noise. Contrary to that, epistemic uncertainty is reducible and a consequence of a lack of knowledge (Murphy, 2022). For instance, utilizing a large amount of data is expected to reduce epistemic uncertainty.[1] Epistemic uncertainty, due to its complexity, is commonly approximated by sampling from a proxy distribution of models (Hüllermeier & Waegeman, 2021).

Combining both in a single model usually results in a so-called second-order distribution (Bengs et al., 2023). On the one hand, it consists of a distribution over model weights capturing epistemic uncertainty. On the other hand, it models a distribution over plausible predictions representing aleatoric uncertainty. Sampling from the model weights and performing a transformation (forward pass) of the input data results in another distribution representing the aleatoric uncertainty. The shape of this second-order distribution limits further analysis, it is difficult to visualize, and it does not admit a closed-form solution of the marginal likelihood of a sample. Therefore, the second-order distribution is typically marginalized and approximated by a single distribution, representing the joint uncertainty.

[1] Here, we refer to the standard interpretation of aleatoric and epistemic uncertainty. However, this distinction is not always clear and subject to discussion (Hüllermeier & Waegeman, 2021; Gruber et al., 2023).

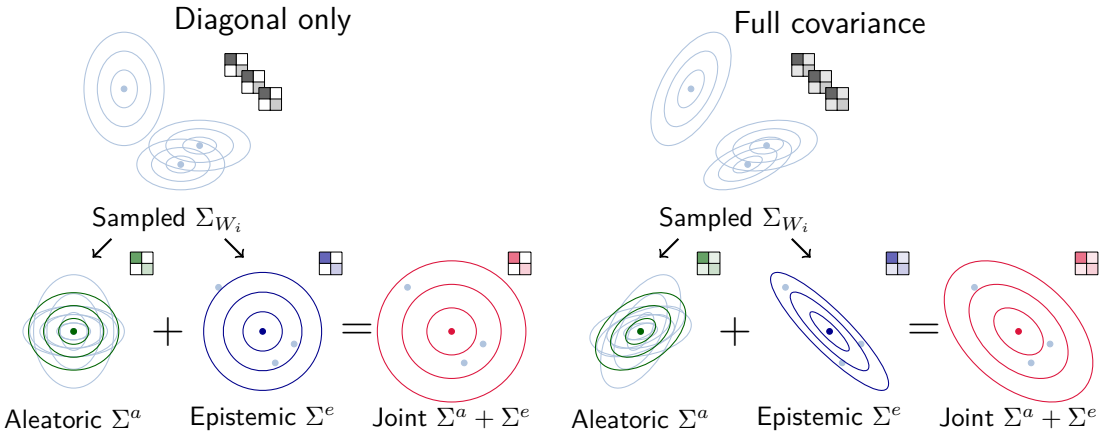


Figure 1: Visualization of covariance matrices for the 2D case. Three samples with corresponding means and covariances are depicted (light blue). The columns show the inferred aleatoric (green), epistemic (blue), and joint uncertainty (red), respectively. On the left, the covariance matrices are purely diagonal, limiting their representational power. To the right, the same matrices are depicted with non-diagonal values kept, allowing them to capture the overall uncertainty in greater detail.

Traditionally, uncertainties across outputs have been jointly represented without considering correlations between outputs (e.g., pixels), assuming independent factorized univariate Gaussian distributions. However, neglecting such correlations can limit a comprehensive understanding of uncertainty, especially in scenarios where dependencies between model outputs exist — such as in pixel-wise semantic segmentation (Monteiro et al., 2020), optical flow estimation, image inpainting, or graph node regression. Figure 1 illustrates the increased representational power of full covariance matrices (right) compared to diagonal ones (left). In both cases, samples from the weight space yield multiple predictions containing a mean and (co-)variance. The expected covariance represents the aleatoric component $\Sigma^a$, while the covariance of means contributes the epistemic component $\Sigma^e$. Their sum yields the joint covariance matrix $\Sigma = \Sigma^a + \Sigma^e$.

Incorporating these correlations efficiently, however, remains challenging. The number of pairwise correlations scales quadratically as $\mathcal{O}(S^2)$ with the number of outputs $S$, resulting in extremely large covariance matrices. This makes many standard operations - such as sampling and computing log-likelihoods - computationally infeasible for high-dimensional outputs.

Consequently, many prominent Bayesian methods focus on low-dimensional output spaces, where exact inference is tractable (Williams & Rasmussen, 1996; Rasmussen & Williams, 2006). Extending these methods to high-dimensional outputs, where considering correlations is essential, remains an open research challenge.

**Related Work** To estimate epistemic uncertainty, various Bayesian frameworks have been developed, including methods like stochastic variational inference (Blundell et al., 2015), Monte Carlo dropout (Gal & Ghahramani, 2016), deep ensembles (Lakshminarayanan et al., 2017), stochastic weight averaging (Maddox et al., 2019), or Laplace approximation (Daxberger et al., 2021). The modeling of heteroscedastic aleatoric uncertainty, where the model predicts all parameters of a target distribution and minimizes the corresponding log-likelihood, has also been well established (Nix & Weigend, 1994; Skafte et al., 2019; Stirn & Knowles, 2020; Seitzer et al., 2022). The latter three works further address the challenge of stabilizing training for such networks — a challenge that becomes more critical when modeling structured forms of uncertainty. Building upon these works, others have unified epistemic and aleatoric uncertainty in a single model (Kendall & Gal, 2017; Depeweg et al., 2018; Stirn et al., 2023; Immer et al., 2024; Valdenegro-Toro & Mori, 2022; Mucsányi et al., 2024; Chan et al., 2024; Wimmer et al., 2023). However, all aforementioned methods either evaluate their method only for prediction tasks with a single output value or approximate the marginalized likelihood as a factorized Gaussian, disregarding inter-pixel correlations.

This simplification neglects the inherent dependencies between output dimensions, often resulting in miscalibrated uncertainties and incoherent predictions in structured settings. Modeling these correlations is therefore crucial and has been explored in various applications, including localization (Russell & Reale, 2021), human pose estimation (Gundavarapu et al., 2019), pixel regression (Dorta et al., 2018a;b; Duff et al., 2023), multi class predictions (Willette et al., 2021), and segmentation (Monteiro et al., 2020).

Some approaches that predict full covariance matrices are limited to low dimensional model output spaces (Russell & Reale, 2021; Gundavarapu et al., 2019). Approaches for handling high-dimensional output spaces typically sparsify the covariance matrix. Yet, certain of these approaches can only model uncertainty in the local neighborhood using a band Cholesky parametrization (Dorta et al., 2018a;b; Duff et al., 2023). Several works (Salinas et al., 2019; Monteiro et al., 2020; Willette et al., 2021; Stoica & Babu, 2023) use a low-rank plus diagonal (LR+D) parametrization, which is capable of capturing global correlations. Nehme et al. (2024); Yair et al. (2024) learn the low-rank factors of aleatoric uncertainty directly without adding a diagonal and create a rank-deficient semi-definite covariance matrix. This may be sufficient for both sampling and analysis, but it does not provide the positive definiteness required for calculating the log-likelihood. Importantly, all these sparse solutions merely focus on aleatoric uncertainty.

One could argue that models implicitly capture correlations through inherent patterns, similar to latent variable models for aleatoric uncertainty (Depeweg et al., 2018) or deep ensembles for epistemic uncertainty (Lakshminarayanan et al., 2017). However, these methods do not explicitly represent or provide those correlations. Zepf et al. (2023) are getting close to this goal and combine aleatoric and epistemic uncertainty with a LR+D representation. However, by partially using the Maximum a posteriori (MAP) solution as a further approximation, they do not account for the influence of the model uncertainty on the estimation of the aleatoric uncertainty, leading overall to a worse uncertainty estimate. Furthermore, unlike our approach, they do not resolve the second-order distribution to provide a joint representation suitable for further analysis, such as log-likelihood calculation, and its usage is limited to consecutive sampling.

In conclusion, while significant advancements have been made in modeling covariances for uncertainty estimation, the existing approaches suffer from limitations such as local sparsification, inadequate joint representations, and neglect of epistemic uncertainty, indicating a need for further research to develop more comprehensive and globally accurate uncertainty estimation methods.

**Contribution** In this work, we propose joint modeling of aleatoric and epistemic uncertainty in a single framework. Unlike existing approaches that approximate the second-order distribution with factorized normals (neglecting output correlations), our method preserves crucial correlations while avoiding the prohibitive space and time costs of full covariance matrices in high-dimensional settings. Our low-rank plus diagonal (LR+D) covariance parameterization reduces memory from $\mathcal{O}(S^2)$ to $\mathcal{O}(SR)$ and reduces log-likelihood computation from $\mathcal{O}(S^3)$ to $\mathcal{O}(SR^2 + R^3)$ ($R \ll S$), enabling joint UQ on 65,000-dimensional outputs like CelebA. Furthermore, the low-rank eigenvectors extracted from the covariance provide interpretable insights into dominant modes of correlated uncertainty. We introduce stabilization techniques for robust training and showcase superior performance on high-dimensional tasks such as MNIST inpainting, CelebA colorization (Liu et al., 2015), and Flying Chairs optical flow (Dosovitskiy et al., 2015).

## 2 Method

We consider supervised learning tasks where we use a neural network $f_w : \mathcal{X} \to \mathcal{Y}$ with an input space $\mathcal{X} \subseteq \mathbb{R}^M$ and a high dimensional output space $\mathcal{Y} \subseteq \mathbb{R}^S$, where $S$ denoting the number of output units, e.g. pixels times the number of output channels. The weights $w \in W \subseteq \mathbb{R}^K$ of the neural network are interpreted probabilistically, meaning we aim to approximate the posterior distribution of the weights $p(w|\mathcal{D}) = p(\mathcal{D}|w)p(w)/p(\mathcal{D})$ allowing to model the epistemic uncertainty after observing a dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$ with $N$ samples. $p(\mathcal{D}|w)$ refers to the likelihood, that represents the epistemic part of the uncertainty, e.g. by modeling the standard deviation besides a mean value. $p(w)$ is the prior distribution over the weights. Computing $p(w|\mathcal{D})$ is generally not given in closed form and has to be approximated. The most popular methods include Monte Carlo Dropout (MCD), flavors of Stochastic Variational Inference (SVI), and more simple methods like Deep Ensembles (DEs). We highlight, that the proposed method is *agnostic* to the method, as long as we can sample from an approximated posterior distribution, i.e. a proxy distribution $q_\theta^*(w)$ over the weight space $W$, parametrized by $\theta$. To represent the joint uncertainty, for the prediction of unseen output $y$ given new input data $x$, one approximates the posterior predictive distribution $p(y|x, \mathcal{D}) = \int_W p(y|x, w)p(w|\mathcal{D})dw$ using $q_\theta^*$ instead of $p(w|\mathcal{D})$ in combination with Monte Carlo sampling. Specifically, we sample $T$ weights $w_i$ using $w_i \sim q_\theta^*$.

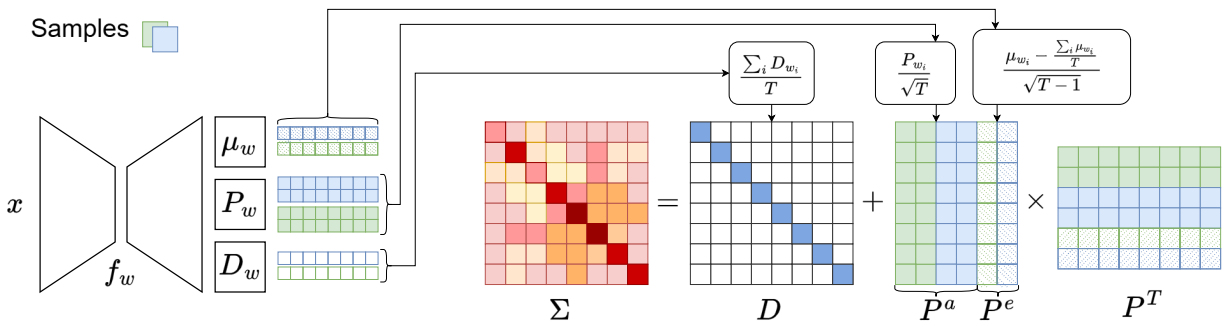


Figure 2: Construction of our LR+D matrix. A network predicts values $\mu_w$, $P_w$, and $D_w$ for two exemplarily sampled weights $w_i$, respectively (green and blue). By averaging the diagonals $D_{w_i}$ and concatenating low rank columns representing the epistemic $P^a$ and aleatoric $P^e$ uncertainty, we build the diagonal $D$ and the low-rank matrix $P$ as parts of our LR+D representation of $\Sigma$. See Section 2 for an in-depth explanation.

### 2.1 Modeling Sparse Joint Uncertainty

To deal with the high dimensionality of $\mathcal{Y}$, we define the likelihood to be a multivariate Gaussian distribution $p(y|x, w) = \mathcal{N}(\mu_W(x), \Sigma_W(x))$, where we keep the spatial complexity of the covariance matrix $\Sigma_W(x)$ low by constructing it in LR+D form. That is, we formulate it as a sum of small matrices, $\Sigma_W(x) = D_W(x) + P_W(x){P_W}^\top(x)$, with $D_W$ denoting a diagonal matrix of shape $S \times S$ and $P_W$ a tall matrix of shape $S \times R^W$. We choose a rank $R^W$ much lower than the number of outputs $R^W \ll S$, such that only the most important directions of the aleatoric covariance are covered. We further enforce $D_W$ to contain strictly positive diagonal entries and since $P_W P_W^\top$ is always symmetric, $\Sigma_W$ is always symmetric positive definite by construction and thus a valid covariance matrix. The ultimate goal of this work is to calculate an efficient yet representative representation of the posterior predictive distribution $p(y|x, \mathcal{D})$.

We start modeling the parameters of the posterior predictive distribution consisting of mean and covariance by using Monte Carlo integration to approximate the expected model output $\mathbb{E}[y|x, \mathcal{D}] \approx \mu(x)$. The empirical mean is given as $\mu(x) = \frac{1}{T}\sum_i^T \mu_{w_i}(x)$, where $T$ represents the number of weight samples drawn from $w_i \sim q_\theta^\star$. The joint covariance matrix can be split into epistemic and aleatoric uncertainty using the law of total variance as

$$\underbrace{\mathrm{Cov}\left[y|x, \mathcal{D}\right]}_{\underset{\text{joint uncertainty}}{\tilde{\approx}\,\Sigma(x)}} \approx \underbrace{\mathrm{Cov}_{q_\theta^*}\left[\mu_W(x)\right]}_{\underset{\text{epistemic uncertainty}}{\tilde{\approx}\,\Sigma^e(x)}} + \underbrace{\mathbb{E}_{q_\theta^*}\left[\Sigma_W(x)\right]}_{\underset{\text{aleatoric uncertainty}}{\tilde{\approx}\,\Sigma^a(x)}} . \tag{1}$$

This suggests that the mean of covariance matrices across forward pass samples captures aleatoric uncertainty, whereas the covariance of the means represents epistemic uncertainty. Unlike previous decompositions (Depeweg et al., 2018; Kendall & Gal, 2017) that use variances, our formulation employs covariance matrices, generalizing to multivariate variables. We provide a complete derivation of equation 1 in the Appendix F.4.

Our objective is to represent the joint uncertainty $\Sigma(x)$ in LR+D form as the sum of aleatoric and epistemic uncertainties,

$$D + PP^\top = (D^e + P^e {P^e}^\top) + (D^a + P^a {P^a}^\top), \tag{2}$$

where $D^a$, $D^e$, and $D$ are diagonal matrices and $P^a$, $P^e$, and $P$ low-rank matrices representing aleatoric, epistemic, and joint uncertainties, respectively. Then, $D = D^e + D^a$ and $P = \begin{bmatrix} P^a & P^e \end{bmatrix}$, where $\begin{bmatrix} & \end{bmatrix}$ denotes columnwise block concatenation. This expression allows us to conveniently represent both aleatoric and epistemic uncertainties in LR+D form, simplifying further analysis and computation. Figure 2 provides an intuitive illustration about the construction of our LR+D matrix components. Starting with $\Sigma^e$, we describe in detail the individual components of our LR+D representations in the following sections.

### 2.2 Epistemic Uncertainty

The epistemic uncertainty is estimated through the distribution over weights. To derive its covariance, we employ empirical sampling from the proxy distribution over model weights (e.g., SVI (Blundell et al., 2015)

or DE (Lakshminarayanan et al., 2017)) as follows:

$$\Sigma^e(x) = \frac{1}{T-1}\sum_i^T \left(\mu_{w_i}(x) - \mu(x)\right)\left(\mu_{w_i}(x) - \mu(x)\right)^\top \quad w_i \sim q_\theta^* \tag{3}$$

Our objective is to avoid the full covariance matrix and instead seek a representation in LR+D form.

To bring the approximated epistemic covariance matrix into LR+D form, we set the diagonal $D^e(x)$ to zero and rewrite the covariance matrix as $\Sigma^e(x) = P^e(x)P^e(x)^\top$, where $P^e(x) \in \mathbb{R}^{S\times R^e}$ has $R^e = T$ columns and is defined as

$$P^e(x) = \frac{1}{\sqrt{T-1}}\left[\mu_{w_1}(x) - \mu(x) \quad \dots \quad \mu_{w_T}(x) - \mu(x)\right]. \tag{4}$$

In high-dimensional scenarios, the number of samples is often significantly lower than the number of output dimensions ($T \ll S$), which renders the empirical covariance matrix $\Sigma^e$ low rank and therefore singular. Acquiring a sufficient number of samples to obtain a full-rank empirical estimate is typically infeasible due to time and space complexity constraints. In our low-rank-plus-diagonal parameterization, the epistemic part is captured purely by the low-rank term $P^e$ and the diagonal is zero ($D^e(x) = \mathbf{0}_S$), while the aleatoric diagonal has strictly positive entries ($D^a_{ii}(x) > 0$; cf. Eq. 6). Consequently, the total covariance $\Sigma(x)$ remains positive definite and thus invertible.

### 2.3 Aleatoric Uncertainty

Similar to epistemic uncertainty, the covariance matrix capturing aleatoric uncertainty $\Sigma^a(x)$ can be approximated through empirical sampling. We calculate the empirical mean of covariance matrix estimations over all sampled model weights via

$$\Sigma^a(x) = \frac{1}{T}\sum_i^T \Sigma_{w_i}(x) \quad w_i \sim q_\theta^*. \tag{5}$$

We here again intend to represent $\Sigma^a(x)$ in LR+D form.

To rewrite the covariance matrix containing the aleatoric uncertainty in LR+D representation, we reformulate $\Sigma^a(x) = D^a(x) + P^a(x)P^a(x)^\top$ using

$$D^a(x) = \frac{1}{T}\sum_i^T D_{w_i}(x) \tag{6}$$

$$P^a(x) = \frac{1}{\sqrt{T}}\left[P_{w_1}(x) \quad \dots \quad P_{w_T}(x)\right]. \tag{7}$$

This yields a $P^a \in \mathbb{R}^{S\times(T\cdot R^W)}$ with $R^a = T \cdot R^W$ columns. Although $R^a$ generally remains far below $S$, it can still become fairly large as the number of Monte Carlo samples $T$ increases. Thus, we suggest reducing the number of columns of $P(x)$.

### 2.4 Truncated Singular Value Decomposition Approximation

The full matrix $P(x)$, representing the joint aleatoric uncertainty, uses $R = T \times (R^W + 1) = R^a + R^e$ columns, where each forward pass $i = 1, \dots, T$ contributes one column from $\mu_{w_i}$ and $R^W$ columns from $P_{w_i}$. In general, increasing the number of forward passes $T$ yields a better uncertainty representation, as more samples enhance the empirical covariance estimate. However, in this naive representation, larger sample sizes also result in quadratic scaling of computational complexity. Hence, we suggest further approximations to cope with moderately high sample sizes.

