## A. Gradient derivation

The $i$-th element of the gradient of the Gaussian, is

$$\nabla_i \mathcal{N}(\boldsymbol{\tau}) = \frac{\partial_i \mathcal{N}(\boldsymbol{\tau})}{\partial \tau_i} \tag{24}$$

$$= \frac{\mathrm{d}\mathcal{N}(\tau_i)}{\mathrm{d}\tau_i} \prod_{\substack{j=1 \\ j \neq i}}^{n} \mathcal{N}(\tau_j) \tag{25}$$

$$= -\frac{\tau_i}{\sigma^2} \mathcal{N}(\tau_i) \prod_{\substack{j=1 \\ j \neq i}}^{n} \mathcal{N}(\tau_j) \tag{26}$$

$$= -\frac{\tau_i}{\sigma^2} \mathcal{N}(\boldsymbol{\tau}), \tag{27}$$

where we use the overloaded convention that $\mathcal{N}(\boldsymbol{\tau})$ takes a vector $\boldsymbol{\tau}$ and $\mathcal{N}(\tau_i)$ the $i$-th element, a scalar $\tau_i$, to produce the one or $n$-dimensional Gaussian.

This differs from Fischer and Ritschel [7] who only blur in the direction in which they differentiate, as in

$$\nabla_i \mathcal{N}(\tau_i) = \frac{\tau_i}{\sigma^2} \mathcal{N}(\tau_i), \tag{28}$$

while it is more consistent with higher-order differentials to blur all dimensions.

## B. Smooth Gradient Marginalization

The sampling of 1D Gaussian gradient was derived by Fischer and Ritschel [7] who constructed a PDF $p^{\mathrm{G}}$. For a $n$-dimensional Gaussian gradient, we need to marginalize and sample the dimensions individually. Marginalization would be integration over all dimensions $j \neq i$, except the one we look for $i$, so:

$$\int_{\boldsymbol{\tau}_{j \neq i}} p^{\mathrm{G}}(\boldsymbol{\tau}) \mathrm{d}\boldsymbol{\tau}_{j \neq i}. \tag{29}$$

Writing out the CDF, a positivized and scaled version of the PDF $p$, where $1/Z$ is the partition function, gives

$$\frac{1}{Z} \int_{\boldsymbol{\tau}_{j \neq i}} \frac{1}{2} |\nabla_i \mathcal{N}(\boldsymbol{\tau})| \mathrm{d}\boldsymbol{\tau}_{j \neq i} = \tag{30}$$

$$\frac{1}{Z} \int_{\boldsymbol{\tau}_{j \neq i}} |\nabla_i \mathcal{N}(\boldsymbol{\tau})| \mathrm{d}\boldsymbol{\tau}_{j \neq i}. \tag{31}$$

Writing the $n$-D Gaussian as product of $n$ 1D Gaussians

$$\frac{1}{Z} \int_{\boldsymbol{\tau}_{j \neq i}} |\nabla_i \prod_{j=1}^{N} \mathcal{N}(\tau_j)| \mathrm{d}\tau_{j \neq i}. \tag{32}$$

As we differentiate only by $\tau_i$, all other factors are 1, so

$$\frac{1}{Z} \int_{\tau_{j \neq i}} |\nabla_i \mathcal{N}(\tau_i)| \mathrm{d}\tau_{j \neq i}. \tag{33}$$

As we are integrating over all $\tau_{j \neq 1}$, integration becomes multiplication with the domain's measure, 1.

$$|\nabla_i \mathcal{N}(\tau_i)|. \tag{34}$$

## C. Hessian derivation

The diagonal elements of the Hessian of the Gaussian are

$$\nabla^2 \mathcal{N}(\boldsymbol{\tau})_{ii} = \frac{\partial_i^2 \mathcal{N}(\boldsymbol{\tau})}{\partial^2 \tau_i} \tag{35}$$

$$= \frac{\partial}{\partial \tau_i} \left( -\frac{\tau_i}{\sigma^2} \mathcal{N}(\tau_i) \prod_{\substack{j=1 \\ j \neq i}}^{n} \mathcal{N}(\tau_j) \right) \tag{36}$$

$$= \frac{\partial}{\partial \tau_i} \left( -\frac{\tau_i}{\sigma^2} \mathcal{N}(\tau_i) \right) \prod_{\substack{j=1 \\ j \neq i}}^{n} \mathcal{N}(\tau_j) \tag{37}$$

$$= \left( -\frac{1}{\sigma^2} + \frac{\tau_i^2}{\sigma^4} \right) \prod_{\substack{j=1 \\ j \neq i}}^{n} \mathcal{N}(\tau_j) \tag{38}$$

$$= \left( -\frac{1}{\sigma^2} + \frac{\tau_i^2}{\sigma^4} \right) \mathcal{N}(\boldsymbol{\tau}). \tag{39}$$

