Appendix

The Kalman filter (Kalman et al., 1960) works by iteratively applying two steps, predict and update. It assumes additive Gaussian noise with zero mean and covariances $\mathbf{\Sigma}^{\text{trans}}$ and $\mathbf{\Sigma}^{\text{obs}}$ on both transitions and observations, which need to be given to the filter. During the prediction step the transition model $\mathbf{A}$ is used to infer the next prior state estimate $\left(\mathbf{x}_{t+1}^-, \mathbf{\Sigma}_{t+1}^-\right)$, i.e., a-priori to the observation, from the current posterior estimate $\left(\mathbf{x}_t^+, \mathbf{\Sigma}_t^+\right)$, by

$$\begin{aligned} \mathbf{x}_{t+1}^- &= \mathbf{A}\mathbf{x}_t^+ \\ \text{and} \quad \mathbf{\Sigma}_{t+1}^- &= \mathbf{A}\mathbf{\Sigma}_t^+\mathbf{A}^T + \mathbf{\Sigma}^{\text{trans}}. \end{aligned}$$

The prior estimate is then updated using the current observation $\mathbf{w_t}$ and the observation model $\mathbf{H}$ to obtain the posterior estimate $\left(\mathbf{x}_t^+, \mathbf{\Sigma}_t^+\right)$, i.e.,

$$\begin{aligned} \mathbf{x}_t^+ &= \mathbf{x}_t^- + \mathbf{Q}_t\left(\mathbf{w}_t - \mathbf{H}\mathbf{x}_t^-\right) \\ \text{and} \quad \mathbf{\Sigma}_t^+ &= \left(\mathbf{I} - \mathbf{Q}_t\mathbf{H}\right)\mathbf{\Sigma}_t^- \\ \text{, with} \quad \mathbf{Q}_t &= \mathbf{\Sigma}_t^-\mathbf{H}^T\left(\mathbf{H}\mathbf{\Sigma}_t^-\mathbf{H}^T + \mathbf{\Sigma}^{\text{obs}}\right)^{-1}, \end{aligned}$$

where $\mathbf{I}$ denotes the identity matrix. The matrix $\mathbf{Q}_t$ is referred to as the Kalman gain. The whole update step can be interpreted as a weighted average between state and observation estimate, where the weighting, i.e., $\mathbf{Q}_t$, depends on the uncertainty about those estimates.

If currently no observation is present or future states should be predicted, the update step is omitted.

A. Simplified Kalman Filter Formulas

As stated above the simple latent observation model $\mathbf{H} = \begin{bmatrix} \mathbf{I}_m & \mathbf{0}_{m\times(n-m)} \end{bmatrix}$, as well as the assumed factorization of the covariance matrices allow us to simplify the Kalman Filter equations.

A.1. Notation

In the following derivations we neglect the time indices t and $t+1$ for brevity. For any matrix $\mathbf{M}$, $\hat{\mathbf{M}}$ denotes a diagonal matrix with the same diagonal as $\mathbf{M}$, $\mathbf{m}$ denotes a vector containing those diagonal elements and $M^{(ij)}$ denotes the entry at row i and column j. Similarly, $v^{(i)}$ denotes the i-th entry of a vector $\mathbf{v}$. The point wise product between two vectors of same length (Hadamat Product) will be denoted by $\odot$ and the point wise division by $\oslash$.

A.2. Prediction Step

$$\begin{aligned} &\text{Mean:} && \mathbf{z}^- = \mathbf{A}\mathbf{z}^+ \\ &\text{Covariance:} && \mathbf{\Sigma}^- = \mathbf{A}\mathbf{\Sigma}^+\mathbf{A}^T + \hat{\mathbf{\Sigma}}^{\text{trans}} \end{aligned}$$

The computation of the mean can not be further simplified, however, depending on the state size and bandwidth, sparse matrix multiplications may be exploited. For the covariance, let $\mathbf{T} = \mathbf{A}\mathbf{\Sigma}^+$. Then,

