Deep Sketched Output Kernel Regression for Structured Prediction

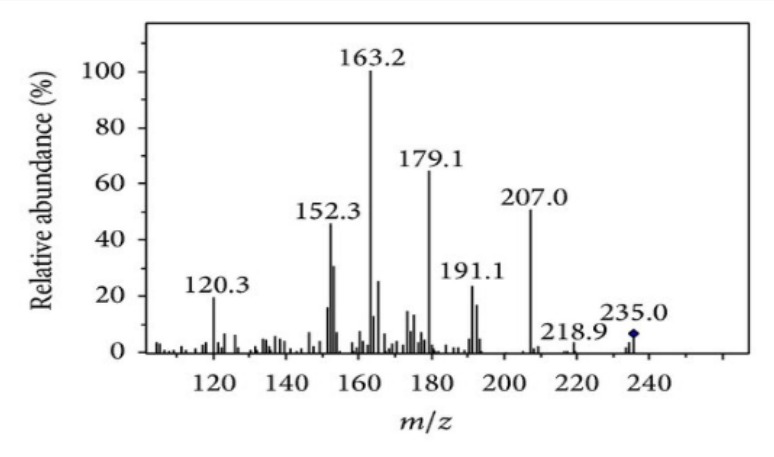
ECML PKDD 2024

Tamim El Ahmad*[1], Junjie Yang*[1], Pierre Laforgue[2], Florence d'Alché-Buc[1]
⋆ Equal contribution
1 LTCI, Télécom Paris, Institut Polytechnique de Paris
2 Università degli Studi di Milano

September 10, 2024

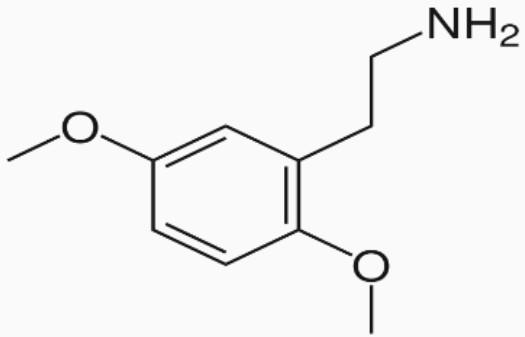
Structured prediction

Emblematic example of metabolite identification (Brouard et al., 2016a; Schymanski et al., 2017):



x

$f = ?$



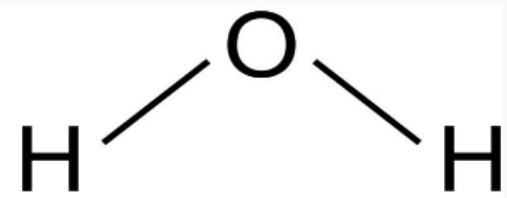
y

Structured Prediction with complex inputs

Goal of this work: solve structured prediction tasks with **complex inputs** such as texts

Water is an oxygen hydride consisting of an oxygen atom that is covalently bonded to two hydrogen atoms.

$$f = ?$$

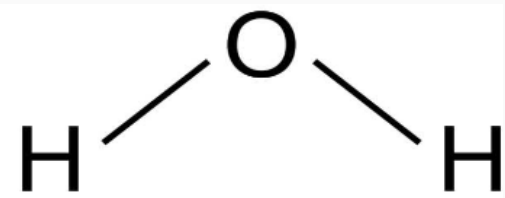


Structured Prediction with complex inputs

Goal of this work: solve structured prediction tasks with **complex inputs** such as texts

Water is an oxygen hydride consisting of an oxygen atom that is covalently bonded to two hydrogen atoms.

$$f = ? \longrightarrow$$



$\Longrightarrow$ need of **expressive** models such as **deep neural networks**

Motivation

Build a versatile and expressive estimator able to tackle a wide variety of structured prediction tasks and learn representations from complex inputs.

Table of contents

Output Kernel Regression

Structured prediction in supervised settings

Supervised settings: n i.i.d. training sample $(x_i, y_i)_{i=1}^n \in (\mathcal{X}, \mathcal{Y})^n \sim \rho$
Given a loss function $\Delta : \mathcal{Y}^2 \to \mathbb{R}$

$$f^* = \operatorname*{arg\,inf}_{f:\mathcal{X}\to\mathcal{Y}} \mathbb{E}_{(x,y)\sim\rho}[\Delta(f(x), y)] \approx \operatorname*{arg\,inf}_{f:\mathcal{X}\to\mathcal{Y}} \frac{1}{n}\sum_{i=1}^{n} \Delta(f(x_i), y_i) = \hat{f}$$

Structured prediction in supervised settings

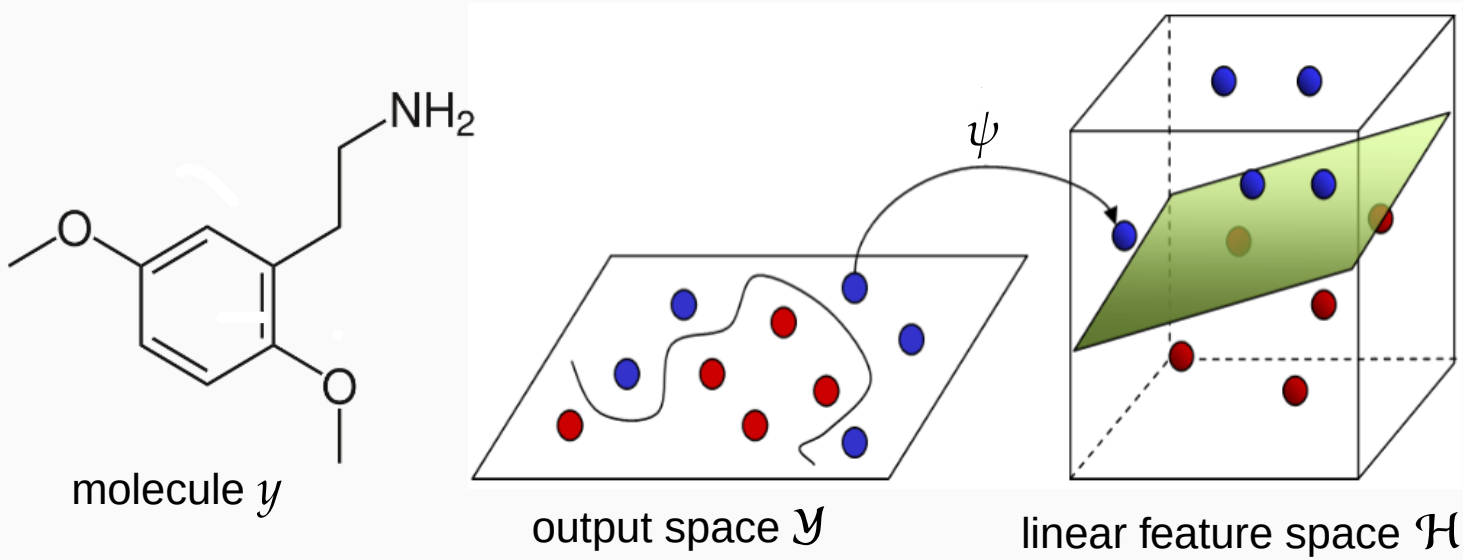
Supervised settings: n i.i.d. training sample $(x_i, y_i)_{i=1}^n \in (\mathcal{X}, \mathcal{Y})^n \sim \rho$
Given a loss function $\boldsymbol{\Delta} : \mathcal{Y}^2 \to \mathbb{R}$

$$f^* = \operatorname*{arg\,inf}_{f:\mathcal{X}\to\mathcal{Y}} \mathbb{E}_{(x,y)\sim\rho}[\boldsymbol{\Delta}(f(x), y)] \approx \operatorname*{arg\,inf}_{f:\mathcal{X}\to\mathcal{Y}} \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\Delta}(f(x_i), y_i) = \hat{f}$$

How to design a loss $\boldsymbol{\Delta}$ taking into account the structure of $\mathcal{Y}$?

