A. MNIST-like Experiments

The plots presented in this appendix support the conclusions made in the main text and provide an overview of the experiments conducted on MNIST-like datasets.

Each figure shows the compression-accuracy trade-off of a particular method and input features for SimpleConvModel and TwoLayerDenseModel models for all four of the studied datasets (described in the main text): EMNIST-Letters on the top-left, KMNIST - top-right, Fashion MNIST – bottom-left, and MNIST on the bottom-right. Figures 7, 8, 9, and 10 present \( \mathbb{R} \) and \( \mathbb{C} \) models with the same intermediate feature sizes.

We compare \( \mathbb{R} \) networks against \( \frac{1}{2} \mathbb{C} \) with half the number of parameters for raw input features on figures 13, and 14, and \( 2\mathbb{R} \) with double the number of parameters against \( \mathbb{C} \) for Fourier input features on figures 11 and 12.

B. Complex-valued Local Reparameterization

In this section we show (11).

By \( e_i \in \mathbb{R} \to \mathbb{C} \) we denote the \( i \)-th unit vector of dimensionality conforming to the matrix-vector expression it is used in. \([M]\) denotes row-major flattening of a matrix \( M \) into a vector, i.e. in lexicographic order of its indices. Furthermore \( \text{diag}(\cdot) \) embeds vectors into matrices with zeros everywhere except the diagonal, and \( \otimes \) is the Kronecker product, for which we note the following identities \([PQR] = (P \otimes R^\top)[Q], (P \otimes Q)^\top = (P^\top \otimes Q^\top), \) and \((P \otimes R)(C \otimes S) = PQ \otimes RS \) (Petersen and Pedersen, 2012).

If we assume a factorized \( \mathbb{C} \)-Gaussian approximation (10) for \( W \in \mathbb{C}^{n \times m} \), then \([W]\) is \( \mathbb{C} \)-Gaussian vector with

\[
[W] \sim \mathcal{CN}_{nm}(\mu_\|=, \text{diag}[\Sigma], \text{diag}[C]),
\]

where with \( C_{ij} = \Sigma_{ij}\xi_{ij}, \Sigma_{ij} \geq 0, \) and \( |C_{ij}|^2 \leq \Sigma_{ij} \). Then for any \( x \in \mathbb{C}^m \) and \( b \in \mathbb{C}^n \) we have \( y = Wx + b = (I_n \otimes x^\top)[W] + b \), whence the covariance and relation matrices of \( y \) are

\[
(I_n \otimes x^\top)\text{diag}[\Sigma](I_n \otimes x^\top)^\top = \sum_{ij}(I_n \otimes x^\top)^\top(e_i \otimes e_j)\Sigma_{ij}(e_i \otimes e_j)^\top(I_n \otimes x^\top)^\top \\
= \sum_{ij}(e_i \otimes x^\top e_j)\Sigma_{ij}(e_i \otimes x^\top e_j)^\top \\
= \sum_{i=1}^n(e_i e_i^\top)\left\{\sum_{j=1}^m \Sigma_{ij}|x_j|^2\right\},
\]

(18)

\[
(I_n \otimes x^\top)\text{diag}[C](I_n \otimes x^\top)^\top = \sum_{ij}(I_n \otimes x^\top)^\top(e_i \otimes e_j)C_{ij}(e_i \otimes e_j)^\top(I_n \otimes x^\top)^\top \\
= \sum_{i=1}^n(e_i e_i^\top)\left\{\sum_{j=1}^m C_{ij}x_j^2\right\}.
\]

(19)

Since (18) and (19) are diagonal, the vector \( y \) has independent univariate \( \mathbb{C} \)-Gaussian components, whence (11) follows.

