Characterizing The Role of A Single Coupling Layer in Affine Normalizing Flows

Felix Draxler[1,2,3], Jonathan Schwarz[1,3], Christoph Schnörr[1,3], and Ullrich Köthe[1,2]

[1] Heidelberg Collaboratory for Image Processing, Heidelberg University, Germany
[2] Visual Learning Lab, Heidelberg University, Germany
[3] Image and Pattern Analysis Group, Heidelberg University, Germany

Abstract. Deep Affine Normalizing Flows are efficient and powerful models for high-dimensional density estimation and sample generation. Yet little is known about how they succeed in approximating complex distributions, given the seemingly limited expressiveness of individual affine layers. In this work, we take a first step towards theoretical understanding by analyzing the behaviour of a *single* affine coupling layer under maximum likelihood loss. We show that such a layer estimates and normalizes conditional moments of the data distribution, and derive a tight lower bound on the loss depending on the orthogonal transformation of the data before the affine coupling. This bound can be used to identify the optimal orthogonal transform, yielding a layer-wise training algorithm for deep affine flows. Toy examples confirm our findings and stimulate further research by highlighting the remaining gap between layer-wise and end-to-end training of deep affine flows.

1 Introduction

Affine Normalizing Flows such as RealNVP [4] are widespread and successful tools for density estimation. They have seen recent success in generative modeling [3,4,9], solving inverse problems [1], lossless compression [6], out-of-distribution detection [12], better understanding adversarial examples [7] and sampling from Boltzmann distributions [13].

These flows approximate arbitrary data distributions $\mu(\mathbf{x})$ by learning an invertible mapping $T(\mathbf{x})$ such that given samples are mapped to normally distributed latent codes $\mathbf{z} := T(\mathbf{x})$. In other words, they reshape the data density μ to form a normal distribution.

While being simple to implement and fast to evaluate, affine flows appear not very expressive at first glance. They consist of invertible layers called coupling blocks. Each block leaves half of the dimensions untouched and subjects the other half to just parameterized translations and scalings.

Explaining the gap between theory and applications remains an unsolved challenge. Taking the problem apart, a single layer consists of a rotation and an affine nonlinearity. It is often hand-wavingly argued that the deep model's

expressivity comes from the rotations between the couplings by allowing different dimensions to influence one another [4].

In this work, we open a rigorous branch of explanation by characterizing the normalizing flow generated by a single affine layer. More precisely, we contribute:

- A single affine layer under maximum likelihood (ML) loss learns first- and second-order moments of the conditional distribution of the changed (active) dimensions given the unchanged (passive) dimensions (Section 3.2).
- From this insight, we derive a tight lower bound on how much the affine nonlinearity can reduce the loss for a given rotation (Section 3.3). This is visualized in Figure 1 where the bound is evaluated for different rotations of the data.
- We formulate a layer-wise training algorithm that determines rotations using the lower bound and nonlinearities using gradient descent in turn (Section 3.4).
- We show that such a single affine layer under ML loss makes the active independent of the passive dimensions if they are generated by a certain rule (Section 3.5).

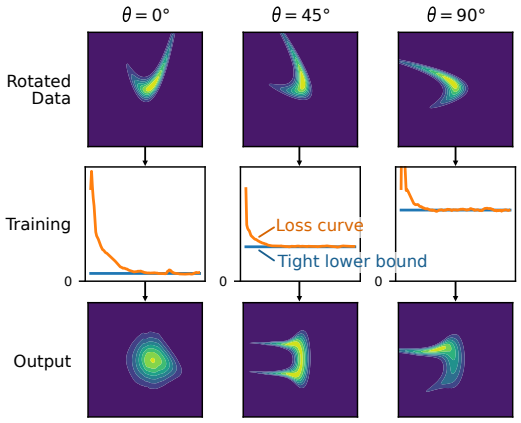

Fig. 1. An affine coupling layer pushes the input density towards standard normal. Its success depends on the rotation of the input *(top row)*. We derive a lower bound for the error that is actually attained empirically *(center row, blue and orange curves)*. The solution with lowest error is clearly closest to standard normal *(bottom row, left)*.

Finally, we show empirically in Section 4 that while improving the training of shallow flows, the above new findings do not yet explain the success of deep affine flows and stimulate further research.

2 Related Work

The connection between affine transformations and the first two moments of a distribution is well-known in the Optimal Transport literature. When the function space of an Optimal Transport (OT) problem with quadratic ground cost is reduced to affine maps, the best possible transport matches mean and covariance of the involved distributions [17]. In the case of conditional distributions, affine maps become conditional affine maps [16]. We show such maps to have the same minimizer under maximum likelihood loss (KL divergence) as under OT costs.

It has been argued before that a single coupling or autoregressive block [14] can capture the moments of conditional distributions. This is one of the motivations for the SOS flow [8], based on a classical result on degree-3 polynomials by [5]. However, they do not make this connection explicit. We are able to give a direct correspondence between the function learnt by an affine coupling and the first two moments of the distribution to be approximated.

Rotations in affine flows are typically chosen at random at initialization and left fixed during training [3,4]. Others have tried training them via some parameterization like a series of Householder reflections [15]. The stream of work most closely related to ours explores the idea to perform layer-wise training. This allows an informed choice of the rotation based on the current estimate of the latent normal distribution. Most of these works propose to choose the least Gaussian dimensions as the active subspace [2,11] . We argue that this is inapplicable to affine flows due to their limited expressivity when the passive dimensions are not informative. To the best of our knowledge, our approach is the first to take the specific structure of the coupling layer into account and derive a tight lower bound on the loss as a function of the rotation.