Kernel methods: output representation

Linear method after embedding through feature map $\psi : \mathcal{Y} \to \mathcal{H}$:
choice of kernel $\iff$ choice of representation



$\langle \psi(y), \psi(y') \rangle_{\mathcal{H}} = k(y, y')$: relevant similarity measure over $\mathcal{Y}$

Output Kernel Regression for structured prediction

$$\implies \boldsymbol{\Delta}(y, y') = \|\boldsymbol{\psi}(y) - \boldsymbol{\psi}(y')\|^2_{\boldsymbol{\mathcal{H}}} = 2 - 2\boldsymbol{k}(y, y')$$

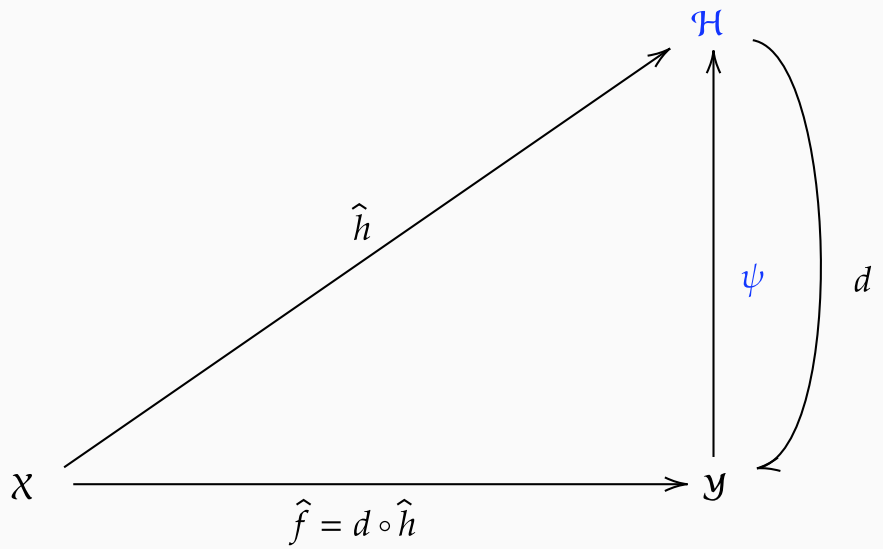
($\forall\, y \in \mathcal{Y}$, $\|\boldsymbol{\psi}\|_{\boldsymbol{\mathcal{H}}} = 1$ without loss of generality)

Versatility: tackle various tasks through an appropriate choice of $\boldsymbol{\psi}$ (e.g. SOTA performance on metabolite identification (Brouard et al., 2016a) and label ranking (Korba et al., 2018) datasets)

Output Kernel Regression: a surrogate approach

Surrogate (2-step) method (Weston et al., 2003; Cortes et al., 2005; Brouard et al., 2011; Kadri et al., 2013):

1. $\hat{h} = \underset{h:\mathcal{X}\to\boldsymbol{\mathcal{H}}}{\arg\min} \frac{1}{n}\sum_{i=1}^{n} \|h(x_i) - \boldsymbol{\psi}(y_i)\|^2_{\boldsymbol{\mathcal{H}}}$ **(training step)**
2. $\hat{f}(x) = d \circ \hat{h}(x) = \underset{y\in\mathcal{Y}}{\arg\min} \|\hat{h}(x) - \boldsymbol{\psi}(y)\|^2_{\boldsymbol{\mathcal{H}}}$ **(inference step)**



Output Kernel Regression: linear estimator

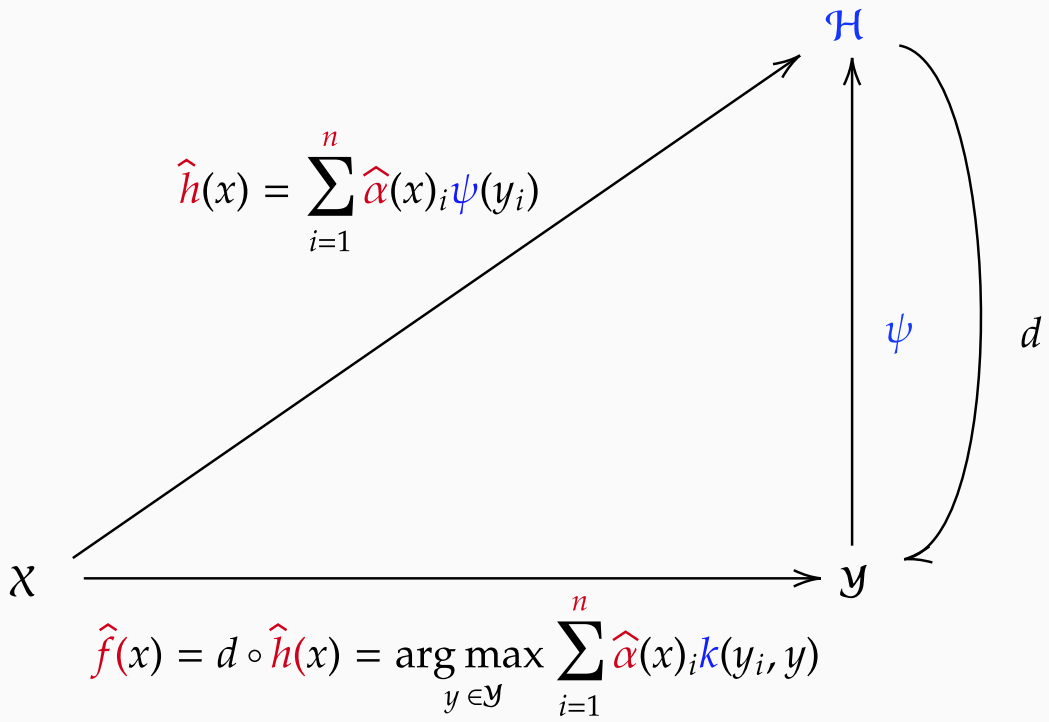
$$\hat{h} : x \mapsto \sum_{i=1}^{n} \hat{\boldsymbol{\alpha}}(x)_i \boldsymbol{\psi}(y_i)$$

where $\hat{\boldsymbol{\alpha}} : \mathcal{X} \to \mathbb{R}^n$ usually obtained by non-parametric methods (e.g. input kernel (**Input Output Kernel Regression**) (Brouard et al., 2016b), input tree (Geurts et al., 2006))

Output Kernel Regression: linear estimator

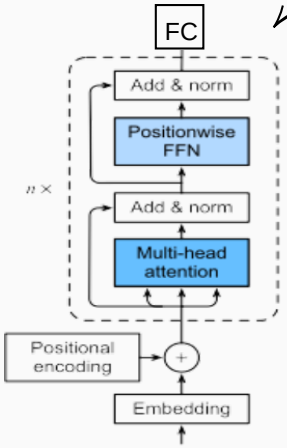
$$\hat{h} : x \mapsto \sum_{i=1}^{n} \hat{\boldsymbol{\alpha}}(x)_i \boldsymbol{\psi}(y_i)$$

where $\hat{\boldsymbol{\alpha}} : \mathcal{X} \to \mathbb{R}^n$ usually obtained by non-parametric methods (e.g. input kernel (**Input Output Kernel Regression**) (Brouard et al., 2016b), input tree (Geurts et al., 2006))