$$\begin{aligned} \mathbf{T} = \mathbf{A}\mathbf{\Sigma}^+ &= \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{bmatrix}\begin{bmatrix} \hat{\mathbf{\Sigma}}^{\text{u},+} & \hat{\mathbf{\Sigma}}^{\text{s},+} \\ \hat{\mathbf{\Sigma}}^{\text{s},+} & \hat{\mathbf{\Sigma}}^{\text{l},+} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{B}_{11}\hat{\mathbf{\Sigma}}^{\text{u},+} + \mathbf{B}_{12}\hat{\mathbf{\Sigma}}^{\text{s},+} & \mathbf{B}_{11}\hat{\mathbf{\Sigma}}^{\text{s},+} + \mathbf{B}_{12}\hat{\mathbf{\Sigma}}^{\text{l},+} \\ \mathbf{B}_{21}\hat{\mathbf{\Sigma}}^{\text{u},+} + \mathbf{B}_{22}\hat{\mathbf{\Sigma}}^{\text{s},+} & \mathbf{B}_{21}\hat{\mathbf{\Sigma}}^{\text{s},+} + \mathbf{B}_{22}\hat{\mathbf{\Sigma}}^{\text{l},+} \end{bmatrix} \end{aligned}$$

and

$$\begin{aligned} \mathbf{\Sigma}^- &= \mathbf{T}\mathbf{A}^T + \hat{\mathbf{\Sigma}}^{\text{trans}} \\ &= \begin{bmatrix} \mathbf{B}_{11}\hat{\mathbf{\Sigma}}^{\text{u},+} + \mathbf{B}_{12}\hat{\mathbf{\Sigma}}^{\text{s},+} & \mathbf{B}_{11}\hat{\mathbf{\Sigma}}^{\text{s},+} + \mathbf{B}_{12}\hat{\mathbf{\Sigma}}^{\text{l},+} \\ \mathbf{B}_{21}\hat{\mathbf{\Sigma}}^{\text{u},+} + \mathbf{B}_{22}\hat{\mathbf{\Sigma}}^{\text{s},+} & \mathbf{B}_{21}\hat{\mathbf{\Sigma}}^{\text{s},+} + \mathbf{B}_{22}\hat{\mathbf{\Sigma}}^{\text{l},+} \end{bmatrix}\cdots \\ \cdots &\begin{bmatrix} \mathbf{B}_{11}^T & \mathbf{B}_{21}^T \\ \mathbf{B}_{12}^T & \mathbf{B}_{22}^T \end{bmatrix} + \hat{\mathbf{\Sigma}}^{\text{trans}} = \begin{bmatrix} \hat{\mathbf{\Sigma}}^{\text{u},-} & \hat{\mathbf{\Sigma}}^{\text{s},-} \\ \hat{\mathbf{\Sigma}}^{\text{s},-} & \hat{\mathbf{\Sigma}}^{\text{l},-} \end{bmatrix}, \end{aligned}$$

with

$$\begin{aligned} \mathbf{\Sigma}^{\text{u},-} =& \left(\mathbf{B}_{11}\hat{\mathbf{\Sigma}}^{\text{u},+} + \mathbf{B}_{12}\hat{\mathbf{\Sigma}}^{\text{s},+}\right)\mathbf{B}_{11}^T\ldots \\ &\ldots + \left(\mathbf{B}_{11}\hat{\mathbf{\Sigma}}^{\text{s},+} + \mathbf{B}_{12}\hat{\mathbf{\Sigma}}^{\text{l},+}\right)\mathbf{B}_{12}^T + \hat{\mathbf{\Sigma}}^{\text{u,trans}} \\ =& \mathbf{B}_{11}\hat{\mathbf{\Sigma}}^{\text{u},+}\mathbf{B}_{11}^T + \mathbf{B}_{12}\hat{\mathbf{\Sigma}}^{\text{s},+}\mathbf{B}_{11}^T\ldots \\ &\ldots + \mathbf{B}_{11}\hat{\mathbf{\Sigma}}^{\text{s},+}\mathbf{B}_{12}^T + \mathbf{B}_{12}\hat{\mathbf{\Sigma}}^{\text{l},+}\mathbf{B}_{12}^T + \hat{\mathbf{\Sigma}}^{\text{u,trans}} \\ \mathbf{\Sigma}^{\text{l},-} =& \left(\mathbf{B}_{21}\hat{\mathbf{\Sigma}}^{\text{u},+} + \mathbf{B}_{22}\hat{\mathbf{\Sigma}}^{\text{s},+}\right)\mathbf{B}_{21}^T\ldots \\ &\ldots + \left(\mathbf{B}_{21}\hat{\mathbf{\Sigma}}^{\text{s},+} + \mathbf{B}_{22}\hat{\mathbf{\Sigma}}^{\text{l},+}\right)\mathbf{B}_{22}^T + \hat{\mathbf{\Sigma}}^{\text{l,trans}} \\ =& \mathbf{B}_{21}\hat{\mathbf{\Sigma}}^{\text{u},+}\mathbf{B}_{21}^T + \mathbf{B}_{22}\hat{\mathbf{\Sigma}}^{\text{s},+}\mathbf{B}_{21}^T\ldots \\ &\ldots + \mathbf{B}_{21}\hat{\mathbf{\Sigma}}^{\text{s},+}\mathbf{B}_{22}^T + \mathbf{B}_{22}\hat{\mathbf{\Sigma}}^{\text{l},+}\mathbf{B}_{22}^T + \hat{\mathbf{\Sigma}}^{\text{l,trans}} \\ \mathbf{\Sigma}^{\text{s},-} =& \left(\mathbf{B}_{21}\hat{\mathbf{\Sigma}}^{\text{u},+} + \mathbf{B}_{22}\hat{\mathbf{\Sigma}}^{\text{s},+}\right)\mathbf{B}_{11}^T\ldots \\ &\ldots + \left(\mathbf{B}_{21}\hat{\mathbf{\Sigma}}^{\text{s},+} + \mathbf{B}_{22}\hat{\mathbf{\Sigma}}^{\text{l},+}\right)\mathbf{B}_{12}^T \\ =& \mathbf{B}_{21}\hat{\mathbf{\Sigma}}^{\text{u},+}\mathbf{B}_{11}^T + \mathbf{B}_{22}\hat{\mathbf{\Sigma}}^{\text{s},+}\mathbf{B}_{11}^T\ldots \\ &\ldots + \mathbf{B}_{21}\hat{\mathbf{\Sigma}}^{\text{s},+}\mathbf{B}_{12}^T + \mathbf{B}_{22}\hat{\mathbf{\Sigma}}^{\text{l},+}\mathbf{B}_{12}^T \end{aligned}$$

