Explainable Cognitive Models for Audiovisual Speaker Localization

12th AABBA General Meeting

Christopher Schymura and Dorothea Kolossa

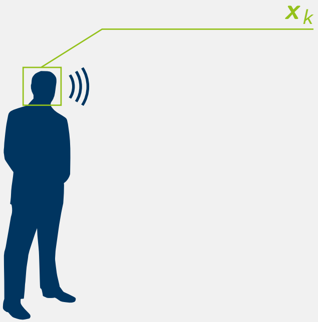
16th January 2020

RUHR UNIVERSITÄT BOCHUM

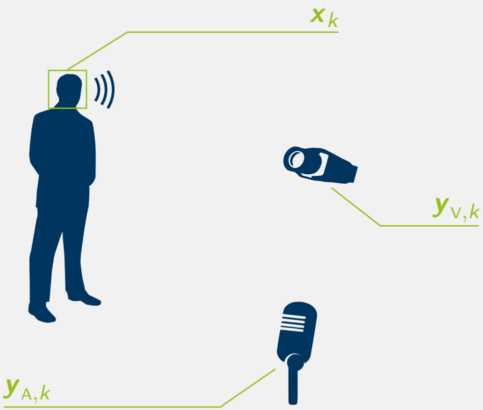
Problem statement

Problem statement

Problem statement



Problem statement

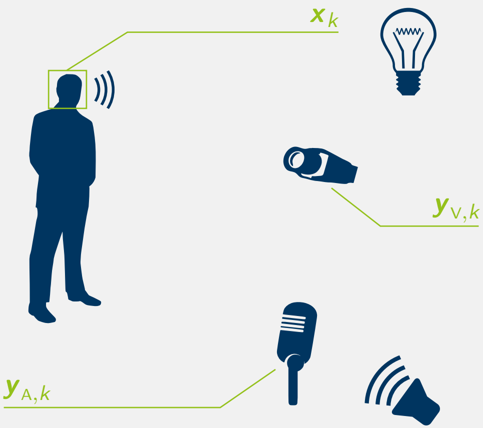


Observation functions:

$$\boldsymbol{y}_{\mathrm{A},k} = h_{\mathrm{A}}(\boldsymbol{x}_k)$$

$$\boldsymbol{y}_{\mathrm{V},k} = h_{\mathrm{V}}(\boldsymbol{x}_k)$$

Problem statement

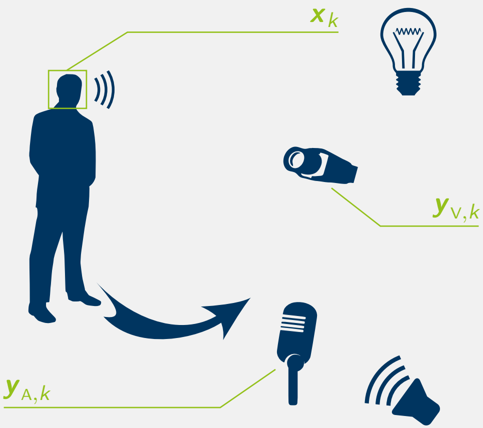


Observation functions:

$$\boldsymbol{y}_{\mathrm{A},k} = h_{\mathrm{A}}(\boldsymbol{x}_k) + \boldsymbol{w}_{\mathrm{A},k}$$

$$\boldsymbol{y}_{\mathrm{V},k} = h_{\mathrm{V}}(\boldsymbol{x}_k) + \boldsymbol{w}_{\mathrm{V},k}$$

Problem statement



State transition function:

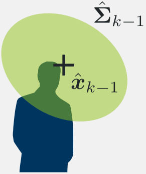
$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}) + \boldsymbol{v}_k$$

Observation functions:

$$\boldsymbol{y}_{\mathrm{A},k} = h_{\mathrm{A}}(\boldsymbol{x}_k) + \boldsymbol{w}_{\mathrm{A},k}$$

$$\boldsymbol{y}_{\mathrm{V},k} = h_{\mathrm{V}}(\boldsymbol{x}_k) + \boldsymbol{w}_{\mathrm{V},k}$$