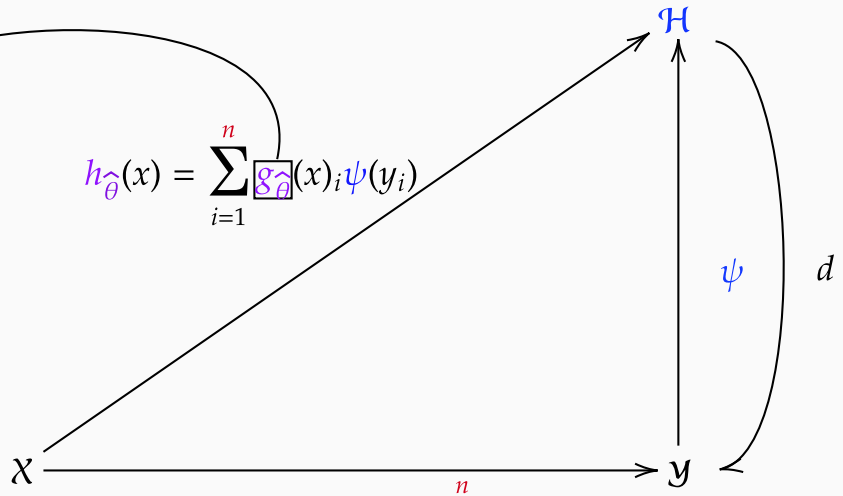


Deep Sketched Output Kernel Regression

Goal

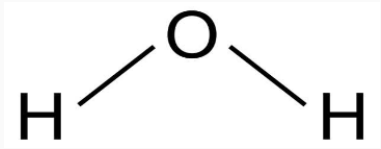


$$h_{\hat{\theta}}(x) = \sum_{i=1}^{n} g_{\hat{\theta}}(x)_i \psi(y_i)$$

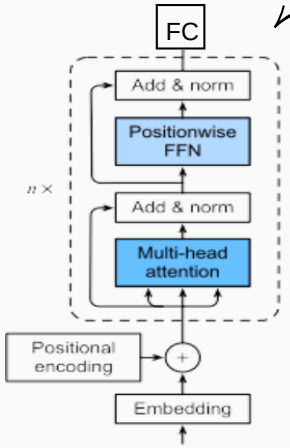


$$f_{\hat{\theta}}(x) = d \circ h_{\hat{\theta}}(x) = \arg\max_{y \in \mathcal{Y}} \sum_{i=1}^{n} g_{\hat{\theta}}(x)_i k(y_i, y)$$

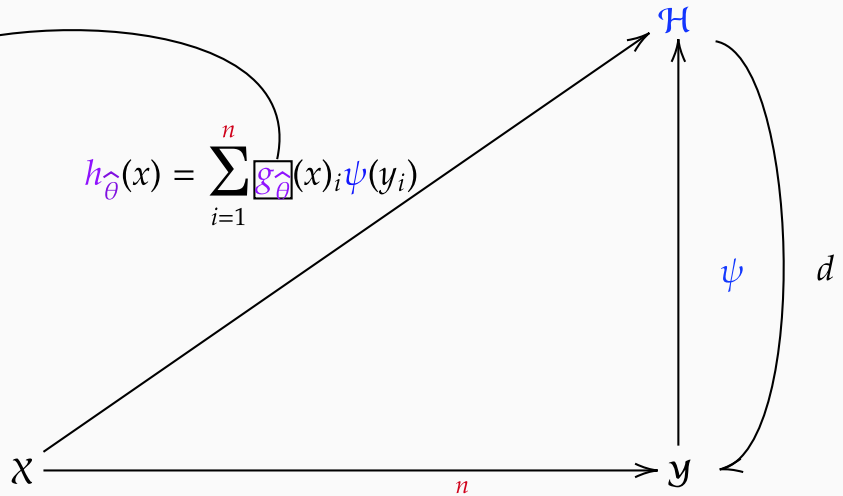
"Water is an oxygen hydride consisting of an oxygen atom that is covalently bonded to two hydrogen atoms."



Goal

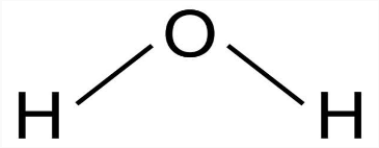


$$h_{\hat{\theta}}(x) = \sum_{i=1}^{n} g_{\hat{\theta}}(x)_i \psi(y_i)$$



$$f_{\hat{\theta}}(x) = d \circ h_{\hat{\theta}}(x) = \arg\max_{y \in \mathcal{Y}} \sum_{i=1}^{n} g_{\hat{\theta}}(x)_i k(y_i, y)$$

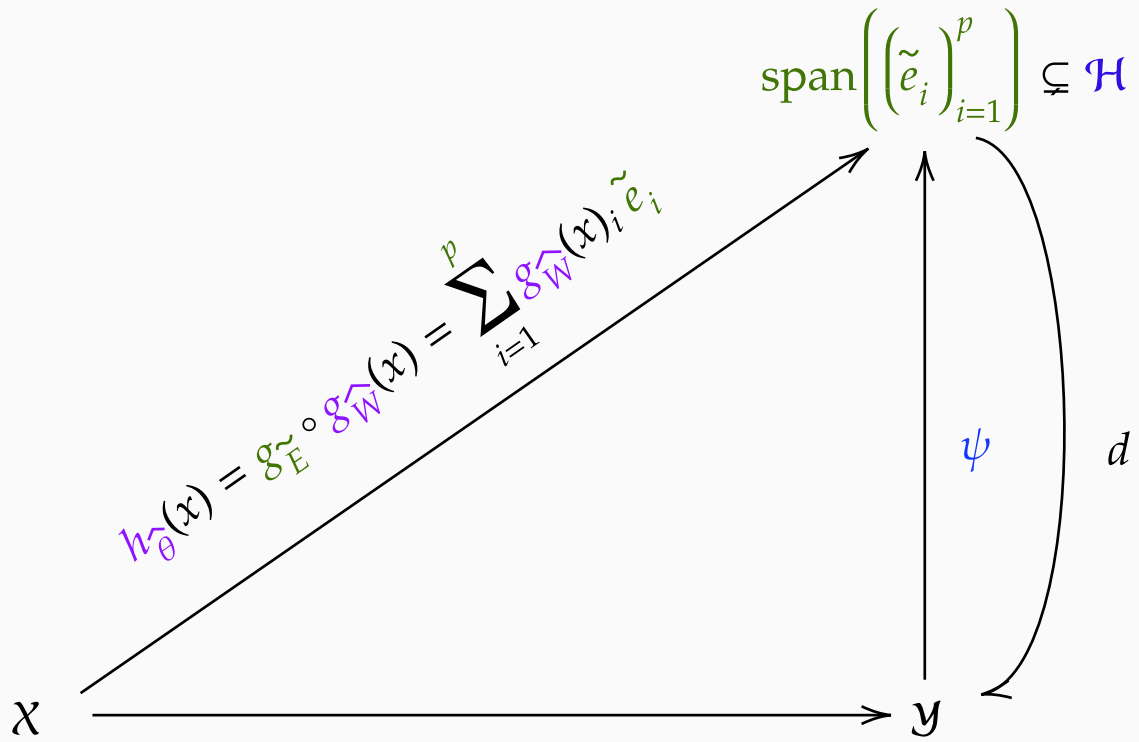
"Water is an oxygen hydride consisting of an oxygen atom that is covalently bonded to two hydrogen atoms."



Goal: reduce the size of the linear combination to unlock the use of deep neural networks within the Output Kernel Regression framework

DSOKR: a basis approach

$$h_\theta(x) := g_{\widetilde{E}} \circ g_W(x) = \sum_{j=1}^{p} g_W(x)_j \tilde{e}_j$$



$$f_{\widehat{\theta}}(x) = d \circ h_{\widehat{\theta}}(x) = \arg\max_{y \in \mathcal{Y}} \sum_{i=1}^{p} g_{\widehat{W}}(x)_i \langle \tilde{e}_i, \psi(y) \rangle_{\mathcal{H}}$$

How do we obtain this basis $\widetilde{E} = (\tilde{e}_1, \ldots, \tilde{e}_p)$?