3 Single Affine Coupling Layer

3.1 Architecture

Normalizing flows approximate data distributions μ available through samples $\mathbf{x} \in \mathbb{R}^D \sim \mu$ by learning an invertible function $T(\mathbf{x})$ such the latent codes $\mathbf{z} := T(\mathbf{x})$ follow an isotropic normal distribution $\mathbf{z} \in \mathbb{R}^D \sim \mathcal{N}(0, \mathbf{1})$. When such a function is found, the data distribution $\mu(\mathbf{x})$ can be approximated using the change-of-variables formula:

$$\mu(\mathbf{x}) = \mathcal{N}(T(\mathbf{x}))|\det \mathbf{J}| =: (T_\sharp^{-1}\mathcal{N})(\mathbf{x}), \tag{1}$$

where $\mathbf{J} = \nabla T(\mathbf{x})$ is the Jacobian of the invertible function, and "$\cdot_\sharp$" is the push-forward operator. New samples $\mathbf{x} \sim \mu$ can be easily generated by drawing $\mathbf{z}$ from the latent Gaussian and transporting them backward through the invertible function:

$$\mathbf{z} \sim \mathcal{N}(0, \mathbf{1}) \quad \Longleftrightarrow \quad \mathbf{x} =: T^{-1}(\mathbf{z}) \sim \mu(\mathbf{x}). \tag{2}$$

Affine Normalizing Flows are a particularly efficient way to parameterize such an invertible function T: They are simple to implement and fast to evaluate in

both directions $T(\mathbf{x})$ and $T^{-1}(\mathbf{z})$, along with the Jacobian determinant $\det \mathbf{J}$ [1]. Like most normalizing flow models, they consist of the composition of several invertible layers $T(\mathbf{x}) = (T_L \circ \cdots \circ T_1)(\mathbf{x})$. The layers are called coupling blocks and modify the distribution sequentially. We recursively define the push-forward of the first l blocks as

$$\mu_l = (T_l)_\sharp \mu_{l-1}, \quad \mu_0 = \mu. \tag{3}$$

Each block $T_l, l = 1, \ldots, L$ contains a rotation $\mathbf{Q}_l \in \mathrm{SO}(D)$ and a nonlinear transformation τ_l:

$$\mathbf{x}_l = T_l(\mathbf{x}_{l-1}) = (\tau_l \circ \mathbf{Q}_l)(\mathbf{x}_{l-1}), \quad \mathbf{x}_0 = \mathbf{x}. \tag{4}$$

The nonlinear transformation τ_l is given by:

$$\tau_l(\mathbf{y}) = \tau_l\left(\begin{bmatrix}\mathbf{p}\\ \mathbf{a}\end{bmatrix}\right) = \begin{bmatrix}\mathbf{p}\\ \mathbf{a} \odot e^{s_l(\mathbf{p})} + t_l(\mathbf{p})\end{bmatrix} =: \begin{bmatrix}\mathbf{p}\\ \mathbf{a}'\end{bmatrix} = \mathbf{y}'. \tag{5}$$

Here, $\mathbf{y} = \mathbf{Q}_l \mathbf{x}_{l-1} \sim (\mathbf{Q}_l)_\sharp \mu_{l-1}$ is the rotated input to the nonlinearity (dropping the index l on $\mathbf{y}$ for simplicity) and $\odot$ is element-wise multiplication. An affine nonlinearity first splits its input into *passive* and *active* dimensions $\mathbf{p} \in \mathbb{R}^{D_P}$ and $\mathbf{a} \in \mathbb{R}^{D_A}$. The passive subspace is copied without modification to the output of the coupling. The active subspace is scaled and shifted as a function of the passive subspace, where s_l and $t_l : \mathbb{R}^{D_P} \to \mathbb{R}^{D_A}$ are represented by a single generic feed forward neural network [9] and need not be invertible themselves. The affine coupling design makes inversion trivial by transposing $\mathbf{Q}_l$ and rearranging terms in τ_l.

Normalizing Flows, and affine flows in particular, are typically trained using the Maximum Likelihood (ML) loss [3]. It is equivalent to the Kullback-Leibler (KL) divergence between the push-forward of the data distribution μ and the latent normal distribution [10]:

$$\mathcal{D}_{\mathrm{KL}}(T_\sharp \mu || \mathcal{N}) = -H[\mu] + \frac{D}{2}\log(2\pi) + \mathbb{E}_{\mathbf{x}\sim\mu}\left[\frac{1}{2}\left\|T(\mathbf{x})\right\|^2 - \log|\det \mathbf{J}(\mathbf{x})|\right] \tag{6}$$

$$= -H[\mu] + \frac{D}{2}\log(2\pi) + \mathrm{ML}(T_\sharp \mu || \mathcal{N}), \tag{7}$$

The two differ only by terms independent of the trained model (the typically unknown entropy $H[\mu]$ and the normalization of the normal distribution).

It is unknown whether affine normalizing flows can push arbitrarily complex distributions to a normal distribution [14]. In the remainder of the section, we shed light on this by considering an affine flow that consists of just a single coupling as defined in Equation (5). Since we only consider one layer, we're dropping the layer index l for the remainder of the section. In Section 4, we will discuss how these insights on isolated affine layers transfer to deep flows.

3.2 KL Divergence Minimizer

We first derive the exact form of the ML loss in Equation (6) for an isolated affine coupling with a fixed rotation $\mathbf{Q}$ as in Equation (4).

The Jacobian for this coupling has a very simple structure: It is a triangular matrix whose diagonal elements are $\mathbf{J}_{ii} = 1$ if i is a passive dimension and $\mathbf{J}_{ii} = \exp(s_i(\mathbf{p}))$ if i is active. Its determinant is the product of the diagonal elements, so that $\det \mathbf{J}(\mathbf{x}) > 0$ and $\log\det \mathbf{J}(\mathbf{x}) = \sum_{i=1}^{D_A} s_i(\mathbf{p})$. The ML loss thus reads:

$$\mathrm{ML}(T_\sharp\mu||\mathcal{N}) = \mathbb{E}_{\mathbf{p},\mathbf{a}\sim\mathbf{Q}_\sharp\mu}\left[\frac{1}{2}\|\mathbf{p}\|^2 + \frac{1}{2}\left\|\mathbf{a}\odot e^{s(\mathbf{p})} + t(\mathbf{p})\right\|^2 - \sum_{i=1}^{D_A} s_i(\mathbf{p})\right]. \tag{8}$$

We now derive the minimizer of this loss:

Lemma 1 (Optimal single affine coupling). *Given a distribution μ and a single affine coupling layer T with a fixed rotation $\mathbf{Q}$. Like in Equation (5), call $(\mathbf{a},\mathbf{p}) = \mathbf{Q}\mathbf{x}$ the rotated versions of $\mathbf{x}\sim\mu$. Then, at the unique minimum of the ML loss (Equation (8)), the functions $s,t:\mathbb{R}^{D_P}\to\mathbb{R}^{D_A}$ as in Equation (4) take the following value:*

$$e^{s_i(\mathbf{p})} = \frac{1}{\sqrt{\mathrm{Var}_{a_i|\mathbf{p}}[a_i]}} = \sigma^{-1}_{A_i|\mathbf{p}}, \tag{9}$$

$$t_i(\mathbf{p}) = -\mathbb{E}_{a_i|\mathbf{p}}[a_i]e^{s_i(\mathbf{p})} = -\frac{m_{A_i|\mathbf{p}}}{\sigma_{A_i|\mathbf{p}}}. \tag{10}$$

We derive this by optimizing for $s(\mathbf{p}), t(\mathbf{p})$ in Equation (8) for each value of $\mathbf{p}$ separately. The full proof can be found in Appendix A.1.