Since we are only interested in the diagonal parts of $\mathbf{\Sigma}^{\text{u},-}, \mathbf{\Sigma}^{\text{l},-}$ and $\mathbf{\Sigma}^{\text{s},-}$ i.e. $\hat{\mathbf{\Sigma}}^{\text{u},-}, \hat{\mathbf{\Sigma}}^{\text{l},-}$ and $\hat{\mathbf{\Sigma}}^{\text{s},-}$, we can further simplify these equations by realizing two properties of the terms above. First, for any matrix $\mathbf{M}, \mathbf{N}$ and a diagonal matrix $\hat{\mathbf{\Sigma}}$ it holds that

$$\left(\mathbf{M}\hat{\mathbf{\Sigma}}\mathbf{N}^T\right)^{(ii)} = \sum_{k=1}^{n} A^{(ik)}B^{(ik)}\sigma^{(k)} = \left(\mathbf{N}\hat{\mathbf{\Sigma}}\mathbf{M}^T\right)_{ii}.$$

Hence, we can simplify the equations for the upper and lower part to

$$\begin{aligned}\boldsymbol{\Sigma}^{u,-} =&\mathbf{B}_{11}\hat{\boldsymbol{\Sigma}}^{\mathrm{u},+}\mathbf{B}_{11}^T + 2\cdot\mathbf{B}_{12}\hat{\boldsymbol{\Sigma}}^{\mathrm{s},+}\mathbf{B}_{11}^T\dots\\ &\dots+\mathbf{B}_{12}\hat{\boldsymbol{\Sigma}}^{\mathrm{l},+}\mathbf{B}_{12}^T+\hat{\boldsymbol{\Sigma}}^{\mathrm{u,trans}},\\ \boldsymbol{\Sigma}^{\mathrm{l},-} =&\mathbf{B}_{21}\hat{\boldsymbol{\Sigma}}^{\mathrm{u},+}\mathbf{B}_{21}^T + 2\cdot\mathbf{B}_{22}\hat{\boldsymbol{\Sigma}}^{\mathrm{s},+}\mathbf{B}_{21}^T\dots\\ &\dots+\mathbf{B}_{22}\hat{\boldsymbol{\Sigma}}^{\mathrm{l},+}\mathbf{B}_{22}^T+\hat{\boldsymbol{\Sigma}}^{\mathrm{l,trans}}.\end{aligned}$$