Assuming that samples are often correlated and exhibit dominant directions of variance, we propose to reduce the dimensionality of $P(x)$ with truncated Singular Value Decomposition (SVD). Keeping only the most informative columns of $P(x)$ will improve the efficiency of further computations without losing much information. However, the calculation of SVD comes with its own computational complexity that has to be

taken into account. Specifically, we decompose the matrix $P$ as $P^\top = U\Psi V^\top$, where $U$ and $V$ are orthogonal matrices, and $\Psi$ is a diagonal matrix containing the singular values in non-decreasing order $\Psi_{1,1} \leq ... \leq \Psi_{S,S}$. Subsequently, we define the matrix $\tilde{P} = V\Psi$ and rewrite the matrix product as $\Sigma = PP^\top = \tilde{P}\tilde{P}^\top$. To reduce dimensionality, we discard the smallest singular values and their associated columns in $V$. However, we keep the univariate variance parts of these dropped columns by transferring them to a new diagonal matrix $\hat{D}$. Hence, the approximated matrix $\hat{\Sigma} = \hat{D} + \hat{P}\hat{P}^\top$ keeps all independent variance and the most important covariances of $\Sigma$. If we keep the $\hat{R}$ largest singular values, the components of $\hat{\Sigma}$ are

$$\hat{P} = \begin{bmatrix} V_{R-\hat{R}} \cdot \Psi_{R-\hat{R},R-\hat{R}} & ... & V_R \cdot \Psi_{R,R} \end{bmatrix} \tag{8}$$

and

$$\hat{D}_{ii} = D_{ii} + \sum_{j=1}^{R-\hat{R}-1} V_{ij}^2 \cdot \Psi_{j,j}^2 \,. \tag{9}$$

The number of columns $\hat{R}$ to retain is determined by reconstruction error and downstream task performance, as validated in our ablation studies (Sec. 3.5). The aforementioned approach enables us to effectively represent joint uncertainty in the LR+D form. For analysis purposes, SVD can also be applied to both of the low rank summands of $P$ namely the aleatoric part $P^a$ and the epistemic part $P^e$ separately. This allows to visualize the most important directions of variance of both components. See Appendix E for pseudocode.

### 2.5 Stability Techniques

As the size of the covariance matrix increases, its condition number $\kappa$ tends to grow, making it more susceptible to numerical instability - particularly during matrix inversion required for log-likelihood calculations. Superior LR+D-parametrized covariance matrices are no exception to this phenomenon. However, for LR+D-parametrized covariance matrices, not only the condition number of the covariance matrix but also that of its internally used capacitance matrix is relevant. This matrix is of size $R \times R$ and therefore much smaller than the full covariance matrix, as $R \ll S$. Instead of inverting the full covariance matrix, the capacitance matrix is inverted internally. The capacitance matrix is given by $C = I_R + P^\top D^{-1} P$ where $I_R$ is an identity matrix.

While the condition number of the capacitance matrix $\kappa(C)$ can be calculated with reasonable complexity, the condition number of the covariance matrix $\kappa(\Sigma)$ cannot be obtained in reasonable time due to the cubic scaling of, for example, SVD. However, the following bounds for $\kappa(\Sigma)$ can be obtained using Weyl's inequality:

$$\frac{\lambda_S(PP^\top) + \lambda_1(D)}{\lambda_{R+1}(D)} \leq \kappa(\Sigma) \leq \frac{\lambda_S(PP^\top) + \lambda_S(D)}{\lambda_1(D)}. \tag{10}$$

where $\lambda_S(PP^\top)$ is the largest eigenvalue of $PP^T$, given by $\lambda_S(PP^\top) = \Psi_S^2$. The smallest eigenvalue of the diagonal matrix $D$ is given by its smallest entry $\lambda_1(D)$, while the largest eigenvalue $\lambda_S(D)$ corresponds to the largest entry of $D$. Further details can be found in the Appendix F.5.

These condition number values and bounds can be used to monitor and mitigate numerical instabilities.

## 3 Experiments

**Proposed Method** We empirically evaluate our method of joint aleatoric and epistemic uncertainty modeling using our LR+D representation in several experiments. In all experiments, we use variants of the U-Net (Ronneberger et al., 2015) architecture. We equip the U-Net with probabilistic outputs via three established Bayesian approximation methods: 1) adding dropout, which we use for MCD (Gal & Ghahramani, 2016), 2) DE (Lakshminarayanan et al., 2017), or 3) by using variational convolutional layers for SVI (Blundell et al., 2015), to estimate a distribution over model weights which estimates epistemic uncertainty. However, we note that our approach is compatible with any Bayesian method as long as it is computationally feasible for considered models. We use a combined model to jointly predict mean and uncertainty because it provides a cleaner, more unified architecture with shared feature learning and simpler training, which can — but does not always — lead to better results. Details and comparisons to variants

with separated mean and uncertainty estimation can be found in Appendix D.3. Further reproducibility details in Appendix B. An anonymized implementation and scripts to reproduce all experiments are available at `https://anonymous.4open.science/r/corr-joint-ae`.

**Datasets and Tasks** We evaluate our method in different settings on the MNIST, CelebA, and Flying Chairs datasets for the tasks of inpainting, colorization, and optical flow.

We train a reconstruction model to inpaint distorted handwritten digits from the MNIST dataset. For the inpainting task, we mask out 5/7 of the image area. We use the official test set and split the training set into 50,000 train and 10,000 validation images.

To evaluate performance on optical flow estimation, we use the Flying Chairs (Dosovitskiy et al., 2015) dataset. This dataset is resized to 192 x 256 and split into 18,297/2,287/2,288 training/validation/test images. We provide visualizations of the predictions as part of the Appendix D.1.

To evaluate our method on facial images, we use the CelebA-HQ dataset, keeping the original splits from celeba (Liu et al., 2015). The original split contains 24,183 images for training, 2,993 for validation, and 2,824 for testing (image size $256 \times 256$). We study two tasks on this dataset: colorization and inpainting.

**Baselines** We evaluate non-Bayesian models alongside various Bayesian methods. In addition to the diagonal (D) covariance matrix approach from (Kendall & Gal, 2017), we introduce a low-rank plus diagonal (LR+D)-parameterized distribution that captures richer output correlations, representing a substantial advancement in uncertainty modeling.

As a further baseline, we follow the approach by Zepf et al. (2023) and approximate the aleatoric uncertainty term of Equation 1 to prevent sampling aleatoric $P$ matrices. This reduces the number of resulting columns from $T \times (R + 1)$ to $T + R$. It is achieved by approximating the expectation of the aleatoric uncertainty $\Sigma^a$ term with the aleatoric covariance prediction of the model with the expected weights:

$$\Sigma^a = \mathbb{E}_{q_\theta^*}\left[\Sigma_W(x)\right] \approx \Sigma_{\mathbb{E}_{q^*}[W]}(x)$$

To compute this term, we require the expected weights of the Bayesian models to be well-defined. For MCD, this is done by turning dropout off and rescaling the activations accordingly. For SVI, where the weights follow Gaussian distributions, the expected weights are simply the means of the Gaussian distributions. For DE, we are unable to define expected weights, hence this approximation is not evaluated in this case. Note that Zepf et al. refer to this approach as MAP solution, which coincides with the *expected weights* solution if the weight uncertainty is modeled with symmetrical unimodal distributions like Gaussians as commonly used by SVI and Laplace Approximation (LA). Furthermore, Zepf et al. do not provide a joint representation, and log-likelihood calculation is only possible using a combination of our methods.

**Model Specifics** Finally, we evaluate our joint LR+D parametrization in combination with all three Bayesian methods. For this case, we let the model predict a matrix $P_W \in \mathbb{R}^{S \times R^W}$ of rank $R^W = 8$ and for epistemic models, we draw $T = 64$ samples. The predictions are multivariate Normal distributions, represented by their LR+D parametrization. Those predictions are joined to a single, LR+D parametrized distribution. For the full joint uncertainty LR+D model, this yields a joint $P$ matrix with $R = T \times (R^W + 1) = 576$ columns, which we optionally compress down with TSVD while keeping the diagonal variance of the dropped columns as described in Section 2. For the expected weights baseline, we perform an additional forward pass using the expected weights and concatenate the aleatoric and epistemic columns, which leads to $R = 72$ in total. All models are trained for the same amount of steps. Prediction errors in Appendix D.2.

**Hyperparameters for Stable Training** Implementing and optimizing heteroscedastic losses can be challenging due to training instability, overfitting, and balancing between prediction accuracy and uncertainty estimation. To address these issues, various stabilization techniques have been proposed, such as architectural decoupling and gradient reweighting based on predicted mean and variance (Stirn et al., 2023; Immer et al., 2024; Seitzer et al., 2022). While some of these approaches may generalize to LR+D-parametrized normal distributions, we opted for a weighted loss and a streamlined architecture with multiple output channels.

| Parameters | Epistemic | Method | MNIST Inpainting ×10 | CelebA Inpainting ×100 | CelebA Colorization ×1000 | Flying Chairs Optical Flow ×100 |
|---|---|---|---|---|---|---|
| D | ✗ | | -3550 ± 22033 | -471 ± 558 | 240 ± 519 | -231 ± 110 |
| LR+D | ✗ | | -2445 ± 14022 | -513 ± 1104 | 495 ± 267 | -184 ± 119 |
| D | MCD | | 72 ± 1004 | -341 ± 495 | 324 ± 249 | -224 ± 85 |
| | SVI | | -159 ± 3636 | -439 ± 621 | 340 ± 254 | -227 ± 104 |
| | DE | | 81 ± 873 | -249 ± 403 | 374 ± 159 | -213 ± 72 |
| LR+D | MCD | $\mathbb{E}[w]$ | -8 ± 1825 | -120 ± 415 | 565 ± 203 | -170 ± 100 |
| | SVI | $\mathbb{E}[w]$ | 41 ± 1383 | -243 ± 599 | 558 ± 222 | -164 ± 103 |
| | MCD | (ours) | 98 ± 439 | 40 ± 177 | 627 ± 67 | -158 ± 70 |
| | SVI | (ours) | 91 ± 573 | -29 ± 263 | **641** ± 66 | **-149** ± 84 |
| | DE | (ours) | **107** ± 275 | **54** ± 170 | 631 ± 60 | -157 ± 68 |

Table 1: Quantitative Results. We evaluate the test log-likelihoods (TLLs) (base 10) of model predictions across various dataset–task combinations, including their test-set variability (standard deviation). Higher values correspond to greater likelihood and therefore better predictive performance. Our approach is assessed on four tasks: inpainting, colorization, and optical flow estimation. Likelihoods scale linearly with output dimensionality and, in the case of masking, are computed only over masked regions. Results are reported for both Bayesian (MCD, SVI, DE) and non-Bayesian (✗) networks with diagonal (D) and low-rank plus diagonal (LR+D) covariance parameterizations. For the combination of LR+D and Bayesian methods, we additionally report outcomes using the expected weights $\mathbb{E}[w]$ approximation. Overall, incorporating epistemic uncertainty and the LR+D representation increases test log-likelihood (TLL), indicating improved predictive fidelity. The relatively high variance in test log-likelihood on MNIST can be attributed to the discrete and multimodal nature of the task: predictions that capture the correct digit mode yield substantially higher likelihood than predictions corresponding to an incorrect digit, leading to large per-sample differences in log-likelihood.

These channels are interpreted as the predictive mean $\mu(x)$, diagonal $D(x)$, and factor $P(x)$. Our loss function $\mathcal{L} = \mathcal{L}_I + \alpha\,\mathcal{L}_{\text{lrd}}$ primarily consists of the log-likelihoods of both the LR+D-parametrized and univariate normal distributions, with the latter effectively reducing to a Mean-Squared-Error loss scaled by a constant.

$$\mathcal{L}_{\text{lrd}} = \sum_i \log \mathcal{N}\left(y_i \mid \mu(x_i),\, D(x_i) + P(x_i)P^\top(x_i)\right) \tag{11}$$

$$\mathcal{L}_I = \sum_i \log \mathcal{N}\left(y_i \mid \mu(x_i),\, I\right) \tag{12}$$

The other problem can arise when the predicted variance of parts of the prediction are very low, which can lead to numerical instabilities. To ensure strict positivity and avoid very small variances, we regularize $\Sigma$ and set $D(x) = \epsilon + \exp(Z(x))$. Therefore, we choose $\epsilon$ empirically. The minimal entry of $D(x)$ is a minimal bound for the lowest eigenvalue of $\Sigma(x)$. $\lambda_1(D) \leq \lambda_1(\Sigma)$. And hence has an effect on the condition number $\kappa(\Sigma)$.

### 3.1 Main Results - Comparison of the Fit of Predictive Distributions

**Quantitative Results** To evaluate the uncertainty estimate, we use the test log-likelihood (TLL), which measures how well a model predicts the observed data while accounting for uncertainty. Higher values indicate that the observed outcomes are more consistent with the model's predictive distribution. Quantitatively, we find that modeling epistemic uncertainty improves the likelihood of unseen test sample predictions, as shown in Table 1. This improvement holds for both modeling the diagonal and modeling the LR+D across all experiments.

Additionally, incorporating covariances via our LR+D approach further improves the likelihood of unseen test sample predictions across all experiments. The effectiveness of the expected weights $\mathbb{E}[W]$ approximation for aleatoric uncertainty $\Sigma^a$ is inconsistent and, for some tasks — such as MNIST — even performs worse than simpler approaches that do not model epistemic uncertainty. In contrast, our method — combining both uncertainties into a joint multivariate representation and leveraging SVD — consistently outperforms all tested Bayesian methods in every experiment.

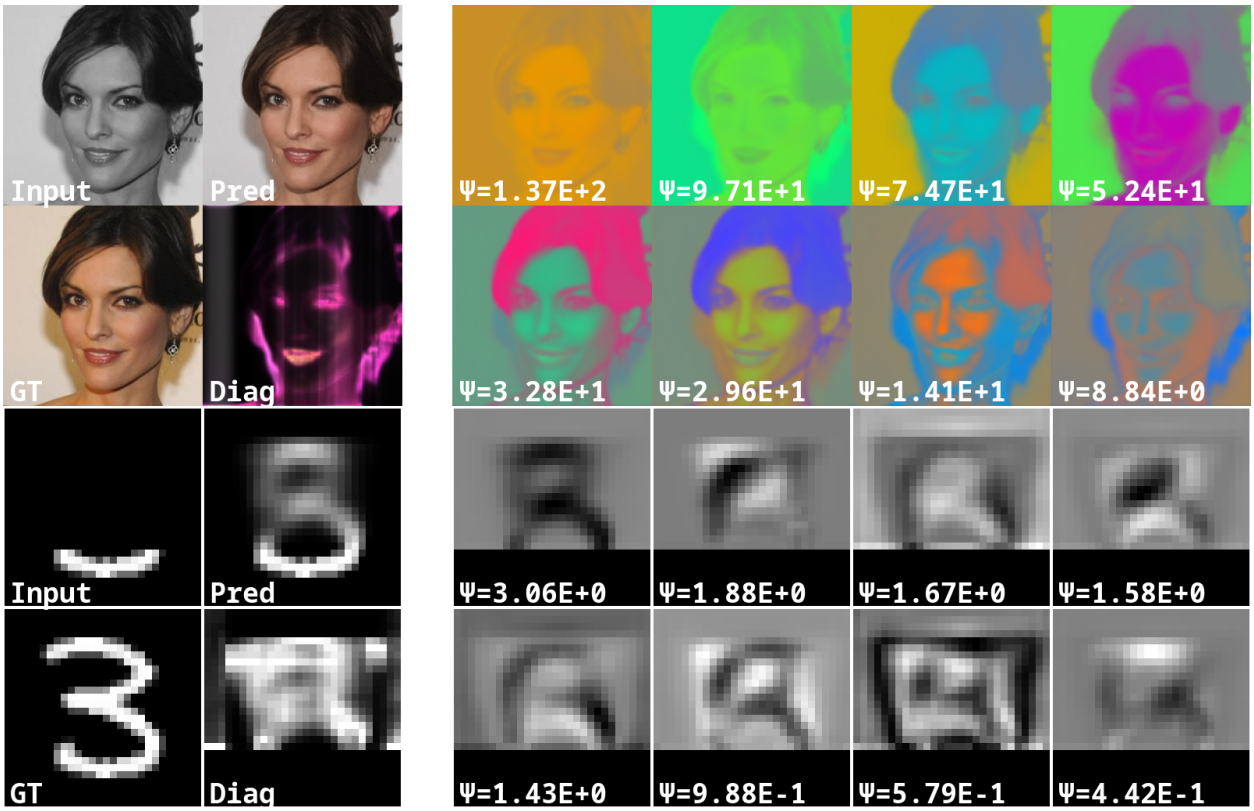


Figure 3: Qualitative Results. Random samples from the test sets depict the input, prediction, ground truth, and parameters of the predictive distribution. The top rows show colorization on CelebA images, while the bottom rows display inpainting of MNIST digits. Our model predicts a mean (Pred), the parameter $D$ (Diag), and a low-rank matrix $P$. In both cases, the predicted joint low-rank matrix $P$ is reduced to the 64 most significant directions (columns) based on their respective eigenvalues. The 8 images in columns 3-6 visualize the 8 most important directions with a random orientation in descending order of the associated eigenvalues. We observe that these columns focus on uncertainty in specific image areas or colors. Additionally, the singular values $\Psi$ measure the importance of the associated direction. For more qualitative results of all datasets and Bayesian methods, please see the Appendix Figures 7-11.

**Qualitative Results** Figure 3 and Appendix Figures 7-10 (Datasets), and 11 (Bayesian Methods) provide qualitative results, where we show how the 8 most important columns in our joint low-rank matrix $P$ describe the areas of correlated uncertainty. For example, in the CelebA inpainting task (top), the visualized $P$ matrix reveals that the first two eigenvectors represent global color shifts across the entire image: the first corresponds to a color axis between orange and blue (complementary colors), and the second to a purple–green axis (also complementary). The third and fourth eigenvectors capture contrast between the foreground and background. The fifth and sixth focus on variations in hair and eye color. Additionally, the singular values $\Psi$ offer insight into the relative importance of these correlations. Visualization of the eigenvectors is only possible with our method, which includes the covariance terms; hence, allowing the identification of image regions with correlated uncertainty. The Parameter D (Diag) captures additional uncertainty, which could not be captured by the Low Rank Covariance Matrix created by $PP^\top$. In summary, these qualitative results can help to intuitively describe the underlying relations of uncertainty on an image level.

### 3.2 Complexity of Covariance Parametrizations

Figure 4 presents the memory (left) and time (right) requirements for computing the log-likelihood of different covariance parameterizations: sparse options like diagonal (D) and low-rank plus diagonal (LR+D), as well as full covariance $\Sigma$ using both naive and lower-triangular parameterizations. The complexity is shown as a function of the number of variables in the covariance matrix, with specific points marking the number of

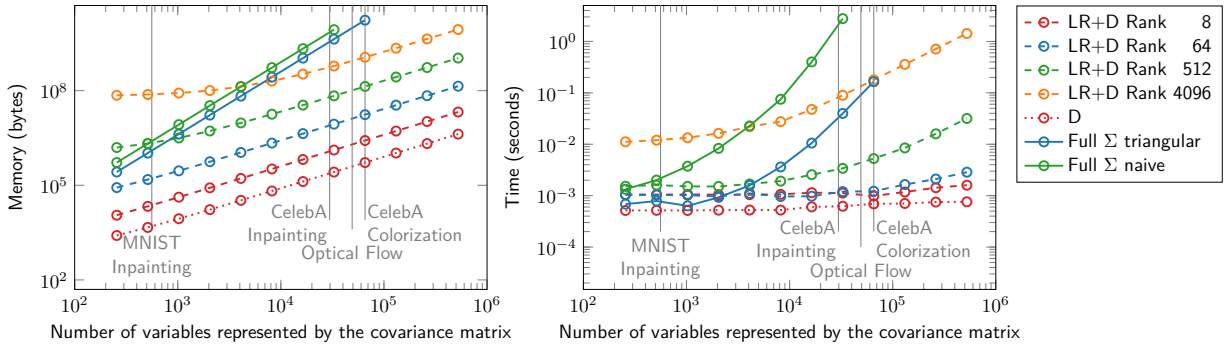


Figure 4: Empirical memory (left) and time (right, avg. of 100 calculations) for log-likelihood across covariance parameterizations. Memory scales linearly for LR+D, quadratically for full $\Sigma$ (until GPU limit). Random parameters used independent of datasets. LR+D handles larger matrices.