We insert the optimal s and t to find the active part of the globally optimal affine nonlinearity:

$$\tau(\mathbf{a}|\mathbf{p}) = \mathbf{a}\odot e^{s(\mathbf{p})} + t(\mathbf{p}) = \frac{1}{\sigma_{A|\mathbf{p}}}\odot(\mathbf{a} - \mathbf{m}_{A|p}). \tag{11}$$

It normalizes $\mathbf{a}$ for each $\mathbf{p}$ by shifting the mean of $\mu(\mathbf{a}|\mathbf{p})$ to zero and rescaling the individual standard deviations to one.

Example 1. Consider a distribution where the first variable p is uniformly distributed on the interval $[-2,2]$. The distribution of the second variable a is normal, but its mean $m(p)$ and standard deviation $\sigma(p)$ are varying depending on p:

$$\mu(p) = \mathcal{U}([-2,2]), \quad \mu(a|p) = \mathcal{N}(m(p),\sigma(p)). \tag{12}$$

$$m(p) = \frac{1}{2}\cos(\pi p), \quad \sigma(p) = \frac{1}{8}(3 - \cos(8\pi/3\, p)). \tag{13}$$

We call this distribution "W density". It is shown in Figure 2a.

We now train a single affine nonlinearity τ by minimizing the ML loss, setting $\mathbf{Q} = \mathbf{1}$. As hyperparameters, we choose a subnet for s,t with one hidden layer and a width of 256, a learning rate of 10^{-1}, a learning rate decay with factor 0.9 every 100 epochs, and a weight decay of 0. We train for 4096 epochs with 4096 i.i.d. samples from μ each using the Adam optimizer.

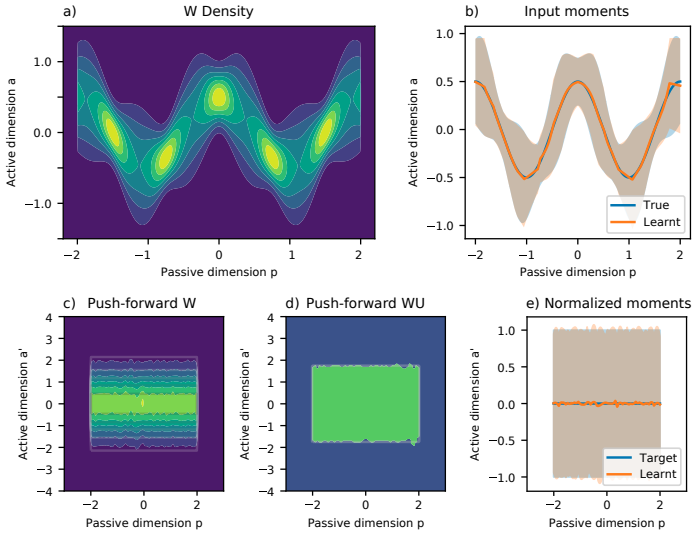

Fig. 2. *(a)* W density contours. *(b)* The conditional moments are well approximated by a single affine layer. *(c, d)* The learnt push-forwards of the W (Example 1) and WU (Example 2) densities remain normal respectively uniform distributions. *(e)* The moments of the transported distributions are close to zero mean and unit variance, shown for the layer trained on the W density.

We solve s, t in Lemma 1 for the estimated mean $\hat{m}(p)$ and standard deviation $\hat{\sigma}(p)$ as predicted by the learnt $\hat{s}$ and $\hat{t}$. Upon convergence of the model, they closely follow their true counterparts $m(p)$ and $\sigma(p)$ as shown in Figure 2b.

Example 2. This example modifies the previous to illustrate that the learnt conditional density $\tau_\sharp \mu(\mathbf{a}|\mathbf{p})$ is not necessarily Gaussian at the minimum of the loss.

The W density from above is transformed to the "WU density" by replacing the conditional normal distribution by a conditional uniform distribution with the same conditional mean $m(p)$ and standard deviation $\sigma(p)$ as before.

$$\mu(p) = \mathcal{U}([-2, 2]), \tag{14}$$

$$\mu(a|p) = \mathcal{U}([m(p) - \sqrt{3}\sigma(p), m(p) + \sqrt{3}\sigma(p)]). \tag{15}$$

One might wrongly believe that the KL divergence favours building a distribution that is *marginally* normal while ignoring the conditionals, i.e. $\tau_\sharp \mu(p) = \mathcal{N}$. Lemma 1 predicts the correct result, resulting in the following uniform push-forward density depicted in Figure 2d:

$$T_\sharp \mu(p) = \mu(p) = \mathcal{U}([-2, 2]), \tag{16}$$

$$T_\sharp \mu(a|p) = \mathcal{U}([-\sqrt{3}, \sqrt{3}]). \tag{17}$$

Note how $\tau_\sharp \mu(a|p)$ does not depend on p, which we later generalize in Lemma 2.

3.3 Tight Bound on Loss

Knowing that a single affine layer learns the mean and standard deviation of $\mu(a_i|\mathbf{p})$ for each $\mathbf{p}$, we can insert this minimizer into the KL divergence. This yields a tight lower bound on the loss after training. Even more, it allows us to compute a tight upper bound on the loss improvement by the layer, which we denote $\Delta \geq 0$. This loss reduction can be approximated using samples without training.