C. Backpropagation through \( \mathbb{C} \)-networks

Wirtinger (\( \mathbb{C}R \)) calculus relies on the natural identification of \( \mathbb{C} \) with \( \mathbb{R}^2 \), and regards \( f: \mathbb{C} \to \mathbb{C} \) as an algebraically equivalent function \( F: \mathbb{R}^2 \to \mathbb{C} \) defined \( f(z) = f(u + jv) = F(u,v) \). It enables general treatment of functions of vector \( \mathbb{C} \)-argument that possess partial derivatives with respect to real and imaginary parts, yet are not required to satisfy Cauchy-Riemann conditions. In \( \mathbb{C}R \) calculus the complex argument \( z \) and its conjugate \( \bar{z} \) act as independent variables and \( f(z) \) is treated as \( f(z, \bar{z}) \) by way of geometric transformations \( z = u + jv \) and \( \bar{z} = u - jv \).

Wirtinger partial derivative operators are formally defined as \( \frac{\partial}{\partial z} = \frac{1}{2}(\frac{\partial}{\partial u} - j\frac{\partial}{\partial v}) \) and \( \frac{\partial}{\partial \bar{z}} = \frac{1}{2}(\frac{\partial}{\partial u} + j\frac{\partial}{\partial v}) \) and differentials are \( dz = du + jdv \) and \( d\bar{z} = du - jdv \). In this paradigm the usual rules of calculus, like chain and product rules, follow.
Acknowledging that the same result was obtained by Hron et al. (2018, eq. (5)), we provide this appendix for the sake of completeness.

At the same time the Cauchy-Riemann conditions (21) converges for any \( \lambda \) by \( \psi \), i.e. \( f \) is a holomorphic function, since \( C \)-calculus of holomorphic functions, since \( f(z) = f(\bar{z}) \) is constant with respect to \( \bar{z} \) in the latter.

Therefore the gradient of (21) is given by

\[
\nabla \approx \lambda \frac{\partial f}{\partial z} = \frac{\partial f}{\partial x} + j \frac{\partial f}{\partial y},
\]

At the same time the Cauchy-Riemann conditions \(-j\frac{\partial f}{\partial z} = \frac{\partial f}{\partial z} \) can be expressed as \( \frac{\partial f}{\partial z} = 0 \). Thus \( \mathbb{C} \)-\( \mathbb{R} \) calculus subsumes the usual \( \mathbb{C} \)-calculus of holomorphic functions, since \( f(z) = f(z, \bar{z}) \) is constant with respect to \( \bar{z} \) in the latter.

In optimization-related tasks the objective is \( f: \mathbb{C} \rightarrow \mathbb{R} \), meaning that if it were to satisfy the Cauchy-Riemann conditions, then it necessarily should have been constant. Nevertheless, the expression of the \( \mathbb{C} \)-\( \mathbb{R} \) gradient is compatible with what is expected, when \( f \) is treated like a \( \mathbb{R}^2 \) function. For such \( f \) we have \( \bar{f} = f \), which implies \( \frac{\partial f}{\partial z} = \frac{\partial f}{\partial z} = \frac{\partial f}{\partial z} \), whence

\[
\frac{\partial f}{\partial z} = \frac{\partial f}{\partial z} + \frac{\partial f}{\partial z} = \frac{\partial f}{\partial z} + \frac{\partial f}{\partial z} = 2\Re \left( \frac{\partial f}{\partial z} \right).
\]

Therefore the gradient of \( f \) at \( z \) is given by \( \nabla \approx f(z) = \frac{\partial f}{\partial z} + \frac{\partial f}{\partial z} + \frac{\partial f}{\partial z} \). The identification \( \mathbb{C} \simeq \mathbb{R}^2 \), backed by Wirtinger calculus, and emulation of \( \mathbb{C} \)-arithmetic in computational graphs with \( \mathbb{R} \)-valued operations makes it possible to reuse \( \Re \) back-propagation and existing auto-differentiation frameworks.

### D. Gradient of the KL-divergence in \( \mathbb{R} \) case

In this appendix we study the approximation proposed by Molchanov et al. (2017) for the KL divergence term (4) for \( \mathbb{R} \)-valued Variational Dropout. Following the logic of Lapidoth and Moser (2003) we derive the expression for \( d \frac{d}{d\log \alpha} K(\alpha) \). Acknowledging that the same result was obtained by Hron et al. (2018, eq. (5)), we provide this appendix for the sake of completeness.

