WHITE PAPER

Hit the mark: train smarter AI with target specifications

A 5x performance increase in design of formulations, chemicals, and materials

intellegens.com | info@intellegens.com
Intellegens, The Studio, Chesterton Mill, Cambridge, CB4 3NP, UK

intellegens

Executive summary

Machine learning shows great potential to design and understand **formulations, chemicals, materials**, and **biopharmaceuticals**. Being focused on the end goal is key: to find the mixture that will fulfil a target specification – for example, to maximize strength, minimize cost, or minimize carbon footprint.

We demonstrate how Alchemite™ offers a machine learning workflow that uniquely exploits the target specification to train a machine learning model that is optimized for this specific goal. This allows a **5x increase in performance** over a model trained for accuracy over all formulations, saving time and money in the design process.

Introduction

The Alchemite™ machine learning software is engineered to **extract all available information** from R&D data: leveraging property-property correlations to overcome sparse data, extracting information from the amplitude of noise, and enabling deep analysis of trends and interactions. These capabilities already enable Alchemite™ to excel at working with the sparse, messy datasets typical of formulation and materials projects.

An additional, and so far untapped, source of information is the **target specification** set by the scientist. Typically, machine learning models are trained to perform well over the entire range of the data on which they are trained. However, with knowledge of the project goals we can train a model to make especially accurate predictions at the specific design variables of interest. This leads to a **5x increase in accuracy** and so to a reduction in cost and time to develop formulations.

Measuring model quality

To train a useful machine learning model, we need both a clear objective and a way to measure success. Machine learning always starts with data, which is typically split into a set for training and a separate set for validation to evaluate the model performance. This quality is measured as the average model error over all the available validation data, often using the coefficient of determination:

$$R^2 = 1 - \frac{\sum_n (y_n - p_n)^2}{\sum_n (y_n - \bar{y})^2},$$

where index n sums over all of the data $\{y\}$ in the validation set, with predictions from machine learning of $\{p\}$. The model parameters are then optimized to maximize the coefficient of determination to give the very best model to deliver high performance across the design space.

However, in most real-world R&D projects the key objective is not to understand the whole of a system equally: instead, the project seeks to discover formulations or materials that achieve commercially valuable performance, for example, seeking outputs $y > t$ that are greater than the target t. To reflect this in model training, we developed a target weighted quality metric

$$R_t^2 = 1 - \frac{\sum_n w(y_n, p_n, \sigma_n, t)(y_n - p_n)^2}{\sum_n w(y_n, p_n, \sigma_n, t)(y_n - \bar{y})^2},$$

where the weighting function $w(y_n, p_n, \sigma_n, t)$ is a function of the true values, predictions, predicted uncertainties, and the target. The function puts more emphasis on predictions that are the wrong side of the target t, ensuring the model can more accurately discriminate between formulations or materials that achieve, or do not achieve, the commercially valuable performance level. An example function is shown in Figure 1, where the weighting function is shown in blue and the target by the orange line.

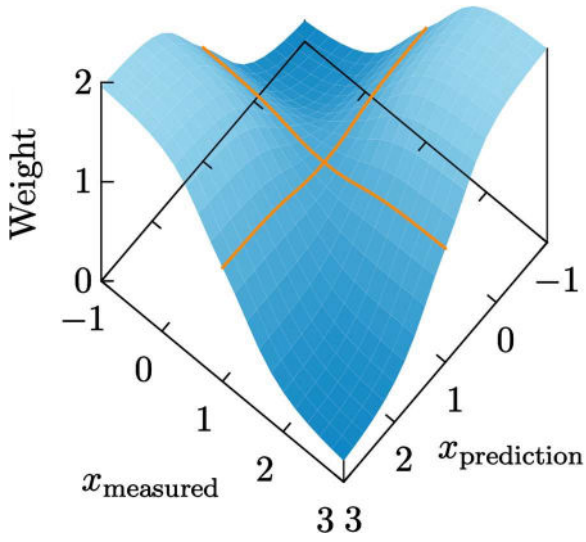


Figure 1. The weighting function with changing predicted and measured values is shown in blue and targets in orange.