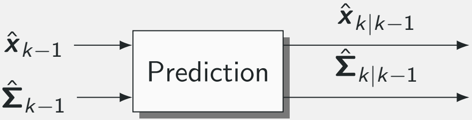
Recursive state estimation



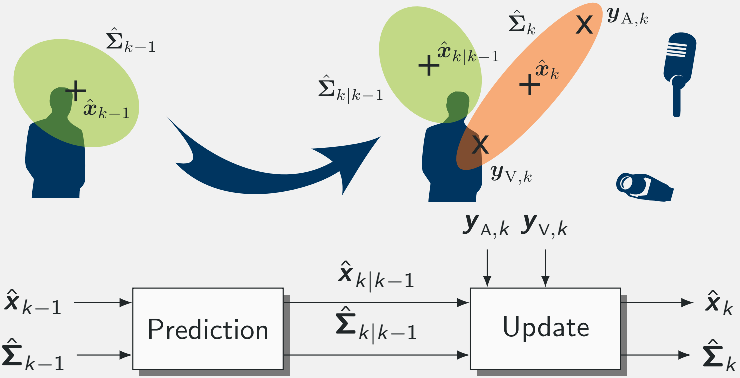
$\hat{\boldsymbol{x}}_{k-1}$

$\hat{\boldsymbol{\Sigma}}_{k-1}$

Recursive state estimation

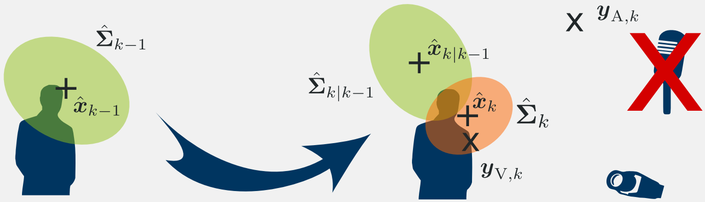


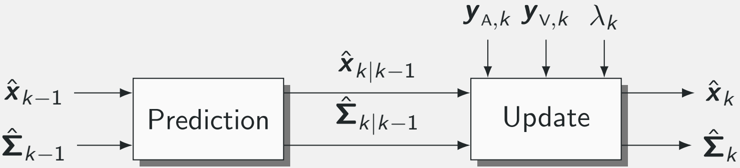
Recursive state estimation



$$\underbrace{p(\boldsymbol{x}_k | \boldsymbol{Y}_{A,1:k}, \boldsymbol{Y}_{V,1:k})}_{\text{Posterior}} \propto \underbrace{p(\boldsymbol{x}_k | \boldsymbol{Y}_{A,1:k-1}, \boldsymbol{Y}_{V,1:k-1})}_{\text{Prior}} \underbrace{p(\boldsymbol{y}_{A,k}, \boldsymbol{y}_{V,k} | \boldsymbol{x}_k)}_{\text{Sensor model}}$$

Recursive state estimation

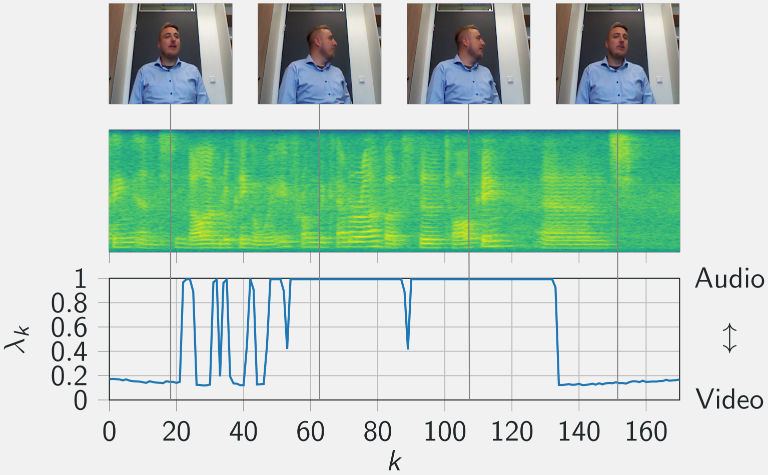




$$\underbrace{p(\mathbf{x}_k|\mathbf{Y}_{A,1:k},\mathbf{Y}_{V,1:k})}_{\text{Posterior}} \propto \underbrace{p(\mathbf{x}_k|\mathbf{Y}_{A,1:k-1},\mathbf{Y}_{V,1:k-1})}_{\text{Prior}} \underbrace{p(\mathbf{y}_{A,k}|\mathbf{x}_k)^{\lambda_k}\, p(\mathbf{y}_{V,k}|\mathbf{x}_k)^{1-\lambda_k}}_{\text{Sensor model w. stream weights}^{2}}$$

[2] C. Schymura et al.: *Extending linear dynamical systems with dynamic stream weights for audiovisual speaker localization*, IWAENC, 2018

Dynamic stream weights



Dynamic stream weights

Assumption: $\boldsymbol{x}_k$, $\boldsymbol{y}_{\mathrm{A},k}$, $\boldsymbol{y}_{\mathrm{V},k}$, $k = 1, \ldots, K$ fully observed, $\lambda_k \in [0, 1]$ and i.i.d.