Second, since we are only interested in the diagonal of the result it is sufficient to compute only the diagonals of the individual parts of the sums which are almost all of the same structure i.e. $\mathbf{S}=\mathbf{M}\hat{\boldsymbol{\Sigma}}^{+}\mathbf{N}^T$. Let $\mathbf{T}=\mathbf{M}\hat{\boldsymbol{\Sigma}}^{+}$, then each element of $\mathbf{T}$ can be computed as

$$T^{(ij)}=\sum_{k=1}^{n}M^{(ik)}\hat{\Sigma}^{(kj)}=M^{(ij)}\sigma^{(j)}.$$

Consequently, the elements of $\mathbf{S}=\mathbf{T}\mathbf{A}^T$ can be computed as

$$S^{(ij)}=\sum_{k=1}^{n}T^{(ik)}A^{(kj)}=\sum_{k=1}^{n}M^{(ik)}\sigma_k N^{(jk)}.$$

Ultimately, we are not interested in $\mathbf{S}$ but only in $\hat{\mathbf{S}}$

$$\hat{S}^{(ii)}=\sum_{k=1}^{n}M^{(ik)}N^{(ik)}\sigma^{(}k).$$

Using this we obtain can obtain the entries of $\boldsymbol{\sigma}^{u,-}, \boldsymbol{\sigma}^{l,-}$ and $\boldsymbol{\sigma}^{s,-}$ by

$$\begin{aligned}\sigma^{u,-,(i)} =& \qquad (10)\\ &\sum_{k=1}^{m}\left(B_{11}^{(ik)}\right)^2\sigma^{u,+,(i)}+2\sum_{k=1}^{m}B_{11}^{(ik)}B_{12}^{(ik)}\sigma^{s,+,(i)}\dots\\ \dots+&\sum_{k=1}^{m}\left(B_{12}^{(ik)}\right)^2\sigma^{l,+,(i)}+\sigma^{u,\mathrm{trans},(i)}\\ \sigma^{l,-,(i)} =& \qquad (11)\\ &\sum_{k=1}^{m}\left(B_{21}^{(ik)}\right)\sigma^{u,+,(i)}+2\sum_{k=1}^{m}B_{22}^{(ik)}B_{21}^{(ik)}\sigma^{s,+,(i)}\dots\\ \dots+&\sum_{k=1}^{m}\left(B_{22}^{(ik)}\right)^2\sigma^{l,+,(i)}+\sigma^{l,\mathrm{trans},(i)}\\ \sigma^{s,-,(i)} =& \qquad (12)\\ &\sum_{k=1}^{m}B_{21}^{(ik)}B_{11}^{(ik)}\sigma^{u,+,(i)}+\sum_{k=1}^{m}B_{22}^{(ik)}B_{11}^{(ik)}\sigma^{s,+,(i)}\dots\\ \dots+&\sum_{k=1}^{m}B_{21}^{(ik)}B_{12}^{(ik)}\sigma^{s,+,(i)}+\sum_{k=1}^{m}B_{22}^{(ik)}B_{12}^{(ik)}\sigma^{l,+,(i)},\end{aligned}$$

which can be implemented efficiently using elementwise matrix multiplication and sum reduction.

A.3. Update Step

Kalman Gain	$\mathbf{Q}=\boldsymbol{\Sigma}^{-}\mathbf{H}^T\left(\mathbf{H}\boldsymbol{\Sigma}^{-}\mathbf{H}^T+\boldsymbol{\Sigma}^{\mathrm{obs}}\right)^{-1}$
Mean	$\mathbf{z}^{+}=\mathbf{z}^{-}+\mathbf{Q}\left(\mathbf{w}-\mathbf{H}\mathbf{z}^{-}\right)$
Covariance	$\boldsymbol{\Sigma}^{+}=\left(\mathbf{I}-\mathbf{Q}\mathbf{H}\right)\boldsymbol{\Sigma}^{-}$

First, note that

$$\boldsymbol{\Sigma}^{-}\mathbf{H}^T=\begin{bmatrix}\hat{\boldsymbol{\Sigma}}^{\mathrm{u},-}\\ \hat{\boldsymbol{\Sigma}}^{\mathrm{s},-}\end{bmatrix}$$

and

$$\mathbf{H}\boldsymbol{\Sigma}^{-}\mathbf{H}^T+\hat{\boldsymbol{\Sigma}}^{\mathrm{obs}}=\hat{\boldsymbol{\Sigma}}^{u,-}+\hat{\boldsymbol{\Sigma}}^{\mathrm{obs}}$$