The function is symmetric around the identity line: we treat 'false positive' and 'false negative' cases where the predictions are the wrong side of the target value equally. The smooth function shown in Figure 1 aligns with the probabilistic interpretation of material optimization, where we aim to maximize the probability of a new material achieving the target: this makes the training of the model more relevant to the eventual use-case of material optimization than if we used a 'pass/fail' classification of each predicted point. For comparison the standard weighting factor used in the calculation of the coefficient of determination is simply the constant $w = 1$.

Variants of the metric have been developed to work for continuous, ordinal, and categorical variables.

With the focused metric in place, we now demonstrate its utility and performance in example use-cases: improving the accuracy of modelling simple systems, accelerating design of experiments, and an exemplar real-world dataset. These examples showcase how aligning model training with the end goal delivers tangible gains in both model performance and R&D efficiency.

Toy example

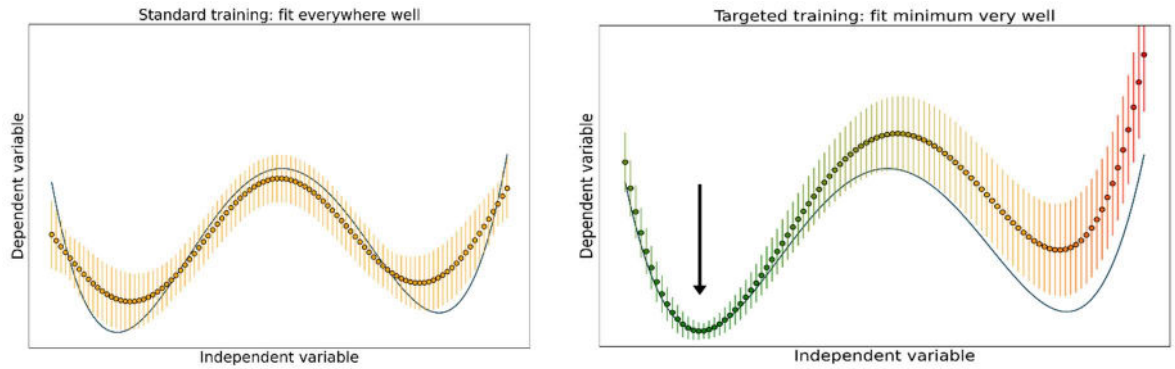


Figure 2: A double-minimum function shown by the black line with a global minimum (to the left of the function) and a subsidiary minimum (to the right of the function). The left-hand panel shows when a machine learning function is fitted across the entire curve by the yellow points with one standard-deviation uncertainty. The right-hand panel shows when a machine learning function is fitted to target low function values, with the arrow denoting the global minimum that we are targeting.

To show the benefits of providing the target specification when training the model, we use a 'toy example' of training a model to fit a one-dimensional curve with a global minimum and a subsidiary minimum. In the end-use of the model, we are particularly interested in understanding the location and depth of the global minimum of the function. The left-hand panel shows a machine learning model that gives a reasonable fit to the function over its entire extent but misses the function minima. However, if we train the machine learning model with the knowledge that we are especially interested in targeting the function global minimum then we can achieve the fit shown in the right-hand panel. Here the machine learning model has better reproduced the function minimum and performed less well elsewhere. But these less well-represented regions are unimportant to the user beyond the fact that they are above the minimum. The machine learning model uncertainties have adjusted to reflect the greater accuracy around the global minimum. The targeted machine learning model is therefore more helpful at answering the crucial question: "*is the function value greater or less than the target value near to the global minimum?*"

Paradigmatic example

To show the performance of the metric on a concrete example, we create a one-dimensional dataset of some 40 entries following $y = x^2$ with increasing noise amplitude. We set a target to find predictions at $y < 0.1$. To assess model quality, we determine how many predictions are above or below the target versus the true function.