Theorem 1 (Improvement by single affine layer). *Given a distribution μ and a single affine coupling layer T with a fixed rotation $\mathbf{Q}$. Like in Equation (5), call $(\mathbf{a}, \mathbf{p}) = \mathbf{Q}\mathbf{x}$ the rotated versions of $\mathbf{x} \sim \mu$. Then, the KL divergence has the following minimal value:*

$$\mathcal{D}_{KL}(T_\sharp\mu||\mathcal{N}) = \mathcal{D}_{KL}(\mu_P||\mathcal{N}) + \mathbb{E}_{\mathbf{p}}\left[\sum_{i=1}^{D_A} H[\mathcal{N}(0, \sigma_{A_i|\mathbf{p}})] - H[\mu(\mathbf{a}|\mathbf{p})]\right] \tag{18}$$

$$= \mathcal{D}_{KL}(\mu||\mathcal{N}) - \Delta. \tag{19}$$

The loss improvement by the optimal affine coupling as in Lemma 1 is:

$$\Delta = \frac{1}{2}\sum_{i=1}^{D_A} \mathbb{E}_{\mathbf{p}}[m^2_{A_i|\mathbf{p}} + \sigma^2_{A_i|\mathbf{p}} - 1 - \log\sigma^2_{A_i|\mathbf{p}}]. \tag{20}$$

To proof, insert the minimizer s, t from Lemma 1 into Equation (8). Then evaluate $\Delta = \mathcal{D}_{\mathrm{KL}}(\mu||\mathcal{N}) - \mathcal{D}_{\mathrm{KL}}(T_\sharp\mu||\mathcal{N})$ to obtain the statement. The detailed proof can be found in Appendix A.2.

The loss reduction by a single affine layer depends solely on the moments of the distribution of the active dimensions conditioned on the passive subspace. Higher order moments are ignored by this coupling design. Together with Lemma 1, this paints the following picture of an affine coupling layer: It fits a Gaussian distribution to each conditional $\mu(a_i|\mathbf{p})$ and normalizes this Gaussian's moments. The gap in entropy between the fit Gaussian and the true conditional distribution cannot be reduced by the affine transformation. This makes up the remaining KL divergence in Equation (18).

We now make the connection explicit that a single affine layer can only achieve zero loss on the active subspace iff the conditional distribution is Gaussian with diagonal covariance:

Corollary 1. *If and only if $(\mathbf{Q}_\sharp\mu)(\mathbf{a}|\mathbf{p})$ is normally distributed for all p with diagonal covariance, that is:*

$$\mu(\mathbf{a}|\mathbf{p}) = \prod_{i=1}^{D_A} \mathcal{N}(a_i|m_{A_i|\mathbf{p}}, \sigma_{A_i|\mathbf{p}}), \tag{21}$$

a single affine block can reduce the KL divergence on the active subspace to zero:

$$\mathcal{D}_{KL}((T_\sharp\mu)(\mathbf{a}|\mathbf{p})||\mathcal{N}) = 0. \tag{22}$$

The proof can be found in Appendix A.3.

Example 3. We revisit the Examples 1 and 2 and confirm that the minimal loss achieved by a single affine coupling layer on the W-shaped densities matches the predicted lower bound. This is the case for both densities. Figure 3 shows the contribution of the conditional part of the KL divergence $\mathcal{D}_{\mathrm{KL}}((T_\sharp\mu)(a|p)||\mathcal{N})$ as a function of p:

For the W density, the conditional $\mu(a|p)$ is normally distributed. This is the situation of Corollary 1 and the remaining conditional KL divergence is zero. The remaining loss for the WU density is the negentropy of a uniform distribution with unit variance.

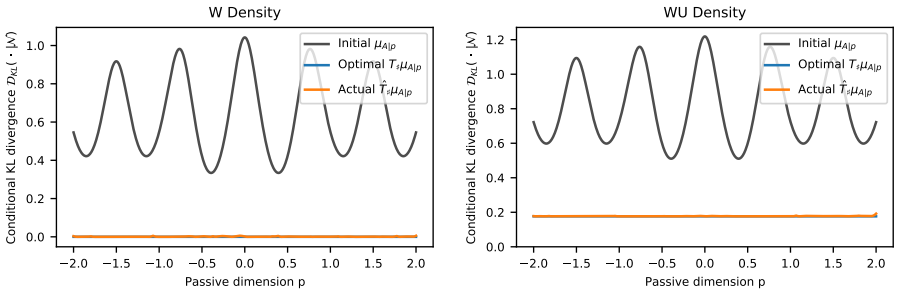

Fig. 3. Conditional KL divergence before *(gray)* and after *(orange)* training for W-shaped densities confirms lower bound *(blue, coincides with orange)*. The plots show the W density from Example 1 *(left)* and the WU density from Example 2 *(right)*.

3.4 Determining the Optimal Rotation

The rotation $\mathbf{Q}$ of the isolated coupling layer determines the splitting into active and passive dimensions and the axes of the active dimensions (the rotation within the passive subspace only rotates the input into s, t and is irrelevant). The bounds in Theorem 1 heavily depend on these choices and thus depend on the chosen rotation $\mathbf{Q}$. This makes it natural to consider the loss improvement as a function of the rotation: $\Delta(\mathbf{Q})$. When aiming to maximally reduce the loss with a single affine layer, one should choose the subspace maximizing this tight upper bound in Equation (20):

$$\arg\max_{\mathbf{Q}\in SO(D)} \Delta(\mathbf{Q}). \tag{23}$$

We propose to approximate this maximization by evaluating the loss improvement for a finite set of candidate rotations in Algorithm 1 "Optimal Affine Subspace (OAS)". Note that Step 5 requires approximating Δ from samples. In the regime of low D_P, one can discretize this by binning samples by their passive coordinate $\mathbf{p}$. Then, one computes mean and variance empirically for each bin. We leave the general solution of Equation (23) for future work.

Algorithm 1 Optimal Affine Subspace (OAS).

Input: $\mathcal{Q} = \{\mathbf{Q}_1, \dots, \mathbf{Q}_C\} \subset SO(D)$, $(\mathbf{x}_j)_{j=1}^N$ i.i.d. samples from μ.
for candidate $\mathbf{Q}_c \in \mathcal{Q}$ **do**
  Rotate samples: $\mathbf{y}_j = \mathbf{Q}_c \mathbf{x}_j$.
  for each active dimension $i = 1, \dots, D_A$ **do**
    Use $(\mathbf{y})_{j=1}^N$ to estimate the conditional mean $m_{A_i|\mathbf{p}}$ and variance $\sigma_{A_i|\mathbf{p}}$ as a function of $\mathbf{p}$. {Example implementation in Example 4}
  end for
  Compute $\Delta_c := \frac{1}{2} \sum_{j=1}^N \sum_{i=1}^{D_A} (m^2_{A_i|\mathbf{p}_j} + \sigma^2_{A_i|\mathbf{p}_j} - 1 - \log \sigma^2_{A_i|\mathbf{p}_j})$ {Equation (20)}.
end for
Return: $\arg\max_{\mathbf{Q}_c \in \mathcal{Q}} \Delta_c$.