and thus the computation of the Kalman Gain only involves diagonal matrices. Hence the Kalman Gain matrix also consists of two diagonal matrices, i.e., $\mathbf{Q}=\begin{bmatrix}\hat{\mathbf{Q}}^{\mathrm{u}}\\ \hat{\mathbf{Q}}^{\mathrm{l}}\end{bmatrix}$ whose diagonals can be computed by

$$\mathbf{q}^{\mathrm{u}}=\boldsymbol{\sigma}^{\mathrm{u},-}\oslash\left(\boldsymbol{\sigma}^{\mathrm{u},-}+\boldsymbol{\sigma}^{\mathrm{obs}}\right) \qquad (13)$$

$$\text{and}\quad \mathbf{q}^{\mathrm{l}}=\boldsymbol{\sigma}^{\mathrm{s},-}\oslash\left(\boldsymbol{\sigma}^{\mathrm{u},-}+\boldsymbol{\sigma}^{\mathrm{obs}}\right). \qquad (14)$$

Using this result, the mean update can be simplified to

$$\mathbf{z}^{+}=\mathbf{z}^{-}+\begin{bmatrix}\mathbf{q}_{\mathrm{u}}\\ \mathbf{q}_{\mathrm{l}}\end{bmatrix}\odot\begin{bmatrix}\mathbf{w}-\mathbf{z}^{\mathrm{u},-}\\ \mathbf{w}-\mathbf{z}^{\mathrm{u},-}\end{bmatrix}. \qquad (15)$$

For the covariance we get:

$$\begin{aligned}&\begin{bmatrix}\hat{\boldsymbol{\Sigma}}^{\mathrm{u},+} & \hat{\boldsymbol{\Sigma}}^{\mathrm{s},+}\\ \hat{\boldsymbol{\Sigma}}^{\mathrm{s},+} & \hat{\boldsymbol{\Sigma}}^{\mathrm{l},+}\end{bmatrix}=\left(\mathbf{I}_n-\mathbf{Q}\mathbf{H}\right)\boldsymbol{\Sigma}^{-}=\\ &\begin{bmatrix}\mathbf{I}_m-\hat{\mathbf{Q}}^{\mathrm{u}} & \mathbf{0}_{m\times m}\\ -\hat{\mathbf{Q}}^{\mathrm{l}} & \mathbf{I}_m\end{bmatrix}\begin{bmatrix}\hat{\boldsymbol{\Sigma}}^{\mathrm{u},-} & \hat{\boldsymbol{\Sigma}}^{\mathrm{s},-}\\ \hat{\boldsymbol{\Sigma}}^{\mathrm{s},-} & \hat{\boldsymbol{\Sigma}}^{\mathrm{l},-}\end{bmatrix}\\ &=\begin{bmatrix}\left(\mathbf{I}_m-\hat{\mathbf{Q}}^{\mathrm{u}}\right)\hat{\boldsymbol{\Sigma}}^{\mathrm{u},-} & \left(\mathbf{I}_m-\hat{\mathbf{Q}}^{\mathrm{u}}\right)\hat{\boldsymbol{\Sigma}}^{\mathrm{s},-}\\ -\hat{\mathbf{Q}}^{\mathrm{l}}\hat{\boldsymbol{\Sigma}}^{\mathrm{u},-}+\hat{\boldsymbol{\Sigma}}^{\mathrm{s},-} & -\hat{\mathbf{Q}}^{\mathrm{l}}\hat{\boldsymbol{\Sigma}}^{\mathrm{s},-}+\hat{\boldsymbol{\Sigma}}^{\mathrm{l},-}\end{bmatrix}.\end{aligned}$$

Hence the diagonals of the individual parts can be computed as

$$\boldsymbol{\sigma}^{\mathrm{u},+}=\left(\mathbf{1}_m-\mathbf{q}^{\mathrm{u}}\right)\odot\boldsymbol{\sigma}^{\mathrm{u},-} \qquad (16)$$

$$\boldsymbol{\sigma}^{\mathrm{s},+}=\left(\mathbf{1}_m-\mathbf{q}^{\mathrm{u}}\right)\odot\boldsymbol{\sigma}^{\mathrm{s},-} \qquad (17)$$

$$\boldsymbol{\sigma}^{\mathrm{l},+}=\boldsymbol{\sigma}^{\mathrm{l},-}-\mathbf{q}^{\mathrm{l}}\odot\boldsymbol{\sigma}^{\mathrm{s},-}. \qquad (18)$$

B. Root Mean Square Error Results

To evaluate the actual prediction performance of our approach we repeated some experiments using the RMSE as loss function. Other than that and removing the variance output of the decoder no changes were made to the model, hyperparameters and learning procedure. The results can be found in Table 7.

