BooVAE: A scalable framework for continual VAE learning under boosting approach

Anna Kuzina & Evgenii Egorov, Evgenii Burnaev
ADASE, Skolkovo Institute of Science and Technology

Summary

- We propose algorithm to train VAE model with data-driven prior
- We propose simple and efficient algorithm for incremental learning which shares prior knowledge between tasks, keeping the single encoder-decoder pair.
- We empirically validate the proposed algorithm on commonly used benchmark datasets (MNIST, and Fashion-MNIST) for both offline and incremental setting.

Objectives

- Use data-driven prior to train VAE
- Construct feasible approximation for the optimal prior, avoiding ovefitting
- Reduce catastrophic forgetting in incremental learning setting, using data-driven prior

Optimal Prior

$$\log p(x) \geq \mathcal{L}(x;\theta;q) = \mathbb{E}_{z\sim q(z)}[\log p_\theta(x|z)] - D_{\mathrm{KL}}[q(z)\|p(z)],$$

Optimal prior in terms of Empirical Bayes:

$$p^*(z) = \arg\max_{p(z)} \mathcal{L} = \frac{1}{N}\sum_{n=1}^{N} q_\phi(z|x_n).$$

Boosting for density estimation

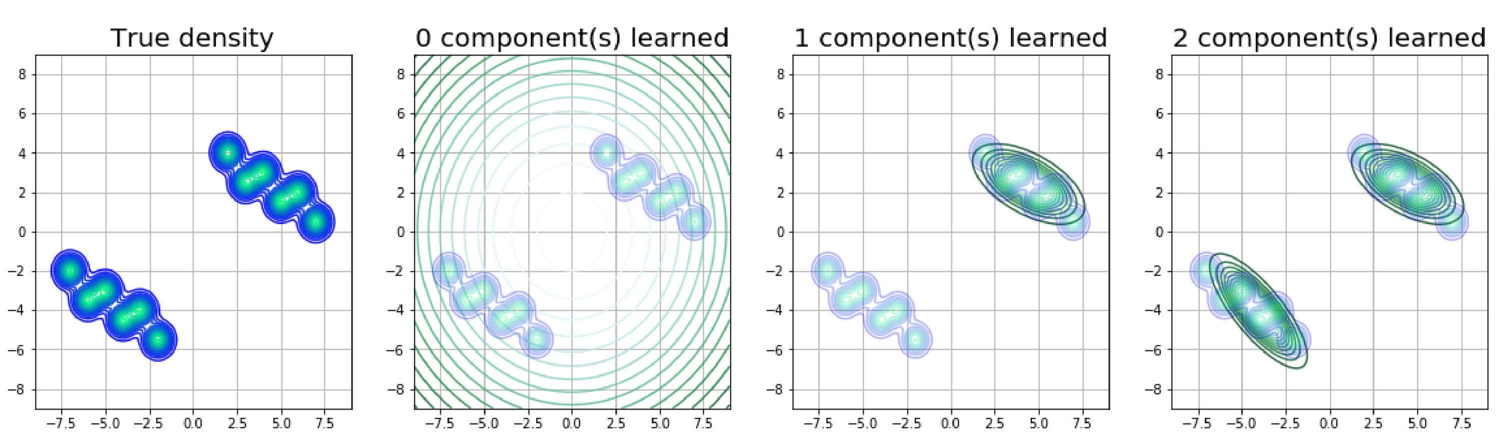
Approximates complex distribution by the simple mixture