Example 4. Consider the following two-component 2D Gaussian Mixture Model:

$$\mu = \frac{1}{2}\big(\mathcal{N}([-\delta; 0], \sigma) + \mathcal{N}([\delta; 0], \sigma)\big). \tag{24}$$

We choose $\delta = 0.95, \sigma = \sqrt{1-\delta^2} = 0.3122...$ so that the mean is zero and the standard deviation along the first axis is one. We now evaluate the loss improvement $\Delta(\theta)$ in Equation (20) as a function of the angle θ with which we rotate the above distribution:

$$\mu(\theta) := \mathbf{Q}(\theta)_\sharp \mu, \quad [p, a] = \mathbf{Q}(\theta)\mathbf{x} \sim \mu(\theta). \tag{25}$$

Analytically, this can be done pointwise for a given p and then integrated numerically. This will not be possible for applications where only samples are available. As a proof of concept, we employ the previously mentioned binning approach. It groups N samples from μ by their p value into B bins. Then, we compute $m_{A|p_b}$ and $\sigma_{A|p_b}$ using the samples in each bin $b = 1, \dots, B$.

Figure 4 shows the upper bound as a function of the rotation angle, as obtained from the two approaches. Here, we used $B = 32$ bins and a maximum of $N = 2^{13} = 8192$ samples. Around $N \approx 256$ samples are sufficient for a good agreement between the analytic and empiric bound on the loss improvement and the corresponding angle at the maximum.

Note: For getting a good density estimate using a single coupling, it is crucial to identify the right rotation. If we naively or by chance decide for $\theta = 90°$, the distribution is left unchanged.

3.5 Independent Outputs

An important step towards pushing a multivariate distribution to a normal distribution is making the dimensions independent of one another. Then, the residual to a global latent normal distribution can be solved with one sufficiently expressive 1D flow per dimension, pushing each distribution independently to a normal distribution. The following lemma shows for which data sets a single affine layer can make the active and passive dimensions independent.

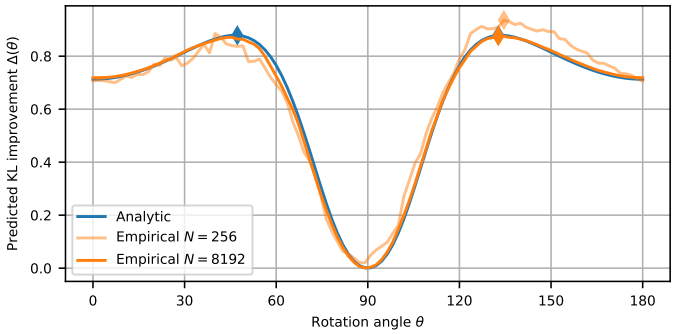

Fig. 4. Tight upper bound given by Equation (20) for two-component Gaussian mixture as a function of rotation angle θ, determined analytically *(blue)* and empirically *(orange)* for different numbers of samples. The diamonds mark the equivalent outputs of the OAS Algorithm 1.

Lemma 2. *Given a distribution μ and a single affine coupling layer T with a fixed rotation $\mathbf{Q}$. Like in Equation* (5)*, call $(\mathbf{a}, \mathbf{p}) = \mathbf{Q}\mathbf{x}$ the rotated versions of $\mathbf{x} \sim \mu$. Then, the following are equivalent:*

1. $\mathbf{a}' := \tau(\mathbf{a}|\mathbf{p}) \perp \mathbf{p}$ *for $\tau(\mathbf{a}|\mathbf{p})$ minimizing the ML loss in Equation* (8)*,*
2. *There exists $\mathbf{n} \perp \mathbf{p}$ such that $\mathbf{a} = f(\mathbf{p}) + \mathbf{n} \odot g(\mathbf{p})$, where $f, g : \mathbb{R}^{D_P} \to \mathbb{R}^{D_A}$.*

The proof can be found in Appendix A.4.

This results shows what our theory can explain about deep affine flows: It is easy to see that $D-1$ coupling blocks with $D_A = 1, D_P = D-1$ can make all variables independent if the data set can be written in the form of $x_i = f(\mathbf{x}_{\neq i}) + x_i g(\mathbf{x}_{\neq i})$. Then, only the aforementioned independent 1D flows are necessary for a push-forward to the normal distribution.

Example 5. Consider again the W-shaped densities from the previous Examples 1 and 2. After optimizing the single affine layer, the two variables p, a' are independent (compare Figure 2c, d):

$$\text{Example 1: } a' \sim \mathcal{N}(0,1) \perp p, \tag{26}$$

$$\text{Example 2: } a' \sim \mathcal{U}([-\sqrt{3}, \sqrt{3}]) \perp p, \tag{27}$$

4 Layer-wise Learning

Do the above single-layer results explain the expressivity of deep affine flows? To answer this question, we construct a deep flow layer by layer using the optimal affine subspace (OAS) algorithm Algorithm 1. Each layer l being added to the flow is trained to minimize the residuum between the current push-forward μ_{l-1} and the latent $\mathcal{N}$. The corresponding rotation $\mathbf{Q}_l$ is chosen by maximizing $\Delta(\mathbf{Q}_l)$ and the nonlinearities τ_l are trained by gradient descent, see Algorithm 2.

Can this ansatz reach the quality of end-to-end affine flows? An analytic answer is out of the scope of this work, and we consider toy examples.

Algorithm 2 Iterative Affine Flow Construction.

1: Initialize $T^{(0)} = \mathrm{id}$.
2: **repeat**
3: Compute $\mathbf{Q}_l$ via OAS (Algorithm 1), using samples from $T_\sharp^{(l-1)}\mu$.
4: Train τ_l on samples $\mathbf{y} = \mathbf{Q}_l \cdot T^{(l-1)}(\mathbf{x})$ for $\mathbf{x} \sim \mu$.
5: Set $T_l = \tau_l \circ \mathbf{Q}_l$.
6: Compose $T^{(l)} = T_l \circ T^{(l-1)}$.
7: **until** convergence, e.g. loss or improvement threshold, max. number of layers.
8: **return** Final transport $T^{(L)}$.

Example 6. We consider a uniform 2D distribution $\mu = \mathcal{U}([-1, 1]^2)$. Figure 5 compares the flow learnt layer-wise using Algorithm 2 to flows learnt layer-wise and end-to-end, but with fixed random rotations. Our proposed layer-wise algorithm performs on-par with end-to-end training despite optimizing only the respective last layer in each iteration, and beats layer-wise random subspaces.