For \( (z_i)_{i=1}^m \sim \mathcal{N}(0, 1) \) iid and \( (\mu_i)_{i=1}^m \in \mathbb{R} \), the random variable \( W = \sum_i (\mu_i + z_i)^2 \) has non-central \( \chi^2 \) distribution with shape \( m \) and non-centrality parameter \( \lambda = \sum_i \mu_i^2 \), i.e. \( W \sim \chi^2_m(\lambda) \). Therefore, the divergence (4) has the form

\[
K(\alpha) \propto \frac{1}{2} \mathbb{E}_{W \sim \chi^2_m(\lambda)} \log W.
\]

\( W \) can alternatively be represented as a Poisson mixture of ordinary \( \chi^2 \) distributions: if \( Z_{i|J} \sim \chi^2_{m+2J} \) for \( J \sim \mathcal{P}_{ois}(\frac{\lambda}{2}) \) then \( W \sim Z \). Therefore, expanding the conditional expectation gives

\[
\mathbb{E}_{W \sim \chi^2_m(\lambda)} \log W = \mathbb{E} \left( \mathbb{E} (\log W \mid J) \right) = \mathbb{E} \left( J \sim \mathcal{P}_{ois}(\frac{\lambda}{2}) \right) \left( \mathbb{E}_{W \sim \chi^2_{m+2J}} \log W \right).
\]

Since \( \chi^2_{\frac{\lambda}{2}} \) is Gamma distribution \( \Gamma \left( \frac{\lambda}{2}, \frac{1}{2} \right) \), it can be shown that the logarithmic moment \( \mathbb{E}_{\chi^2_{\frac{\lambda}{2}}} \log W \) is \( \psi \left( \frac{\lambda}{2} \right) - \log \frac{\lambda}{2} \), where \( \psi \) is the digamma function \( \psi(x) = \frac{d}{dx} \log \Gamma(x) \). By expanding expectation of a Poisson random variable we get

\[
\mathbb{E}_{W \sim \chi^2_m(\lambda)} \log W = \log 2 + g_m \left( \frac{\lambda}{2} \right),
\]

where

\[
g_m(x) = e^{-x} \sum_{j \geq 0} \frac{x^j}{j!} \psi \left( \frac{m + 2J}{2} \right).
\]

Making use of the property \( \psi(x + 1) = \psi(x) + \frac{1}{x} \) of the digamma function for \( x > 0 \), we conclude that the power series in (21) converges for any \( x \geq 0 \). Therefore the derivative of (21) is given by

\[
\frac{d}{dx} g_m(x) = -g_m(x) + e^{-x} \sum_{j \geq 0} \frac{x^j}{j!} \left( \psi \left( \frac{m + 2J}{2} \right) + \frac{2}{m + 2J} \right).
\]

Variance Sparsification of Deep \( \mathbb{C} \)-valued Networks
Bayesian Sparsification of Deep \textbf{C}-valued Networks

By manipulating the partial sums within (22) we get
\[
\frac{d}{dx} g_m(x) = e^{-x} \sum_{j \geq 0} \frac{x^j}{j! j + \frac{m}{2}} = e^{-x} x^{-\frac{m}{2}} \sum_{j \geq 0} \frac{1}{j!} \int_0^x t^{j + \frac{m}{2} - 1} dt.
\] (23)

Furthermore, the functions \( t \mapsto \sum_{j=0}^J \frac{1}{j! t^{j + \frac{m}{2} - 1}} \) are non-decreasing on \((0, x)\) with growing \( J \) and converge to \( t^{\frac{m}{2} - 1} e^t \), which implies by the Monotone Convergence Theorem that
\[
\frac{d}{dx} g_m(x) = e^{-x} x^{-\frac{m}{2}} \int_0^x \sum_{j \geq 0} \frac{1}{j!} t^{j + \frac{m}{2} - 1} dt = e^{-x} x^{-\frac{m}{2}} \int_0^x t^{\frac{m}{2} - 1} e^t dt.
\] (24)