Dynamic stream weights

Assumption: $\boldsymbol{x}_k$, $\boldsymbol{y}_{\mathrm{A},k}$, $\boldsymbol{y}_{\mathrm{V},k}$, $k = 1, \ldots, K$ fully observed, $\lambda_k \in [0, 1]$ and i.i.d.

$$
\begin{aligned}
& p(\boldsymbol{x}_k, \boldsymbol{y}_{\mathrm{A},k}, \boldsymbol{y}_{\mathrm{V},k}, \lambda_k) \propto p(\boldsymbol{y}_{\mathrm{A},k}|\boldsymbol{x}_k)^{\lambda_k} p(\boldsymbol{y}_{\mathrm{V},k}|\boldsymbol{x}_k)^{1-\lambda_k} \\
\Leftrightarrow \quad & \log\{p(\boldsymbol{x}_k, \boldsymbol{y}_{\mathrm{A},k}, \boldsymbol{y}_{\mathrm{V},k}, \lambda_k)\} = \lambda_k \log\{p(\boldsymbol{y}_{\mathrm{A},k}|\boldsymbol{x}_k)\} + (1-\lambda_k) \log\{p(\boldsymbol{y}_{\mathrm{V},k}|\boldsymbol{x}_k)\} + c
\end{aligned}
$$

Dynamic stream weights

Assumption: $\boldsymbol{x}_k$, $\boldsymbol{y}_{\mathrm{A},k}$, $\boldsymbol{y}_{\mathrm{V},k}$, $k = 1, \ldots, K$ fully observed, $\lambda_k \in [0, 1]$ and i.i.d.

$$\begin{aligned} & p(\boldsymbol{x}_k, \boldsymbol{y}_{\mathrm{A},k}, \boldsymbol{y}_{\mathrm{V},k}, \lambda_k) \propto p(\boldsymbol{y}_{\mathrm{A},k}|\boldsymbol{x}_k)^{\lambda_k} p(\boldsymbol{y}_{\mathrm{V},k}|\boldsymbol{x}_k)^{1-\lambda_k} \\ \Leftrightarrow \quad & \log\{p(\boldsymbol{x}_k, \boldsymbol{y}_{\mathrm{A},k}, \boldsymbol{y}_{\mathrm{V},k}, \lambda_k)\} = \lambda_k \log\{p(\boldsymbol{y}_{\mathrm{A},k}|\boldsymbol{x}_k)\} + (1 - \lambda_k) \log\{p(\boldsymbol{y}_{\mathrm{V},k}|\boldsymbol{x}_k)\} + c \end{aligned}$$

Problem: Direct optimization not feasible.

Dynamic stream weights

Assumption: $\boldsymbol{x}_k,\ \boldsymbol{y}_{\mathrm{A},k},\ \boldsymbol{y}_{\mathrm{V},k},\ k = 1, \ldots, K$ fully observed, $\lambda_k \in [0,\ 1]$ and i.i.d.

$$\begin{aligned} & p(\boldsymbol{x}_k, \boldsymbol{y}_{\mathrm{A},k}, \boldsymbol{y}_{\mathrm{V},k}, \lambda_k) \propto p(\boldsymbol{y}_{\mathrm{A},k}|\boldsymbol{x}_k)^{\lambda_k} p(\boldsymbol{y}_{\mathrm{V},k}|\boldsymbol{x}_k)^{1-\lambda_k} \\ \Leftrightarrow \quad & \log\{p(\boldsymbol{x}_k, \boldsymbol{y}_{\mathrm{A},k}, \boldsymbol{y}_{\mathrm{V},k}, \lambda_k)\} = \lambda_k \log\{p(\boldsymbol{y}_{\mathrm{A},k}|\boldsymbol{x}_k)\} + (1-\lambda_k)\log\{p(\boldsymbol{y}_{\mathrm{V},k}|\boldsymbol{x}_k)\} + c \end{aligned}$$

Problem: Direct optimization not feasible.