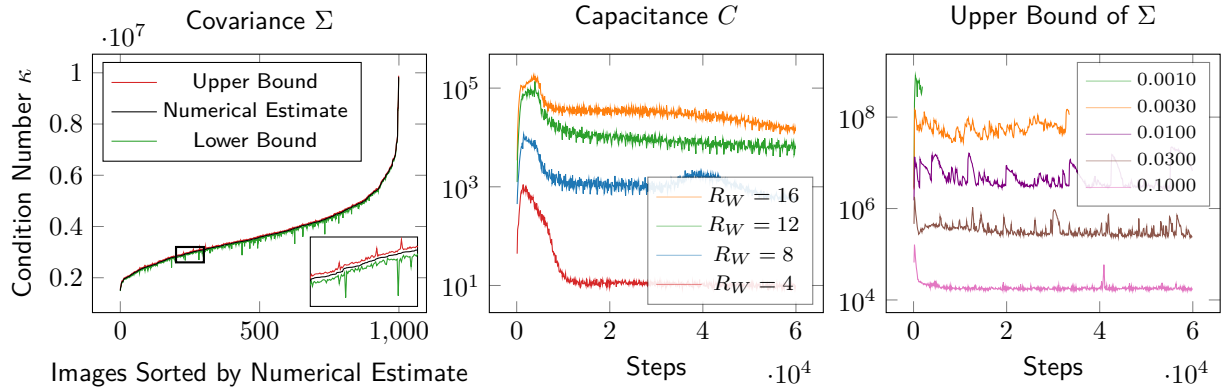


Figure 5: Condition number distributions for $\Sigma$ in MNIST inpainting (left), and effects of model parameters on condition numbers and training stability in CelebA colorization (center and right). The left panel displays condition numbers for samples ordered by their upper bound, showing exact values tightly between the computed upper and lower bounds, which validates the numerical estimates. The center panel plots the numerical condition number estimate of the capacitance matrix over training steps for varying ranks of the low-rank matrix $P_W$; higher ranks increase condition numbers and cause training instability. The right panel shows the upper bound on the covariance matrix condition number versus the minimum diagonal entry $\epsilon$ over training steps; smaller $\epsilon$ worsens conditioning and raises numerical failure risk.

variables for each dataset-task combination. Random numbers were used for all parameters, independent of datasets, for complexity evaluation.

For LR+D, we evaluate various numbers of columns $R$ in the low-rank matrix $P$. We limit our analysis to sizes that fit within a single 48GB GPU. As seen in the figure, the LR+D parameterization (with 64 columns) is significantly more efficient than the naive full covariance, both with respect to memory and time, for all datasets. In larger datasets like CelebA and Flying Chairs, the full covariance matrix approaches the GPU memory limit, even without batching or storing the model and its gradients. Theoretical details on the computational complexity can be found in the Appendix C.

### 3.3 Numerical Validation of Covariance Condition Number Bounds

Figure 5 (left) illustrates condition numbers of the covariance matrix $\kappa(\Sigma)$ for MNIST inpainting. The red and green curves represent the upper and lower bounds on the condition number, respectively, while the black curve shows the numerical estimate computed in double precision (float64) for 1000 images, sorted by their estimated condition number. Despite the inherent rounding issues in finite-precision arithmetic, the exact condition number remains tightly enclosed between the bounds, as seen particularly in the magnified section.

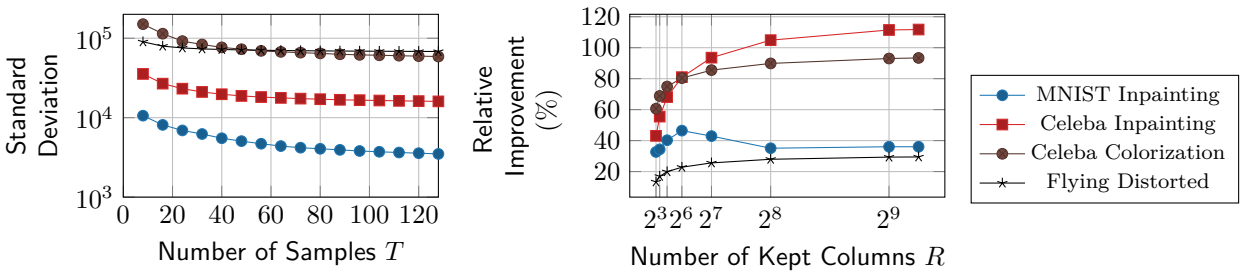


Figure 6: Impact of Sample Count and Low-Rank Truncation on the Stability and Accuracy of LR+D-Parametrized Covariance Approximations. Left: Standard deviation of test log-likelihood decreases with more samples, indicating improved consistency. Right: Relative TLL improvement (%) over diagonal covariance versus retained columns in low-rank approximation. More columns generally improve results, but overparameterization can harm performance in low-dimensional cases like MNIST.

### 3.4 Effect of Training Hyperparameters on Conditioning and Stability

Training low-rank plus diagonal (LR+D) parameterized distributions can be numerically unstable for certain hyperparameter settings. Instabilities arise during Cholesky decomposition and inversion of the capacitance matrix for log-likelihood computation.

Figure 5 (center and right) illustrates how two key parameters affect training stability. The center plot shows the estimated condition number of the capacitance matrix over time for the CelebA colorization task. Increasing the number of columns in the low-rank matrix $P_W$ raises this condition number. Models with 20 or more columns consistently crash because of exploding gradients early in training across datasets. Notably, the condition number peaks early in training and decreases after 10,000 steps, suggesting early instability.

The right plot shows the upper bound of the condition number of the full covariance matrix in MNIST inpainting when varying the minimum diagonal value added to ensure positive definiteness. Smaller values increase the condition number and often cause training failures—values below 0.001 consistently led to crashes across five random seeds. Low diagonal values are often predicted in background regions near image boundaries (see Appendix Figure 13), which are typically black and certain in MNIST. This results in very low predicted uncertainty, small variances, and thus poorly conditioned covariance matrices.

These observations highlight a key trade-off: while small diagonal offsets enable the model to express high certainty, they also increase numerical instability. To prevent this, it is crucial to enforce a small positive $\epsilon$ on the diagonal, ensuring the smallest eigenvalue remains bounded away from zero.

### 3.5 Effect of Evaluation Hyperparameters

For a comprehensive evaluation of our uncertainty framework, we conduct multiple ablations to identify which factors most influence model performance. Specifically, we study the effects of sample count and dimensionality reduction in the LR+D-parametrized covariance, which sparsely models joint uncertainty.

In Figure 6, the left plot shows the average standard deviation of the TLL across the dataset as the number of samples T used to estimate the joint multivariate normal distribution increases. We observe that higher sample counts consistently reduce TLL variability, resulting in more stable and reliable predictions. However, this increases computation for both forward passes and LR+D rank.

In the right plot, we fix the sample count at T=64 and analyze the impact of truncating the number of retained columns in the LR+D covariance structure via TSVD. The y-axis shows the relative improvement in TLL (in %) compared to a purely diagonal covariance. Retaining more columns generally improves performance by capturing a richer covariance structure. However, for low-dimensional outputs such as MNIST Inpainting (560 predicted pixels), this benefit saturates and even reverses as the number of retained columns approaches the number of output dimensions. This indicates that overly expressive low-rank approximations can lead to overfitting or instability. Notably, across the board, any compromise involving a limited number of columns still yields significant improvements over a purely diagonal covariance.

We further ablate various design choices in Appendix D.3 and compare against Nehme et al. (2024) (Tables 5 and 6). First, as shown in Table 7, incorporating the diagonal update defined in Equation 9 leads to more robust predictions. Next, we investigate the impact of the number of columns $R^W$ in $P_W$, which are directly predicted by the model weights, while keeping the number of columns $R$ retained after TSVD fixed (Table 8). We find that the optimal value of $R^W$ is task-dependent. Nevertheless, for simplicity, stability, and computational efficiency, we adopt a fixed value of $R^W = 8$ across all tasks in our remaining experiments. Furthermore, we provide a comprehensive overview of different design choice combinations in the full ablation Table 9. Finally, we analyze the distribution of eigenvalues of $PP^\top$ and $D$ under different approximations (Figure 12), and examine how the hyperparameter $\epsilon$ affects the MNIST inpainting predictions (Figure 13).

## 4 Discussion

**Conclusion** In this work, we have explored the dual nature of uncertainties — aleatoric and epistemic — and their integration in high-dimensional regression tasks. We proposed a novel method that employs a low-rank plus diagonal covariance matrix to approximate joint uncertainty, effectively preserving vital output correlations and significantly reducing the computational demands that are inherent to full covariance matrix representation. Our approach lowers memory usage and improves the efficiency of both sampling and log-likelihood calculations. To address stability during training, we incorporate tools to monitor and regularize the condition number of both the covariance matrix and the internally used capacitance matrix.

Empirically, our approach outperforms the commonly used factorized Gaussian representation. It exhibits a lower negative log-likelihood and produces more reliable uncertainty estimates, demonstrating clear advantages in uncertainty modeling. Beyond quantitative gains, the low-rank structure also exposes interpretable patterns in correlated uncertainties — offering insights into how uncertainty propagates across high-dimensional outputs. These results highlight the method's effectiveness and interpretability in capturing and quantifying uncertainty in large-scale regression tasks.

**Limitations** Our method conceptually extends to any Bayesian framework; however, for simplicity and computational reasons, we restrict our evaluation to using Monte Carlo Dropout, Stochastic Variational Inference and Deep Ensemble. Further investigations into other Bayesian inference techniques should determine their empirical applicability. We expect that more advanced concepts will lead to better overall uncertainty estimation.

Beyond vision, this parameterization is directly applicable to other multivariate regression settings with structured outputs, such as graph-based prediction of node or edge attributes, multivariate time series forecasting of correlated signals (for example, energy demand across regions or multiple physiological channels), and multi-output tabular or scientific regression where several related physical or environmental quantities are predicted jointly. While directly applicable, we lack empirical evaluation on these domains, which we leave for future work.

The method is flexible with regard to the choice in number of columns utilized in the LR+D-parameterization of the covariance matrix. Increasing the number of columns generally leads to improved uncertainty estimation but comes at the cost of additional computational complexity and potential training instability.

Training difficulty also arises from numerical sensitivity associated with the covariance matrix and the internally used capacitance matrix. Specifically, both matrices can become ill-conditioned, resulting in numerical errors, particularly during Cholesky factorization. Monitoring and managing the condition number of these matrices is essential to ensure convergence.

Finally, our method builds upon the assumption that uncertainties in output can be modeled by a single multivariate Gaussian, even though this approximation is often used in the literature Kendall & Gal (2017); Monteiro et al. (2020); Duff et al. (2023). However, multivariate Gaussians may not be a suitable approximation for every task, for example, for uncertainties in translation or rotation in images. Exploring epistemic uncertainty under different distributions is a highly promising research question.

By more expressive approximation of the posterior predictive distribution than traditional joint distributions, our method enhances both the reliability and explainability of predictions from deep learning models.

## References


Neil Band, Tim GJ Rudner, Qixuan Feng, Angelos Filos, Zachary Nado, Michael W Dusenberry, Ghassen Jerfel, Dustin Tran, and Yarin Gal. Benchmarking bayesian deep learning on diabetic retinopathy detection tasks. *arXiv preprint arXiv:2211.12717*, 2022.

Viktor Bengs, Eyke Hüllermeier, and Willem Waegeman. On second-order scoring rules for epistemic uncertainty quantification. In *International Conference on Machine Learning*, pp. 2078–2091. PMLR, 2023.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pp. 1613–1622. PMLR, 2015.

Matthew Albert Chan, Maria J. Molina, and Christopher Metzler. Estimating epistemic and aleatoric uncertainty with a single model. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. Laplace redux-effortless bayesian deep learning. *Advances in Neural Information Processing Systems*, 34:20089–20103, 2021.

Stefan Depeweg, Jose-Miguel Hernandez-Lobato, Finale Doshi-Velez, and Steffen Udluft. Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning. In *International conference on machine learning*, pp. 1184–1193. PMLR, 2018.

Garoe Dorta, Sara Vicente, Lourdes Agapito, Neill DF Campbell, and Ivor Simpson. Structured uncertainty prediction networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5477–5485, 2018a.

Garoe Dorta, Sara Vicente, Lourdes Agapito, Neill DF Campbell, and Ivor Simpson. Training vaes under structured residuals. *arXiv preprint arXiv:1804.01050*, 2018b.

A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

Margaret AG Duff, Ivor JA Simpson, Matthias Joachim Ehrhardt, and Neill DF Campbell. Vaes with structured image covariance applied to compressed sensing mri. *Physics in Medicine & Biology*, 68(16): 165008, 2023.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

Cornelia Gruber, Patrick Oliver Schenk, Malte Schierholz, Frauke Kreuter, and Göran Kauermann. Sources of uncertainty in machine learning–a statisticians' view. *arXiv preprint arXiv:2305.16703*, 2023.

Nitesh B Gundavarapu, Divyansh Srivastava, Rahul Mitra, Abhishek Sharma, and Arjun Jain. Structured aleatoric uncertainty in human pose estimation. In *CVPR Workshops*, volume 2, pp. 2, 2019.

Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine learning*, 110(3):457–506, 2021.

Alexander Immer, Emanuele Palumbo, Alexander Marx, and Julia Vogt. Effective bayesian heteroscedastic regression with deep neural networks. *Advances in Neural Information Processing Systems*, 36, 2024.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015.

L Lopez, Tim GJ Rudner, and Farah E Shamout. Informative priors improve the reliability of multimodal clinical data classification. *arXiv preprint arXiv:2312.00794*, 2023.

Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. *Advances in neural information processing systems*, 32, 2019.

Miguel Monteiro, Loïc Le Folgoc, Daniel Coelho de Castro, Nick Pawlowski, Bernardo Marques, Konstantinos Kamnitsas, Mark van der Wilk, and Ben Glocker. Stochastic segmentation networks: Modelling spatially correlated aleatoric uncertainty. *Advances in neural information processing systems*, 33:12756–12767, 2020.

Bálint Mucsányi, Michael Kirchhof, and Seong Joon Oh. Benchmarking uncertainty disentanglement: Specialized uncertainties for specialized tasks. *arXiv preprint arXiv:2402.19460*, 2024.

Kevin P Murphy. *Probabilistic machine learning: an introduction*. MIT press, 2022.

Elias Nehme, Omer Yair, and Tomer Michaeli. Uncertainty quantification via neural posterior principal components. *Advances in Neural Information Processing Systems*, 36, 2024.

David A Nix and Andreas S Weigend. Estimating the mean and variance of the target probability distribution. In *Proceedings of 1994 ieee international conference on neural networks (ICNN'94)*, volume 1, pp. 55–60. IEEE, 1994.

Carl Edward Rasmussen and Christopher KI Williams. *Gaussian Processes for Machine Learning*, volume 1. MIT press, 2006.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234–241. Springer, 2015.

Rebecca L Russell and Christopher Reale. Multivariate uncertainty in deep learning. *IEEE Transactions on Neural Networks and Learning Systems*, 33(12):7937–7943, 2021.

David Salinas, Michael Bohlke-Schneider, Laurent Callot, Roberto Medico, and Jan Gasthaus. High-dimensional multivariate forecasting with low-rank gaussian copula processes. *Advances in neural information processing systems*, 32, 2019.

Maximilian Seitzer, Arash Tavakoli, Dimitrije Antic, and Georg Martius. On the pitfalls of heteroscedastic uncertainty estimation with probabilistic neural networks. *arXiv preprint arXiv:2203.09168*, 2022.

Nicki Skafte, Martin Jørgensen, and Søren Hauberg. Reliable training and estimation of variance networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Andrew Stirn and David A Knowles. Variational variance: Simple, reliable, calibrated heteroscedastic noise variance parameterization. *arXiv preprint arXiv:2006.04910*, 2020.

Andrew Stirn, Harm Wessels, Megan Schertzer, Laura Pereira, Neville Sanjana, and David Knowles. Faithful heteroscedastic regression with neural networks. In *International Conference on Artificial Intelligence and Statistics*, pp. 5593–5613. PMLR, 2023.

Petre Stoica and Prabhu Babu. Low-rank covariance matrix estimation for factor analysis in anisotropic noise: application to array processing and portfolio selection. *IEEE Transactions on Signal Processing*, 2023.

Ba-Hien Tran, Simone Rossi, Dimitrios Milios, and Maurizio Filippone. All you need is a good functional prior for bayesian deep learning. *Journal of Machine Learning Research*, 23(74):1–56, 2022.

Matias Valdenegro-Toro and Daniel Saromo Mori. A deeper look into aleatoric and epistemic uncertainty disentanglement. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1508–1516. IEEE, 2022.

Jeffrey Willette, Hae Beom Lee, Juho Lee, and Sung Ju Hwang. Meta learning low rank covariance factors for energy-based deterministic uncertainty. *arXiv preprint arXiv:2110.06381*, 2021.

Carl K I Williams and Carl Edward Rasmussen. Gaussian processes for regression. In *Advances in Neural Information Processing Systems*, volume 8, 1996.

Lisa Wimmer, Yusuf Sale, Paul Hofman, Bernd Bischl, and Eyke Hüllermeier. Quantifying aleatoric and epistemic uncertainty in machine learning: Are conditional entropy and mutual information appropriate measures? In *Uncertainty in Artificial Intelligence*, pp. 2282–2292. PMLR, 2023.

Omer Yair, Elias Nehme, and Tomer Michaeli. Uncertainty visualization via low-dimensional posterior projections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11041–11051, 2024.

Kilian Zepf, Selma Wanna, Marco Miani, Juston Moore, Jes Frellsen, Søren Hauberg, Aasa Feragen, and Frederik Warburg. Laplacian segmentation networks: Improved epistemic uncertainty from spatial aleatoric uncertainty. *arXiv preprint arXiv:2303.13123*, 2023.

# A Symbols and Acronyms

## A.1 List of Symbols

| Symbol | Remark |
|---|---|
| $\mathcal{X}$ | Input space, $\mathcal{X} \subseteq \mathbb{R}^M$ |
| $\mathcal{Y}$ | Output space, $\mathcal{Y} \subseteq \mathbb{R}^S$ |
| $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ | Training dataset with $N$ input–output pairs |
| $\boldsymbol{x} = \{x_i\}_{i=1}^N$ | Stacked input sample from train split $\boldsymbol{x} \in \mathbb{R}^{N \times M}$ |
| $\boldsymbol{y} = \{y_i\}_{i=1}^N$ | Stacked input sample from train split $\boldsymbol{x} \in \mathbb{R}^{N \times S}$ |
| $x$ | Single input (test or train) sample |
| $y$ | Single output (test or train) sample |
| $p(w)$ | Prior distribution over weights |
| $p(\mathcal{D} \mid w)$ | Likelihood of the data given weights |
| $p(w \mid \mathcal{D})$ | Posterior distribution over weights |
| $q_\theta^\star(w)$ | Variational (proxy) distribution approximating $p(w \mid \mathcal{D})$ |
| $p(y \mid x, w)$ | Predictive distribution given input $x$ and weights $w$ |
| $p(y \mid x, \mathcal{D})$ | Bayesian predictive distribution (weights marginalized) |
| $p$ | Generic probability distribution |
| $q$ | Generic proxy / variational distribution |
| $N$ | Number of training samples |
| $S$ | Number of output units (e.g., pixels $\times$ channels) |
| $M$ | Number of input units (e.g., pixels $\times$ channels) |
| $T$ | Number of samples drawn from distribution over weights |
| $K$ | Size of weight space |
| $R$ | Number of columns of tall matrix $P$ |
| $\hat{R}$ | Number of columns of $\hat{P}$ after truncation |
| $f_w$ | Neural network mapping $f_w : \mathcal{X} \to \mathcal{Y}$ parameterized by $w$ |
| $W$ | Weight space, $W \in \mathbb{R}^K$ |
| $w$ | Neural network weight vector $w \in W$ |
| $\mu$ | Mean prediction output of the network $x$ |
| $\Sigma$ | Covariance (uncertainty) output of the network |
| $e$ | Prediction error of a sample |
| $D$ | Diagonal matrix used for linear low-rank decomposition (LRD), $D \in \mathbb{R}^{S \times S}$ |
| $\epsilon$ | Minimal variance entry of $D_{ii}$ enforced by implementation |
| $P$ | Tall matrix used for LRD, $P \in \mathbb{R}^{S \times R}$ |
| $C$ | Capacitance matrix used for inversion of $\Sigma$ and log-likelihood calculation, $C \in \mathbb{R}^{R \times R}$ |
| $I_R$ | Identity matrix, $I_R \in \mathbb{R}^{R \times R}$ |
| $\mathbf{0}_S$ | Zero matrix, $\mathbf{0}_S \in \mathbb{R}^{S \times S}$ |
| $U$ | Left singular vectors of $P$, $U \in \mathbb{R}^{R \times R}$ |
| $\Psi$ | Diagonal matrix of singular values of $P$, $\Psi \in \mathbb{R}^{R \times R}$ |
| $\hat{P}$ | Truncated matrix after applying SVD , $\hat{P} \in \mathbb{R}^{S \times \hat{R}}$ |
| $\hat{D}$ | Updated $D$ matrix after applying TSVD to $P$ to keep the variance, $\hat{P} \in \mathbb{R}^{S \times S}$ |
| $\tilde{P}$ | Rotated matrix after applying SVD, $\tilde{P} \in \mathbb{R}^{S \times R}$ |
| $P^\star$ | Orthogonal matrix before normalization (used in ablation), $P^\star \in \mathbb{R}^{S \times R}$ |
| $\bar{P}$ | Orthonormal matrix after normalization (used in ablation), $\bar{P} \in \mathbb{R}^{S \times R}$ |
| $P^W$ | Raw model output matrix (used in ablation), $P^W \in \mathbb{R}^{S \times R}$ |
| $\mathcal{L}$ | Loss function |
| $\mathcal{N}$ | Normal (Gaussian) distribution |
| $\mathbb{E}[\,\cdot\,]$ | Expectation operator |
| $\mathrm{Cov}[\,\cdot\,]$ | Covariance operator |
| $[\,\cdot\;\;\cdot\,]$ | Column-wise block concatenation |
| $(\cdot)^\top$ | Matrix or vector transpose |
| $(\cdot)^a$ | Aleatoric uncertainty only |
| $(\cdot)^e$ | Epistemic uncertainty only |
| $(\cdot)^W$ | Raw model head output from a single forward pass |
| $(\cdot)_w$ | Quantity viewed as a function of the weights $w$ |
| $(\cdot)_{w_i}$ | Quantity evaluated at a particular weight sample $w_i$ |
| $(\cdot)_{i\_}$ | $i^{\text{th}}$ row of a matrix |
| $(\cdot)_{\_i}$ | $i^{\text{th}}$ column of a matrix |
| $\lambda(\cdot)$ | Eigenvalue operator applied to matrix $\cdot$ |
| $\kappa(\cdot)$ | Condition number operator applied to matrix $\cdot$ |
| $\alpha, \beta$ | Hyperparameters scalar controlling trade-off in loss |
| $\mathcal{L}, \mathcal{L}_I, \mathcal{L}_{\text{lrd}}, \mathcal{L}_{\bar{P}}, \mathcal{L}_{P^a}$ | Loss functions as defined in the equations |
| $\lfloor . \rfloor$ | Stop-gradient operator |

Table 2: List of symbols used in the paper.

