Supplementary information for "Optical materials discovery and design with federated databases and machine learning"

Victor Trinquet,*[a] Matthew L. Evans,*[ab] Cameron J. Hargreaves,[a] Pierre-Paul De Breuck,‡[a] and Gian-Marco Rignanese[a]

[a] *UCLouvain, Institut de la Matiere Condensée et des Nanosciences (IMCN), Chemin des Étoiles 8, Louvain-la-Neuve 1348, Belgium*
[b] *Matgenix SRL, 185 Rue Armand Bury, 6534 Gozée, Belgium*

∗ These authors contributed equally to this work.
‡ Present address: Ruhr-Universität Bochum, Universitätsstr. 150, 44801 Bochum, Germany

A Appendix

A.1 Supplementary figures

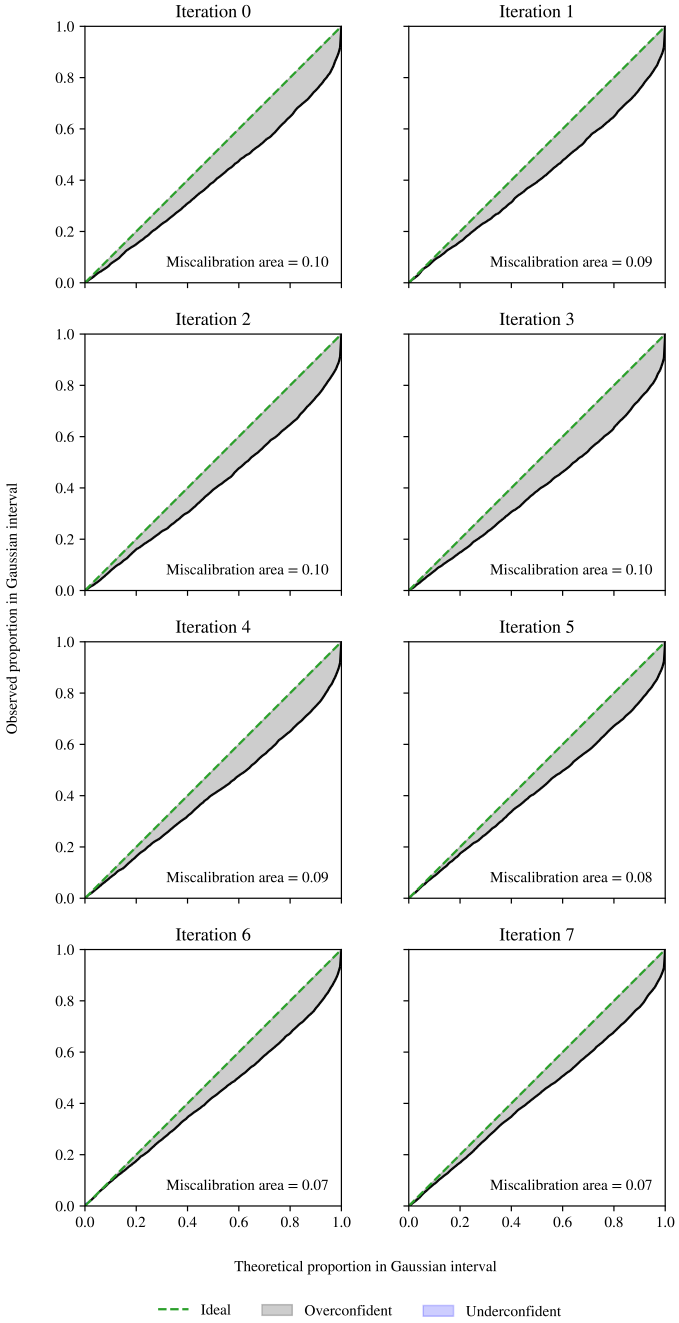

Fig. A.1 Calibration curves for the MODNet model used in the different iterations of the active learning campaign