Substituting \( u^2 = t \) on \([0, \infty] \) with \( 2udu = dt \) and letting \( I_m : x \mapsto e^{-x^2} \int_0^{x^2} u^{m-1} e^{u^2} du \) yields
\[
\frac{d}{d\lambda} \left( \frac{d(20)}{d\lambda} \right) = e^{-x^2} x^{-\frac{m}{2}} \int_0^{x^2} u^{m-1} e^{u^2} du = \left( \sqrt{\frac{2}{\lambda}} \right)^m I_m \left( \sqrt{\frac{2}{\lambda}} \right).
\] (25)

Since \( \alpha \) is non-negative, it is typically parameterized via its logarithm, whence the derivative of (4') with respect to \( \log \alpha \) follows from (25) for \( m = 1 \) and \( \lambda = \frac{1}{\alpha} \):
\[
\frac{dK(\alpha)}{d\log \alpha} = -\frac{1}{\sqrt{2\alpha}} I_1 \left( \sqrt{\frac{1}{2\alpha}} \right).
\] (26)

We compute the Monte-Carlo estimate of (4) on a sample of \( 10^7 \) draws over an equally spaced grid of \( \log \alpha \) in \([-12, +12]\) of size 4096. The approximation proposed by Molchanov et al. (2017) is given in (27), with coefficients \( k_1 = 0.63576 \), \( k_2 = 1.8732 \), and \( k_3 = 1.48695 \). The derivative of the approximation with respect to \( \log \alpha \) follows (26) within 4% of relative tolerance, see fig. 15.

\[
(4) \approx \frac{1}{2} \log \left( 1 + e^{-\log \alpha} \right) + k_1 \sigma \left( -\left( k_2 + k_3 \log \alpha \right) \right),
\] (27)

Similarly, the forward difference estimate of the derivative (26) very closely (up to sampling error). For sake of completeness, we compute a similar Monte-Carlo estimate for the KL divergence term in (13') for \( \text{C-valued Variational Dropout with } \beta = 2 \), fit the best approximation (27), and compare it against the exact \( \log \alpha \) derivative \( \frac{d(13')}{d\log \alpha} = e^{-\frac{1}{\alpha}} - 1 \).
Figure 7. The trade-off of ARD method for $\mathbb{R}$ and $\mathbb{C}$ models using Fourier features.

Figure 8. The trade-off of VD method for $\mathbb{R}$ and $\mathbb{C}$ models using Fourier features.
Bayesian Sparsification of Deep \( C \)-valued Networks

![Graphs showing trade-off on EMNIST-Letters (raw) by ARD (\( \tau = -0.5 \)), Trade-off on KMNIST (raw) by ARD (\( \tau = -0.5 \)), Trade-off on Fashion-MNIST (raw) by ARD (\( \tau = -0.5 \)), and Trade-off on MNIST (raw) by ARD (\( \tau = -0.5 \)).](image)

Figure 9. The trade-off of ARD method for \( R \) and \( C \) models using raw features.

![Graphs showing trade-off on EMNIST-Letters (raw) by VD (\( \tau = -0.5 \)), Trade-off on KMNIST (raw) by VD (\( \tau = -0.5 \)), Trade-off on Fashion-MNIST (raw) by VD (\( \tau = -0.5 \)), and Trade-off on MNIST (raw) by VD (\( \tau = -0.5 \)).](image)

Figure 10. The trade-off of VD method for \( R \) and \( C \) models using raw features.
Bayesian Sparsification of Deep C-valued Networks

Figure 11. The trade-off of ARD method for $2\mathbb{R}$ and C models using Fourier features.

Figure 12. The trade-off of VD method for $2\mathbb{R}$ and C models using Fourier features.
Bayesian Sparsification of Deep C-valued Networks

Figure 13. The trade-off of ARD method for $R$ and $\frac{1}{2}C$ models using raw features.

Figure 14. The trade-off of VD method for $R$ and $\frac{1}{2}C$ models using raw features.
Bayesian Sparsification of Deep $\mathbb{C}$-valued Networks

Figure 15. $\frac{dK(\cdot)}{d\log}$ of the approximation (27), MC estimate of (4), and the exact derivative using (26).