### A.2 List of Acronyms

## Acronyms

Flying Chairs

**CelebA** CelebFaces Attributes

**D** diagonal

**DE** Deep Ensemble

**GT** ground truth

**LA** Laplace Approximation

**LL** log-likelihood

**LR+D** low-rank plus diagonal

**LU** lower-upper

**MAP** Maximum a posteriori

**MC** Monte Carlo

**MCD** Monte Carlo Dropout

**MNIST** Modified National Institute of Standards and Technology database

**NLL** negative log-likelihood

**NPPC** Neural Posterior Principal Components

**SVD** Singular Value Decomposition

**SVI** Stochastic Variational Inference

**TLL** test log-likelihood

**TSVD** Truncated Singular Value Decomposition

**UQ** Uncertainty Quantification

# B Reproducibility

An anonymized implementation and scripts to reproduce all experiments are available at `https://anonymous.4open.science/r/corr-joint-ae`. Checkpoints for all models trained with the proposed method, as well as all baselines, are available upon request. The datasets used in the experiments are publicly accessible and links as well as preprocessing scripts are included in the repository. An extensive schematic, with pseudocode and intuitive description of the method, along with proofs, is also included here. Additionally, qualitative examples are provided to enhance understanding of the method. All experiments were conducted on a single NVIDIA Quadro RTX 8000 GPU with 48 GB of RAM. For all experiments, we fix the random seed to 42. For Deep Ensemble models, we train ensemble members using distinct seeds 42, 43, 44, 45, 46, following standard practice.

# C Computation, Time and Space Complexity

Table 3 gives the theoretical time and memory complexities of various covariance parametrizations and calculations. The sparse representations are more efficient in terms of memory and computational complexity. However, they do not provide all degrees of freedom of a covariance matrix and are limited to either local or the most important global correlations.

| Type | Parametrization | Captured correlation | Precompute | Per-Eval $p(y \mid \mathcal{N})$ $x\Sigma^{-1}x$ | Per-Eval $p(y \mid \mathcal{N})$ $\lvert\Sigma\rvert$ | Per-Eval $y \sim \mathcal{N}$ | Memory |
|---|---|---|---|---|---|---|---|
| full Russell & Reale (2021) | correlation | all | $\mathcal{O}(S^3)$ | $\mathcal{O}(S^2)$ | $\mathcal{O}(S)$ | $\mathcal{O}(S^2)$ | $\mathcal{O}(S^2)$ |
| full Gundavarapu et al. (2019) | lower-triangular Cholesky | all | – | $\mathcal{O}(S^2)$ | $\mathcal{O}(S)$ | $\mathcal{O}(S^2)$ | $\mathcal{O}(S^2)$ |
| sparse Dorta et al. (2018a;b) | inverse band Cholesky ⋆ | local | – | $\mathcal{O}(SR)$ | $\mathcal{O}(S)$ | $\mathcal{O}(SR)$ | $\mathcal{O}(SR)$ |
| sparse Monteiro et al. (2020) | LRD | global | $\mathcal{O}(SR^2 + R^3)$ | $\mathcal{O}(SR)$ | $\mathcal{O}(SR)$ | $\mathcal{O}(SR)$ | $\mathcal{O}(SR)$ |
| factorized Kendall & Gal (2017) | diagonal | none | – | $\mathcal{O}(S)$ | $\mathcal{O}(S)$ | $\mathcal{O}(S)$ | $\mathcal{O}(S)$ |

Table 3: This table depicts the computational complexity for calculations using different parametrizations for covariance matrices. Here $S$ is the output dimensionality (number of variables) and $R$ denotes the structural size: for banded Cholesky $R$ is the bandwidth (number of subdiagonals), and for low-rank plus diagonal (LR+D) $R$ is the factor rank; throughout $R \ll S$ is assumed. We use the sparse LR+D parametrization as the basis for our method. This reduces time and spatial complexity in comparison to the naive or Cholesky decomposition and allows for global correlation in comparison to the sparse inverse band Cholesky parametrization. The type and amount of correlations of different parametrization is different (Captured Corr). Furthermore, the used representation enables for efficient calculation of $\Sigma$ or $\Sigma^-1$ (Parametr. Representation) and needs different amount of memory. The time complexity is given for calculation of the mahalanobis distance $x\Sigma^{-1}x^\top$, determinant $|\Sigma|$ as well as sampling.
⋆ Inverse band Cholesky assumes a direct parametrization of a banded precision factor $L$ with $\Sigma^{-1} = LL^\top$. The per-evaluation costs shown (quadratic form and sampling in $\mathcal{O}(SR)$, log-determinant in $\mathcal{O}(S)$) use precision-based sampling via triangular solves, as in Dorta et al. (2018a;b). Achieving these $\mathcal{O}(SR)$ costs in practice requires implementations that exploit band structure; generic dense linear algebra backends typically treat $L$ as a full $S \times S$ matrix.

# D Additional Results

## D.1 Qualitative Results

We provide additional qualitative results for every performed tasks. Figures 7 presents optical flow on Flying Chairs, 8 depicts CelebA inpainting, 9 shows CelebA colorization, and 10 illustrates MNIST inpainting. Figure 11 compares both, eigenvectors in both random orientations as well as the different used Bayesian methods.

The optical flow visualization of Figure 7 encodes the 2D motion vectors into a color image using a color wheel scheme. Each pixel's hue corresponds to the direction of motion, covering all angles in a circular manner (e.g., red for rightward motion, green for upward, blue for leftward, etc.). The color saturation or intensity represents the magnitude of the motion, with brighter and more saturated colors indicating higher motion speeds. This method allows intuitive interpretation of both the direction and speed of movement in the scene or the direction of the movement uncertainty for the Eigenvectors.

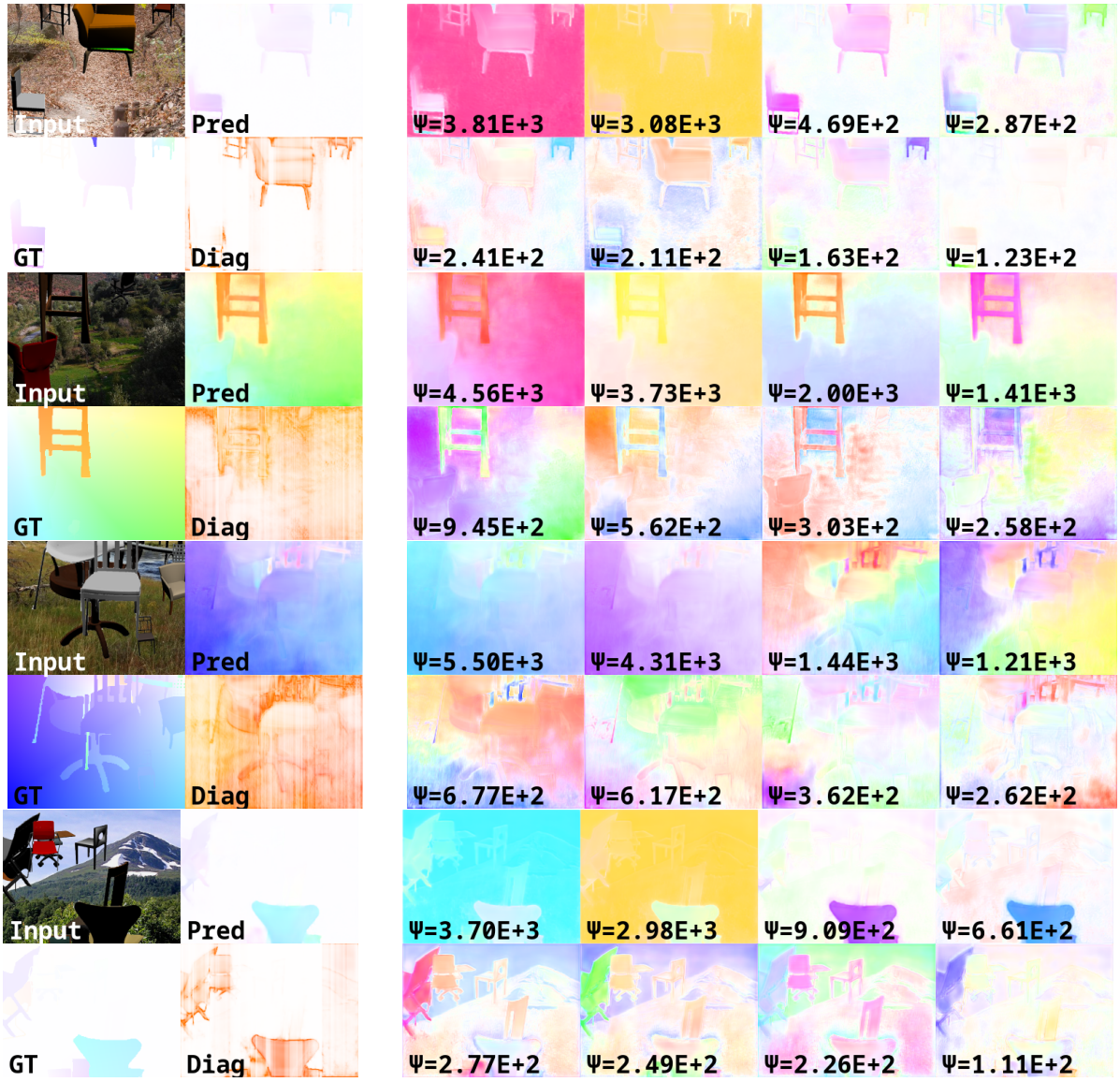


Figure 7: Additional Qualitative Results, Visualizing ' Optical Flow. Random samples from the test sets showing input, prediction, ground truth and parameters of the predictive distribution. The task here is optical flow estimation in the  dataset. The model predicts a mean (Pred), and the parameter $D$ (Diag), as well as a low-rank matrix $P$. In all cases, the predicted joint low-rank matrix $P$ is reduced to the 64 most significant directions (columns) and displayed using the 10 most significant ones in descending order of associated eigenvalues. We can clearly see that the columns focus on uncertainty in certain images areas. Furthermore, the singular values $\Psi$ give a measure of importance of the associated direction. Note that the orientation of the singular vectors is arbitrarily chosen and can be inverted, which results in opposite colors (left) and brightness (right). In these examples, the first singular vectors are more than 10 times as important as the 8th and last visualized singular vectors. One can get insights into the uncertainty priority: For instance, in the first example, the background seems to be the most uncertain in both directions according to the first two eigenvectors (with roughly the same value). In the second example, the first singular values suggest a higher uncertainty in the foreground and the rightmost background of the image. These eigenvectors are only possible to visualize when modeling covariances and show the direction of maximum variability of the data and helps to understand the underlying factors. Furthermore, we show the upper bound of the angles between the directions of the eigenvectors of $PP^\top$ and the eigenvectors of $\Sigma$.

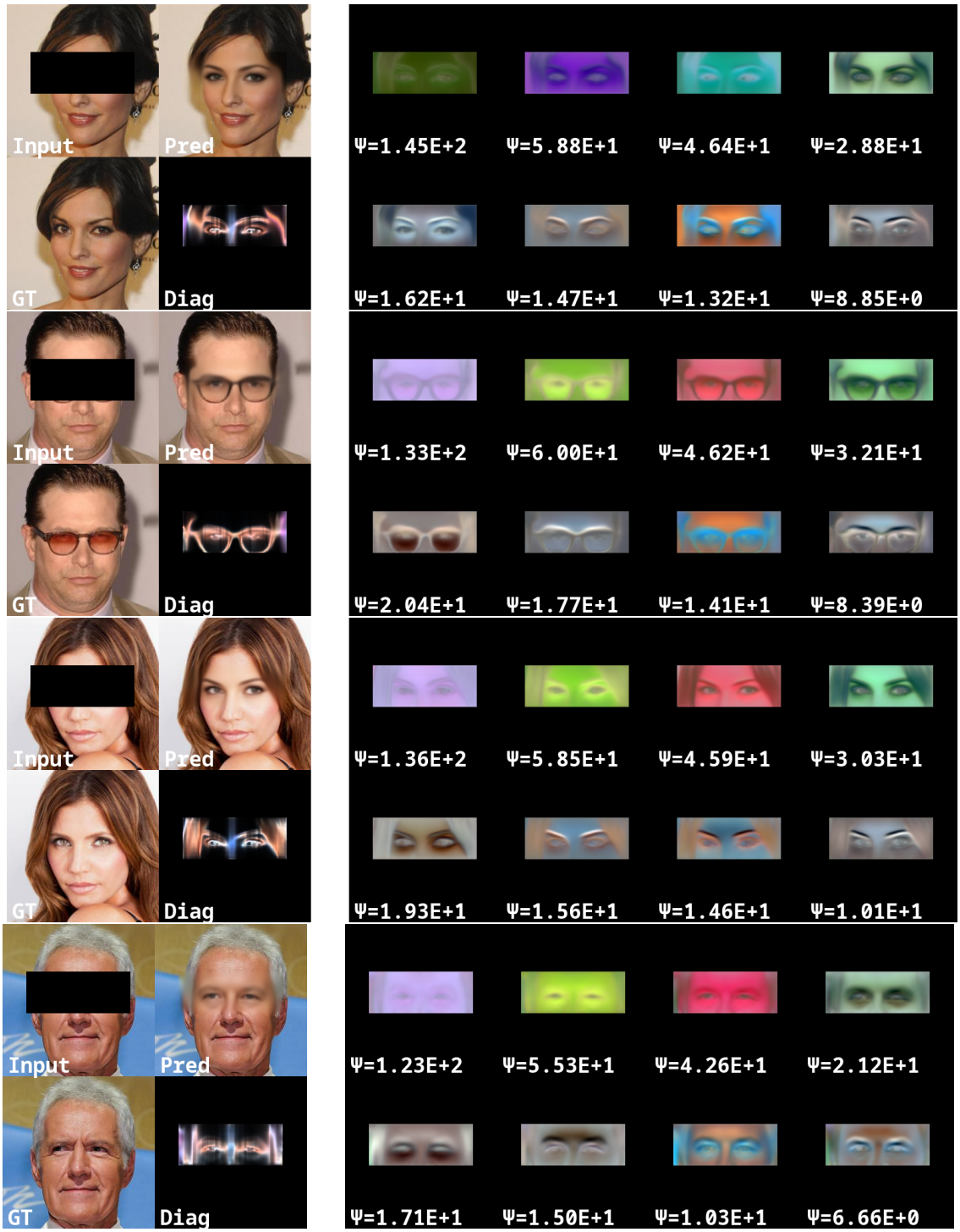


Figure 8: Additional Qualitative Results, Visualizing Inpainting of Eyes of CelebA Faces. Random samples from the test sets showing input, prediction, ground truth and parameters of the predictive distribution. The task here is inpainting of the eyes region of the CelebA faces dataset. The model predicts a mean (Pred), and the parameter $D$ (Diag), as well as a low-rank matrix $P$. In all cases, the predicted joint low-rank matrix $P$ is reduced to the 64 most significant directions (columns) and displayed using the 10 most significant ones in descending order of associated eigenvalues.

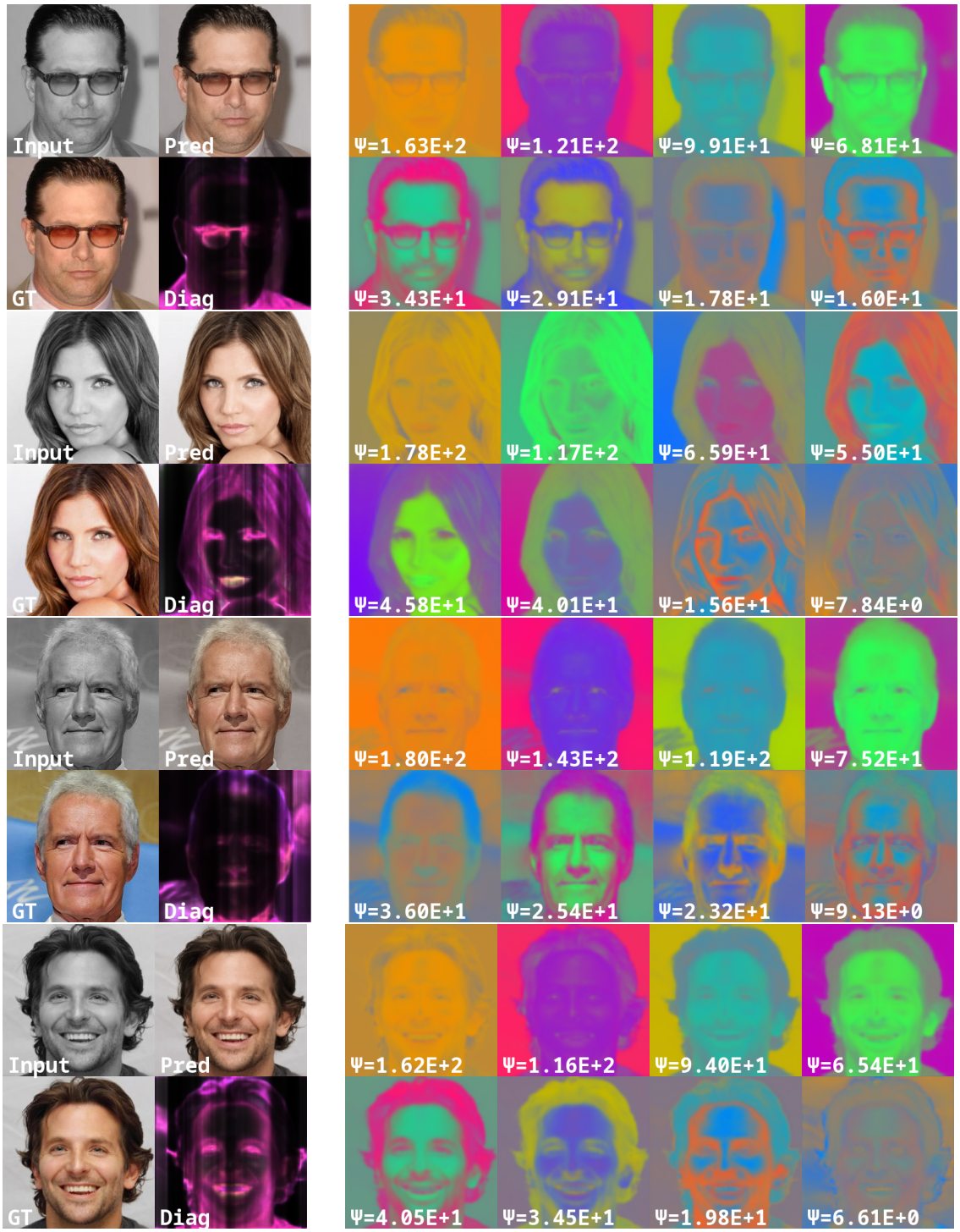


Figure 9: Additional Qualitative Results, Visualizing the Colorization of CelebA Faces. Random samples from the test sets showing input, prediction, ground truth and parameters of the predictive distribution. The task here is colorization of the CelebA faces dataset. The model predicts a mean (Pred), and the parameter $D$ (Diag), as well as a low-rank matrix $P$. In all cases, the predicted joint low-rank matrix $P$ is reduced to the 64 most significant directions (columns) and displayed using the 10 most significant ones in descending order of associated eigenvalues.