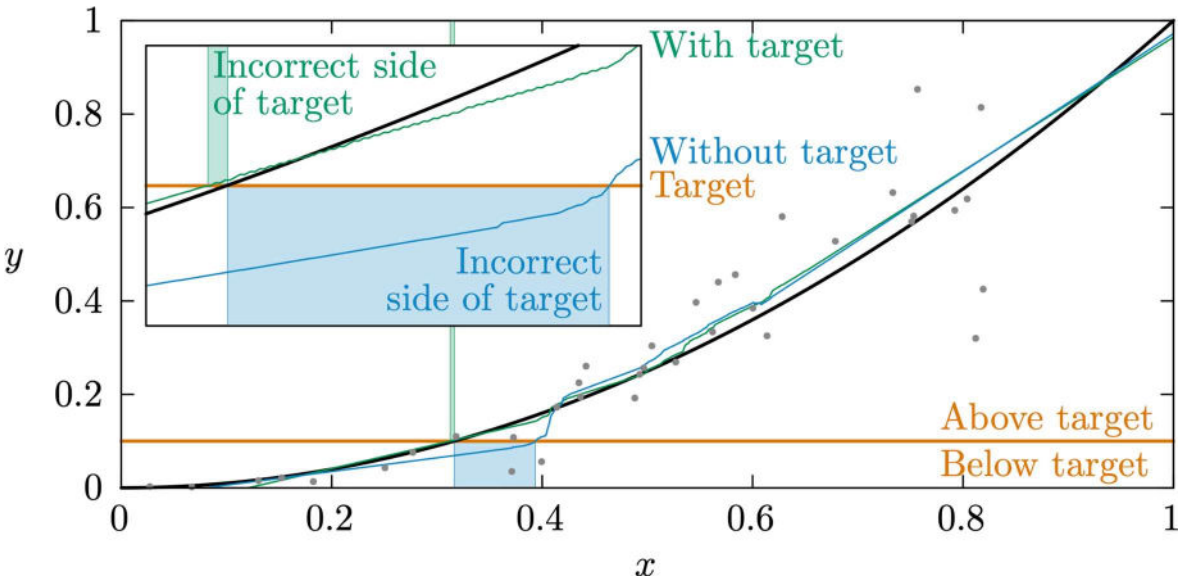


Figure 3: The one-dimensional function (black line) with inset focused on the function near to the target. The gray points are the training data that include noise scattered about the function, and the orange line the target value, y=0.1. The blue line is the model fit without knowledge of the target, and the green line is the fit performed with knowledge of the target. The blue shaded area shows where the model trained without knowledge of the target predicts that the model will not exceed the target, whereas in fact the true curve did. The green shaded area shows where the model trained with knowledge of the target predicts that the model exceeds the target whereas in fact the true curve did not.

We train two machine learning models: one without knowledge of the target requirements, and one with knowledge of the targets. The predictions of the two models are shown in Figure 3. We measure the performance of the models by examining the range of *x* values over which the model makes incorrect predictions. With standard machine learning the model is incorrect in the range $0.317<x<0.394$, whereas when it was trained with the target specification available the model is incorrect in the range $0.313<x<0.317$, corresponding to a 19x improvement.

In summary, knowledge of the target specification whilst training the model has led to around a 20x improvement in the model's ability to discriminate whether a prediction will fulfil the target specification. This offers a really significant improvement in modelling accuracy and efficiency.

Optimization example

Machine learning is increasingly used to guide optimization in R&D, helping scientists identify the best-performing formulations or materials with fewer experiments. Crucially, when the model is trained with the performance target explicitly defined, it becomes more accurate in the high-value region of the design space. This targeted accuracy enables more effective optimization, reducing both experimental cost and time to discovery.

To demonstrate this, we aim to minimize the six hump camel function[1] – a standard benchmark in optimization research. Starting with only 32 rows of training data, we compare three approaches over 25 iterations. These include the industry-standard OnePlusOne evolutionary optimization algorithm from Nevergrad[2]; a standard Alchemite™ model trained without target information; and an Alchemite™ model trained with the optimization target embedded from the outset. In each case, the model proposes a new experiment predicted to improve on the best result so far, and the outcome is added back into the process. The results are shown in Figure 4.

The results are striking. After 20 new experiments, the Alchemite™ model trained with target information improved on the best value found in the initial data by a factor of 250. In comparison, the standard Alchemite™ model achieved a 50-fold improvement and Nevergrad achieved a 35-fold improvement. The fivefold improvement in optimization quality using the target information when compared to standard model training, for exactly the same modelling and experimental cost, promises to accelerate optimization of materials and chemicals.