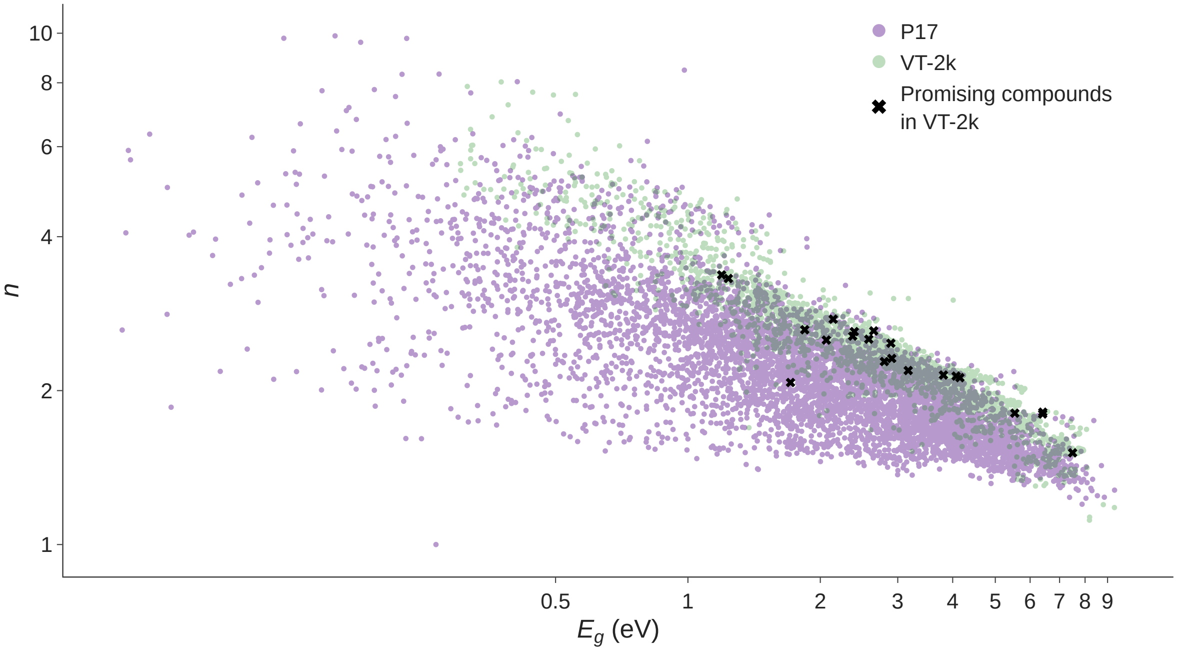

Fig. A.2 The (n, E_g) space spanned by the P17[25] (purple), and VT-2k datasets (green). The black crosses correspond to the materials that pass each of the filtering steps described in **Fig. 2**. They are also listed in Table **2**.