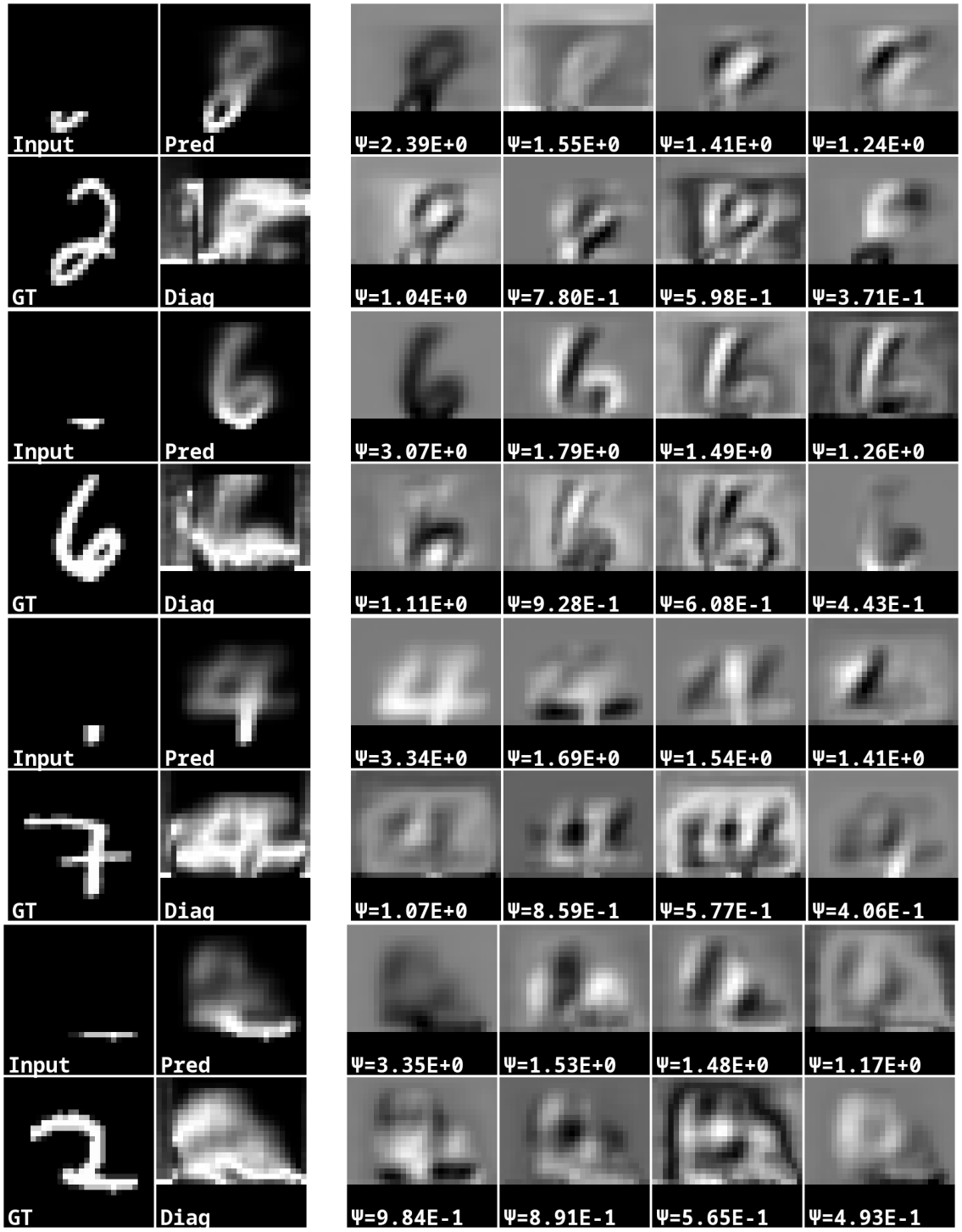


Figure 10: Additional Qualitative Results, Visualizing Inpainting of MNIST Digits. Random samples from the test sets showing input, prediction, ground truth and parameters of the predictive distribution. The task is inpainting MNIST digits. The model predicts a mean (Pred), and the parameter $D$ (Diag), as well as a low-rank matrix $P$. In all cases, the predicted joint low-rank matrix $P$ is reduced to the 64 most significant directions (columns) and displayed using the 10 most significant ones in descending order of associated eigenvalues.

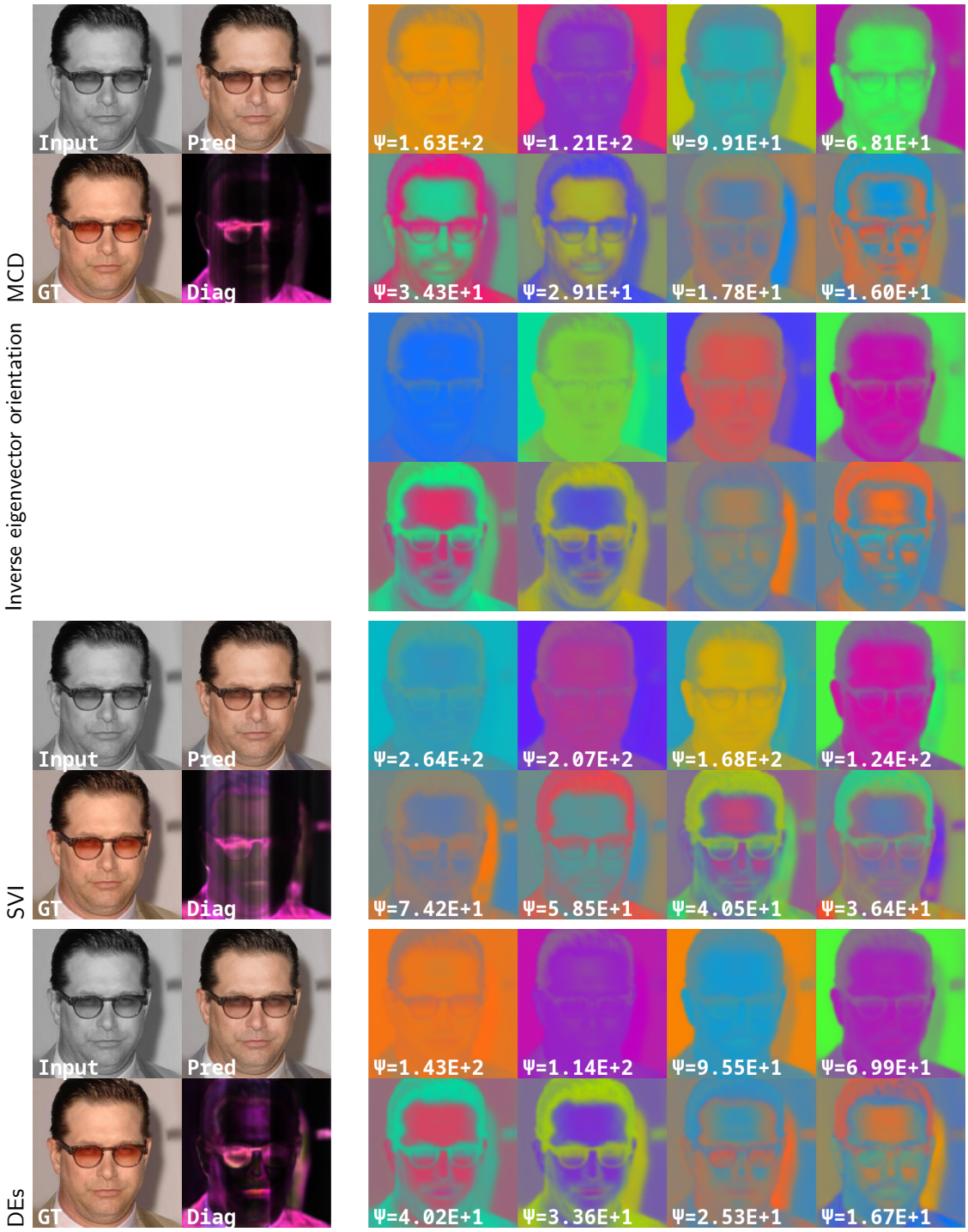


Figure 11: Additional Qualitative Results, comparing Bayesian Methods. A Random sample from the test sets showing input, prediction, ground truth and parameters of the predictive distribution with various Bayesian Methods. For the first method, we also show the eigenvectors with inverse sign. Both signs are mathematically equivalent and one of them is randomly chosen for the visualizations. The task here is colorization of the CelebA faces dataset. The model predicts a mean (Pred), and the parameter $D$ (Diag), as well as a low-rank matrix $P$. In all cases, the predicted joint low-rank matrix $P$ is reduced to the 64 most significant directions (columns) and displayed using the 10 most significant ones in descending order of associated eigenvalues.

### D.2 Quantitative Prediction Errors

Table 4 lists the predicted errors for all Bayesian methods. We aim for similar predictive errors for all models to get mainly evaluate the quality of the uncertainty using test log-likelihood (TLL).

| | | | MNIST | | CelebA | | | | Flying Chairs | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Inpainting | | Colorization | | Inpainting | | Opt. Flow | |
| Param. | $R_W$ | Epistemic | $L_1 \downarrow$ | $L_2 \downarrow$ | $L_1 \downarrow$ | $L_2 \downarrow$ | $L_1 \downarrow$ | $L_2 \downarrow$ | $L_1 \downarrow$ | $L_2 \downarrow$ |
| ✗ | 0 | ✗ | **0.102** | **0.231** | 0.0326 | 0.0467 | 0.354 | **0.451** | 2.48 | 5.50 |
| D | 0 | ✗ | **0.102** | 0.232 | 0.0329 | 0.0473 | 0.355 | 0.452 | 2.67 | 5.59 |
| | | MCD | 0.111 | 0.240 | 0.0339 | 0.0488 | 0.354 | 0.452 | 2.64 | 5.60 |
| | | SVI | 0.107 | 0.234 | 0.0331 | 0.0476 | 0.355 | 0.452 | 2.63 | 5.61 |
| | | DE | 0.107 | 0.233 | **0.0309** | **0.0446** | 0.354 | **0.451** | **2.27** | **5.18** |
| LR+D | 4 | MCD | 0.109 | 0.235 | 0.0315 | 0.0453 | 0.354 | **0.451** | 2.54 | 5.54 |
| | 8 | ✗ | 0.103 | 0.232 | 0.0317 | 0.0457 | 0.355 | 0.452 | 2.48 | 5.48 |
| | | MCD | 0.109 | 0.235 | 0.0313 | 0.0451 | 0.354 | **0.451** | 2.58 | 5.54 |
| | | SVI | 0.109 | 0.236 | 0.0312 | 0.0449 | 0.355 | 0.452 | 2.66 | 5.66 |
| | | DE | 0.108 | 0.232 | 0.0310 | 0.0447 | **0.353** | **0.451** | 2.28 | 5.19 |
| | 12 | MCD | 0.110 | 0.235 | 0.0315 | 0.0454 | 0.354 | **0.451** | 2.57 | 5.57 |
| | 16 | MCD | 0.110 | 0.235 | 0.0315 | 0.0452 | 0.354 | **0.451** | 2.49 | 5.47 |

Table 4: Comparison of reconstruction or prediction errors of all methods. We use the same loss for the prediction between those methods. The last convolutional layer of models with LR+D parametrization has more channels in comparison to models with D parametrization. Furthermore, row 0 shows the result for models trained without extra channels for aleatoric uncertainty prediction. The uncertainty channels of the other models receive gradients from different negative log-likelihood functions. Bayesian models (Epistemic) include additional Dropout layers or variational convolutional layers and are evaluated using $T = 64$ weight samples. Essentially, the presented study shows our robust, better uncertainty quantification towards the quality of the prediction. This is important to evaluate because the negative-log-likelihood is affected by both prediction and uncertainty estimation.

### D.3 Additional Ablation Study

**Comparison with Nehme et al. (2024)** We trained additional models to compare with the approach of Nehme et al. (2024), as summarized in Tables 5, 6, and 4. To enable the use of Nehme's Neural Posterior Principal Components (NPPC) loss for the covariance factor, several modifications were necessary.

First, to improve training stability, the mean and uncertainty predictions were separated into two models, with the uncertainty model optionally receiving the mean model's output as an additional input (indicated by the *Combined* column: ✓vs. ✗). Second, Gram–Schmidt orthogonalization was applied to enforce orthogonality of the covariance directions (*Gram–Schmidt* column: ✓vs. ✗), a prerequisite for the NPPC loss. Finally, the NPPC loss was applied to the covariance factor $P$, while the diagonal covariance terms remained optimized with the LR+D loss using a stop-gradient on $P$.

Due to instability, NPPC training was only viable with separate models and Gram–Schmidt projection; combined mean–uncertainty models with NPPC were not stable. Thus, we present results showing a stepwise transition from the baseline LR+D model to the NPPC configuration.

Before any subsequent loss computation or evaluation, the raw low-rank model output matrix $P^W$ consists of columns $p_r^W$ that represent initial (non-orthogonal) low-rank directions. These columns are first orthogonalized using the Gram–Schmidt process to obtain the orthogonalized directions $p_r^a$. Normalizing these columns produces the orthonormal directions $\bar{p}_r$.

In matrix form, these are arranged as:

$$P^W = \begin{bmatrix} p_1^W & p_2^W & \cdots & p_R^W \end{bmatrix}, \quad P^\star = \begin{bmatrix} p_1^\star & p_2^\star & \cdots & p_R^\star \end{bmatrix}, \quad \bar{P} = \begin{bmatrix} \bar{p}_1 & \bar{p}_2 & \cdots & \bar{p}_R \end{bmatrix}.$$

Each $p_r^W, p_r^\star, \bar{p}_r$ is a column vector in these respective matrices representing different stages of processing for the basis directions in the low-rank decomposition.

**Algorithm 1** Gram-Schmidt

**Require:** Model output vectors $p_1^W$, $p_2^W$, …, $p_R^W$
**Ensure:** Orthogonal vectors $p_1^\star$, $p_2^\star$, …, $p_R^\star$
**Ensure:** Orthonormal vectors $\bar{p}_1$, $\bar{p}_2$, …, $\bar{p}_R$

1: $p_1^\star = p_1^W$
2: $\bar{p}_1 = \dfrac{p_1^\star}{\|p_1^\star\|}$
3: **for** $k = 2$ **to** $K$ **do**
4: Form the matrix of previous orthonormal columns:

$$\bar{P}^{(r-1)} = \begin{bmatrix} \bar{p}_1 & \cdots & \bar{p}_{r-1} \end{bmatrix}$$

5: Orthogonalize:

$$p_r^\star = p_r^W - \bar{P}^{(r-1)} \left( (\bar{P}^{(r-1)})^\top p_r^\star \right)$$

6: Normalize:

$$\bar{p}_r = \frac{p_r^\star}{\|p_r^\star\|}$$

7: **end for**

The NPPC objectives are calculated using the detached prediction error $e(x_i) = \lfloor \mu(x_i) - y_i \rfloor$, and decompose into direction and variance losses:

$$\mathcal{L}_{\bar{P}} = 1 - \sum_i \left\| \bar{P}^\top(x_i) e(x_i) \right\|^2$$

$$\mathcal{L}_{P^\star} = \sum_i \underbrace{\frac{1}{\|e_i\|^4}}_{\text{normalization}} \sum_{r=1}^{R^W} \left( \|p_r^\star(x_i)\|^2 - \left( \bar{p}_r^\top(x_i) e(x_i) \right)^2 \right)^2$$

Whereas the normalization $\frac{1}{\|e_i\|^4}$, is not explicitly mentioned in Nehme et al. (2024) while the accompanying implementation of the NPPC actually normalizes the variance loss $L_{P^\star}$. Therefore, we stick with their implementation and adapt the formulas accordingly. The joint loss becomes:

$$\mathcal{L}_{\text{lrd}^\star} = \sum_i \log \mathcal{N}\left( y_i \mid \lfloor \mu(x_i) \rfloor,\, D(x_i) + \lfloor P(x_i) P^\top(x_i) \rfloor \right) \tag{13}$$

$$\mathcal{L} = \alpha\, \mathcal{L}_{\text{lrd}^\star} + \beta\, \mathcal{L}_{\bar{P}} + \beta\, \mathcal{L}_{p^\star} \tag{14}$$

We evaluate models using TLL (Table 5) and relative reconstruction error

$$e = \mu(x) - y, \quad \frac{e - PP^\top e}{e},$$

which quantifies the fraction of residual error unexplained by the covariance directions (Table 6). Prediction errors from the previous section (Table 4) serve as baseline references.

The LR+D-trained models consistently outperform NPPC variants in TLL, indicating better likelihood-based prediction. Conversely, NPPC reduces the relative reconstruction error, suggesting improved capture of residual structure by the covariance factor. These results highlight a trade-off: optimizing for TLL favors LR+D models, while NPPC better models orthogonal residual errors, guiding metric-dependent optimization choices.

Models without combined prediction use the same mean model as the first row of Table 4, allowing direct reading of their prediction errors from that reference.

| Combined | Gram Schmid | Loss | MNIST Inpainting ×10 | CelebA Inpainting ×100 | CelebA Colorization ×1000 | Flying Chairs Optical Flow ×1000 |
|---|---|---|---|---|---|---|
| ✓ | ✗ | | **105** ± 158 | **-65** ± 239 | **585** ± 116 | -173 ± 73 |
| | ✓ | | 97 ± 463 | -91 ± 341 | 582 ± 113 | -167 ± 75 |
| ✗ | ✗ | | 99 ± 63 | -79 ± 162 | 566 ± 58 | **-91** ± 132 |
| | ✓ | | 99 ± 48 | -75 ± 156 | 565 ± 64 | -95 ± 120 |
| | | NPPC | 95 ± 162 | -96 ± 284 | 571 ± 99 | -100 ± 206 |

Table 5: TLL for each model configuration across tasks. The table compares the impact of combined versus separate mean and uncertainty models (*Combined*), application of Gram–Schmidt orthogonalization (*Gram Schmid*), and different loss functions, including NPPC. Values are reported as mean ± standard deviation, scaled as indicated.

| Combined | Gram Schmid | Loss | MNIST Inpainting | CelebA Inpainting | CelebA Colorization | Flying Chairs Optical Flow |
|---|---|---|---|---|---|---|
| ✓ | ✗ | | 0.39 ± 0.06 | **0.36** ± 0.13 | 0.63 ± 0.08 | 0.62 ± 0.13 |
| | ✓ | | 0.39 ± 0.06 | **0.36** ± 0.13 | 0.62 ± 0.08 | 0.63 ± 0.13 |
| ✗ | ✗ | | 0.41 ± 0.06 | 0.37 ± 0.13 | 0.68 ± 0.08 | 0.62 ± 0.14 |
| | ✓ | | 0.38 ± 0.06 | 0.37 ± 0.13 | 0.67 ± 0.08 | 0.61 ± 0.14 |
| | | NPPC | **0.31** ± 0.07 | **0.36** ± 0.12 | **0.61** ± 0.07 | **0.53** ± 0.13 |

Table 6: Relative reconstruction error $\left(\frac{e-PP^\top e}{e}\right)$ for each model configuration on the same test sets. This metric captures the proportion of the prediction residual error not explained by the covariance model directions, providing insight into the quality of uncertainty estimation. Column organization matches Table 5 for direct comparison.