Sketching: linear random projections

Let $m \ll n$, $R \in \mathbb{R}^{m \times n}$ sampled from a random distribution

Basic idea:

$$\underbrace{\widehat{\mathcal{H}} = \operatorname{span}\left((\psi(y_i))_{i=1}^{n}\right)}_{\dim = n} \leftarrow \underbrace{\operatorname{span}\left(\left(\sum_{j=1}^{n} [R]_{ij}\psi(y_j)\right)_{i=1}^{m}\right)}_{\dim = p \leq m} = \widetilde{\mathcal{H}}$$

Sketching: linear random projections

Let $m \ll n$, $R \in \mathbb{R}^{m \times n}$ sampled from a random distribution

Basic idea:

$$\underbrace{\widehat{\mathcal{H}} = \text{span}\left((\psi(y_i))_{i=1}^{n}\right)}_{\dim = n} \leftarrow \underbrace{\text{span}\left(\left(\sum_{j=1}^{n}[R]_{ij}\psi(y_j)\right)_{i=1}^{m}\right) = \widetilde{\mathcal{H}}}_{\dim = p \leq m}$$

Examples:

1. **Sub-sampling sketching (a.k.a. Nyström approximation):** rows of R sampled from I_n

   $$\implies \widetilde{\mathcal{H}} = \text{span}\left((\psi(\tilde{y}_i))_{i=1}^{m}\right) \quad \text{where} \quad \{(\tilde{y}_i)_{i=1}^{m}\} \subset \{(y_i)_{i=1}^{n}\}$$

2. **Gaussian sketching:** $R_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1/m)$

What is the orthonormal basis of $\widetilde{\mathcal{H}}$?

Construction of the orthonormal basis $\widetilde{E}$

- $\widehat{C} = (1/n) \sum_{i=1}^{n} \psi(y_i) \otimes \psi(y_i) \in \widehat{\mathcal{H}}^{\mathcal{H}}$
- $\widetilde{C} = \frac{1}{n} \sum_{l=1}^{m} \left(\sum_{i=1}^{n} R_{li} \psi(y_i) \right) \otimes \left(\sum_{j=1}^{n} R_{lj} \psi(z_j) \right) \in \widetilde{\mathcal{H}}^{\mathcal{H}}$
- $K = (k(y_i, y_j))_{1 \leq i,j \leq n} \in \mathbb{R}^{n \times n}$
- $\widetilde{K} = RKR^\top \in \mathbb{R}^{m \times m}$, and $\left\{ \left(\sigma_i(\widetilde{K}), \tilde{\mathbf{v}}_i \right), i \in [m] \right\}$ its eigenpairs
- $p = \text{rank}\left(\widetilde{K}\right)$, $\forall\, 1 \leq i \leq p$, $\tilde{\boldsymbol{e}}_i = \sqrt{\frac{n}{\sigma_i(\widetilde{K})}} \sum_{j=1}^{n} [R^\top \tilde{\mathbf{v}}_i]_j \psi(y_j) \in \mathcal{H}$

Construction of the orthonormal basis $\widetilde{E}$

- $\widehat{C} = (1/n)\sum_{i=1}^{n} \psi(y_i) \otimes \psi(y_i) \in \widehat{\mathcal{H}}^{\mathcal{H}}$
- $\widetilde{C} = \frac{1}{n}\sum_{l=1}^{m} \left(\sum_{i=1}^{n} R_{li}\psi(y_i)\right) \otimes \left(\sum_{j=1}^{n} R_{lj}\psi(z_j)\right) \in \widetilde{\mathcal{H}}^{\mathcal{H}}$
- $K = (k(y_i, y_j))_{1\leq i,j\leq n} \in \mathbb{R}^{n\times n}$
- $\widetilde{K} = RKR^\top \in \mathbb{R}^{m\times m}$, and $\left\{\left(\sigma_i(\widetilde{K}), \tilde{\mathbf{v}}_i\right), i \in [m]\right\}$ its eigenpairs
- $p = \text{rank}\left(\widetilde{K}\right)$, $\forall\, 1 \leq i \leq p$, $\tilde{e}_i = \sqrt{\frac{n}{\sigma_i(\widetilde{K})}} \sum_{j=1}^{n} [R^\top \tilde{\mathbf{v}}_i]_j \psi(y_j) \in \mathcal{H}$

Proposition (El Ahmad et al., 2024)

The $\tilde{e}_i$s are the **eigenfunctions**, associated to the eigenvalues $\sigma_i(\widetilde{K})/n$, of $\widetilde{C}$, whose range is $\widetilde{\mathcal{H}}$.
Then, $\widetilde{E} = (\tilde{e}_1, \ldots, \tilde{e}_p)$ is an **orthonormal basis** of $\widetilde{\mathcal{H}}$.

Related works on Nyström: Yang et al. (2012); Rudi et al. (2015)

Solving the surrogate problem

$$\min_{w \in \mathcal{W}} \frac{1}{n} \sum_{i=1}^{n} \| g_{\widetilde{E}} \circ g_w(x_i) - \psi(y_i) \|_{\mathcal{H}}^2$$

Solving the surrogate problem

$$\min_{w \in \mathcal{W}} \frac{1}{n} \sum_{i=1}^{n} \| g_{\widetilde{E}} \circ g_w(x_i) - \psi(y_i) \|_{\mathcal{H}}^2$$

$$\left\| g_{\widetilde{E}} \circ g_w(x) - \psi(y) \right\|_{\mathcal{H}}^2 = \left\| \sum_{i=1}^{p_Y} g_w(x)_j \tilde{e}_j^Y - \psi(y) \right\|_{\mathcal{H}}^2$$

$$= \left\| g_w(x) - \tilde{\psi}(y) \right\|_2^2 - \left(\left\| \tilde{\psi}(y) \right\|_2^2 + k(y, y) \right)$$

where

- $\tilde{\psi}(y) = \widetilde{D}_p^{-1/2} \widetilde{V}_p^\top R k^y \in \mathbb{R}^p$
- $\widetilde{V}_p \widetilde{D}_p \widetilde{V}_p^\top = \underbrace{\widetilde{K}}_{m \times m} = R K R^\top$ (SVD of $\widetilde{K}$)
- $k^y = (k(y, y_1), \ldots, k(y, y_n))$

Deep Sketched Output Kernel Regression: inference

$$f_{\hat{\theta}}(x) = \arg\max_{y \in \mathcal{Y}} \sum_{i=1}^{p} g_{\hat{W}}(x)_i \langle \tilde{e}_i^Y, \psi(y) \rangle_{\mathcal{H}} = \arg\max_{y \in \mathcal{Y}} g_{\hat{W}}(x)^\top \tilde{\psi}(y)$$

- **Test set:** $X^{te} = \{x_1^{\text{te}}, \ldots, x_{n_{\text{te}}}^{\text{te}}\}$ of size n_{te}
- **Candidate set:** $Y^{\text{c}} = \{y_1^{\text{c}}, \ldots, y_{n_{\text{c}}}^{\text{c}}\}$ of size n_{c}

$$f_{\hat{\theta}}(x_i^{\text{te}}) = y_j^{\text{c}} \quad \text{where} \quad j = \arg\max_{1 \leq j \leq n_{\text{c}}} g_{\hat{W}}(x_i^{\text{te}})^\top \tilde{\psi}(y_j^{\text{c}})$$

DSOKR: summary

1. Training. a. Computations for the basis $\widetilde{E}$.
   - SVD of $\widetilde{K} = RKR^\top \in \mathbb{R}^{m \times m} \to \left\{ \left(\sigma_i(\widetilde{K}), \tilde{\mathrm{v}}_i \right), i \in [m] \right\}$
   - $\widetilde{M} = \widetilde{D}_p{}^{-1/2} \widetilde{V}_p{}^\top \in \mathbb{R}^{p \times m}$, where $\widetilde{V}_p = (\tilde{\mathrm{v}}_1, \ldots, \tilde{\mathrm{v}}_p)$, $\widetilde{D}_p = \mathrm{diag}(\sigma_1(\widetilde{K}), \ldots, \sigma_p(\widetilde{K}))$

1. Training. b. Solving the surrogate problem.
   - $\{(x_i, y_i)\}_{i=1}^n \leftarrow \{(x_i, \tilde{\psi}(y_i))\}_{i=1}^n$, $\{(x_i^{\mathrm{val}}, y_i^{\mathrm{val}})\}_{i=1}^{n_{\mathrm{val}}} \leftarrow \{(x_i, \tilde{\psi}(y_i^{\mathrm{val}}))\}_{i=1}^{n_{\mathrm{val}}}$, where $\tilde{\psi}(y) = \widetilde{M} R k^y$
   - $g_{\hat{W}} = \underset{g_W, W \in \mathcal{W}}{\arg\min} \frac{1}{n} \sum_{i=1}^n \left\| g_{\hat{W}}(x_i) - \tilde{\psi}(y_i) \right\|_2^2$