Table 7. RMSE Results

Model	RMSE
Pendulum	
RKN ($m = 15, b = 3, K = 15$)	0.0779 ± 0.0082
RKN ($m = b = 15, K = 15$)	0.0758 ± 0.0094
LSTM ($m = 50$)	0.0920 ± 0.0774
LSTM ($m = 6$)	0.0959 ± 0.0100
GRU ($m = 50$)	0.0821 ± 0.0084
GRU ($m = 8$)	0.0916 ± 0.0087
Multiple Pendulums	
RKN ($m = 45, b = 3, k = 15$)	0.0878 ± 0.0036
LSTM ($m = 50$)	0.098 ± 0.0036
LSTM ($m = 12$)	0.104 ± 0.0043
GRU ($m = 50$)	0.112 ± 0.0371
GRU ($m = 14$)	0.105 ± 0.0055
Quad Link (without additional noise)	
RKN ($m = 100, b = 25, k = 15$)	0.103 ± 0.00076
LSTM ($m = 100$)	0.175 ± 0.182
LSTM ($m = 25$)	0.118 ± 0.0049
GRU ($m = 100$)	0.278 ± 0.105
GRU ($m = 25$)	0.121 ± 0.0021
Quad Link (with additional noise)	
RKN ($m = 100, b = 25, k = 15$)	0.171 ± 0.0039
LSTM ($m = 75$)	0.175 ± 0.0022
GRU ($m = 25$)	0.204 ± 0.0023

C. Visualization of Imputation Results

Exemplary results of the data imputation experiments conducted for the Pendulum and Quad Link experiment can be found in Figure 5

D. Network Architectures and Hyper Parameters

For all experiments Adam (Kingma & Ba, 2014) with default parameters ($\alpha = 10^{-3}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 10^{-8}$) was used as an optimizer. The gradients were computed using (truncated) Backpropagation Trough Time (BpTT) (Werbos, 1990). Further in all (transposed) convolutional layers layer normalization (LN) (Ba et al., 2016) was employed to normalize the filter responses. "Same" padding was used. The elu activation function (Clevert et al., 2015) plus a constant 1 is denoted by $(\text{elu} + 1)$ was used to ensure that the variance outputs are positive.

D.1. Pendulum and Multiple Pendulum Experiments

Observations. *Pendulum*: Grayscale images of size 24×24 pixels. *Multiple Pendulum*: RGB images of size 24×24 pixels. See Figure 6 for examples.

Dataset. 1000 Train and 500 Test sequences of length 150. For the filtering experiments noise according to section E was added , for imputation 50% of the images were removed randomly.

Encoder. 2 convolution + 1 fully connected + linear output & $(\text{elu} + 1)$ output:

- Convolution 1: 12, 5×5 filter, ReLU, 2×2 max pool with 2×2 stride
- Convolution 2: 12, 3×3 filter with 2×2 stride, ReLU, 2×2 max pool with 2×2 stride
- *Pendulum*: Fully Connected 1: 30, ReLU
- *Multiple Pendulum*: Fully Connected 1: 90, ReLU

Transition Model *Pendulum*: 15 dimensional latent observation, 30 dimensional latent state. *Multiple Pendulum*: 45 dimensional latent observation, 90 dimensonal latent state. *Both:* bandwidth: 3, number of basis: 15

- $\alpha(\mathbf{z}_t)$: No hidden layers - softmax output

Decoder (for $\mathbf{s}_t^+$). 1 fully connected + linear output:

- Fully Connected 1: 10, ReLU

Decoder (for $\mathbf{o}_t^+$). : 1 fully connected + 2 transposed convolution + transposed convolution output:

- Fully Connected 1: 144 ReLU
- Transposed Convolution 1: 16, 5×5 filter with 4×4 stride, ReLU
- Transposed Convolution 2: 12, 3×3 filter with 2×2 stride, ReLU
- Transposed Convolution Out: *Pendulum*: 1 *Multiple Pendulum*: 3, 3×3 filter with 1×1 stride, Sigmoid

Decoder (for $\boldsymbol{\sigma}_t^+$ or σ_t^+). 1 fully connected + $(\text{elu} + 1)$:

- Fully Connected 1: 10, ReLU

D.2. Quad Link

Observations. Grayscale images of size $48x48$ pixels.