Solution: Impose prior on λ_k, e.g. Gaussian or symmetric Beta[3] distribution.

$$J(\lambda_k) = \lambda_k \log\{p(\boldsymbol{y}_{\mathrm{A},k}|\boldsymbol{x}_k)\} + (1-\lambda_k)\log\{p(\boldsymbol{y}_{\mathrm{V},k}|\boldsymbol{x}_k)\} + \log\{p(\lambda_k)\}$$

[3] C. Schymura et al.: *Audiovisual speaker tracking using nonlinear dynamical systems with dynamic stream weights*, arXiv, 2019

Symmetric Beta prior

$$J(\lambda_k) = \lambda_k \log\{p(\boldsymbol{y}_{\mathrm{A},k}|\boldsymbol{x}_k)\} + (1-\lambda_k)\log\{p(\boldsymbol{y}_{\mathrm{V},k}|\boldsymbol{x}_k)\} + \log\{p(\lambda_k)\}$$

with

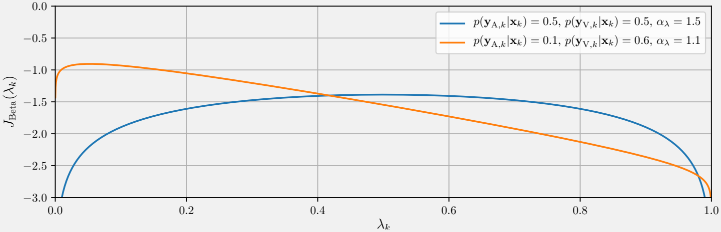
$$p(\lambda_k) = \frac{1}{\mathrm{B}(\alpha_\lambda,\, \alpha_\lambda)} \lambda_k^{\alpha_\lambda - 1}(1-\lambda_k)^{\alpha_\lambda - 1}$$

yields

$$\begin{aligned} J_{\mathrm{Beta}}(\lambda_k) = \lambda_k \log\{p(\boldsymbol{y}_{\mathrm{A},k}|\boldsymbol{x}_k)\} + (1-\lambda_k)\log\{p(\boldsymbol{y}_{\mathrm{V},k}|\boldsymbol{x}_k)\} \\ + (\alpha_\lambda - 1)\Big(\log\{\lambda_k\} + \log\{1-\lambda_k\}\Big) + \mathrm{const.} \end{aligned}$$

$$\Rightarrow \quad \lambda_k^\star = \max_{\lambda_k} J_{\mathrm{Beta}}(\lambda_k) \qquad \text{s. t.} \quad 0 < \lambda_k < 1$$

Symmetric Beta prior

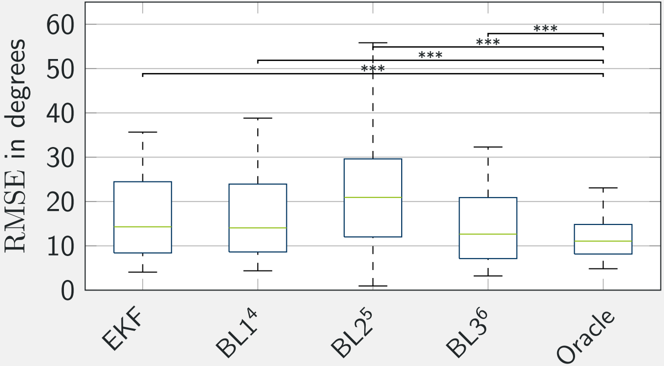


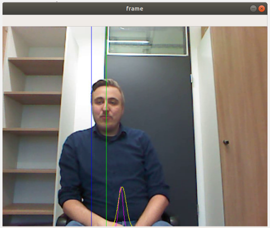
$J_{\text{Beta}}(\lambda_k)$ is a concave function:

$$\frac{dJ_{\text{Beta}}(\lambda_k)}{d\,\lambda_k} = \log\left\{\frac{p(\boldsymbol{y}_{\text{A},k}|\boldsymbol{x}_k)}{p(\boldsymbol{y}_{\text{V},k}|\boldsymbol{x}_k)}\right\} + (\alpha_\lambda - 1)\left(\frac{1}{\lambda_k} + \frac{1}{\lambda_k - 1}\right)$$

$$\frac{d^2 J_{\text{Beta}}(\lambda_k)}{d\,\lambda_k^2} = (\alpha_\lambda - 1)\left(\frac{1}{\lambda_k^2} + \frac{1}{(\lambda_k - 1)^2}\right) < 0 \;\forall\, \alpha_\lambda > 1$$

Results I





[$* * *$] $p < 0.001$

[4] T. Gehrig et al.: *Kalman filters for audio-video source localization*, WASPAA, 2005

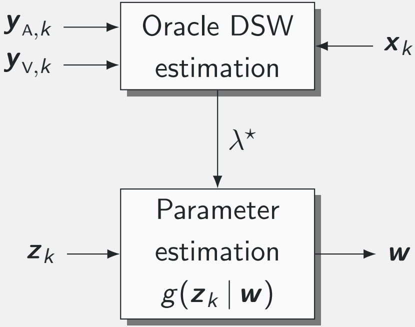
[5] S. Gerlach et al.: *2D audio-visual localization in home environments using a particle filter*, ITG Symp., 2012