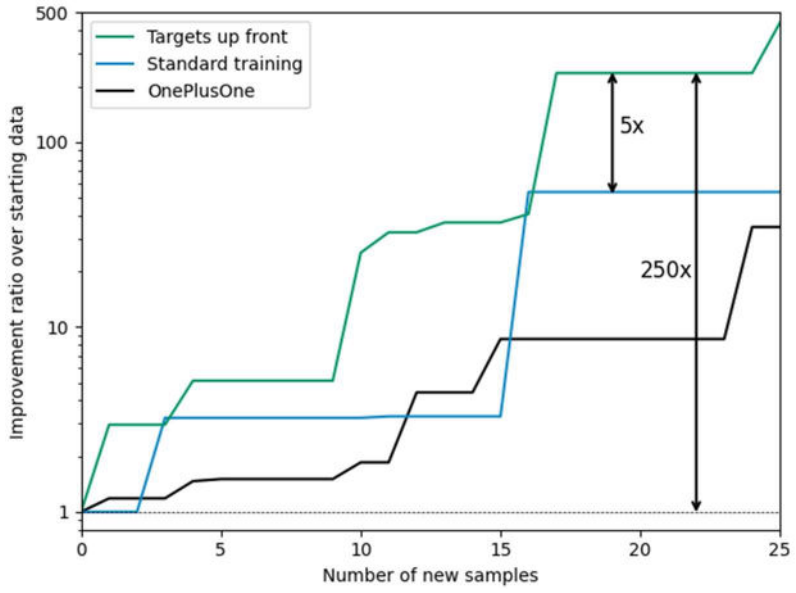


Figure 4: Improvement in minimum value found of the six hump camel function using different optimization algorithms: setting the optimization target up front when training the model (green line), versus optimization without knowledge of the targets (blue line), versus the industry leading OnePlusOne algorithm from Nevergrad (black line); optimization with the target up front delivers the best improvement over the starting data.

[1] https://www.sfu.ca/~ssurjano/camel6.html
[2] https://github.com/FacebookResearch/Nevergrad

Real-world example

To test the approach on a more complex and realistic dataset, we applied Alchemite™ to the challenge of predicting the presence of Alzheimer's disease based on the publicly available OASIS dataset from Washington University[3]. The model takes as input key variables thought to affect the onset of Alzheimer's, including: *Male or Female?; Age; Education; Socioeconomic Status; Mini-Mental State Exam; Estimated Total Intracranial Volume; Normalised Whole Brain Volume; Atlas Scaling Factor* to predict the *Clinical Dementia Rating (CDR).*

The clinical dementia rating ranges from non-demented through to demented in an ordinal scale: we set the objective as distinguishing non-demented patients from all other levels. Two models were compared: one trained using conventional methods without a specific diagnostic target, and one trained with the target embedded upfront — prioritizing accurate classification near the decision threshold.

Including the diagnostic target during training led to a notable improvement in distinguishing non-demented patients from others. Misclassifications dropped by approximately 20%, and the Matthews Correlation Coefficient more than doubled, demonstrating a clear benefit from focusing the model on the clinical decision threshold. This underscores the broader relevance of target-aware training in high-stakes, data-limited environments like healthcare.

Figure 5 shows the Influence of each of the input variables on the targeted models' understanding of CDR. The targeted model identifies that the Mini-Mental State Exam is a key predictor for distinguishing non-demented patients from others, highlighting this as a key measurement to make for patients to determine their dementia status.