2. Inference.
   - $\{y_i^{\mathrm{c}}\}_{i=1}^{n_{\mathrm{c}}} \leftarrow \{\tilde{\psi}(y_i^{\mathrm{c}})\}_{i=1}^{n_{\mathrm{c}}}$
   - $f_{\hat{\theta}}(x_i^{\mathrm{te}}) = y_j^{\mathrm{c}}$ where $j = \underset{1 \le j \le n_{\mathrm{c}}}{\arg\max}\, g_{\hat{W}}(x_i^{\mathrm{te}})^\top \tilde{\psi}(y_j^{\mathrm{c}})$

Experiments

Synthetic least squares regression

1) $n = 50\,000$, $\mathcal{X} = \mathbb{R}^{2\,000}$, $\mathcal{Y} = \mathbb{R}^{1\,000}$, k linear kernel $\implies$
$\mathcal{H} = \mathcal{Y} = \mathbb{R}^{1\,000}$

Goal: build this dataset such that the outputs lie in a subspace of $\mathcal{Y}$ of dimension $d = 50 < 1000$

Synthetic least squares regression

1) $n = 50\,000$, $\mathcal{X} = \mathbb{R}^{2\,000}$, $\mathcal{Y} = \mathbb{R}^{1\,000}$, k linear kernel $\implies$
$\mathcal{H} = \mathcal{Y} = \mathbb{R}^{1\,000}$

Goal: build this dataset such that the outputs lie in a subspace of $\mathcal{Y}$ of dimension $d = 50 < 1000$

2) Draw $H = (H_{ij})_{1 \le i \le d, 1 \le j \le 2\,000} \in \mathbb{R}^{d \times 2\,000}$ s.t. $H_{ij} \sim \mathcal{N}(0, 1)$, $x_i \sim \mathcal{N}(0, C_{\mathcal{X}})$, where $(\sigma_j(C_{\mathcal{X}}) = j^{-1/2})_{j=1}^{2\,000}$, $\varepsilon_i \sim \mathcal{N}(0, \sigma^2 I_{1\,000})$ with $\sigma^2 = 0.01$,

$$y_i = UHx_i + \varepsilon_i \,,$$

where $U = (u_1, \ldots, u_d) \in \mathbb{R}^{1\,000 \times d}$ and $(u_j)_{j=1}^d$ are d randomly drawn orthonormal vectors

Synthetic least squares regression: results

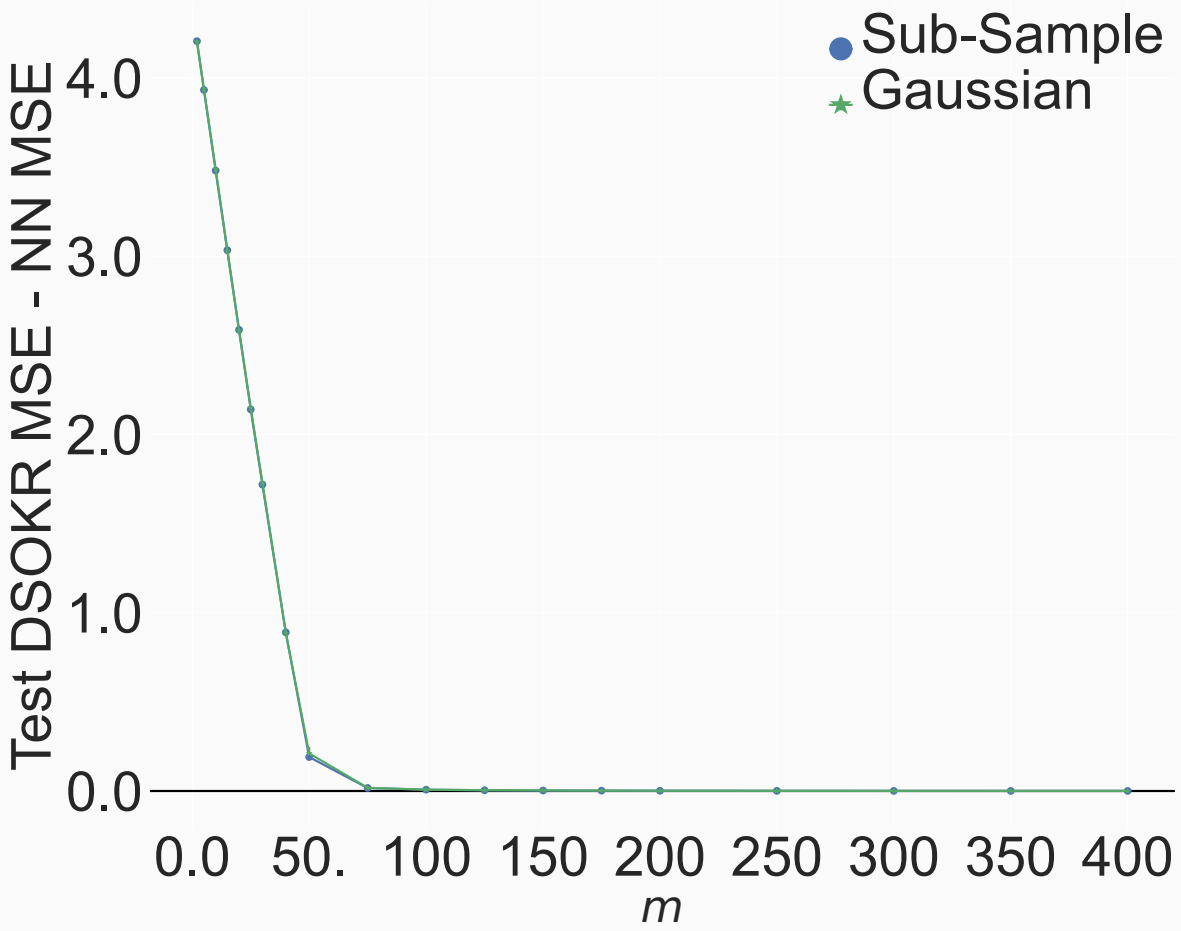


Figure 1: Difference between test MSE of DSOKR and NN w.r.t. m.

Text to molecule

ChEBI-20 dataset (Edwards et al., 2021)

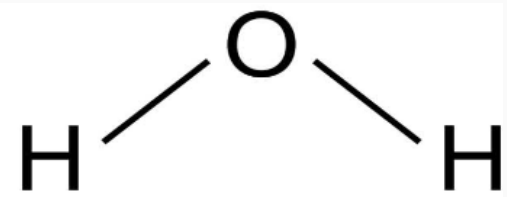
$n = 26\,408$, $n_{\text{te}} = 3\,301$, $n_{\text{c}} = 33\,010$

Inputs: texts (mean/median number of words per description is 55/51)

Outputs: molecules as graphs (mean/median number of atoms per molecule is 32/25)

Water is an oxygen hydride consisting of an oxygen atom that is covalently bonded to two hydrogen atoms.

$$f = ?$$



Text to molecule

ChEBI-20 dataset (Edwards et al., 2021)

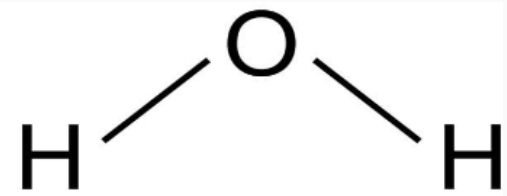
$n = 26\,408$, $n_{\text{te}} = 3\,301$, $n_{\text{c}} = 33\,010$

Inputs: texts (mean/median number of words per description is 55/51)

Outputs: molecules as graphs (mean/median number of atoms per molecule is 32/25)

Water is an oxygen hydride consisting of an oxygen atom that is covalently bonded to two hydrogen atoms.