The non-diagonals of the Hessian of the Gaussian are

$$\nabla^2 \mathcal{N}(\boldsymbol{\tau})_{ij} = \frac{\partial^2 \mathcal{N}(\boldsymbol{\tau})}{\partial_i \tau_i \partial_j \tau_j} \tag{40}$$

$$= \frac{\partial}{\partial \tau_j} \left( -\frac{\tau_i}{\sigma^2} \mathcal{N}(\tau_i) \prod_{\substack{k=1 \\ k \neq i}}^{n} \mathcal{N}(\boldsymbol{\tau}_k) \right) \tag{41}$$

$$= \frac{\tau_i}{\sigma^2} \mathcal{N}(\tau_i) \frac{\tau_j}{\sigma^2} \mathcal{N}(\tau_j) \prod_{\substack{k=1 \\ k \neq i,j}}^{n} \mathcal{N}(\boldsymbol{\tau}_k) \tag{42}$$

$$= \frac{\tau_i}{\sigma^2} \frac{\tau_j}{\sigma^2} \mathcal{N}(\tau_i) \mathcal{N}(\tau_j) \prod_{\substack{k=1 \\ k \neq i,j}}^{n} \mathcal{N}(\boldsymbol{\tau}_k) \tag{43}$$

$$= \frac{\tau_i \tau_j}{\sigma^4} \mathcal{N}(\boldsymbol{\tau}). \tag{44}$$

A remark: It might appear, that diagonal is a special case of off-diagonal, but for differentiation, that is not true, as on the diagonal, the variable we differentiate in respect to appears twice, in the sense that $\mathrm{d}uv/\mathrm{d}u = v$ and $\mathrm{d}uv/\mathrm{d}v = u$, but $\mathrm{d}uu/\mathrm{d}u = 2u$.

## D. Sampling diagonal of Hessian

Similar to Sec. B, for the diagonal of the Hessian, we can sample each dimension independently. Thus we can first derive the valid distribution of the second-order derivative of the one-dimensional Gaussian by positivization and scaling, and it will apply to higher dimensions: The one-dimensional Gaussian's second-order derivative is

$$\left( -\frac{1}{\sigma^2} + \frac{\tau_i^2}{\sigma^4} \right) \mathcal{N}(\tau_i) \tag{45}$$

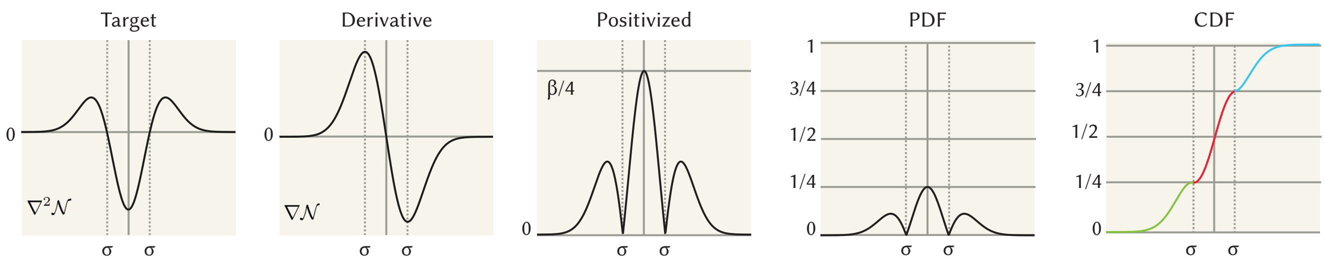


Figure 6. Detailed plots of the functions involved in the derivation of the CDF for the diagonal elements of $\nabla^2\mathcal{N}$, extending Fig. 3.