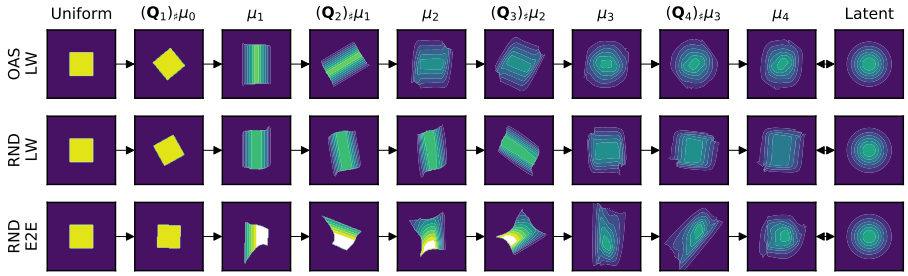

Fig. 5. Affine flow trained layer-wise "LW", using optimal affine subspaces "OAS" *(top)* and random subspaces "RND" *(middle)*. After a lucky start, the random subspaces do not yield a good split and the flow approaches the latent normal distribution significantly slower. End-to-end training "E2E" *(bottom)* chooses a substantially different mapping, yielding a similar quality to layer-wise training with optimal subspaces.

Example 7. We now provide more examples on a set of toy distributions. As before, we train layer-wise using OAS and randomly selected rotations, and end-to-end. Additionally, we train a mixed variant of OAS and end-to-end: New layers are still added one by one, but Algorithm 2 is modified such that iteration l optimizes *all* layers 1 through l in an end-to-end fashion. We call this training "progressive" as layers are progressively activated and never turned off again.

We obtain the following results: Optimal rotations always outperform random rotations in layer-wise training. With only a few layers, they also outperform end-to-end training, but are eventually overtaken as the network depth increases. Progressive training continues to be competitive also for deep networks.

Figure 6 shows the density estimates after twelve layers. At this point, none of the methods show a significant improvement by adding layers. Hyperparameters were optimized for each training configuration to obtain a fair comparison.

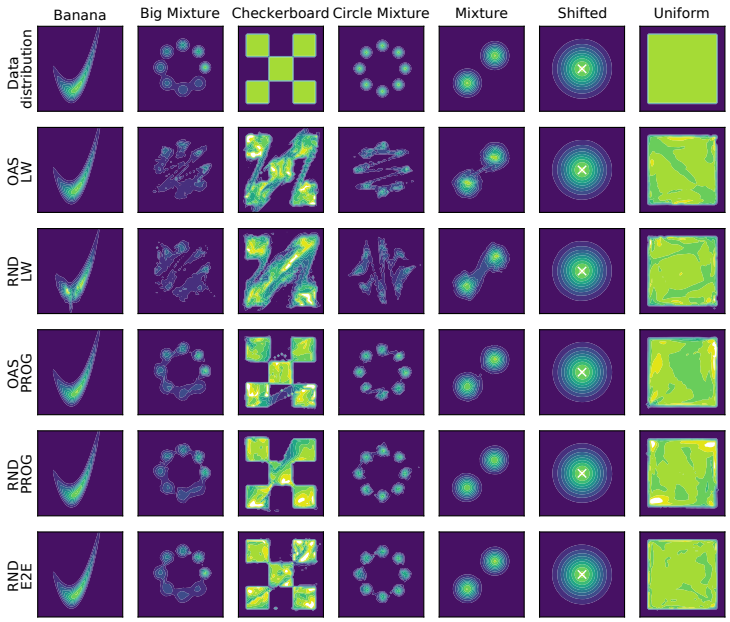

Fig. 6. Affine flows trained on different toy problems *(top row)*. The following rows depic different training methods: layer-wise "LW" *(rows 2 and 3)*, progressively "PROG" *(rows 4-5)* and end-to-end "E2E" *(last row)*. Rotations are "OAS" when determined by Algorithm 1 *(row 2 and 4)* or randomly selected "RND" *(rows 3, 5 and 6)*.

Densities obtained by layer-wise training exhibit significant spurious structure for both optimal and random rotations, with an advantage for optimally chosen subspaces.

5 Conclusion

In this work, we showed that an isolated affine coupling learns the first two moments of the conditioned data distribution $\mu(\mathbf{a}|\mathbf{p})$. Using this result, we derived a tight upper bound on the loss reduction that can be achieved by such a layer. We then used this to choose the best rotation of the coupling.

We regard our results as a first step towards a better understanding of deep affine flows. We provided sufficient conditions for a data set that can be exactly solved with layer-wise trained affine couplings and a single layer of D independent 1D flows.

Our results can be seen analogously to the classification layer at the end of a multi-layer classification network: The results from Section 3 directly apply to the last coupling in a deep normalizing flow. This raises a key question for future work: How do the first $L-1$ layers prepare the distribution μ_{L-1} such that the final layer can perfectly push the data to a Gaussian?

Acknowledgement

This work is supported by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC-2181/1 - 390900948 (the Heidelberg STRUCTURES Cluster of Excellence).