Dataset. 4000 Train and 1000 Test sequences of length 150. For the filtering with additional noise experiments noise according to section D was added, for imputation 50% of the images were removed randomly.

Encoder. 2 convolution + 1 fully connected + linear output & $(\text{elu}+1)$ output:

- Convolution 1: 12, 5×5 filter with 2×2 stride, ReLU, 2×2 max pool with 2×2 stride
- Convolution 2: 12, 3×3 filter with 2×2 stride, ReLU, 2×2 max pool with 2×2 stride
- Fully Connected 1: 200 ReLU

Transition Model. 100 dimensional latent observation, 200 dimensional latent state, bandwidth: 3, number of basis: 15

- $\alpha(\mathbf{z}_t)$: No hidden layers - softmax output

Decoder (for $\mathbf{s}_t^+$). 1 fully connected + linear output:

- Fully Connected 1: 10, ReLU

Decoder (for $\mathbf{o}_t^+$). 1 fully connected + 2 transposed convolution + transposed convolution output:

- Fully Connected 1: 144 ReLU
- Transposed Convolution 1: 16, 5×5 filter with 4×4 stride, ReLU
- Transposed Convolution 2: 12, 3×3 filter with 4×4 stride, ReLU
- Transposed Convolution Out: 1, 1×1 stride, Sigmoid

Decoder (for $\boldsymbol{\sigma}_t^+$ or σ_t^+. 1 fully connected + $(\text{elu}+1)$:

- Fully Connected 1: 10, ReLU

D.3. Kitti

Observation and Data Set. For this experiment, our encoder is based on the pose network proposed by (Zhou et al., 2017) which helps us to speed-up the training process. Specifically we extract features from the conv6 layer of the pose network by running the model on the KITTI odometry dataset. The training dataset for this experiment comprised of sequences 00, 01, 02, 08, 09. Sequences 03, 04, 05, 06, 07, 10 were used for testing.

Encoder. Pose Network of (Zhou et al., 2017) up to layer conv6 + 1 Convolution

- Convolution 1: 50, 1x1 filter,with 1x1 stride

Transition Model. 50 dimensional latent observations, 100 dimensional latent state, bandwidth 1, number of basis 16

- $\alpha(\mathbf{z}_t)$: No hidden layers - softmax output

Decoder (for $\mathbf{s}_t^+$). 2 fully connected + linear output:

- Fully Connected 1: 50, ReLU

D.4. Pneumatic Brook Robot Arm

Observations and Data Set. 6 sequences of $30,000$ samples each of input currents and observed joint positions, sampled at 100Hz. 5 sequences were used for training, 1 for testing.

Encoder. 1 fully connected + linear output & $(\text{elu}+1)$ output.