**Joint vs. separate training of mean and uncertainty models** While the design choice to jointly train mean prediction and uncertainty estimation models is conceptually cleaner and yields superior results on three out of four benchmark tasks, an exception arises in the Flying Chairs dataset. For this dataset, the separate model setup performs significantly better. In the split configuration, the uncertainty model receives as input the concatenation of both the original input and the output of the mean prediction model.

This architectural difference may explain the observed discrepancy in performance. Unlike image inpainting and colorization tasks—where the model has no direct way to verify the correctness of its predictions—the Flying Chairs task inherently allows the uncertainty model to implicitly "check" prediction accuracy. Specifically, the uncertainty model can leverage the input images and predicted flow outputs to learn pixel shifts that align with ground truth, effectively evaluating the prediction quality.

We hypothesize that this direct feedback mechanism enables the separate uncertainty model in Flying Chairs to better estimate prediction errors, whereas joint training suffices or outperforms for other tasks where such verification is unavailable or less direct. This is supported by the negative log-likelihood comparisons reported in Table 5, where the split model outperforms the combined model specifically on the Flying Chairs dataset.

This suggests that task-specific characteristics and the degree of prediction observability should guide the choice between joint and separated uncertainty modeling architectures.

**Retaining variance of TSVD** One component of our proposed method is to retain the variance of the removed columns after dimensionality reduction using SVD, see 9. In Table 7 we ablate this design choice. The column $\hat{D}$ indicates whether the diagonal $D$ is updated (✓) after performing SVD according to Equation 9, or if the original $D$ is retained (✗) as per Equation 6. Our ablation shows that updating the diagonal $D$ appears to slightly improve the average, TLL while also enhancing prediction consistency and reducing test set variability. This is consistent for three different configurations of dimensionality reductions, see Table 7.

| $R$ | $\hat{D}$ | MNIST Inpainting ×10 | CelebA Inpainting ×100 | CelebA Colorization ×1000 | Flying Chairs Optical Flow ×1000 |
|---|---|---|---|---|---|
| 8 | ✓ | 95 ± 78 | -194 ± 281 | 521 ± 138 | -194 ± 84 |
| | ✗ | 48 ± 1341 | -294 ± 556 | 519 ± 241 | -186 ± 109 |
| 16 | ✓ | 96 ± 88 | -152 ± 263 | 548 ± 139 | -186 ± 82 |
| | ✗ | 53 ± 1255 | -211 ± 439 | 550 ± 203 | -178 ± 98 |
| 32 | ✓ | 101 ± 102 | -109 ± 249 | 567 ± 129 | -179 ± 76 |
| | ✗ | 60 ± 1098 | -143 ± 368 | 571 ± 171 | -172 ± 87 |
| 64 | ✓ | **105** ± 158 | -65 ± 239 | 585 ± 116 | -173 ± 73 |
| | ✗ | 72 ± 883 | -82 ± 311 | 589 ± 141 | -167 ± 80 |
| 128 | ✓ | 103 ± 308 | -22 ± 223 | 602 ± 87 | -167 ± 72 |
| | ✗ | 84 ± 662 | -28 ± 253 | 604 ± 96 | -163 ± 75 |
| 256 | ✓ | 97 ± 446 | 17 ± 201 | 616 ± 75 | -161 ± 71 |
| | ✗ | 95 ± 482 | 16 ± 207 | 617 ± 78 | -160 ± 72 |
| 512 | ✓ | 98 ± 439 | 39 ± 179 | 626 ± 68 | -158 ± 70 |
| | ✗ | 98 ± 439 | 39 ± 179 | 626 ± 68 | -158 ± 70 |
| 576 | - | 98 ± 439 | **40** ± 177 | **627** ± 67 | **-158** ± 70 |

Table 7: Comparison between adapting the diagonal $D$ after performing the SVD according to Equation 9 or not. Here $R$ denotes the number of columns in the resulting representation. The columns $P^a$ and $P^e$ denote if and to what degree the dimensionality is reduced after sampling using SVD. The numbers in brackets denote the kept singular vectors, which result in columns $R$. The column $\hat{D}$ indicates whether the diagonal $D$ is updated (✓) after performing SVD according to Equation 9, or if the original $D$ is retained as per Equation 6, despite the dimensionality reduction of $P$. We show in the first three rows that updating the diagonal $D$ appears to slightly improve the average TLL while also enhancing prediction consistency and reducing test set variability.

| $R$ | $R_W$ | MNIST Inpainting ×10 | CelebA Inpainting ×100 | CelebA Colorization ×1000 | Flying Chairs Optical Flow ×1000 |
|---|---|---|---|---|---|
| 16 | 4 | 91 ± 402 | -218 ± 429 | 537 ± 140 | **-180** ± 90 |
| | 8 | 96 ± 88 | -152 ± 263 | **548** ± 139 | -186 ± 82 |
| | 12 | 103 ± 80 | -135 ± 236 | 526 ± 76 | -190 ± 70 |
| | 16 | **106** ± 147 | **-129** ± 155 | 535 ± 72 | -192 ± 71 |
| 32 | 4 | 83 ± 760 | -168 ± 398 | 556 ± 118 | **-174** ± 85 |
| | 8 | 101 ± 102 | -109 ± 249 | 567 ± 129 | -179 ± 76 |
| | 12 | 105 ± 106 | **-98** ± 229 | 555 ± 87 | -182 ± 68 |
| | 16 | **112** ± 163 | -102 ± 167 | **568** ± 76 | -182 ± 70 |
| 64 | 4 | 83 ± 781 | -114 ± 348 | 573 ± 100 | **-167** ± 82 |
| | 8 | 105 ± 158 | -65 ± 239 | 585 ± 116 | -173 ± 73 |
| | 12 | 108 ± 148 | **-58** ± 224 | 582 ± 81 | -174 ± 68 |
| | 16 | **113** ± 173 | -65 ± 177 | **594** ± 72 | -172 ± 71 |
| 128 | 4 | 82 ± 794 | -66 ± 299 | 589 ± 86 | **-161** ± 79 |
| | 8 | 103 ± 308 | -22 ± 223 | 602 ± 87 | -167 ± 72 |
| | 12 | 109 ± 245 | **-14** ± 215 | 607 ± 67 | -166 ± 69 |
| | 16 | **115** ± 238 | -21 ± 184 | **617** ± 68 | -162 ± 72 |

Table 8: Comparison between different number of columns used during training the model. While $R^W$ denotes the number of columns produced by the model without sampling., $R$ denotes the number of columns in the resulting representation. The column $P$ show how SVD is used dimensionality is reduced. The number in the brackets denote the kept singular vectors, which result as columns $R$. The results tend to be better with a higher number of learned columns $R^W$. In general, increasing $R^W$ can cause increasing training time. However, we also experienced instabilities during training for 3 of those tasks when using $R^W = 32$.

**Influence of output layer columns** Furthermore, Table 8 evaluates the impact of the number of columns predicted by the model's output layer. Increasing the number of columns benefits the TLL for the 3 tasks with larger images and therefore a higher dimensional output space (CelebA inpainting and colorization as well as optical flow estimation on flying chairs). However, this increases computational complexity and lead to numerical instabilities during training, as models with $R^W \geq 20$ columns fail for all tasks. Balancing these trade-offs, we chose a rank of 8, which aligns close with the choice of Monteiro et al. (2020).

Finally, Table 9 presents an extended ablation of various parameters, non-Bayesian with various Bayesian Models (epis) and both kinds of distribution parametrizations (Param). Therefore, it compares a purely diagonal (D) uncertainty with our LR+D parametrization. The representation took $T$ Bayesian samples, and results in $R$ columns of the low-rank matrix. The aleatoric matrix $P^a$ is in some rows approximated using the expected weights $\mathbb{E}[W]$ approximation. The number of columns of the matrix ($P$) is optionally reduced using TSVD. The matrix is not existant for D Parametriations (-) truncated (✓) or kept (✗). The column $\hat{D}$ indicates whether the diagonal $D$ is updated (✓) after performing TSVD according to Equation 9, or if the original $D$ is retained as per Equation 6, despite the dimensionality reduction of $P$.

Table 9:

| Epis | Param | $R^W$ | $T$ | $P^a$ | TSVD | $\hat{D}$ | $R$ | MNIST Inpainting ×10 | CelebA Inpainting ×100 | CelebA Colorization ×1000 | Flying Chairs Optical Flow ×1000 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ✗ | D | 0 | 0 + 1 | - | - | - | 0 | -3550 ± 22033 | -471 ± 558 | 240 ± 519 | -231 ± 110 |
| ✗ | LR+D | 8 | 0 + 1 | - | ✗ | - | 8 | -2445 ± 14022 | -513 ± 1104 | 495 ± 267 | -184 ± 119 |
| MCD | D | 0 | 64 + 0 | - | - | - | 0 | 72 ± 1004 | -341 ± 495 | 324 ± 249 | -224 ± 85 |
| MCD | D | 0 | 64 + 1 | $\mathbb{E}[W]$ | - | - | 0 | 22 ± 1599 | -362 ± 530 | 317 ± 265 | -224 ± 90 |
| MCD | LR+D | 4 | 64 + 0 | - | ✓ | ✗ | 4 | 55 ± 1295 | -373 ± 652 | 456 ± 272 | -196 ± 125 |
| MCD | LR+D | 4 | 64 + 0 | - | ✓ | ✓ | 4 | 90 ± 144 | -298 ± 472 | 478 ± 163 | -194 ± 97 |
| MCD | LR+D | 4 | 64 + 0 | - | ✓ | ✗ | 8 | 57 ± 1279 | -317 ± 609 | 506 ± 202 | -186 ± 113 |
| MCD | LR+D | 4 | 64 + 0 | - | ✓ | ✓ | 8 | 91 ± 190 | -260 ± 452 | 514 ± 152 | -187 ± 94 |
| MCD | LR+D | 4 | 64 + 0 | - | ✓ | ✗ | 16 | 59 ± 1246 | -259 ± 558 | 534 ± 168 | -178 ± 104 |
| MCD | LR+D | 4 | 64 + 0 | - | ✓ | ✓ | 16 | 91 ± 402 | -218 ± 429 | 537 ± 140 | -180 ± 90 |
| MCD | LR+D | 4 | 64 + 0 | - | ✓ | ✗ | 32 | 63 ± 1167 | -193 ± 481 | 556 ± 134 | -171 ± 95 |
| MCD | LR+D | 4 | 64 + 0 | - | ✓ | ✓ | 32 | 83 ± 760 | -168 ± 398 | 556 ± 118 | -174 ± 85 |
| MCD | LR+D | 4 | 64 + 0 | - | ✓ | ✗ | 64 | 69 ± 1054 | -126 ± 390 | 574 ± 108 | -165 ± 87 |
| MCD | LR+D | 4 | 64 + 0 | - | ✓ | ✓ | 64 | 83 ± 781 | -114 ± 348 | 573 ± 100 | -167 ± 82 |
| MCD | LR+D | 4 | 64 + 0 | - | ✓ | ✗ | 128 | 75 ± 930 | -69 ± 312 | 589 ± 89 | -160 ± 81 |
| MCD | LR+D | 4 | 64 + 0 | - | ✓ | ✓ | 128 | 82 ± 794 | -66 ± 299 | 589 ± 86 | -161 ± 79 |
| MCD | LR+D | 4 | 64 + 0 | - | ✓ | ✗ | 256 | 81 ± 815 | -30 ± 257 | 601 ± 77 | -156 ± 76 |
| MCD | LR+D | 4 | 64 + 0 | - | ✓ | ✓ | 256 | 81 ± 811 | -30 ± 256 | 601 ± 77 | -156 ± 76 |
| MCD | LR+D | 4 | 64 + 1 | $\mathbb{E}[W]$ | ✗ | - | 68 | -9 ± 1947 | -144 ± 438 | 548 ± 183 | -169 ± 105 |
| MCD | LR+D | 8 | 8 + 0 | - | ✗ | - | 72 | 58 ± 1060 | -139 ± 355 | 576 ± 150 | -173 ± 90 |
| MCD | LR+D | 8 | 16 + 0 | - | ✗ | - | 144 | 75 ± 812 | -63 ± 268 | 596 ± 114 | -167 ± 80 |
| MCD | LR+D | 8 | 24 + 0 | - | ✗ | - | 216 | 82 ± 691 | -23 ± 232 | 607 ± 92 | -164 ± 75 |
| MCD | LR+D | 8 | 32 + 0 | - | ✗ | - | 288 | 86 ± 623 | 0 ± 211 | 614 ± 83 | -162 ± 73 |
| MCD | LR+D | 8 | 40 + 0 | - | ✗ | - | 360 | 91 ± 551 | 16 ± 197 | 618 ± 77 | -161 ± 72 |
| MCD | LR+D | 8 | 48 + 0 | - | ✗ | - | 432 | 94 ± 507 | 26 ± 188 | 622 ± 73 | -159 ± 71 |
| MCD | LR+D | 8 | 56 + 0 | - | ✗ | - | 504 | 96 ± 470 | 34 ± 182 | 625 ± 69 | -159 ± 71 |
| MCD | LR+D | 8 | 64 + 0 | - | ✓ | ✗ | 8 | 48 ± 1341 | -294 ± 556 | 519 ± 241 | -186 ± 109 |
| MCD | LR+D | 8 | 64 + 0 | - | ✓ | ✓ | 8 | 95 ± 78 | -194 ± 281 | 521 ± 138 | -194 ± 84 |
| MCD | LR+D | 8 | 64 + 0 | - | ✓ | ✗ | 16 | 53 ± 1255 | -211 ± 439 | 550 ± 203 | -178 ± 98 |
| MCD | LR+D | 8 | 64 + 0 | - | ✓ | ✓ | 16 | 96 ± 88 | -152 ± 263 | 548 ± 139 | -186 ± 82 |
| MCD | LR+D | 8 | 64 + 0 | - | ✓ | ✗ | 32 | 60 ± 1098 | -143 ± 368 | 571 ± 171 | -172 ± 87 |
| MCD | LR+D | 8 | 64 + 0 | - | ✓ | ✓ | 32 | 101 ± 102 | -109 ± 249 | 567 ± 129 | -179 ± 76 |
| MCD | LR+D | 8 | 64 + 0 | - | ✓ | ✗ | 64 | 72 ± 883 | -82 ± 311 | 589 ± 141 | -167 ± 80 |
| MCD | LR+D | 8 | 64 + 0 | - | ✓ | ✓ | 64 | 105 ± 158 | -65 ± 239 | 585 ± 116 | -173 ± 73 |
| MCD | LR+D | 8 | 64 + 0 | - | ✓ | ✗ | 128 | 84 ± 662 | -28 ± 253 | 604 ± 96 | -163 ± 75 |
| MCD | LR+D | 8 | 64 + 0 | - | ✓ | ✓ | 128 | 103 ± 308 | -22 ± 223 | 602 ± 87 | -167 ± 72 |
| MCD | LR+D | 8 | 64 + 0 | - | ✓ | ✗ | 256 | 95 ± 482 | 16 ± 207 | 617 ± 78 | -160 ± 72 |
| MCD | LR+D | 8 | 64 + 0 | - | ✓ | ✓ | 256 | 97 ± 446 | 17 ± 201 | 616 ± 75 | -161 ± 71 |
| MCD | LR+D | 8 | 64 + 0 | - | ✓ | ✗ | 512 | 98 ± 439 | 39 ± 179 | 626 ± 68 | -158 ± 70 |
| MCD | LR+D | 8 | 64 + 0 | - | ✓ | ✓ | 512 | 98 ± 439 | 39 ± 179 | 626 ± 68 | -158 ± 70 |
| MCD | LR+D | 8 | 64 + 0 | - | ✗ | - | 576 | 98 ± 439 | 40 ± 177 | 627 ± 67 | -158 ± 70 |
| MCD | LR+D | 8 | 72 + 0 | - | ✗ | - | 648 | 99 ± 419 | 45 ± 174 | 629 ± 66 | -158 ± 70 |
| MCD | LR+D | 8 | 80 + 0 | - | ✗ | - | 720 | 100 ± 406 | 48 ± 171 | 631 ± 64 | -157 ± 69 |
| MCD | LR+D | 8 | 88 + 0 | - | ✗ | - | 792 | 101 ± 394 | 51 ± 168 | 632 ± 63 | -157 ± 69 |
| MCD | LR+D | 8 | 96 + 0 | - | ✗ | - | 864 | 101 ± 381 | 54 ± 166 | 633 ± 62 | -156 ± 69 |
| MCD | LR+D | 8 | 104 + 0 | - | ✗ | - | 936 | 102 ± 372 | 56 ± 164 | 634 ± 61 | -156 ± 69 |
| MCD | LR+D | 8 | 112 + 0 | - | ✗ | - | 1008 | 102 ± 364 | 58 ± 163 | 635 ± 60 | -156 ± 68 |
| MCD | LR+D | 8 | 120 + 0 | - | ✗ | - | 1080 | 103 ± 357 | 60 ± 162 | 636 ± 59 | -156 ± 68 |
| MCD | LR+D | 8 | 128 + 0 | - | ✗ | - | 1152 | 103 ± 351 | 61 ± 160 | 637 ± 59 | -155 ± 68 |
| MCD | LR+D | 8 | 64 + 1 | $\mathbb{E}[W]$ | ✗ | - | 72 | -8 ± 1825 | -120 ± 415 | 565 ± 203 | -170 ± 100 |
| MCD | LR+D | 12 | 64 + 0 | - | ✓ | ✗ | 8 | 58 ± 1327 | -308 ± 596 | 475 ± 296 | -184 ± 93 |
| MCD | LR+D | 12 | 64 + 0 | - | ✓ | ✓ | 8 | 87 ± 59 | -233 ± 93 | 366 ± 55 | -206 ± 68 |