References

1. Ardizzone, L., Kruse, J., Wirkert, S., Rahner, D., Pellegrini, E.W., Klessen, R.S., Maier-Hein, L., Rother, C., Köthe, U.: Analyzing inverse problems with invertible neural networks. arXiv preprint arXiv:1808.04730 (2018)
2. Bigoni, D., Zahm, O., Spantini, A., Marzouk, Y.: Greedy inference with layers of lazy maps. arXiv preprint arXiv:1906.00031 (2019)
3. Dinh, L., Krueger, D., Bengio, Y.: Nice: Non-linear independent components estimation. arXiv preprint arXiv:1410.8516 (2014)
4. Dinh, L., Sohl-Dickstein, J., Bengio, S.: Density estimation using real nvp. arXiv preprint arXiv:1605.08803 (2016)
5. Fleishman, A.I.: A method for simulating non-normal distributions. Psychometrika 43(4), 521–532 (1978)
6. Hoogeboom, E., Peters, J., van den Berg, R., Welling, M.: Integer discrete flows and lossless compression. In: Advances in Neural Information Processing Systems. pp. 12134–12144 (2019)
7. Jacobsen, J.H., Behrmann, J., Zemel, R., Bethge, M.: Excessive invariance causes adversarial vulnerability. arXiv preprint arXiv:1811.00401 (2018)
8. Jaini, P., Selby, K.A., Yu, Y.: Sum-of-squares polynomial flow. arXiv preprint arXiv:1905.02325 (2019)
9. Kingma, D.P., Dhariwal, P.: Glow: Generative flow with invertible 1x1 convolutions. In: Advances in neural information processing systems. pp. 10215–10224 (2018)
10. Marzouk, Y., Moselhy, T., Parno, M., Spantini, A.: Sampling via measure transport: An introduction. Handbook of Uncertainty Quantification pp. 1–41 (2016)
11. Meng, C., Ke, Y., Zhang, J., Zhang, M., Zhong, W., Ma, P.: Large-scale optimal transport map estimation using projection pursuit. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 8116–8127. Curran Associates, Inc. (2019), http://papers.nips.cc/paper/9023-large-scale-optimal-transport-map-estimation-using-projection-pursuit.pdf
12. Nalisnick, E., Matsukawa, A., Teh, Y.W., Lakshminarayanan, B.: Detecting out-of-distribution inputs to deep generative models using a test for typicality. arXiv preprint arXiv:1906.02994 (2019)
13. Noé, F., Olsson, S., Köhler, J., Wu, H.: Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. Science 365(6457), eaaw1147 (2019)
14. Papamakarios, G., Nalisnick, E., Rezende, D.J., Mohamed, S., Lakshminarayanan, B.: Normalizing flows for probabilistic modeling and inference. arXiv preprint arXiv:1912.02762 (2019)
15. Putzky, P., Welling, M.: Invert to learn to invert. In: Advances in Neural Information Processing Systems. pp. 446–456 (2019)

16. Tabak, E.G., Trigila, G.: Conditional expectation estimation through attributable components. Information and Inference: A Journal of the IMA 7(4), 727–754 (2018)
17. Trigila, G., Tabak, E.G.: Data-driven optimal transport. Communications on Pure and Applied Mathematics 69(4), 613–648 (2016)

A Proofs

A.1 Proof of Lemma 1

Proof. Minimizing the KL divergence is equivalent to minimizing Equation (4). The affine nonlinearity τ leaves the passive dimensions $\mathbf{p}$ unchanged. This leaves us with the following minimization problem:

$$\min_{s,t:\mathbb{R}^{D_P}\to\mathbb{R}^{D_A}} \mathbb{E}_{\mathbf{p},\mathbf{a}}\left[\frac{1}{2}\left\|\mathbf{a}\odot e^{s(\mathbf{p})}+t(\mathbf{p})\right\|^2-\sum_{i=1}^{D_A}s_i(\mathbf{p})\right], \tag{28}$$

where $\mathbb{E}_{\mathbf{p},\mathbf{a}}$ is shorthand for $\mathbb{E}_{\mathbf{p},\mathbf{a}\sim\mathbf{Q}_\sharp\mu}$.

Under the assumption that s,t are arbitrary functions without smoothness constraints, the above minimization problem decouples into one for each value of $\mathbf{p}$. We fix $\mathbf{p}$ for what follows and write $\mathbf{s}=s(\mathbf{p}),\mathbf{t}=t(\mathbf{p})\in\mathbb{R}^{D_A}$ instead of the corresponding functions and obtain:

$$\min_{\mathbf{s},\mathbf{t}\in\mathbb{R}^{D_A}} \mathbb{E}_{a_i|\mathbf{p}}\left[\frac{1}{2}\|\mathbf{a}\odot e^{\mathbf{s}}+\mathbf{t}\|^2-\sum_{i=1}^{D_A}s_i(\mathbf{p})\right]. \tag{29}$$

This can be decoupled into D_A independent minimization problems, indexed by $i=1,\dots,D_A$:

$$\min_{s_i,t_i\in\mathbb{R}} \mathbb{E}_{a_i|\mathbf{p}}\left[\frac{1}{2}(a_ie^{s_i}+t_i)^2-s_i\right]. \tag{30}$$

At an extremal point, we find

$$\partial_{s_i}\mathbb{E}_{a_i|\mathbf{p}}\left[\frac{1}{2}(e^{s_i}a_i+t_i)^2-s_i\right]=\mathbb{E}_{a_i|\mathbf{p}}\big[(e^{s_i}a_i+t_i)e^{s_i}a_i-1\big]=0, \tag{31}$$

$$\partial_{t_i}\mathbb{E}_{a_i|\mathbf{p}}\left[\frac{1}{2}(e^{s_i}a_i+t_i)^2-s_i\right]=\mathbb{E}_{a_i|\mathbf{p}}\big[(e^{s_i}a_i+t_i)\big]=0. \tag{32}$$

We can solve the second equation for t_i:

$$t_i=-\mathbb{E}_{a_i|\mathbf{p}}[a_i]e^{s_i}. \tag{33}$$

Insert this into the condition on s_i:

$$\mathbb{E}_{a_i|\mathbf{p}}\Big[2(e^{s_i}a_i-\mathbb{E}_{a_i|\mathbf{p}}[a_i]e^{s_i})e^{s_i}a_i-1\Big]=e^{2s_i}\operatorname{Var}_{a_i|\mathbf{p}}[a_i]-1=0, \tag{34}$$

and find:

$$\begin{aligned} e^{2s_i}\operatorname{Var}[a_i]&=1,\\ e^{s_i}&=\frac{1}{\sigma_{A_i|\mathbf{p}}}. \end{aligned} \tag{35}$$

This is the statement, written componentwise and for a fixed $\mathbf{p}$. □

A.2 Proof of Theorem 1

Proof. Inserting the minimizer from Equation (11), we find the minimal KL divergence of a single affine layer:

$$
\begin{aligned}
&\mathcal{D}_{\mathrm{KL}}(T_\sharp \mu || \mathcal{N}) = \mathcal{D}_{\mathrm{KL}}(\mu || T^\sharp \mathcal{N}) \\
&= \mathcal{D}_{\mathrm{KL}}(\mu_P || \mathcal{N}) + \mathbb{E}_{\mathbf{p},\mathbf{a}}\left[\log \mu(\mathbf{a}|\mathbf{p}) + D \log \sqrt{2\pi} + \frac{1}{2}(\mathbf{a}')^2 - \log |\nabla_{\mathbf{a}} \mathbf{a}'|\right] \\
&= \mathcal{D}_{\mathrm{KL}}(\mu_P || \mathcal{N}) + \mathbb{E}_{\mathbf{p},\mathbf{a}}\left[\log \mu(\mathbf{a}|\mathbf{p}) + \sum_{i=1}^{D_A}\left(\frac{(a_i - m_{A_i|\mathbf{p}})^2}{2\sigma^2_{A_i|\mathbf{p}}} + \log(\sigma_{A_i|\mathbf{p}}\sqrt{2\pi})\right)\right] \\
&= \mathcal{D}_{\mathrm{KL}}(\mu_P || \mathcal{N}) + \mathbb{E}_{\mathbf{p}}\left[\mathbb{E}_{\mathbf{a}|\mathbf{p}}[\log \mu(\mathbf{a}|\mathbf{p})] + \sum_{i=1}^{D_A}\Big(1 + \log(\sigma_{A_i|\mathbf{p}}\sqrt{2\pi})\Big)\right] \\
&= \mathcal{D}_{\mathrm{KL}}(\mu_P || \mathcal{N}) + \mathbb{E}_{\mathbf{p}}\left[\sum_{i=1}^{D_A} H[\mathcal{N}(0, \sigma_{A_i|\mathbf{p}})] - H[\mu(\mathbf{a}|\mathbf{p})]\right]. \qquad (36)
\end{aligned}
$$

Compare this to the KL divergence without transport:

$$
\begin{aligned}
&\mathcal{D}_{\mathrm{KL}}(\mu || \mathcal{N}) \qquad (37) \\
&= \mathcal{D}_{\mathrm{KL}}(\mu_P || \mathcal{N}) + \mathbb{E}_{\mathbf{p}}\left[\mathbb{E}_{\mathbf{a}|\mathbf{p}}[\log \mu(\mathbf{a}|\mathbf{p})] + D_A \log \sqrt{2\pi} + \frac{1}{2}\sum_{i=1}^{D_A} \mathbb{E}_{a|\mathbf{p}}[a_i^2]\right] \qquad (38) \\
&= \mathcal{D}_{\mathrm{KL}}(\mu_P || \mathcal{N}) + \mathbb{E}_{\mathbf{p}}\left[\frac{1}{2}\sum_{i=1}^{D_A}(m^2_{A_i|\mathbf{p}} + \sigma^2_{A_i|\mathbf{p}} + 2\log\sqrt{2\pi}) - H[\mu(\mathbf{a}|\mathbf{p})]\right]. \qquad (39)
\end{aligned}
$$

Subtracting the two, we find an improvement in KL divergence by the single affine layer of:

$$
\begin{aligned}
\Delta &= \mathcal{D}_{\mathrm{KL}}(\mu || \mathcal{N}) - \mathcal{D}_{\mathrm{KL}}(T_\sharp \mu || \mathcal{N}) \qquad (40) \\
&= \frac{1}{2}\sum_{i=1}^{D_A} \mathbb{E}_{\mathbf{p}}[m^2_{A_i|\mathbf{p}} + \sigma^2_{A_i|\mathbf{p}} - 1 - \log \sigma^2_{A_i|\mathbf{p}}] \qquad (41)
\end{aligned}
$$

which can be computed independently for the D_A active dimensions. □

A.3 Proof of Corollary 1

Proof. "⇒": Insert this particular choice of μ into Equation (18) to obtain the result.

"⇐": The push-forward $T_\sharp \mu(\mathbf{a}|\mathbf{p})$ can only be written as a product of its marginals if $\mu(\mathbf{a}|\mathbf{p})$ was a product distribution for each $\mathbf{p}$. Then, Equation (18) decouples into contributions from each $\mu(a_i|\mathbf{p})$. Each contribution is the negentropy of $\mu(a_i|\mathbf{p})$ which is only zero if $\mu(a_i|\mathbf{p})$ is Gaussian. □

Side note: Iff $\mathbf{Q}$ is chosen such that $\mu(\mathbf{a}|\mathbf{p})$ has diagonal covariance for all $\mathbf{p}$ (i.e. the relations between the dimensions are the same, they are just scaled differently for different $\mathbf{p}$), then it can be chosen such that the total KL divergence is zero after the layer.

A.4 Proof of Lemma 2

Proof. $1 \Rightarrow 2$: Rewriting Equation (11), we find:

$$\mathbf{a} = m_{A|\mathbf{p}} + \sigma_{A|\mathbf{p}} \odot \mathbf{a}' =: f(\mathbf{p}) + g(\mathbf{p}) \odot \mathbf{n}. \tag{42}$$

By assumption, $\mathbf{n} = \mathbf{a}' \perp \mathbf{p}$ and and we obtain the statement.

$2 \Rightarrow 1$: In the following, we omit "$\odot$" and all multiplications are element-wise.

We first identify the solution as in Equation (11) and then show that the resulting variable is independent of $\mathbf{p}$. In the following, we write $f = f(\mathbf{p})$ and $g = g(\mathbf{p})$.

$$\mathbb{E}_{\mathbf{n}}[A] = f + \mathbb{E}_{\mathbf{n}}[\mathbf{n}]g, \tag{43}$$

$$\mathbb{E}_{\mathbf{n}}[A^2] = f^2 + 2fg\mathbb{E}_{\mathbf{n}}[\mathbf{n}] + g^2\mathbb{E}[\mathbf{n}^2]. \tag{44}$$

We combine:

$$m_{A|\mathbf{p}} = f + \mathbb{E}_{\mathbf{n}}[\mathbf{n}]g, \tag{45}$$

$$\sigma_{A|\mathbf{p}} = \sqrt{f^2 + 2fg\mathbb{E}_{\mathbf{n}}[\mathbf{n}] + g^2\mathbb{E}[\mathbf{n}^2] - (f^2 + 2fg\mathbb{E}_{\mathbf{n}}[\mathbf{n}] + g^2\mathbb{E}_{\mathbf{n}}[\mathbf{n}]^2)} \tag{46}$$

$$= g\sigma_{\mathbf{n}}. \tag{47}$$

The resulting $\mathbf{a}'$ from this transport reads:

$$A' = T(A|\mathbf{p}) = \frac{1}{\sigma_{A|\mathbf{p}}}(\mathbf{a} - m_{A|\mathbf{p}}) \tag{48}$$

$$= \frac{1}{g\sigma_N}(f + \mathbf{n}g - (f + \mathbb{E}_{\mathbf{n}}[\mathbf{n}]g)) \tag{49}$$

$$= \frac{1}{\sigma_N}(\mathbf{n} - \mathbb{E}_{\hat{\mathbf{n}}}[\hat{\mathbf{n}}]). \tag{50}$$

This is independent of $\mathbf{p}$. □