$f = ?$ →



Input neural network: SciBERT (transformer) (Beltagy et al., 2019)

Output kernel: cosine applied to Mol2vec (Jaeger et al., 2018) (for normalization)

Sketching: Sub-Sample and Gaussian, $m = 100$

Text to molecule: results

	Hits@1 ↑	Hits@10 ↑	MRR ↑
SISOKR	0.4%	2.8%	0.015
SciBERT Regression	16.8%	56.9%	0.298
CMAM - MLP	34.9%	84.2%	0.513
CMAM - GCN	33.2%	82.5%	0.495
CMAM - Ensemble (MLP×3)	39.8%	87.6%	0.562
CMAM - Ensemble (GCN×3)	39.0%	87.0%	0.551
CMAM - Ensemble (MLP×3 + GCN×3)	44.2%	**88.7**%	0.597
DSOKR - SubSample Sketch	48.2%	87.4%	0.624
DSOKR - Gaussian Sketch	49.0%	87.5%	0.630
DSOKR - Ensemble (SubSample×3)	**51.0**%	88.2%	**0.642**
DSOKR - Ensemble (Gaussian×3)	50.5%	87.9%	**0.642**
DSOKR - Ensemble (SubSample×3 + Gaussian×3)	50.0%	88.3%	0.640

Conclusion

Conclusion

- DSOKR: sketching on the output kernel to unlock the use of Deep Neural Networks within OKR framework
- Basis obtained via a sketch-based Kernel PCA
- Any DNN architecture can be considered and its layers will always be fully connected regardless of the output data at hand
- Experiments: DSOKR outperforms SOTA method on a text-to-molecule dataset
- Code publicly available at https://github.com/tamim-el/dsokr

Perspectives

- **Excess risk bound for DSOKR**:
  - ▷ theory of OKR with sketching and input kernel (El Ahmad et al., 2024)
  - ▷ excess risk of MLP with ReLU activations (Schmidt-Hieber, 2017)

- **End-to-end version of DSOKR**:
  - ▷ direct risk minimization (Belanger et al., 2017) together with differentiable approximation (Berthet et al., 2020; Niculae and Martins, 2020) technique
  - ▷ inference neural network (decoder) (Tu and Gimpel, 2018)

- **DSOKR for unsupervised learning**:
  - ▷ basis approach on both first and last layers
  - ▷ auto-encoder for **structured objects** (Laforgue et al., 2019)

References

Belanger, D., Yang, B., and McCallum, A. (2017). End-to-end learning for structured prediction energy networks. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 429–439. PMLR.

Beltagy, I., Lo, K., and Cohan, A. (2019). SciBERT: A Pretrained Language Model for Scientific Text. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

References ii

Berthet, Q., Blondel, M., Teboul, O., Cuturi, M., Vert, J.-P., and Bach, F. (2020). Learning with differentiable perturbed optimizers.

Brouard, C., d'Alché-Buc, F., and Szafranski, M. (2011). Semi-supervised penalized output kernel regression for link prediction. In *International Conference on Machine Learning (ICML)*, pages 593–600.

Brouard, C., Shen, H., Dührkop, K., d'Alché-Buc, F., Böcker, S., and Rousu, J. (2016a). Fast metabolite identification with input output kernel regression. *Bioinformatics*, 32(12):28–36.

Brouard, C., Szafranski, M., and D'Alché-Buc, F. (2016b). Input output kernel regression: supervised and semi-supervised structured output prediction with operator-valued kernels. *The Journal of Machine Learning Research*, 17(1):6105–6152.

References iii

Cortes, C., Mohri, M., and Weston, J. (2005). A general regression technique for learning transductions. In *International Conference on Machine Learning (ICML)*, pages 153–160.

Edwards, C., Zhai, C., and Ji, H. (2021). Text2Mol: Cross-Modal Molecule Retrieval with Natural Language Queries. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 595–607, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

El Ahmad, T., Brogat-Motte, L., Laforgue, P., and d'Alché Buc, F. (2024). Sketch in, sketch out: Accelerating both learning and inference for structured prediction with kernels. In Dasgupta, S., Mandt, S., and Li, Y., editors, *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 109–117. PMLR.

References iv

Geurts, P., Wehenkel, L., and d'Alché Buc, F. (2006). Kernelizing the output of tree-based methods. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 345–352, New York, NY, USA. Association for Computing Machinery.

Jaeger, S., Fulle, S., and Turk, S. (2018). Mol2vec: Unsupervised Machine Learning Approach with Chemical Intuition. *Journal of Chemical Information and Modeling*, 58(1):27–35.

Kadri, H., Ghavamzadeh, M., and Preux, P. (2013). A generalized kernel approach to structured output learning. In Dasgupta, S. and McAllester, D., editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 471–479, Atlanta, Georgia, USA. PMLR.

References v

Korba, A., Garcia, A., and d'Alché-Buc, F. (2018). A structured prediction approach for label ranking. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Laforgue, P., Clémençon, S., and d'Alché-Buc, F. (2019). Autoencoding any data through kernel autoencoders. In Chaudhuri, K. and Sugiyama, M., editors, *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, volume 89 of *Proceedings of Machine Learning Research*, pages 1061–1069. PMLR.

References vi

Niculae, V. and Martins, A. (2020). LP-SparseMAP: Differentiable relaxed optimization for sparse structured prediction. In III, H. D. and Singh, A., editors, *Proceedings of International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7348–7359.

Nikolentzos, G., Meladianos, P., Limnios, S., and Vazirgiannis, M. (2018). A Degeneracy Framework for Graph Similarity. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 2595–2601. International Joint Conferences on Artificial Intelligence Organization.

Ramakrishnan, R., Dral, P. O., Rupp, M., and von Lilienfeld, O. A. (2014). Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1. Publisher: Nature Publishing Group.

Ruddigkeit, L., van Deursen, R., Blum, L. C., and Reymond, J.-L. (2012). Enumeration of 166 Billion Organic Small Molecules in the Chemical Universe Database GDB-17. *Journal of Chemical Information and Modeling*, 52(11):2864–2875. Publisher: American Chemical Society.

Rudi, A., Camoriano, R., and Rosasco, L. (2015). Less is more: Nyström computational regularization. *Advances in Neural Information Processing Systems*, 28.

Schmidt-Hieber, J. (2017). Nonparametric regression using deep neural networks with relu activation function. *Annals of Statistics*, 48.

Schymanski, E., Ruttkies, C., and Krauss, M. e. a. (2017). Critical assessment of small molecule identification 2016: automated methods. *J Cheminform*, 9:22.

References viii

Tu, L. and Gimpel, K. (2018). Learning approximate inference networks for structured prediction. In *International Conference on Learning Representations*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is All you Need. In *Advances in Neural Information Processing Systems (NIPS)*, pages 5998–6008.

Weston, J., Chapelle, O., Vapnik, V., Elisseeff, A., and Schölkopf, B. (2003). Kernel dependency estimation. In Becker, S., Thrun, S., and Obermayer, K., editors, *Advances in Neural Information Processing Systems 15*, pages 897–904. MIT Press.

Yang, T., Li, Y.-f., Mahdavi, M., Jin, R., and Zhou, Z.-H. (2012). Nyström method vs random fourier features: A theoretical and empirical comparison. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

DSOKR Inference: Ensemble Approach

Let $T > 1$, and for $1 \leq t \leq T$, let R_t be a randomly drawn sketching matrix, $h_{\hat{\theta}_t} = g_{\tilde{E}_t} \circ g_{\hat{W}_t}$ denotes the trained DSOKR neural network based on R_t

$$f_{\hat{\theta}}^{\text{mean}}(x) = \underset{y \in \mathcal{Y}_c}{\arg\max} \sum_{t=1}^{T} \omega_t \, g_{\hat{W}_t}(x)^\top \tilde{\psi}_t(y) \quad \text{with} \quad \sum_{t=1}^{T} \omega_t = 1$$

or

$$f_{\hat{\theta}}^{\max}(x) = \underset{y \in \mathcal{Y}_c}{\arg\max} \, \underset{1 \leq t \leq T}{\arg\max} \, g_{\hat{W}_t}(x)^\top \tilde{\psi}_t(y)$$

Sketching size selection strategy

Goal: set the minimal value of m s.t. it captures the information contained in the empirical covariance operator $\widehat{C} = \frac{1}{n} \sum_{i=1}^{n} \psi(y_i) \otimes \psi(y_i)$

However: computing the SVD of $\widehat{C}$ is costing, i.e. $\mathcal{O}(n^3)$ in time.