Table 9 – *Continued on next page*

Table 9 – *Continued from previous page*

| Epis | Param | $R^W$ | $T$ | $P^a$ | TSVD | $\hat{D}$ | $R$ | MNIST Inpainting ×10 | CelebA Inpainting ×100 | CelebA Colorization ×1000 | Flying Chairs Optical Flow ×1000 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| MCD | LR+D | 12 | 64 + 0 | - | ✓ | ✗ | 12 | 61 ± 1299 | -252 ± 530 | 522 ± 255 | -178 ± 89 |
| MCD | LR+D | 12 | 64 + 0 | - | ✓ | ✓ | 12 | 100 ± 68 | -151 ± 236 | 504 ± 58 | -194 ± 71 |
| MCD | LR+D | 12 | 64 + 0 | - | ✓ | ✗ | 16 | 64 ± 1222 | -216 ± 498 | 533 ± 243 | -174 ± 86 |
| MCD | LR+D | 12 | 64 + 0 | - | ✓ | ✓ | 16 | 103 ± 80 | -135 ± 236 | 526 ± 76 | -190 ± 70 |
| MCD | LR+D | 12 | 64 + 0 | - | ✓ | ✗ | 24 | 69 ± 1094 | -176 ± 463 | 550 ± 219 | -170 ± 83 |
| MCD | LR+D | 12 | 64 + 0 | - | ✓ | ✓ | 24 | 104 ± 97 | -114 ± 232 | 543 ± 84 | -185 ± 69 |
| MCD | LR+D | 12 | 64 + 0 | - | ✓ | ✗ | 32 | 74 ± 975 | -148 ± 432 | 563 ± 204 | -168 ± 81 |
| MCD | LR+D | 12 | 64 + 0 | - | ✓ | ✓ | 32 | 105 ± 106 | -98 ± 229 | 555 ± 87 | -182 ± 68 |
| MCD | LR+D | 12 | 64 + 0 | - | ✓ | ✗ | 48 | 79 ± 856 | -110 ± 385 | 579 ± 179 | -165 ± 79 |
| MCD | LR+D | 12 | 64 + 0 | - | ✓ | ✓ | 48 | 107 ± 123 | -75 ± 226 | 571 ± 89 | -178 ± 68 |
| MCD | LR+D | 12 | 64 + 0 | - | ✓ | ✗ | 64 | 83 ± 788 | -84 ± 354 | 590 ± 134 | -163 ± 77 |
| MCD | LR+D | 12 | 64 + 0 | - | ✓ | ✓ | 64 | 108 ± 148 | -58 ± 224 | 582 ± 81 | -174 ± 68 |
| MCD | LR+D | 12 | 64 + 0 | - | ✓ | ✗ | 96 | 88 ± 696 | -48 ± 302 | 604 ± 99 | -160 ± 75 |
| MCD | LR+D | 12 | 64 + 0 | - | ✓ | ✓ | 96 | 109 ± 196 | -33 ± 219 | 597 ± 71 | -169 ± 68 |
| MCD | LR+D | 12 | 64 + 0 | - | ✓ | ✗ | 128 | 91 ± 624 | -23 ± 273 | 613 ± 84 | -158 ± 74 |
| MCD | LR+D | 12 | 64 + 0 | - | ✓ | ✓ | 128 | 109 ± 245 | -14 ± 215 | 607 ± 67 | -166 ± 69 |
| MCD | LR+D | 12 | 64 + 0 | - | ✓ | ✗ | 192 | 96 ± 540 | 7 ± 239 | 624 ± 69 | -156 ± 73 |
| MCD | LR+D | 12 | 64 + 0 | - | ✓ | ✓ | 192 | 106 ± 345 | 11 ± 208 | 620 ± 62 | -161 ± 69 |
| MCD | LR+D | 12 | 64 + 0 | - | ✓ | ✗ | 256 | 99 ± 482 | 27 ± 217 | 631 ± 63 | -155 ± 72 |
| MCD | LR+D | 12 | 64 + 0 | - | ✓ | ✓ | 256 | 103 ± 413 | 28 ± 200 | 629 ± 59 | -158 ± 70 |
| MCD | LR+D | 12 | 64 + 0 | - | ✓ | ✗ | 384 | 102 ± 435 | 49 ± 188 | 639 ± 56 | -153 ± 71 |
| MCD | LR+D | 12 | 64 + 0 | - | ✓ | ✓ | 384 | 102 ± 434 | 49 ± 184 | 638 ± 55 | -154 ± 70 |
| MCD | LR+D | 12 | 64 + 0 | - | ✓ | ✗ | 512 | 102 ± 433 | 60 ± 174 | 644 ± 53 | -152 ± 71 |
| MCD | LR+D | 12 | 64 + 0 | - | ✓ | ✓ | 512 | 102 ± 433 | 60 ± 173 | 644 ± 52 | -153 ± 70 |
| MCD | LR+D | 12 | 64 + 0 | - | ✓ | ✗ | 768 | 102 ± 433 | 65 ± 165 | 649 ± 49 | -152 ± 70 |
| MCD | LR+D | 12 | 64 + 0 | - | ✓ | ✓ | 768 | 102 ± 433 | 65 ± 165 | 649 ± 49 | -152 ± 70 |
| MCD | LR+D | 12 | 64 + 1 | $\mathbb{E}[W]$ | ✗ | - | 76 | 9 ± 1703 | -103 ± 410 | 575 ± 199 | -162 ± 83 |
| MCD | LR+D | 16 | 64 + 0 | - | ✓ | ✗ | 16 | 77 ± 1285 | -207 ± 452 | 529 ± 267 | -172 ± 98 |
| MCD | LR+D | 16 | 64 + 0 | - | ✓ | ✓ | 16 | 106 ± 147 | -129 ± 155 | 535 ± 72 | -192 ± 71 |
| MCD | LR+D | 16 | 64 + 0 | - | ✓ | ✗ | 32 | 85 ± 1023 | -151 ± 365 | 575 ± 168 | -163 ± 92 |
| MCD | LR+D | 16 | 64 + 0 | - | ✓ | ✓ | 32 | 112 ± 163 | -102 ± 167 | 568 ± 76 | -182 ± 70 |
| MCD | LR+D | 16 | 64 + 0 | - | ✓ | ✗ | 64 | 94 ± 779 | -91 ± 303 | 603 ± 118 | -157 ± 87 |
| MCD | LR+D | 16 | 64 + 0 | - | ✓ | ✓ | 64 | 113 ± 173 | -65 ± 177 | 594 ± 72 | -172 ± 71 |
| MCD | LR+D | 16 | 64 + 0 | - | ✓ | ✗ | 128 | 101 ± 607 | -31 ± 249 | 624 ± 88 | -152 ± 82 |
| MCD | LR+D | 16 | 64 + 0 | - | ✓ | ✓ | 128 | 115 ± 238 | -21 ± 184 | 617 ± 68 | -162 ± 72 |
| MCD | LR+D | 16 | 64 + 0 | - | ✓ | ✗ | 256 | 108 ± 455 | 22 ± 201 | 641 ± 68 | -147 ± 78 |
| MCD | LR+D | 16 | 64 + 0 | - | ✓ | ✓ | 256 | 111 ± 377 | 24 ± 183 | 638 ± 62 | -152 ± 73 |
| MCD | LR+D | 16 | 64 + 0 | - | ✓ | ✗ | 512 | 111 ± 387 | 58 ± 166 | 655 ± 55 | -144 ± 76 |
| MCD | LR+D | 16 | 64 + 0 | - | ✓ | ✓ | 512 | 111 ± 386 | 58 ± 164 | 654 ± 54 | -146 ± 74 |
| MCD | LR+D | 16 | 64 + 0 | - | ✓ | ✗ | 1024 | 111 ± 386 | 67 ± 152 | 663 ± 49 | -143 ± 74 |
| MCD | LR+D | 16 | 64 + 0 | - | ✓ | ✓ | 1024 | 111 ± 386 | 67 ± 152 | 663 ± 49 | -143 ± 74 |
| MCD | LR+D | 16 | 64 + 1 | $\mathbb{E}[W]$ | ✗ | - | 80 | 44 ± 1419 | -83 ± 359 | 581 ± 195 | -157 ± 97 |
| SVI | D | 0 | 64 + 0 | - | - | - | 0 | -159 ± 3636 | -439 ± 621 | 340 ± 254 | -227 ± 104 |
| SVI | D | 0 | 64 + 1 | $\mathbb{E}[W]$ | - | - | 0 | -197 ± 3967 | -457 ± 619 | 333 ± 266 | -229 ± 105 |
| SVI | LR+D | 8 | 8 + 0 | - | ✗ | - | 72 | 68 ± 952 | -268 ± 500 | 565 ± 175 | -167 ± 102 |
| SVI | LR+D | 8 | 16 + 0 | - | ✗ | - | 144 | 76 ± 820 | -167 ± 382 | 594 ± 128 | -160 ± 95 |
| SVI | LR+D | 8 | 24 + 0 | - | ✗ | - | 216 | 82 ± 717 | -115 ± 331 | 610 ± 106 | -156 ± 91 |
| SVI | LR+D | 8 | 32 + 0 | - | ✗ | - | 288 | 85 ± 665 | -84 ± 305 | 620 ± 90 | -154 ± 89 |
| SVI | LR+D | 8 | 40 + 0 | - | ✗ | - | 360 | 88 ± 632 | -63 ± 289 | 628 ± 82 | -152 ± 87 |
| SVI | LR+D | 8 | 48 + 0 | - | ✗ | - | 432 | 89 ± 613 | -49 ± 277 | 633 ± 75 | -151 ± 86 |
| SVI | LR+D | 8 | 56 + 0 | - | ✗ | - | 504 | 90 ± 588 | -38 ± 269 | 637 ± 70 | -150 ± 85 |
| SVI | LR+D | 8 | 64 + 0 | - | ✓ | ✗ | 8 | 55 ± 1238 | -472 ± 842 | 481 ± 324 | -183 ± 116 |
| SVI | LR+D | 8 | 64 + 0 | - | ✓ | ✓ | 8 | 109 ± 261 | -311 ± 388 | 483 ± 64 | -191 ± 101 |
| SVI | LR+D | 8 | 64 + 0 | - | ✓ | ✗ | 16 | 58 ± 1176 | -424 ± 782 | 516 ± 273 | -174 ± 109 |
| SVI | LR+D | 8 | 64 + 0 | - | ✓ | ✓ | 16 | 107 ± 316 | -301 ± 413 | 536 ± 156 | -182 ± 98 |
| SVI | LR+D | 8 | 64 + 0 | - | ✓ | ✗ | 32 | 64 ± 1056 | -356 ± 707 | 553 ± 218 | -168 ± 103 |
| SVI | LR+D | 8 | 64 + 0 | - | ✓ | ✓ | 32 | 107 ± 329 | -269 ± 417 | 560 ± 149 | -174 ± 94 |
| SVI | LR+D | 8 | 64 + 0 | - | ✓ | ✗ | 64 | 71 ± 922 | -266 ± 588 | 583 ± 157 | -162 ± 98 |
| SVI | LR+D | 8 | 64 + 0 | - | ✓ | ✓ | 64 | 101 ± 430 | -218 ± 413 | 584 ± 123 | -166 ± 91 |
| SVI | LR+D | 8 | 64 + 0 | - | ✗ | - | 576 | 91 ± 573 | -29 ± 263 | 641 ± 66 | -149 ± 84 |
| SVI | LR+D | 8 | 72 + 0 | - | ✗ | - | 648 | 92 ± 561 | -22 ± 258 | 644 ± 63 | -148 ± 84 |
| SVI | LR+D | 8 | 80 + 0 | - | ✗ | - | 720 | 92 ± 552 | -16 ± 253 | 646 ± 60 | -148 ± 83 |
| SVI | LR+D | 8 | 88 + 0 | - | ✗ | - | 792 | 93 ± 547 | -12 ± 250 | 648 ± 58 | -147 ± 83 |
| SVI | LR+D | 8 | 96 + 0 | - | ✗ | - | 864 | 93 ± 542 | -8 ± 247 | 650 ± 57 | -147 ± 83 |
| SVI | LR+D | 8 | 104 + 0 | - | ✗ | - | 936 | 93 ± 539 | -4 ± 245 | 652 ± 55 | -146 ± 82 |
| SVI | LR+D | 8 | 112 + 0 | - | ✗ | - | 1008 | 94 ± 535 | -2 ± 242 | 653 ± 54 | -146 ± 82 |
| SVI | LR+D | 8 | 120 + 0 | - | ✗ | - | 1080 | 94 ± 530 | 1 ± 241 | 654 ± 53 | -146 ± 82 |
| SVI | LR+D | 8 | 64 + 1 | $\mathbb{E}[W]$ | ✗ | - | 72 | 41 ± 1383 | -243 ± 599 | 558 ± 222 | -164 ± 103 |
| DE | D | 0 | 64 + 0 | - | - | - | 0 | 81 ± 873 | -249 ± 403 | 374 ± 159 | -213 ± 72 |
| DE | LR+D | 8 | 8 + 0 | - | ✗ | - | 72 | 60 ± 1025 | -124 ± 352 | 578 ± 130 | -171 ± 86 |
| DE | LR+D | 8 | 16 + 0 | - | ✗ | - | 144 | 73 ± 827 | -54 ± 266 | 597 ± 103 | -166 ± 78 |
| DE | LR+D | 8 | 24 + 0 | - | ✗ | - | 216 | 82 ± 687 | -17 ± 231 | 608 ± 88 | -163 ± 75 |
| DE | LR+D | 8 | 32 + 0 | - | ✗ | - | 288 | 97 ± 486 | 13 ± 205 | 617 ± 75 | -160 ± 72 |
| DE | LR+D | 8 | 40 + 0 | - | ✗ | - | 360 | 101 ± 409 | 29 ± 193 | 623 ± 68 | -159 ± 71 |
| DE | LR+D | 8 | 48 + 0 | - | ✗ | - | 432 | 103 ± 369 | 40 ± 185 | 627 ± 65 | -158 ± 70 |

Table 9 – *Continued on next page*

Table 9 – *Continued from previous page*

| Epis | Param | $R^W$ | $T$ | $P^a$ | TSVD | $\hat{D}$ | $R$ | MNIST Inpainting ×10 | CelebA Inpainting ×100 | CelebA Colorization ×1000 | Flying Chairs Optical Flow ×1000 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| DE | LR+D | 8 | 56 + 0 | - | ✗ | - | 504 | 105 ± 319 | 49 ± 177 | 629 ± 61 | -157 ± 69 |
| DE | LR+D | 8 | 64 + 0 | - | ✓ | ✗ | 8 | 79 ± 1006 | -244 ± 519 | 529 ± 224 | -178 ± 83 |
| DE | LR+D | 8 | 64 + 0 | - | ✓ | ✓ | 8 | 80 ± 28 | -182 ± 115 | 479 ± 70 | -205 ± 62 |
| DE | LR+D | 8 | 64 + 0 | - | ✓ | ✗ | 16 | 85 ± 864 | -198 ± 461 | 554 ± 190 | -173 ± 80 |
| DE | LR+D | 8 | 64 + 0 | - | ✓ | ✓ | 16 | 85 ± 33 | -128 ± 191 | 538 ± 92 | -190 ± 66 |
| DE | LR+D | 8 | 64 + 0 | - | ✓ | ✗ | 32 | 92 ± 669 | -134 ± 384 | 576 ± 148 | -168 ± 76 |
| DE | LR+D | 8 | 64 + 0 | - | ✓ | ✓ | 32 | 93 ± 42 | -97 ± 230 | 568 ± 99 | -180 ± 67 |
| DE | LR+D | 8 | 64 + 0 | - | ✓ | ✗ | 64 | 98 ± 498 | -67 ± 308 | 596 ± 101 | -164 ± 73 |
| DE | LR+D | 8 | 64 + 0 | - | ✓ | ✓ | 64 | 102 ± 71 | -51 ± 221 | 590 ± 82 | -172 ± 67 |
| DE | LR+D | 8 | 64 + 0 | - | ✗ | - | 576 | 107 ± 275 | 54 ± 170 | 631 ± 60 | -157 ± 68 |
| DE | LR+D | 8 | 72 + 0 | - | ✗ | - | 648 | 108 ± 258 | 58 ± 165 | 633 ± 58 | -157 ± 68 |
| DE | LR+D | 8 | 80 + 0 | - | ✗ | - | 720 | 109 ± 244 | 62 ± 162 | 634 ± 57 | -157 ± 67 |
| DE | LR+D | 8 | 88 + 0 | - | ✗ | - | 792 | 110 ± 232 | 66 ± 158 | 635 ± 55 | -157 ± 67 |
| DE | LR+D | 8 | 96 + 0 | - | ✗ | - | 864 | 110 ± 225 | 70 ± 156 | 636 ± 54 | -157 ± 67 |
| DE | LR+D | 8 | 104 + 0 | - | ✗ | - | 936 | 111 ± 221 | 72 ± 155 | 637 ± 53 | -157 ± 67 |
| DE | LR+D | 8 | 112 + 0 | - | ✗ | - | 1008 | 112 ± 210 | 75 ± 148 | 637 ± 52 | -157 ± 66 |
| DE | LR+D | 8 | 120 + 0 | - | ✗ | - | 1080 | 112 ± 206 | 77 ± 145 | 638 ± 52 | -157 ± 66 |

Table 9: Extended Ablation. We compare non-Bayesian networks with aleatoric uncertainty only and various Bayesian networks with both kinds of uncertainties and various hyperparameters.

### D.4 Eigenvalues Distribution

The figure shows the normalized eigenvalues of the low-rank matrix $PP^\top$ and the diagonal matrix $D$ (colored lines), along with a minimum diagonal threshold (dashed line), for four datasets (columns) and three different approximation methods (rows). Both axes—eigenvalue magnitude and eigenvalue index—are displayed on a logarithmic scale, with eigenvalues sorted from largest to smallest.

The top row corresponds to the expected weights $\mathbb{E}[w]$ approximation (similar to Zepf et al. (2023)), the middle row shows truncated SVD, and the bottom row displays the naive approach without truncation. The diagonal elements for the naive and expected weights rows are identical, whereas in the truncated SVD row, the diagonal is updated post-truncation.

The number of eigenvalues of the low-rank matrix depends on the rank (number of columns in $P$) and thus varies across approximation methods. The minimum diagonal entry acts as a lower bound for the diagonals in the non-updated rows (top and bottom), and the smallest eigenvalues approach this minimum. This effect is particularly pronounced in the MNIST dataset, where some border pixels have high certainty, leading to very low predicted variances. To maintain training stability, a minimum diagonal threshold is enforced.

### D.5 Low Variance Analysis

Figure 12 shows that several entries on the diagonal of the matrix $D$, predicted during MNIST inpainting, are limited by the lower bound $\epsilon$. This implies that the model's predicted uncertainty for many pixels is artificially constrained to be higher, rather than reflecting actual confidence levels.

Further insight is provided by Figure 13, which visualizes the eigenvalues close to $\epsilon$. These values suggest that, in such cases, the model's uncertainty is defined more by the engineered lower bound than by the data itself. The bottom row highlights in grey the pixels whose corresponding entries in $D$ lie close to this minimum.

A clear spatial trend emerges: the affected pixels are mostly located near the image borders. This indicates that the model is artificially constrained to be less confident in these regions, likely due to reduced contextual information or edge effects during training and inference.

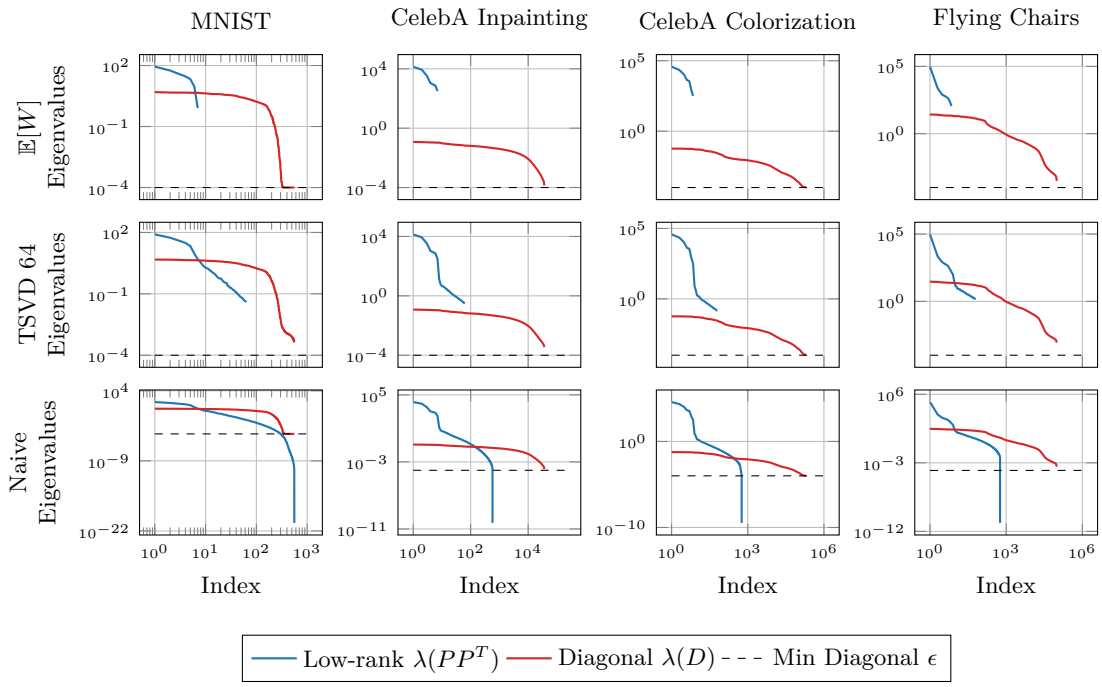


Figure 12: Normalized eigenvalues of low-rank and diagonal matrices across datasets and approximation methods.

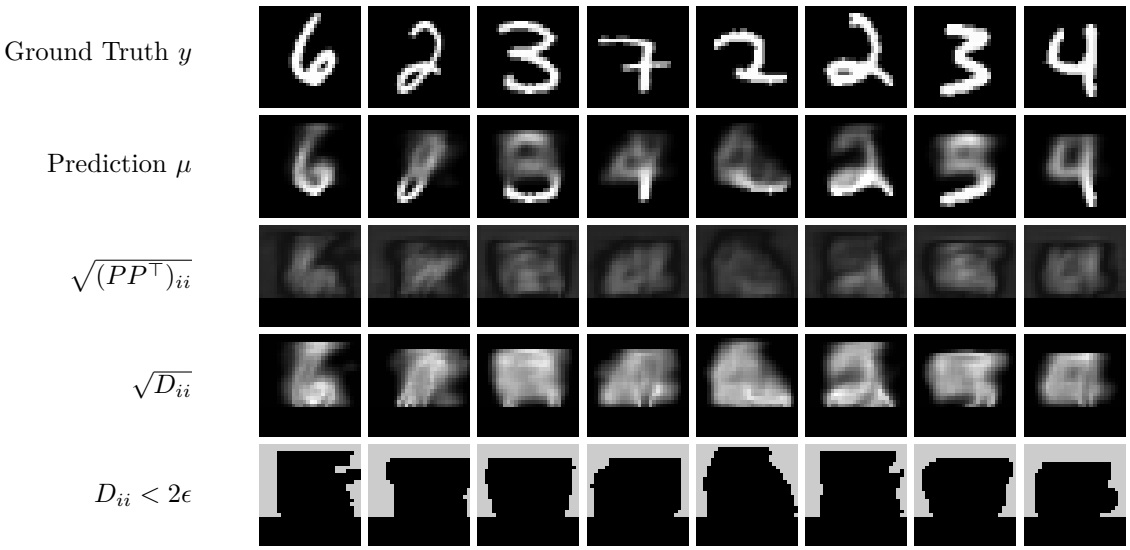


Figure 13: Analysis of very low entries on the diagonal of $D$, approaching the $\epsilon$ hyperparameter, which acts as a lower bound for $\lambda_i(D)$. Rows show: the ground truth, the prediction, the aggregated standard deviation estimated from the columns of $P$, the per-pixel standard deviation from $D$, and a mask marking pixels where $D$ is close to $\epsilon$. Notably, pixels near the image borders tend to fall into this constrained region, indicating artificially elevated uncertainty.