[6] X. Qian et al.: *3D audio-visual speaker tracking with an adaptive particle filter*, ICASSP, 2017

Learning dynamic stream weights

Supervised learning approach

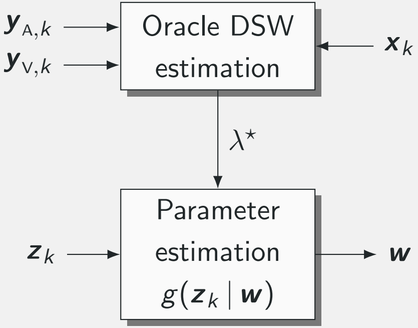
Oracle DSW serve as targets



Learning dynamic stream weights

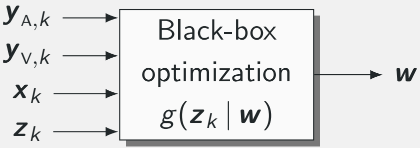
Supervised learning approach

Oracle DSW serve as targets



Evolutionary[7] learning approach

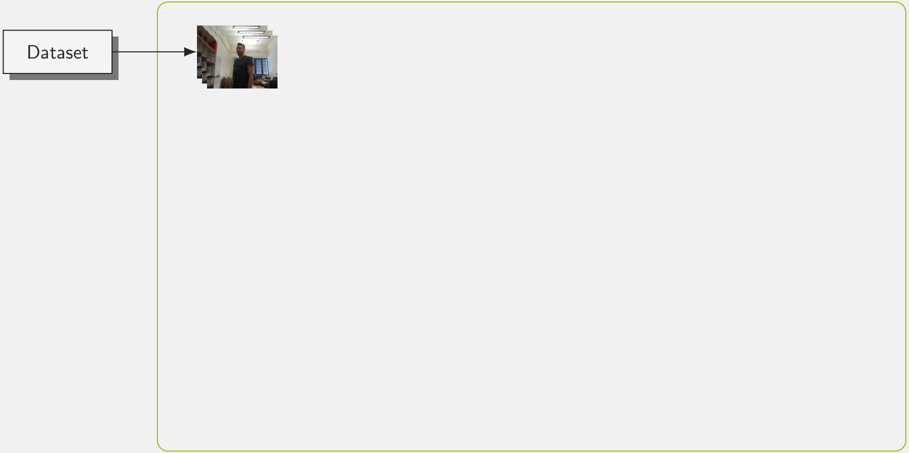
Direct optimization of localization error



[7] D. Wierstra et al.: *Natural evolution strategies*, Journal of machine learning research, vol. 15, 2014

Learning dynamic stream weights

Training procedure[8]

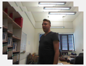


[8] C. Schymura et al.: *Learning dynamic stream weights for linear dynamical systems using natural evolution strategies*, ICASSP, 2019

Learning dynamic stream weights

Training procedure[8]

Dataset

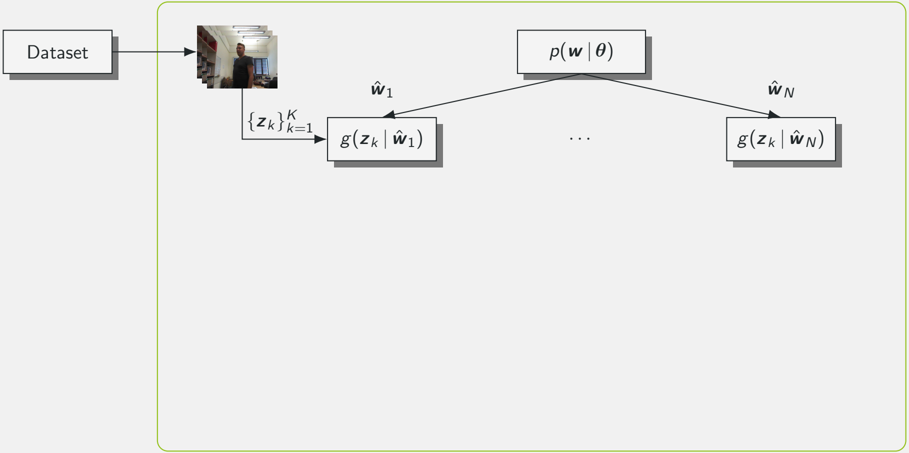


$p(\boldsymbol{w} \mid \boldsymbol{\theta})$

[8] C. Schymura et al.: *Learning dynamic stream weights for linear dynamical systems using natural evolution strategies*, ICASSP, 2019