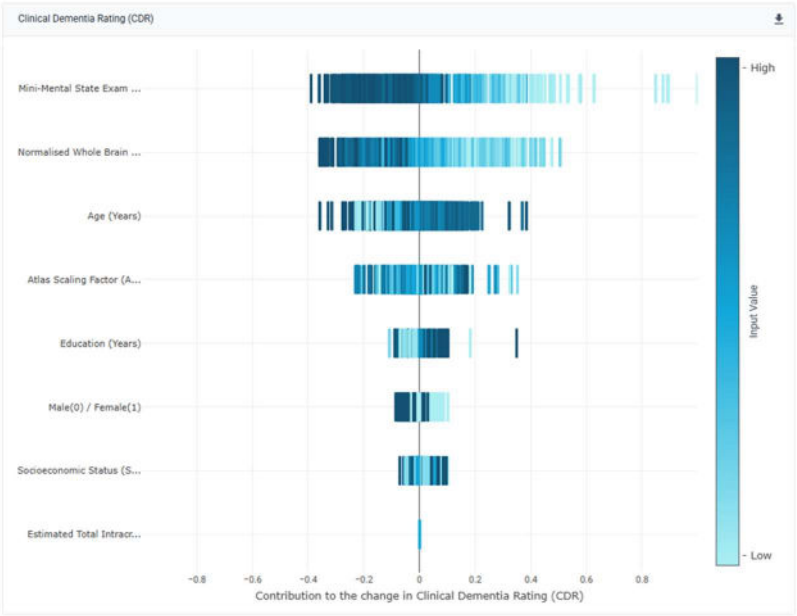


Figure 5: Influence of the input variables on the targeted model's understanding of CDR. Each small vertical line represents a patient. Dark blue lines represent higher values of the input, and lines to the right of the centre line represent an increase in the predicted CDR.

[3] https://sites.wustl.edu/oasisbrains/

Workflow benefits

Including property targets in ML model training improves R&D workflows in four key ways:

Focus on project end goals: inputting target end goals at the start of the training process focuses the user on achieving those goals and on whether the model is sufficiently accurate around the target values, avoiding distractions connected to accuracy in less relevant regions. The user can shift the end goals part-way through the project as machine learning helps them understand the system better and their priorities change.

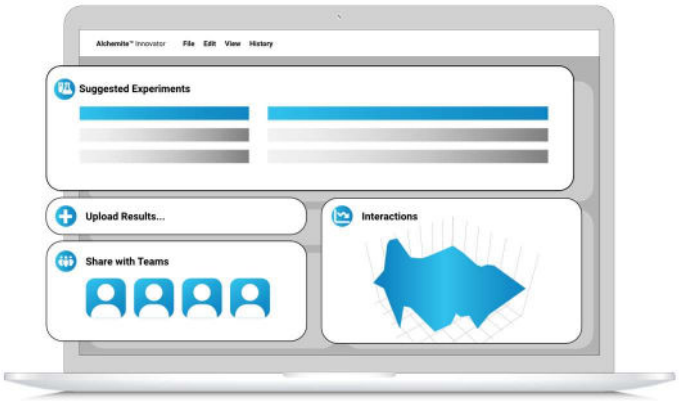
Increase accuracy: we have seen how training a model in the presence of targets delivers improved quality of predictions in those key regions. Better predictions help users make better decisions and optimize products more effectively.

Reduce data wrangling: training data typically represents previously performed experiments. Models are usually trained to best fit over that distribution. The nature of R&D means there is often little data in the region now of interest, and so model training deprioritises that region. Data scientists spend significant time deliberately removing or augmenting data to better focus training. The methodology described here relieves them of this task, saving that time.

Improve design of experiments: The improved accuracy of predictions is accompanied by improved understanding of the uncertainty in those predictions. This is vital to ensure that during Design of Experiments cycles effort is focused where it is needed most: promising but uncertain results in which confidence could be grown by performing additional experiments.

Alchemite™ software

The **Alchemite™ Suite** is a range of easy-to-use R&D tools, each focused on a key challenge for R&D managers, scientists, experimentalists, or data scientists. Give the right app to the right team member, speeding and informing their work. Then share results and collaborate across your team, creating an integrated machine learning solution for your R&D organization.



Alchemite™ Innovator combines predictive tools with a quick and easy method to design experimental programmes, for a complete project toolset. Using the powerful Alchemite™ method, you can instantly generate a machine learning model from your data, even when that

data has gaps or is noisy, where other ML methods fail. You can optionally specify targets, as discussed in this paper, before the machine learning model is trained. Then apply the model to empower your research.

More at intellegens.com/solutions/innovator/

About Intellegens

Our vision is that machine learning will drive innovation and deliver value wherever data is used in R&D. We aim for best-in-class easy-to-use machine learning software for data analysis in chemicals, materials, life science, and manufacturing. Our Alchemite™ technology originated at the University of Cambridge and development is on-going at Intellegens, in close collaboration with our growing community of customer organizations. These represent sectors including additive manufacturing, aerospace, alloys, batteries, biopharmaceuticals, ceramics, chemical processes, composites, consumer products, cosmetics, drug discovery, energy, food and beverage, formulated products, paints, plastics, and printing technology.

www.intellegens.com | info@intellegens.com