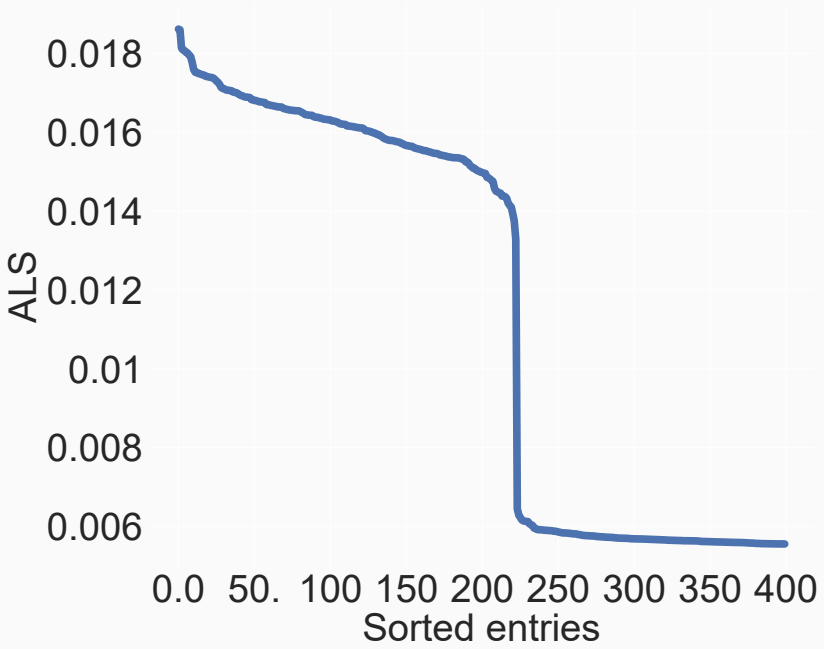
1. Approximate leverage scores of $\widehat{C}$

2. Set the optimal m according to the performance of the *perfect h* estimator on the validation set, i.e.

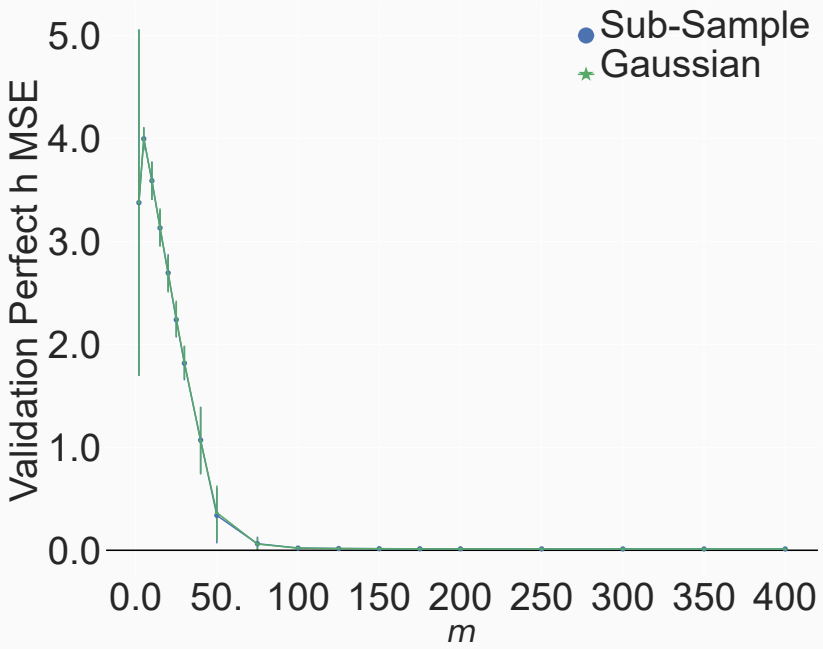
$$h : (x, y) \mapsto \sum_{j=1}^{p} \langle \tilde{e}_j, \psi(y) \rangle_{\mathcal{H}} \, \tilde{e}_j = \sum_{j=1}^{p} \tilde{\psi}(y)_j \, \tilde{e}_j \, . \tag{1}$$

$\Longrightarrow$ **allows to cope with the neural net training phase!**

Synthetic least squares regression: sketching size selection



(a) Sorted 400 highest ALS.



(b) Validation MSE of *Perfect h* w.r.t. *m*.

Smiles to molecule

QM9 molecule dataset (Ruddigkeit et al., 2012; Ramakrishnan et al., 2014)

$n = n_c = 131\,382$, $n_{te} = 2\,000$

Inputs: strings (smiles)

Outputs: graphs (molecules)

O=C1CC2NC2CC1O $\xrightarrow{f = ?}$ 

Input neural network: transformer (Vaswani et al., 2017)

Output kernel: core Weisfeiler-Lehman subtree kernel (CORE-WL) (Nikolentzos et al., 2018)