The roots of Eq. 45 are the $\tau_i$ for which

$$\left(-\frac{1}{\sigma^2}+\frac{\tau_i^2}{\sigma^4}\right)\mathcal{N}(\tau_i)=0. \tag{46}$$

As $\mathcal{N}(\tau_i)>0$ for all $\tau_i$, the product can be 0 only if

$$-\frac{1}{\sigma^2}+\frac{\tau_i^2}{\sigma^4}=0 \qquad \text{and hence} \tag{47}$$

$$\tau_i=\pm\sigma. \tag{48}$$

The function value between $-\sigma<\tau_i\leq\sigma$ is negative and hence needs to be positivised. Since the second-order derivative should integrate to the gradient of the Gaussian, we know that it reaches zero as $\tau_i$ reaches infinity. In conjunction with the fact that the second-order derivative is symmetric about the $y$-axis, we can conclude that the integral of the interval $-\sigma<\tau_i\leq\sigma$ should be twice the size of the integral of $\tau_i\leq-\sigma=\tau_i>\sigma$. Thus, after positivisation the CDF should be scaled, such that it is $\frac{1}{4}$ at $\tau_i=-\sigma$. Solving for these equalities, we can get:

$$\beta\nabla\mathcal{N}(-\sigma)=\frac{1}{4} \tag{49}$$

$$\beta=\frac{1}{4\nabla\mathcal{N}(-\sigma)}. \tag{50}$$

So, for the positivised rescaled second-order derivative as a PDF of the distribution:

$$p_{ii}^{\mathrm{H}}=|\beta\nabla^2\mathcal{N}(\tau_i)|. \tag{51}$$

We can get the integrating constant by flipping and translating the scaled gradient of Gaussian to arrive at the CDF function for the intervals:

$$P_{ii}^{\mathrm{H}}(\tau_i)=\begin{cases}\beta\nabla\mathcal{N}(\tau_i) & \text{if } \tau_i<-\sigma,\\ \frac{1}{2}+\beta\nabla\mathcal{N}(\tau_i) & \text{if } \tau_i\in[-\sigma,\sigma]\\ 1-\beta\nabla\mathcal{N}(\tau_i) & \text{if } \tau_i>\sigma.\end{cases} \tag{52}$$

## E. Grey-box differentials

Sometimes, differentials are in respect to a function that is a composition $\mathbf{z}=f(\mathbf{y}=g(\mathbf{x}))$ of an inner function with known analytic differentials $g$ (white box) and an outer function $f$ with differentials that need to be sampled (black box). For first order (gradient), this is

$$\nabla_{\mathbf{z}}(\mathbf{z}=f(g(\mathbf{x})))=(\nabla_{\mathbf{x}}g(\mathbf{x}))^{\mathsf{T}}\cdot\nabla_{\mathbf{y}}f(\mathbf{y}=g(\mathbf{x})),$$

which means to take the Jacobian (as both $g$ and $f$ in general are vector-valued) of the inner function $g$ in respect to the inner argument $\mathbf{x}$ and vector-matrix multiply this with the gradient of the outer function $f$ but in respect to the outer argument $\mathbf{y}$. For the second order it is

$$\nabla_{\mathbf{z}}^2 f(g(\mathbf{x}))\approx(\nabla_{\mathbf{x}}g(\mathbf{x}))^{\mathsf{T}}\cdot\nabla_{\mathbf{y}}^2 f(g(\mathbf{x}))\cdot\nabla_{\mathbf{x}}g(\mathbf{x})$$

which means again to take the gradient of the inner function, but multiply it with the Hessian, instead of the Jacobian of the composition in respect to the outer argument [15].

The aim of this exercise is to have the sampled gradients handle only the black-box part and the analytic gradients handle the non-sampled [parts. As the analytic parts are typically large (*e.g.*, in the order of the size of a neural network) compared to the number of physical rendering parameters (placement of light, cameras or objects) this can provide substantial advantages.

## F. BFGS/LBFGS method

Quasi-Newton methods are also a way to utilize the second order information for optimization, however they approximates these information with zero or first order information. We tested the a family of algorithms from the quasi-Newton methods that is known to be most effective, the BFGS algorithms [28]. For this family of algorithms, the vanilla BFGS algorithm [27], along with BFGS with Armijo-Wolfe line search [20], LBFGS [20], were tested but they only converge for the QUAD task. This is probably due to the noisy nature of the derivative estimation. To this end, damped BFGS [23], and adaptive finite difference BFGS [1] were also added but neither changed the convergence of other tasks.