Learning dynamic stream weights

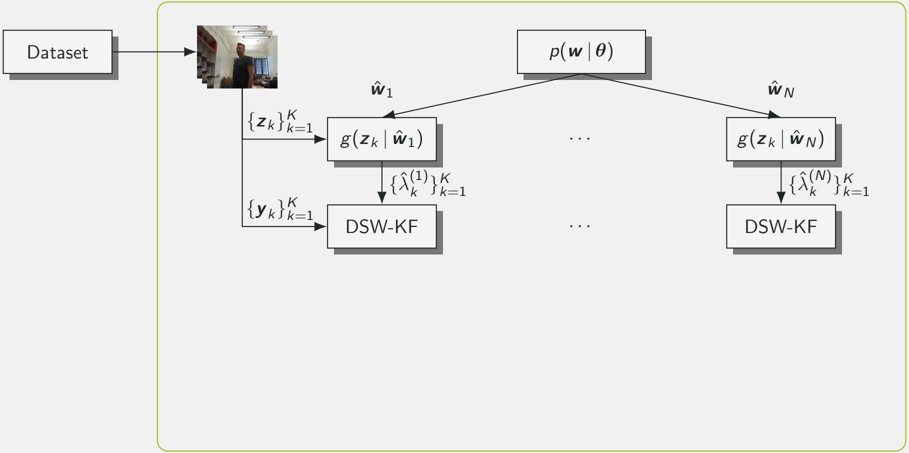
Training procedure[8]



[8] C. Schymura et al.: *Learning dynamic stream weights for linear dynamical systems using natural evolution strategies*, ICASSP, 2019

Learning dynamic stream weights

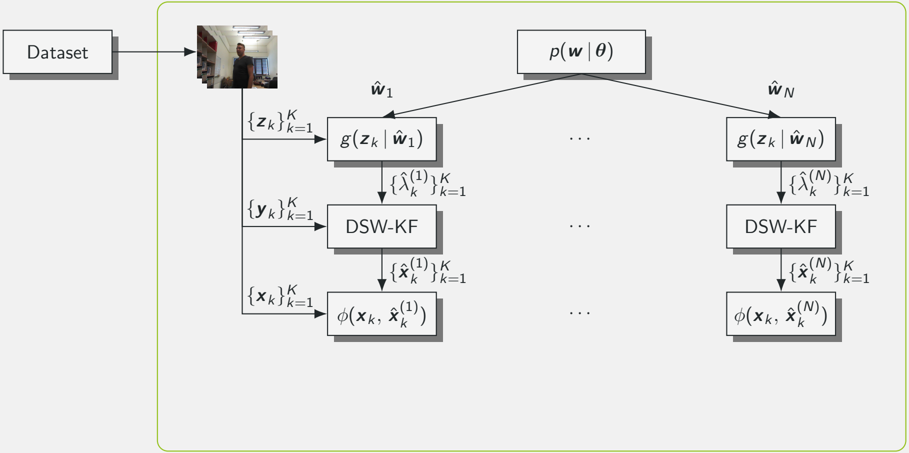
Training procedure[8]



[8] C. Schymura et al.: *Learning dynamic stream weights for linear dynamical systems using natural evolution strategies*, ICASSP, 2019

Learning dynamic stream weights

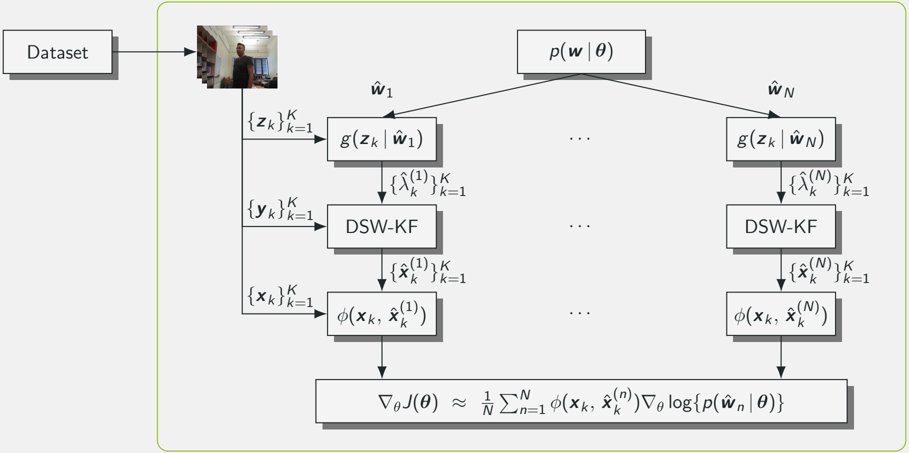
Training procedure[8]