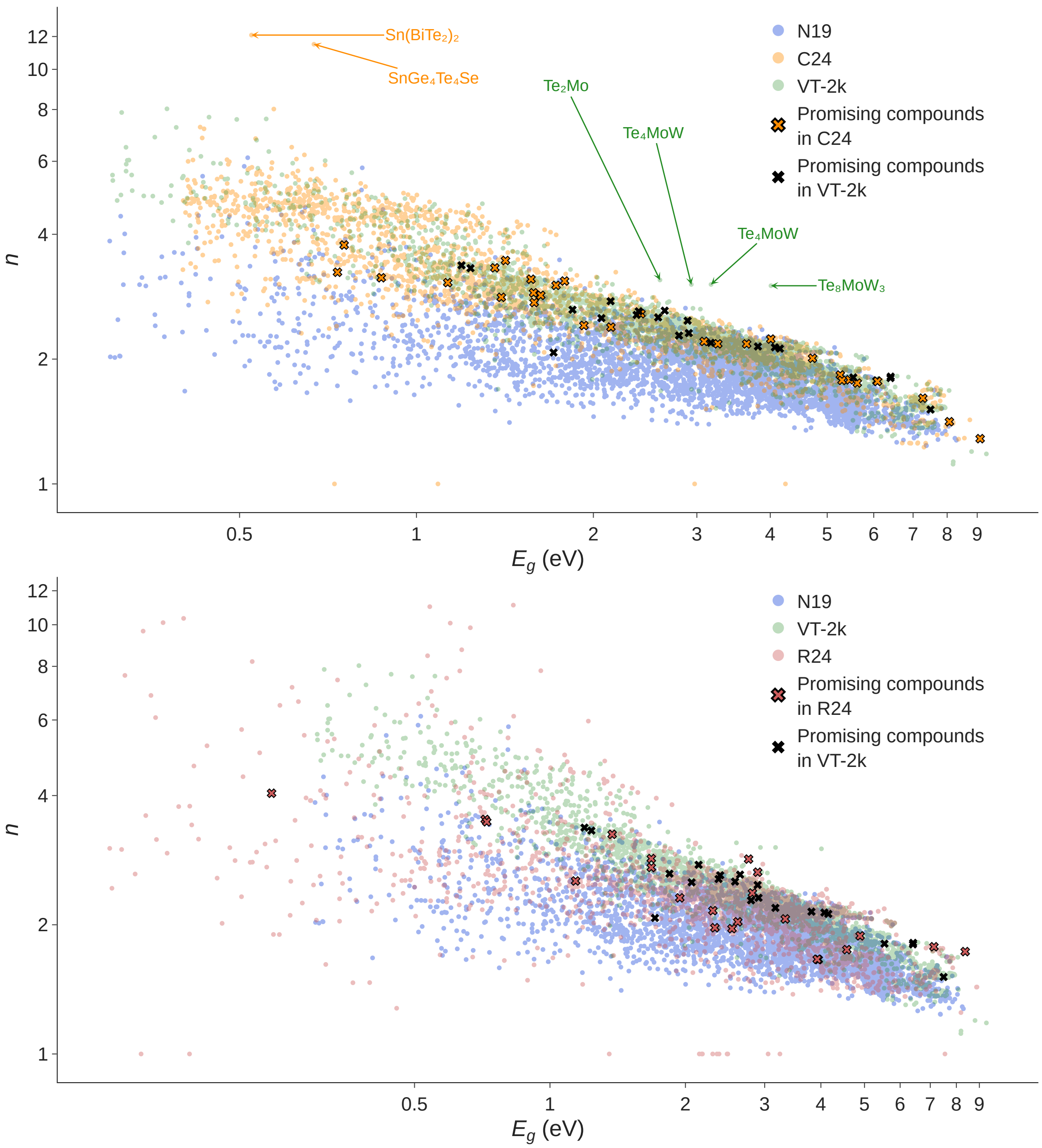

Fig. A.3 The filters described in **Fig. 2** applied to the external datasets of C24[26] (orange crosses, upper panel) and R24[28] (red crosses, lower panel), in comparison with the filtered materials from VT-2k (black crosses). Exceptional outlier materials (Te-W-Mo and Sn-Te containing compounds) from each dataset are labelled explicitly.

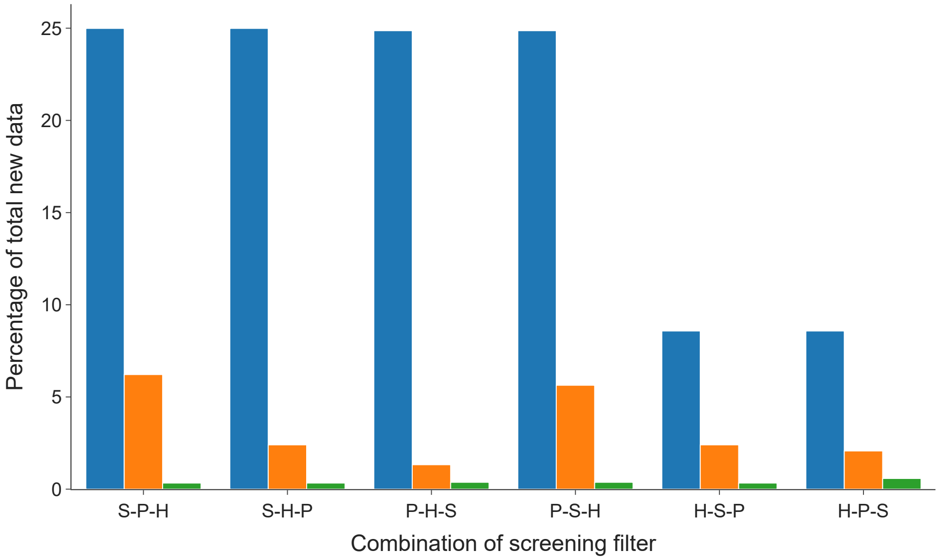

Fig. A.4 Effect of the order of the filters on the final screening of promising high-n materials. The y-axis shows the relative quantity of selected materials with respect to the present dataset (2,413 entries). The different orders of filters are situated along the horizontal axis where 'S' stands for the criterion on synthesisability, 'P' for the restriction to the Pareto neighbourhood, and 'H' for the exclusion of high HHI elements.