$$p^* \approx \sum_{i=1}^{K} \alpha_i p^{(i)} = p_K$$

New component h is learned greedily, using MaxEntropy approach

$$\max_{h\in Q} \mathcal{H}(h)$$ + linearization

$$D_{\mathrm{KL}}(p_{t-1}|p^*) - D_{\mathrm{KL}}(p_t|p^*) > 0$$



BooVAE

Input: Dataset : $\{(x_i)\}_{i=1}^N$
Input: λ, Maximal number of components K
Choose random subset $\mathcal{M} \subset \mathcal{D}$
Initialize prior $p_0 = \mathcal{N}(\mu_0, \Sigma_0)$
$\theta^*, \phi^*, \mu_0, \Sigma_0 = \mathcal{L}(p_0, \theta, \phi)$
k = 1
while not converged **do**
  Update network parameters $\theta^*, \phi^* = \arg\max \mathcal{L}(p_{k-1}, \theta, \phi)$
  if $k < K$ **then**
    Update optimal prior $p^*(z) = \frac{1}{n}\sum_{x\in\mathcal{M}} q_{\phi^*}(z|x)$
    Add new component $p_k = \alpha h + (1-\alpha)p_{k-1}$
      $h = \arg\min D_{\mathrm{KL}}\left(h \| \left[\frac{p^*}{p_{k-1}}\right]^\lambda\right)$
      $\alpha = \arg\min D_{\mathrm{KL}}\left(\alpha h + (1-\alpha)p_{k-1} \| p^*\right)$
    $k = k + 1$
  end if
end while
return p_K, θ^*, ϕ^*

Results

# comp.	MNIST Vamp	MNIST Boo	Fashion MNIST Vamp	Fashion MNIST Boo
10	90.39	**89.98**	232.53	**231.94**
20	89.97	**89.78**	232.22	**231.84**
50	89.40	**89.16**	232.19	**231.63**
100	89.16	**88.90**	232.01	**231.55**
500	88.82	**88.68**	**231.67**	231.85

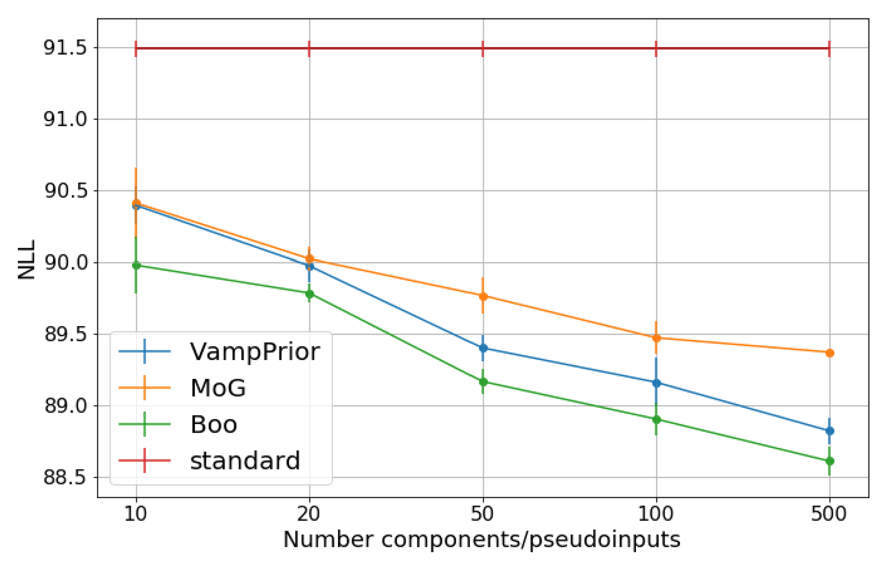
Table: NLL, Offline setting

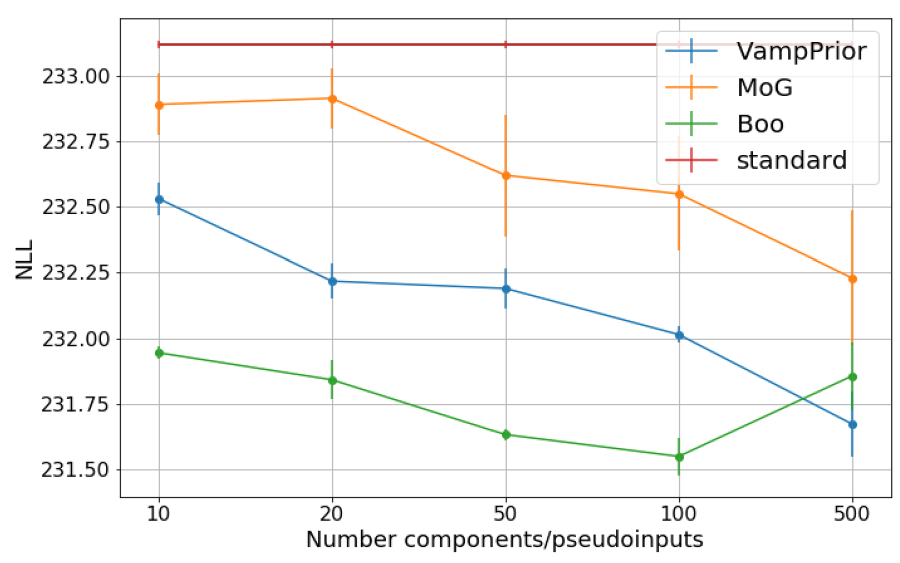
# Tasks	MNIST EWC	MNIST Boo	Fashion MNIST EWC	Fashion MNIST Boo
2	256.55	**100.11**	271.14	**227.83**
5	192.84	**132.08**	270.44	**253.12**
8	189.06	**140.80**	565.81	**260.05**
10	170.26	**142.92**	427.83	**284.86**