[8] C. Schymura et al.: *Learning dynamic stream weights for linear dynamical systems using natural evolution strategies*, ICASSP, 2019

Learning dynamic stream weights

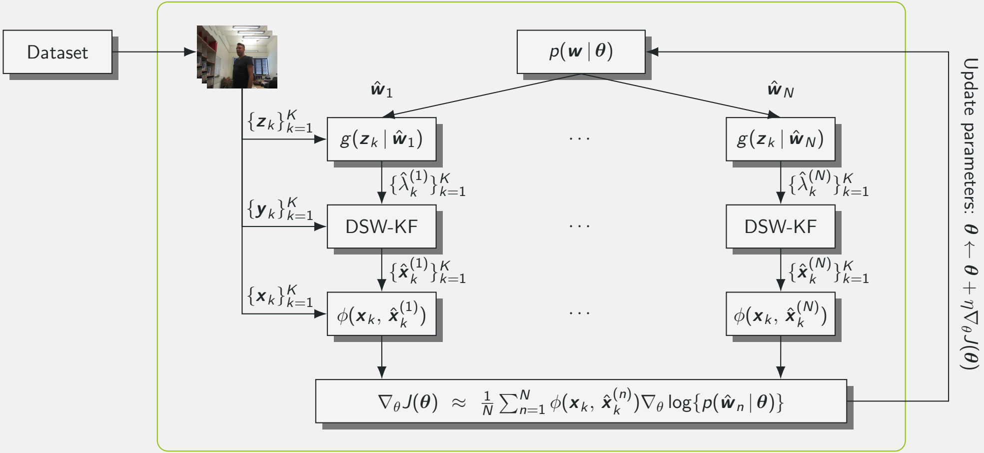
Training procedure[8]



[8] C. Schymura et al.: *Learning dynamic stream weights for linear dynamical systems using natural evolution strategies*, ICASSP, 2019

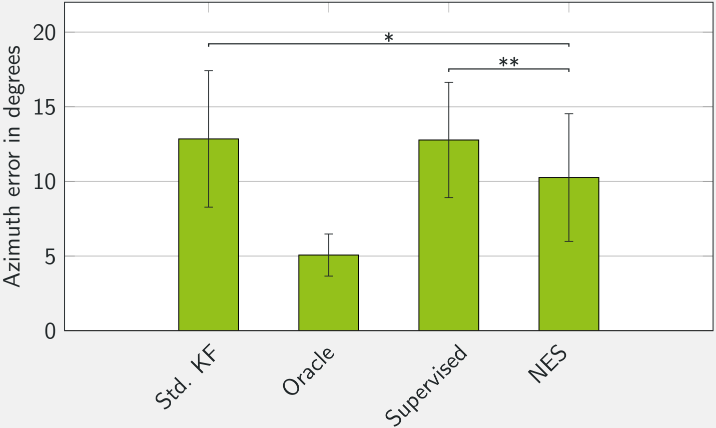
Learning dynamic stream weights

Training procedure[8]



[8] C. Schymura et al.: *Learning dynamic stream weights for linear dynamical systems using natural evolution strategies*, ICASSP, 2019