Input/output sketching: Sub-sample, $m = 3\,200$

Smiles to molecule: Perfect h strategy

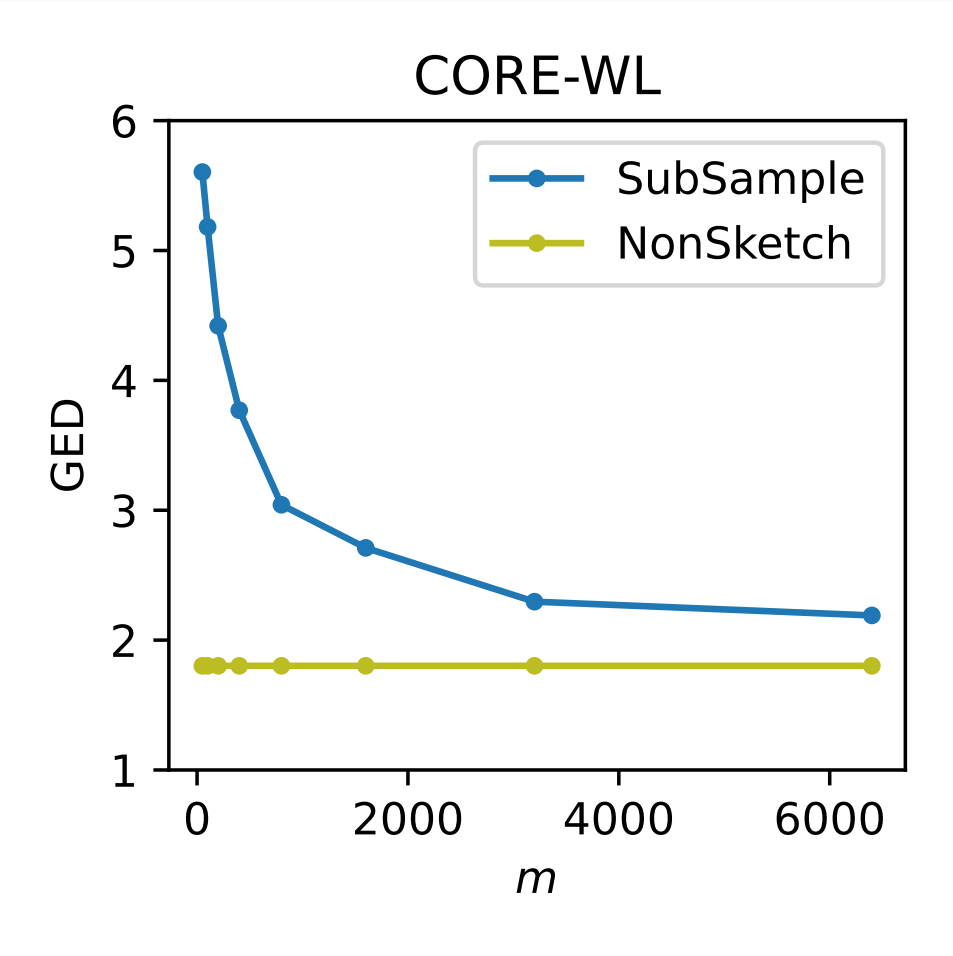
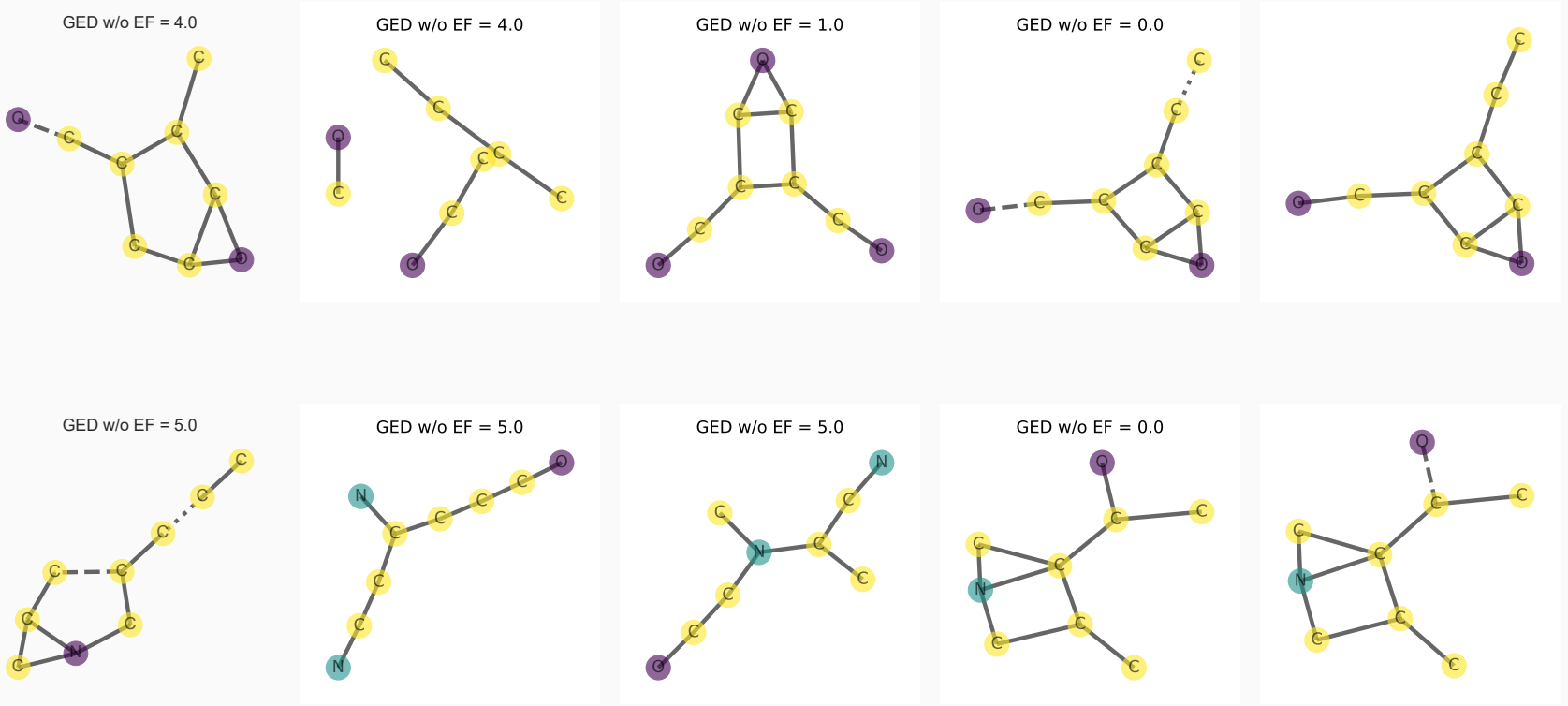


Figure 3: The GED w/ edge feature w.r.t. thes ketching size m for *Perfect h* for the CORE-WL output kernel on SMI2Mol ($m > 6400$ is too costly computationally).

Smiles to molecule: results

	GED w/o edge feature ↓	GED w/ edge feature ↓
NNBary-FGW	5.115 ± 0.129	-
Sketched ILE-FGW	2.998 ± 0.253	-
SISOKR	3.330 ± 0.080	4.192 ± 0.109
DSOKR	$\mathbf{1.951 \pm 0.074}$	$\mathbf{2.960 \pm 0.079}$

Smiles to Molecule: some nice figures



(a) SISOKR (b) NNBary (c) ILE (d) DSOKR (e) True target

Figure 4: Predicted molecules on the SMI2Mol dataset.

Text to molecule: Perfect h strategy

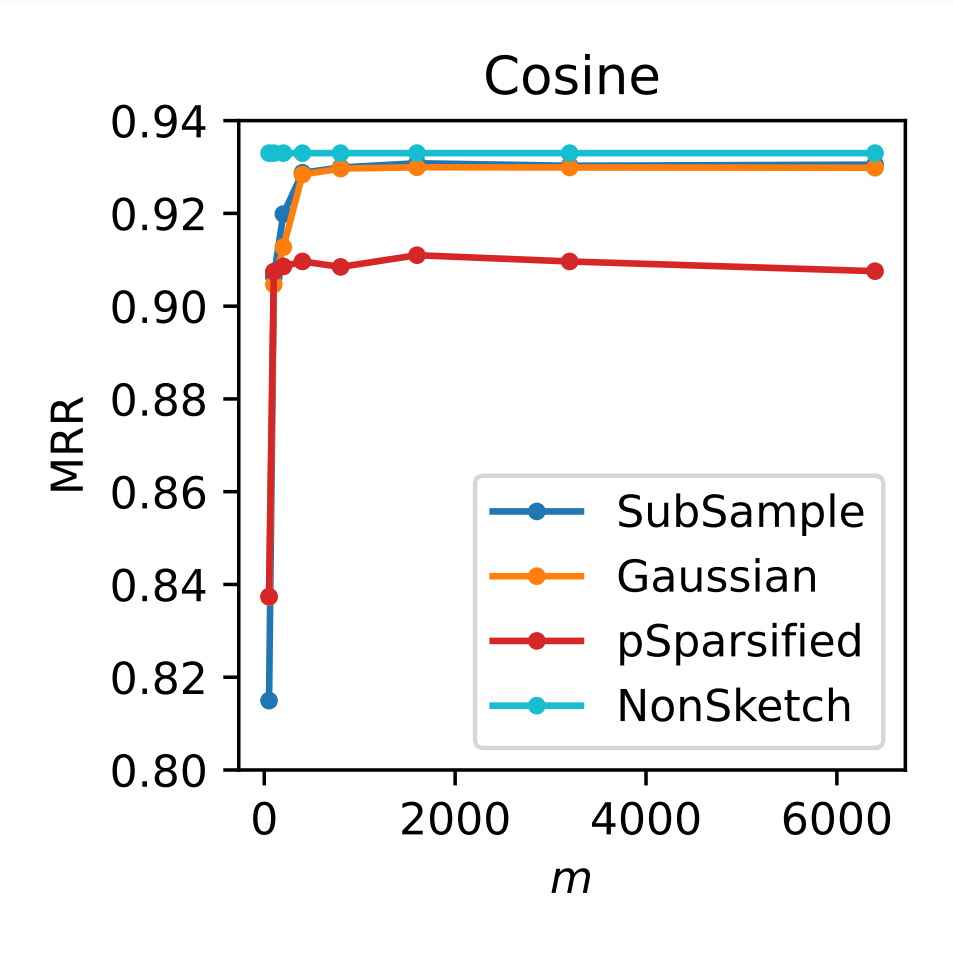


Figure 5: The MRR scores on ChEBI-20 validation set w.r.t. m for *Perfect h* when the output kernel is Cosine on the ChEBI-20 dataset.