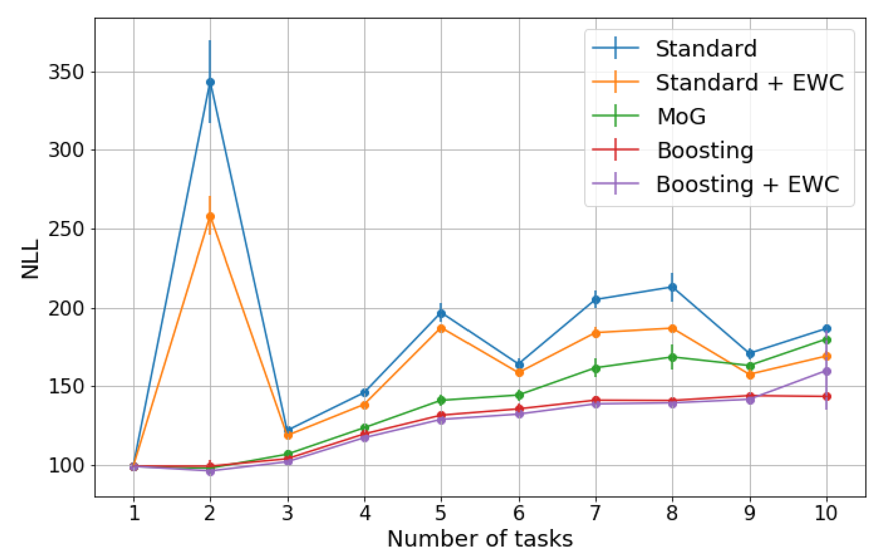
Table: NLL, Incremental setting

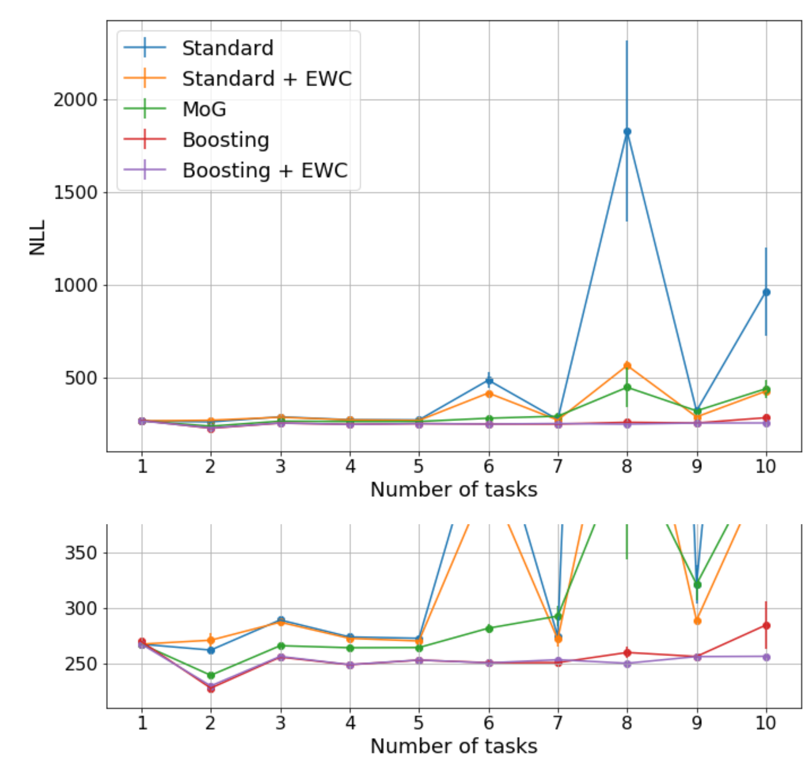
MNIST | Fashion MNIST

IWAE bound on NLL



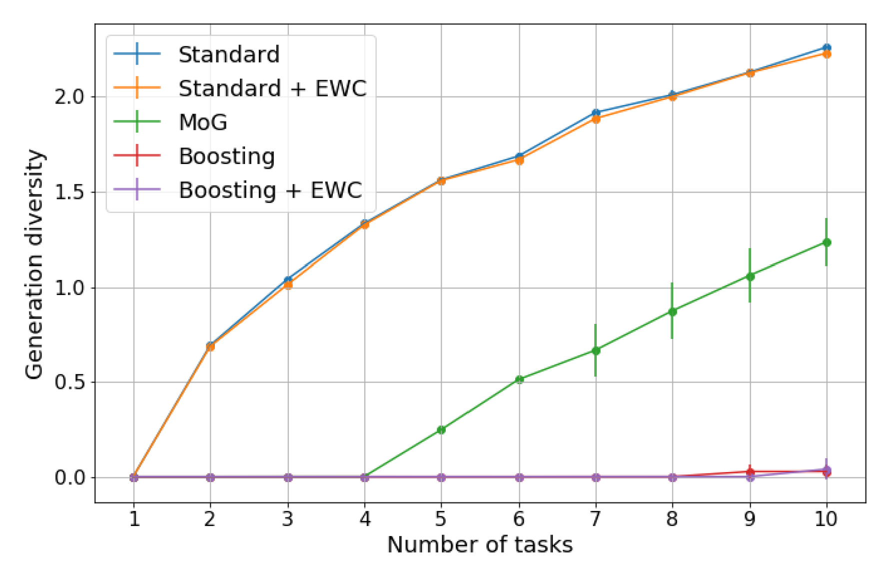


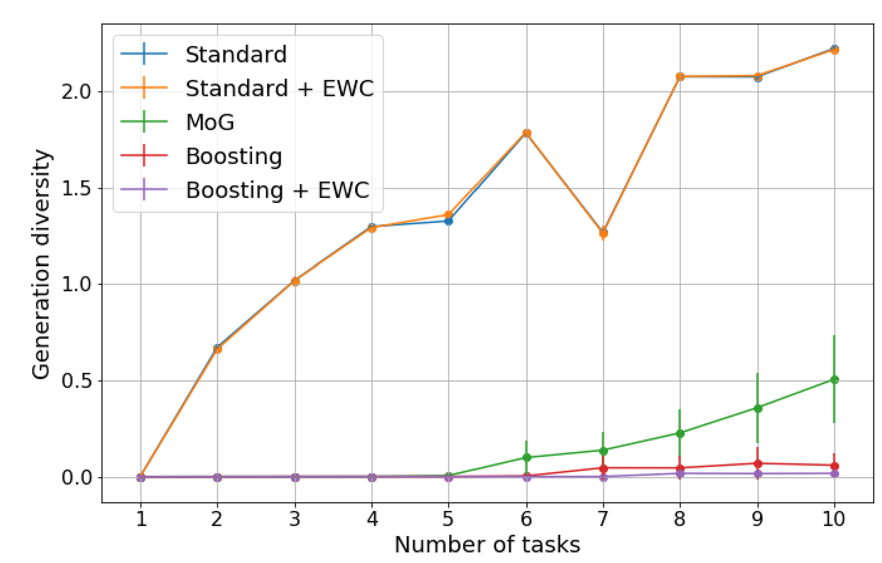




Generation diversity

$$\sum_k D_{\mathrm{KL}}\left(u \| \bar{x}_k\right), \; u \sim \mathrm{Be}\left(\frac{1}{K}\right), \; \bar{x}_k \sim \mathrm{Be}\left(\frac{N_k}{N}\right)$$





Generation after seeing 10 tasks (EWC and Boo)