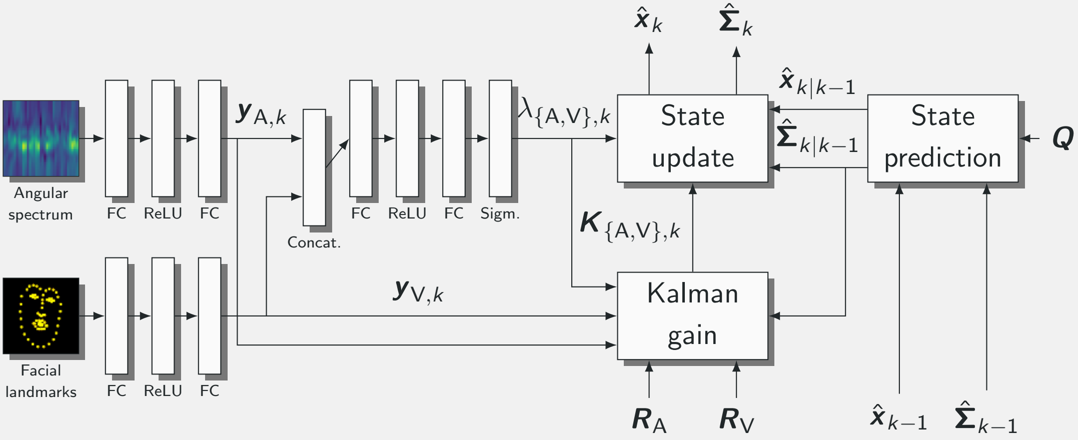
Results II



[*] $p < 0.05$
[**] $p < 0.01$

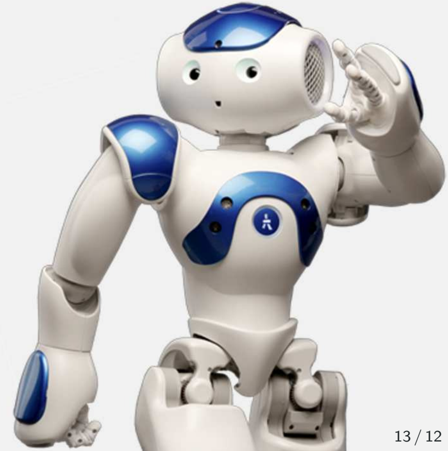
Ongoing work (in collaboration with NTT)

End-to-end optimization in a deep learning framework:



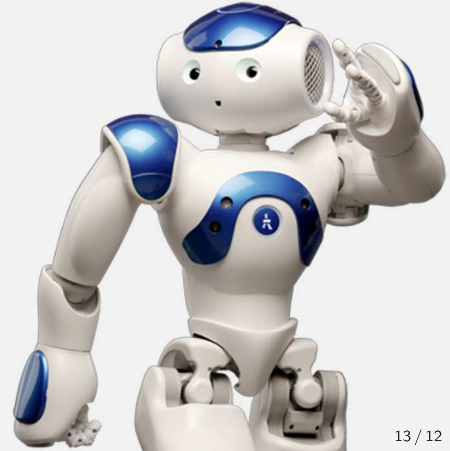
Conclusions

- Dynamic stream weights can benefit audiovisual speaker localization performance and provide an **additional notion of uncertainty**.



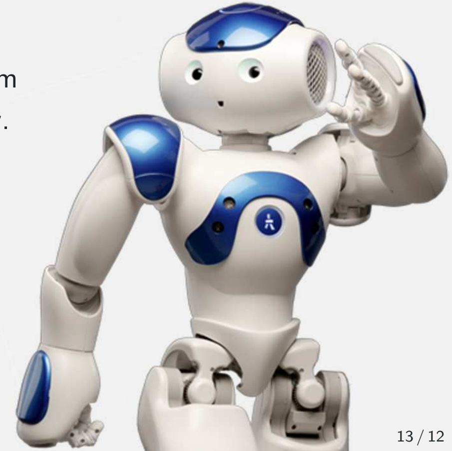
Conclusions

- Dynamic stream weights can benefit audiovisual speaker localization performance and provide an **additional notion of uncertainty**.
- Models for estimating dynamic stream weights can be trained effectively using **evolutionary optimization**.



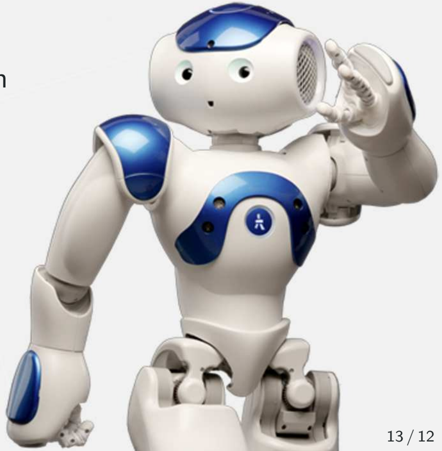
Conclusions

- Dynamic stream weights can benefit audiovisual speaker localization performance and provide an **additional notion of uncertainty**.
- Models for estimating dynamic stream weights can be trained effectively using **evolutionary optimization**.
- State error covariance matrices and dynamic stream weights jointly contribute to **model explainability**.



Conclusions

- Dynamic stream weights can benefit audiovisual speaker localization performance and provide an **additional notion of uncertainty**.
- Models for estimating dynamic stream weights can be trained effectively using **evolutionary optimization**.
- State error covariance matrices and dynamic stream weights jointly contribute to **model explainability**.
- Promising direction for future research: How to integrate modern deep learning techniques without sacrificing explainability?



Conclusions

- Dynamic stream weights can benefit audiovisual speaker localization performance and provide an **additional notion of uncertainty**.
- Models for estimating dynamic stream weights can be trained effectively using **evolutionary optimization**.
- State error covariance matrices and dynamic stream weights jointly contribute to **model explainability**.
- Promising direction for future research: How to integrate modern deep learning techniques without sacrificing explainability?

Thank you for your attention!