## E Algorithm

The algorithm constructs a low-rank plus diagonal covariance matrix using Monte Carlo sampling from a proxy posterior distribution over network weights. It begins by performing $T$ forward passes, each with weights sampled from the proxy distribution $q_\theta^*$. For each pass, the predictive mean $\mu_{w_i}(x)$, aleatoric uncertainty factor $P_{w_i}^a(x)$, and diagonal covariance $D_{w_i}(x)$ are computed. These estimates are then aggregated by averaging over samples to obtain $\mu(x)$ and $D(x)$.

Next, the epistemic uncertainty factor $P^e(x)$ is constructed by stacking the centered deviations of the predictive means from their average, scaled appropriately. The aleatoric factor $P^a(x)$ is either averaged or stacked similarly to capture inherent noise uncertainty. The total low-rank factor matrix $P(x)$ is formed by concatenating these epistemic and aleatoric components.

Optionally, truncated SVD can be applied to $P(x)$ to reduce its rank and improve computational efficiency. The final covariance estimate $\Sigma(x)$ is assembled as the sum of the low-rank product $P(x)P(x)^\top$ and the diagonal $D(x)$. This low-rank decomposition enables scalable and expressive uncertainty quantification by efficiently combining both epistemic and aleatoric sources.

**Algorithm 2** LR+D Covariance Construction with MC Sampling

**Require:** Proxy posterior $q_\theta^*$
**Require:** Input $x$
**Ensure:** Mean $\mu(x)$, diagonal $D(x)$, low rank factor $P(x)$, covariance $\Sigma(x)$

// **Step 1: Monte Carlo Sampling**
**for** $i = 1$ **to** $T$ **do**
  Sample weights $w_i \sim q_\theta^*$
  Compute $\mu_{w_i}(x)$, $P_{w_i}(x)$, $D_{w_i}(x)$
**end for**
// **Step 2: Aggregate Means**
$\mu(x) = \frac{1}{T}\sum_{i=1}^{T} \mu_{w_i}(x)$
$D(x) = \frac{1}{T}\sum_{i=1}^{T} D_{w_i}(x)$
// **Step 3: Epistemic Factor**
$P^e(x) = \frac{1}{\sqrt{T-1}} [\mu_{w_1}(x) - \mu(x) \quad \mu_{w_2}(x) - \mu(x) \quad ... \quad \mu_{w_T}(x) - \mu(x)]$
// **Step 4: Aleatoric Factor**
$P^a(x) = \frac{1}{T} [P_{w_1}(x) \quad P_{w_2}(x) \quad ... \quad P_{w_T}(x)]$
// **Step 5: Combine Factors**
$P(x) \leftarrow [P^a(x) \quad P^e(x)]$
// **Step 6: Optional SVD**
**if** use SVD **then**
  $[U, S, \_] = \text{svd}(P(x))$
  $\hat{P}(x) \leftarrow U_{:,1:r} S_{1:r,1:r}$
  $\hat{D}_{ii}(x) \leftarrow D_{ii}(x) + \sum_{j=r+1}^{R} U_{ij}^2 S_{jj}^2$
**end if**
// **Final Covariance**
$\Sigma(x) = P(x)P(x)^\top + D(x)$

## F Derivations in Detail

### F.1 Exploiting LR+D for efficient computation of matrix determinant and inverse

Both the likelihood function $p(y|x, w) = \mathcal{N}\left(\mu_W^a(x), \Sigma_W^a(x)\right)$ as well as the approximate posterior predictive distribution $p(y|x, \mathcal{D}) \approx \mathcal{N}\left(\mu(x), \Sigma(x)\right)$ are multivariate normal distributions parametrized by covariance matrices $\Sigma_W^a$ and $\Sigma$, respectively, where in the following, we only consider $\Sigma$ for clarity. Denoting by $S$ the

output dimension, the normal distribution is then defined as

$$\mathcal{N}\left(\mu\left(x\right), \Sigma\left(x\right)\right) = \frac{1}{\sqrt{|\Sigma(x)|(2\pi)^S}} \exp\left(-\frac{1}{2}(\mu(x)-y)^\top \Sigma^{-1}(x)(\mu(x)-y)\right) \tag{15}$$

which requires computation of the covariance matrix' determinant $|\Sigma|$ and inverse $\Sigma^{-1}$ for sampling and evaluation of the log-likelihood. For full covariance matrices $\Sigma \in \mathbb{R}^{S\times S}$ with large $S$, these are very expensive, if not impossible, to compute directly. Instead, we exploit our LR+D representation for efficient computation of the matrix determinant and inverse.

We compute the determinant as

$$\begin{aligned} |\Sigma| &= |D + PP^\top| && (16)\\ &= |I_R + P^\top D^{-1} P||D| && (17)\\ &= |C||D| && (18) \end{aligned}$$

where we first substituted $\Sigma$ with its LR+D representation and subsequently applied the matrix determinant lemma. With $D \in \mathbb{R}^{S\times S}$, $P \in \mathbb{R}^{S\times R}$ and $I_R \in \mathbb{R}^R$, the so-called capacitance $C = I_R + P^\top D^{-1} P$ is an $R \times R$ matrix. Since $R \ll S$, the determinant of the capacitance matrix is very cheap to compute.

To compute the inverse, we use the Woodbury matrix identity, again by exploiting the LR+D representation.

$$\begin{aligned} \Sigma^{-1} &= (D + PP^\top)^{-1} && (19)\\ &= D^{-1} - D^{-1}P(I_R + P^\top D^{-1} P)^{-1} P^\top D^{-1} && (20)\\ &= D^{-1} - D^{-1} P C^{-1} P^\top D^{-1} && (21) \end{aligned}$$

As before, the capacitance matrix $C \in \mathbb{R}^{R\times R}$ is very small and thus its inverse easy to compute.

### F.2 Full derivation of SVD

We apply dimensionality reduction using SVD on our tall $P$ matrices. This involves decomposing into three separate matrices: $U$, $\Psi$, and $V^\top$. The $U$ matrix represents an arbitrary not further used rotation, $\Psi$ is a diagonal matrix containing the singular values, and $V^\top$ contains the columns of the transformed matrix.

By selecting the top $R$ singular values and corresponding vectors, we can approximate the original matrix. This approximation is achieved by truncating the matrices $U$ and $V^\top$ to retain only the top $R$ singular values and vectors. This reduces the dimensionality of the data while preserving its essential structure.

The reduced dimensionality representation, denoted as $\hat{P}$, is computed by taking the product of the truncated matrices $V$ and $\Psi$. Additionally, a diagonal matrix $\hat{D}$ captures the by the dimensionality reduction removed variance of $PP^\top$, with each element representing the contribution of the omitted singular values to the overall uncertainty. We use $\hat{D}$ to update our diagonal for the final LR+D representation.

$$\begin{aligned} P^\top &= U\Psi V^\top && (22)\\ PP^\top &= (U\Psi V^\top)^\top (U\Psi V^\top) && (23)\\ &= V\Psi U^\top U\Psi V^\top && (24)\\ &= V\Psi\Psi V^\top && (25)\\ \hat{\Sigma} &= \hat{D} + \hat{P}\hat{P}^\top && (26)\\ \hat{P} &= \begin{bmatrix} V_{R-\hat{R}} \cdot \Psi_{R-\hat{R},R-\hat{R}} & \dots & V_R \cdot \Psi_{R,R} \end{bmatrix} && (27)\\ \hat{D}_{ii} &= \sum_{j=1}^{R-\hat{R}-1} V_{ij}^2 \cdot \Psi_{j,j}^2 && (28) \end{aligned}$$

### F.3 Loss Definition

For regression problems we intend to maximize the data likelihood $p(\boldsymbol{y}|\boldsymbol{x}, w) = \prod_i p(y_i|x_i, w)$, where we assumed all dataset samples to be i.i.d. Equivalently, we can minimize the negative log-likelihood $p(\boldsymbol{y}|\boldsymbol{x}, w) = \sum_i -\log p(y_i|x_i, w)$. We further assume the network predictions to be distributed around the true value $y$ following a Gaussian distribution with mean $\mu_w(x)$ and covariance $\Sigma_w(x)$.

The subscript $()_w$ implies given weights, as in a classical frequentist interpretation of neural networks. To account for the epistemic part of uncertainty, this weight is not fixed, and a probabilistic interpretation is used, implying a $p(w)$. Later, we approximate the posterior of $p(w|\mathcal{D})$ by using Monte Carlo integration. If we drop $\sigma_w$ as nuisance parameter, we would recover a standard squared error loss as it is often used in regression.

The loss function for a single training sample can then be defined as

$$\begin{aligned}
\mathcal{L}_{lrd} &= -\frac{1}{S}\log p(y|x, w) \\
&= -\frac{1}{S}\log\left(\frac{1}{\sqrt{|\Sigma_w|(2\pi)^S}}\exp\left(-\frac{1}{2}(\mu_w - y)^\top \Sigma_w^{-1}(\mu_w - y)\right)\right) \\
&= \frac{1}{S}\left(\log\sqrt{|\Sigma_w|(2\pi)^S} + \frac{1}{2}(\mu_w - y)^\top \Sigma_w^{-1}(\mu_w - y)\right) \\
&= \frac{1}{S}\left(\frac{1}{2}\log|\Sigma_w| + \frac{S}{2}\log(2\pi) + \frac{1}{2}(\mu_w - y)^\top \Sigma_w^{-1}(\mu_w - y)\right)
\end{aligned}$$

where we normalized by the output dimensionality $S$.

Dropping constant terms, we are left with:

$$\mathcal{L}_{lrd} = \frac{1}{2S}\log|\Sigma_w| + \frac{1}{2S}(\mu_w - y)^\top \Sigma_w^{-1}(\mu_w - y)$$

We can see that evaluating $\mathcal{L}$ involves computing the determinant and inverse of the covariance matrix. To achieve this, we exploit our LR+D representation as described in the previous section F.1

### F.4 Full derivation of mean vector and covariance matrix

The expectation of the posterior predictive distribution is given by:

$$\begin{aligned}
\mathbb{E}[y|x, \mathcal{D}] &= \mathbb{E}_{p(w|\mathcal{D})}\left[\mathbb{E}\left[y|x, w\right]\right] && (29)\\
&\approx \mathbb{E}_{q_\theta^*}\left[\mathbb{E}\left[y|x, w\right]\right] && (30)\\
&= \mathbb{E}_{q_\theta^*}\left[\mu_W^a\left(x\right)\right] && (31)\\
&\approx \frac{1}{T}\sum_i^T \mu_{w_i}^a(x) \quad w_i \sim q_\theta^* && (32)\\
&= \mu(x) && (33)
\end{aligned}$$

The covariance of the posterior predictive distribution is given by:

$$\begin{aligned}
\mathrm{Cov}\left[y|\, x, \mathcal{D}\right] &= \mathrm{Cov}_{p(w|\mathcal{D})}\left[\mathbb{E}_{p(w|\mathcal{D})}\left[y|x, w\right]\right] + \mathbb{E}_{p(w|\mathcal{D})}\left[\mathrm{Cov}\left[y|x, w\right]\right] && (34)\\
&\approx \mathrm{Cov}_{q_\theta^*}\left[\mathbb{E}_{q_\theta^*}\left[y|x, w\right]\right] + \mathbb{E}_{q_\theta^*}\left[\mathrm{Cov}\left[y|x, w\right]\right] && (35)\\
&= \underbrace{\mathrm{Cov}_{q_\theta^*}\left[\mu_W^a(x)\right]}_{epistemic} + \underbrace{\mathbb{E}_{q_\theta^*}\left[\Sigma_W^a(x)\right]}_{aleatoric} && (36)\\
&\approx \Sigma^e(x) + \Sigma^a(x) && (37)\\
&= \Sigma(x) && (38)
\end{aligned}$$

$$\Sigma^e(x) = \frac{1}{T-1}\sum_i^T \left(\mu_{w_i}(x) - \mu(x)\right)\left(\mu_{w_i}(x) - \mu(x)\right)^\top \quad w_i \sim q_\theta^* \tag{39}$$

$$\Sigma^a(x) = \frac{1}{T}\sum_i^T \Sigma_{w_i}(x) \quad w_i \sim q_\theta^*\,. \tag{40}$$

In above transformations of expectation and variance, we applied the law of total expectation or variance, respectively, and subsequently approximated them using the proxy distribution $q_\theta^\star(w)$. The expectation over $y$, denoted by $\mathbb{E}[y|x,w]$, is given by the mean of the predicted normal distribution $\mu_W^a(x)$, whereas the covariance over $y$, denoted as $\mathrm{Cov}[y|x,w]$, is given by the covariance matrix of the predicted normal distribution. Finally, the expectation – and in some suggested solutions also the covariance – over the proxy distribution $q_\theta^\star(w)$ is approximated using Monte Carlo integration, i.e. sampling $T$ weights from the proxy $w \sim q_\theta^\star$.

### F.5 Bounds for the Condition Number

The exact condition number of $\Sigma$ can be calculated using the following equation:

$$\kappa(\Sigma) = \frac{\lambda_S(\Sigma)}{\lambda_1(\Sigma)}$$

Therefore, we need to estimate both the smallest and largest eigenvalues of $\Sigma$.

We begin by analyzing the eigenvalues of the individual components of the LR+D decomposition. The eigenvalues of the diagonal matrix $D$ are simply its diagonal entries,

$$\lambda_i(D) = \{D_{ii} | D_{i-1,i-1} \geq D_{ii} \geq D_{i+1,i+1} > 0\}, \tag{41}$$

where the diagonal entries are assumed to be sorted in non-increasing order without loss of generality.

The symmetric matrix $PP^\top$ is rank-deficient, with at most $R$ non-zero singular values. Using the Singular Value Decomposition (SVD), its eigenvalues are given by

$$\lambda_i(PP^\top) = \begin{cases} 0, & \text{if } i \leq S-R, \\ \Psi_{jj}^2, & \text{otherwise, where } j = i+S-R \end{cases} \tag{42}$$

where the non-zero eigenvalues are sorted in non-increasing order.

Direct computation of the eigenvalues of the full covariance matrix $\Sigma = PP^\top + D$ is computationally prohibitive. Instead, we approximate or bound them using Weyl's inequality, which provides an efficient way to estimate the eigenvalues of matrix sums. For $1 \leq i \leq S$, the eigenvalues of $\Sigma$ satisfy the following sandwich inequality:

$$\lambda_{i-j+1}(PP^\top) + \lambda_j(D) \leq \lambda_i(\Sigma) \leq \lambda_{i+k}(PP^\top) + \lambda_{S-k}(D), \tag{43}$$

where $j$ and $k$ can be chosen freely within the ranges $1 \leq j \leq i$ and $0 \leq k \leq S-i$.

To simplify the calculation, we assume that the $R$ largest eigenvalues of $PP^\top$ dominate the diagonal entries of $D$. Furthermore, the remaining eigenvalues of $PP^\top$ are zero and thus smaller than any positive eigenvalue of $D$. Based on this assumption, we propose fixed choices of $j$ and $k$ to yield the following bounds:

$$\left.\begin{array}{rr} & \lambda_1(D) \\ & \lambda_{i-R}(D) \\ \lambda_i(PP^\top) & + \quad \lambda_1(D) \end{array}\right\} \leq \lambda_i(\Sigma) \leq \left\{\begin{array}{rr} & \lambda_{i+R}(D) \quad \text{if } i \leq R \\ & \lambda_{i+R}(D) \quad \text{if } i \leq S-R \\ \lambda_i(PP^\top) & + \quad \lambda_S(D) \quad \text{if } S-R < i \end{array}\right. \tag{44}$$

Finally, the condition number $\kappa(\Sigma)$ can be bounded by substituting the eigenvalue bounds:

$$\frac{\lambda_S(PP^\top) + \lambda_1(D)}{\lambda_{R+1}(D)} \leq \kappa(\Sigma) \leq \frac{\lambda_S(PP^\top) + \lambda_S(D)}{\lambda_1(D)}. \tag{45}$$

These bounds provide a computationally efficient means of estimating the condition number without requiring exact eigenvalue decomposition, making them well-suited for large-scale covariance matrices in deep learning applications.

### F.6 Alternative Approximation of the Condition Number

In the paper, we suggest using bounds to estimate both the eigenvalues and the condition number. However, an approximate solution using power iteration is also available. To estimate the largest and smallest eigenvalues, we first need to approximate their corresponding eigenvectors $v_S$ and $v_1$. This approximation requires the covariance matrix and its inverse, respectively. One of the main advantages of the LR+D parametrization is its efficient inverse. The following equations approximate the desired eigenvectors using $i$ as iteration step.

$$v_S^{i+1} = \frac{\Sigma v_S^i}{||\Sigma v_S^i||_2} \tag{46}$$

$$v_1^{i+1} = \frac{\Sigma^{-1} v_S^i}{||\Sigma^{-1} v_S^i||_2} \tag{47}$$

After converged, the eigenvalues can be calculated $\lambda_S = ||\Sigma v_S||$ and $\lambda_1 = ||\Sigma v_1||$. The cost of the matrix multiplication $v\Sigma$ and $v_\Sigma^{-1}$ can be reduced by the creation of intermediate arrays. In the naive form. Due to further optimization, we can reduce the memory $\mathcal{O}(S^2)$ and time $\mathcal{O}(\max(R^2, S)SI)$ complexity to $\mathcal{O}(SR)$ and $\mathcal{O}(SR\max(I, R))$, where $I$ is the number of iteration steps. After the optimization, The equation for the eigenvector corresponding to the largest eigenvalue looks like:

$$\begin{aligned} \Sigma v_i &= (D + PP^\top)v_i && (48)\\ &= Dv_i + P(P^\top v_i), && (49) \end{aligned}$$

where the equation for the smallest eigenvalue looks like:

$$\begin{aligned} \Sigma^{-1} v_i &= (D + PP^\top)^{-1} v_i && (50)\\ &= (D^{-1} - (D^{-1}P(I_R + P^\top D^{-1}P)^{-1})(P^\top D^{-1}))v_i && (51)\\ &= \underbrace{D^{-1}}_{\substack{\mathbb{R}^{SxS}\\ diagonal}} v_i - \underbrace{(D^{-1}P(I_R + P^\top D^{-1}P)^{-1})}_{\mathbb{R}^{SxR}}\underbrace{((P^\top D^{-1}))}_{\mathbb{R}^{RxS}}v_i)\,. && (52) \end{aligned}$$

The underbraced parts can be precomputed and reused for every iteration.

Stop the power iteration when the residual norm falls below a tolerance:

$$\|\Sigma v_S^i - \lambda_S^i v_S^i\|_2 < \epsilon_\kappa \tag{53}$$

or equivalently via the Rayleigh quotient change:

$$\left| \frac{v_S^{i\top} \Sigma v_S^i}{v_S^{i\top} v_S^i} - \frac{v_S^{i-1\top} \Sigma v_S^{i-1}}{v_S^{i-1\top} v_S^{i-1}} \right| < \epsilon_\kappa\,. \tag{54}$$

The same criteria apply analogously to $v_1^i$, replacing $\Sigma$ with $\Sigma^{-1}$ and $\lambda_S$ with $\lambda_1$.

The convergence is geometric and the ratio for the largest eigenvalue is given by $c = \frac{\lambda_S}{\lambda_{S-1}}$, whereas for the smallest eigenvalue it is given by $c = \frac{\lambda_2}{\lambda_1}$. The problem of this approximation is, that the convergence of the smallest eigenvalue is pretty slow as the small eigenvalues are pretty similar $\lambda_2 \sim \lambda_1$.