- Fully Connected, 30 ReLU

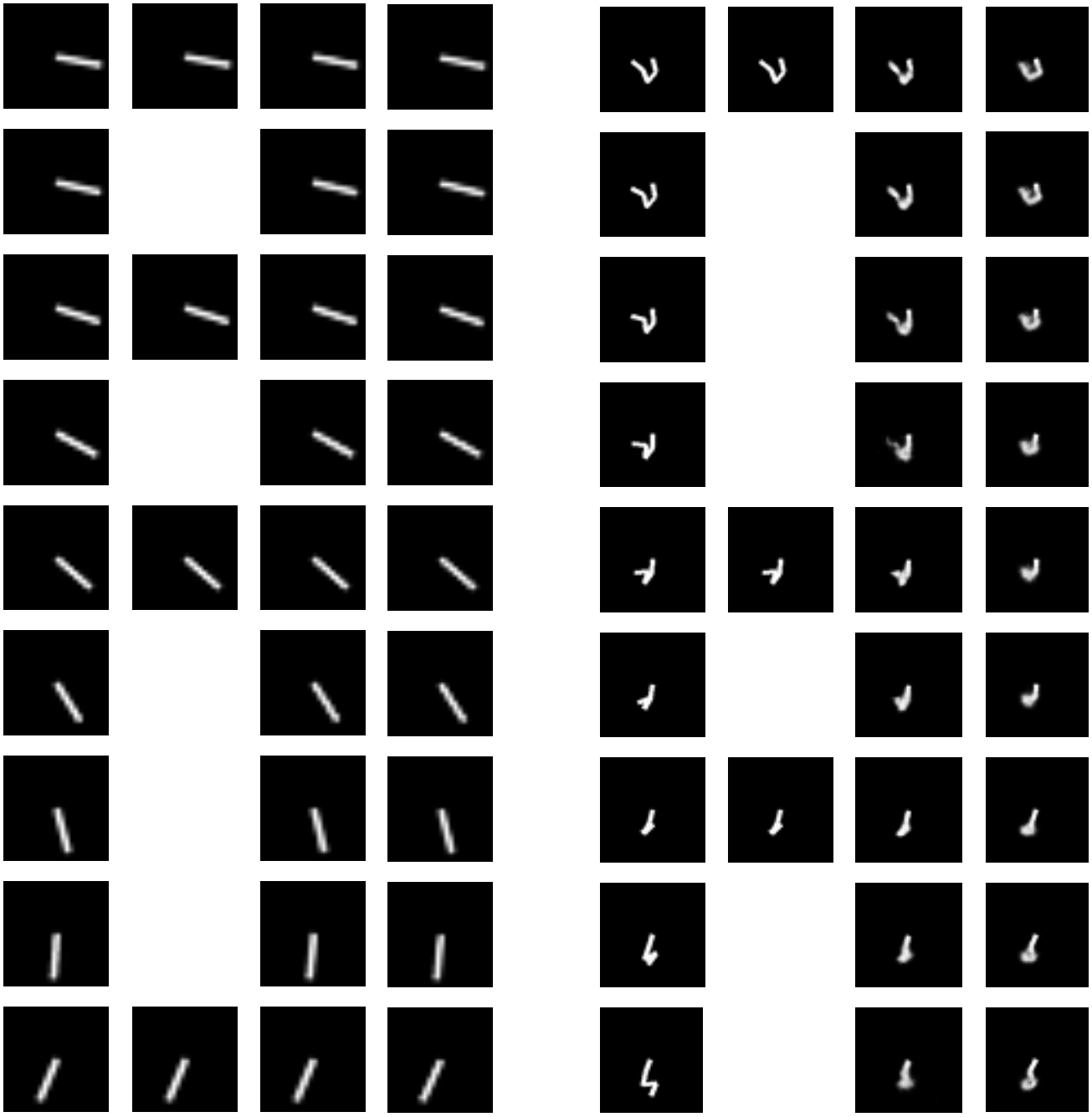
Transition Model. 30 dimensional latent observation, 60 dimensional latent state, bandwidth 3, number of basis 32

Decoder (for $\mathbf{s}_t^+$. 1 fully connected + linear output

- 30 ReLU

E. Observation Noise generation process

Let $\mathcal{U}(x,y)$ denote the uniform distribution from x to y.To generate the noise for the pendulum task for each sequence a sequence of factors f_t of same length was generated. To correlate the factors they were sampled as $f_0 \sim \mathcal{U}(0,1)$ and $f_{t+1} = \min(\max(0, f_t + r_t), 1)$ with $r_t \sim \mathcal{U}(-0.2, 0.2)$. Afterwards, for each sequence two thresholds $t_1 \sim \mathcal{U}(0.0, 0.25)$ and $t_2 \sim \mathcal{U}(0.75, 1)$ were sampled. All $f_t < t_1$ were set to 0, all $f_t > t_2$ to 1 and the rest was linearly mapped to the interval $[0,1]$. Finally, for each image $\mathbf{i}_t$ an image consisting of pure uniformly distributed noise $\mathbf{i}_t^{noise}$ was sampled and the observation computed as $\mathbf{o}_t = f_t \cdot \mathbf{i}_t + (1-f_t)\cdot \mathbf{i}_t^{noise}$.



(a) Pendulum (b) Quad Link

Figure 5. Each of (a) and (b) shows from left to right: true images, input to the models, imputation results for RKF, imputation results for KVAE(Fraccaro et al., 2017).

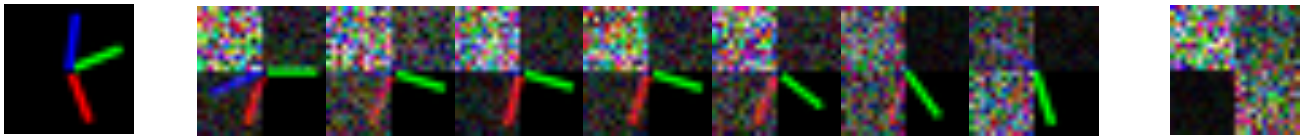


Figure 6. Example images for the multiple pendulum experiments. **Left**: Noise free image. **Middle**: sequence of images showing how the noise affects different pendulums differently. **Right**: Image without useful information.