A.2 Active versus static learning

The advantage of active learning with respect to other common search methods is demonstrated using the P17 dataset. Initially, a subset of 1,000 labeled samples from P17 is designated as the starting set $\mathscr{L}$ (simulating a known dataset), while the remaining $\sim$ 5,000 data points form the candidate pool $\mathscr{P}$ and remain unlabeled (simulating known structures with unknown characteristics). Different strategies are then employed to query b points (as defined by the budget, up to 1,000) from $\mathscr{P}$. These points are subsequently labeled using the P17 dataset (simulating an oracle). The quality of the selected points is quantified by computing the top-k score over ω_{eff}:

$$\text{top}_k(\omega_{\text{eff}}) = \frac{1}{k}\sum_{i=1}^{k} \omega_{\text{eff},(i)},$$

where $\omega_{\text{eff},(i)}$ represents the i-th highest score among the newly labeled samples. This metric is calculated over the k samples with the highest ω_{eff}, where k is an adjustable parameter.

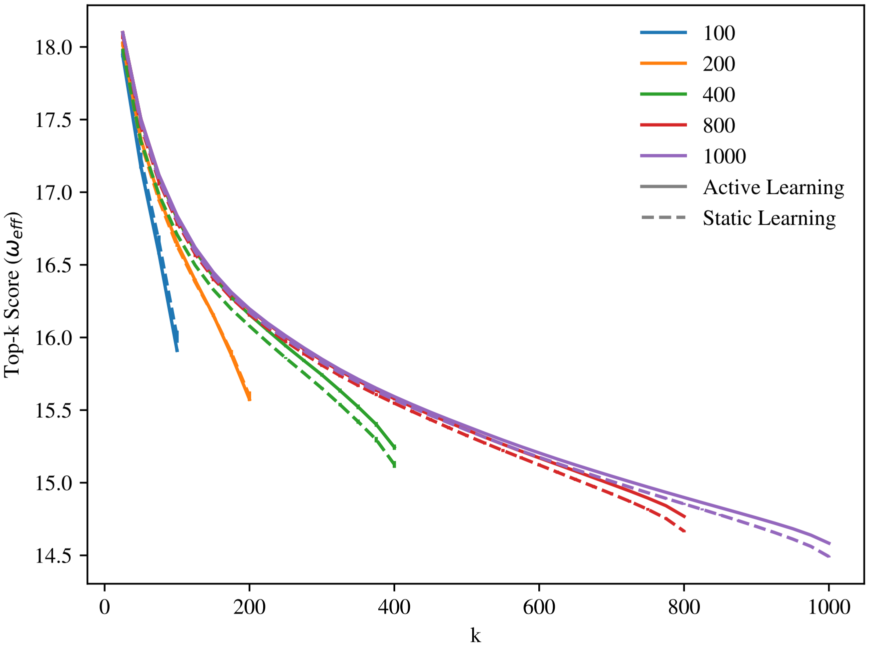

Fig. A.5 Top-k score as a function of k for different search budgets, comparing AL (Active Learning) and SL (Static Learning) algorithms.

As a baseline strategy, one might randomly select points from $\mathscr{P}$. The P17 dataset presents average values of $n = 2.29 \pm 0.86$, $E_g = 2.44 \pm 1.67$, and $\omega_{\text{eff}} = 11.91 \pm 3.61$ ($\pm$ denoting the standard deviation). For a random search baseline, the expected value of the top-k score, calculated over the highest k scores from randomly selected points, would approximate the mean of ω_{eff}, i.e., 11.91. The variance of this score would decrease with increasing k, scaling approximately as $1/k^2$.

A more commonly used approach is to train a surrogate machine learning (ML) model on $\mathscr{L}$, and then use it to screen $\mathscr{P}$ for promising candidates, selecting the b candidates with the highest predicted ω_{eff}. We refer to this method as static learning (SL). In contrast, active learning (AL) iteratively updates its ML model while exploring $\mathscr{P}$. Both strategies are allowed to query b new samples from $\mathscr{P}$, as defined by the budget.

Figure A.5 represents the top-k score as a function of k, for different budget sizes. It is observed that AL systematically outperforms SL. Although the advantage is small, it is noteworthy that with a budget of 800, AL achieves a similar or better top-k score than SL with a budget of 1000, across almost the entire range of k. This represents a non-negligible saving of resources of 20%.

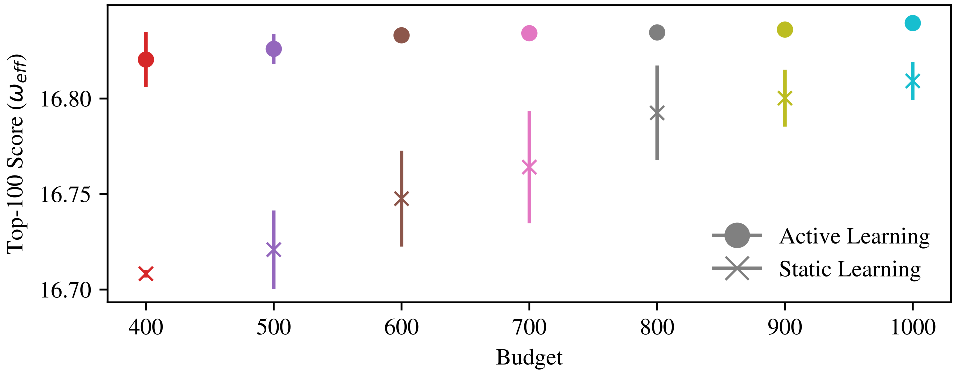

Fig. A.6 Top-100 score as a function of iterations (i.e., search budget) for AL (Active Learning) and SL (Static Learning) search algorithms. Error bars represent the standard deviation over 5 repeated experiments.

Figure A.6 displays the Top-100 score for various budgets, while also depicting the observed standard deviation over repeated experiments. A slight yet consistent advantage is observed for active learning, with a notably smaller variance among selected candidates. Notably, a budget of 1000 is necessary for static learning to achieve a Top-100 score comparable to that reached by active learning with just a budget of 400.

Finally, Figure A.7 shows the Mean Absolute Error (MAE) on a hold-out test set as a function of the AL iterations (i.e., search budget). As expected, it demonstrates that in AL, the model progressively improves with each search iteration. In contrast, for SL, the accuracy remains constant, as the model does not update during the search process.

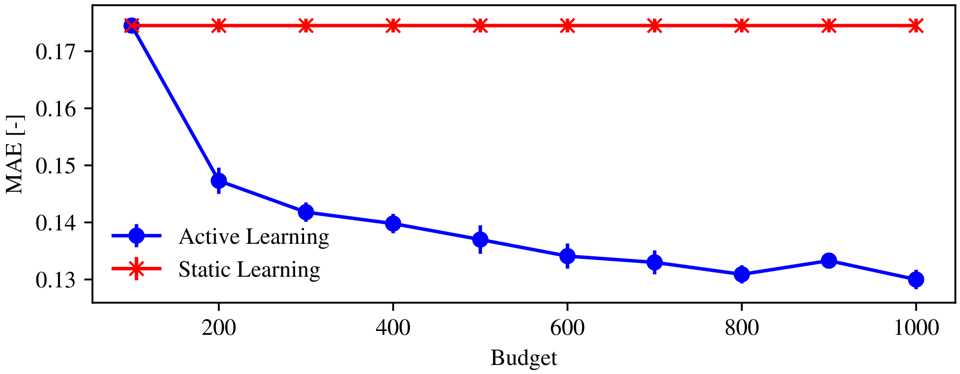

Fig. A.7 Mean Absolute Error (MAE) of the models used in Active Learning (AL) and Static Learning (SL) on a hold-out test set, as a function of iterations (i.e., search budget).