Science Advances
AAAS

advances.sciencemag.org/cgi/content/full/6/9/eaay6913/DC1

Supplementary Materials for

Automated structure discovery in atomic force microscopy

Benjamin Alldritt, Prokop Hapala, Niko Oinonen, Fedor Urtev, Ondrej Krejci, Filippo Federici Canova, Juho Kannala, Fabian Schulz, Peter Liljeroth*, Adam S. Foster*

*Corresponding author. Email: peter.liljeroth@aalto.fi (P.L.); adam.foster@aalto.fi (A.S.F.)

Published 26 February 2020, *Sci. Adv.* **6**, eaay6913 (2020)
DOI: 10.1126/sciadv.aay6913

This PDF file includes:

Section S1. Image representations of output molecular structure

The chosen image representation is even more important when considering non-ideal experimental conditions and the inherent approximations present in any simulation model generating the training examples. The information that we try to reconstruct should be not only present in AFM data, but it must also be robust with respect to minor errors and unknowns. In particular, the determination of atoms which are too deep and therefore do not significantly contribute to the AFM signal is an ill-posed problem, and we should therefore avoid output representations that pretend to provide information about such atoms.

To aid in validation, we should also output a representation which is convenient for perception of the human user as well as for design of the neural network architecture. Representation of the molecular structure as a list of atomic coordinates is generally considered problematic for machine learning, as the size of such an output vector would differ for each molecule, and it does not respect permutation symmetry (i.e. exchange of atoms corresponding to the same element in the list produces apparently different descriptors, despite representing physically identical structures). Various methods were developed in the past to overcome these problems (e.g Atom-centered symmetry functions (*68*), or smooth overlap of atomic positions (*69*)), but do not present molecular geometry in a human-readable form. Image-like representations are an intuitive choice for humans, projecting the system geometry into a scalar or complex field sampled on a regular real-space grid (*70*).

The output property has to be generated together with simulated AFM data during training, therefore the definition of the output property is closely related to the algorithm by which it is generated. In this work, we designed three distinct image descriptors representing 2D image projections of molecular geometries (see Fig. S1).

Height Map For each pixel of the image, we calculated the depth at which the vertical component of forces between sample and tip apex becomes more repulsive than some constant value (typically $\approx 0.1eV/\text{Å}$). The resulting image (see Fig. S1e,h) should roughly correspond to a hypothetical AFM image obtained in constant-force mode with a non-flexible tip (unlike CO). It also roughly corresponds to the concept of the solvent-accessible surface introduced in biochemistry (*71*), indicating areas accessible to the probe particle. This is also very useful to rationalize the formation of the imaging contrast and supramolecular interactions in general. In the future we plan also to map the electrostatic potential on top of this surface.

van der Waals Spheres While the Height Map corresponds to an isosurface of aggregate force (with contributions from all atoms), a representation of molecular structure by van der Waals spheres is commonly used in chemical visualization programs (e.g. Jmol), as it is intuitive for the human user. This also follows naturally from the LJ-potential based PP method used in simulations. The vdW-Spheres representation (see Fig. S1f,i) shows creases between atoms, in contrast to the blunt shapes of the Height Map. This was one motivation to introduce this output descriptor - it allows us to see how well discrete atoms can be recognized from the images. In the nomenclature of neural networks, we can roughly relate the Height Map to a soft-max-operation and vdW-Spheresto simple max-operation over the contributions of atoms to the force field.

By its nature, the van der Waals Spheres representation encodes the depth of an atom's position by brightness of sphere and atomic radius (which is connected with atomic type) by sphere size. Therefore we considered modification of the CNN learning process for the vdW-Spheres representation such that different sphere radii would be split by colour for more convenient atomic type identification - we call this modification *type map*. In its current form, it allows identification of 3 different categories of atoms according to sphere radii: hydrogen (red), non-hydrogen peripheral (green) and carbon backbone (blue).

Atomic Disks This descriptor goes one step further from a quantity directly related to the force field towards a more abstract representation. Here we render small disks (with the brightness conically decreasing from the center) onto positions of atomic nuclei (see Fig. S1g,j). The brightness indicates the height of atoms, while the size of the disk is proportional to the covalent radius of atom.

S1.1 Comparison of the different output representations of molecular structure generated from simulated AFM input

Here (Fig. S1 – Fig. S3) we compare predictions of other representations for molecules from Fig. 2 of the main text. In general it can be said that decreasing the size of atomic features (from Height Map to Atomic Disks) helps to better recognize the atoms by a human user, but also it makes the training more demanding and output less robust. Hence, the selection remains a compromise, and we are planning to implement simultaneous calculation of all descriptors (these and other descriptors in development) during the simulation process.

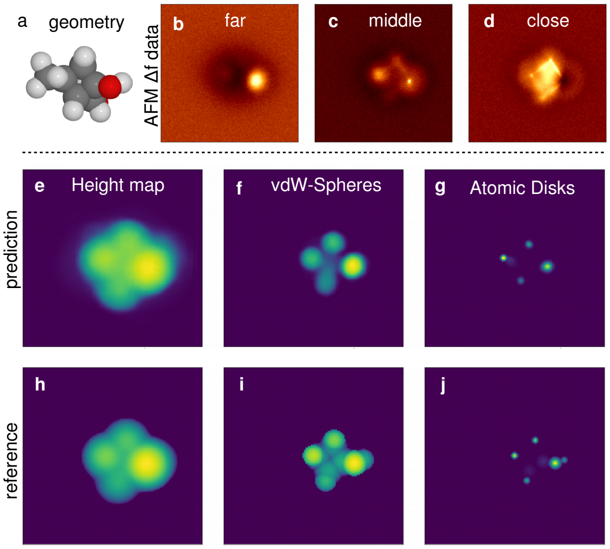

Fig. S1. **Different 2D image representations of the output geometry X for simulated AFM images of a $C_7H_{10}O_2$ molecule from the training set.** **a** 3D render of molecular geometry; **b-d** simulated AFM Δf images with decreasing tip-sample distance; **e-g** 2D Image representation predicted by our CNN; **h-j** Reference 2D Image representation produced directly from the molecule geometry.

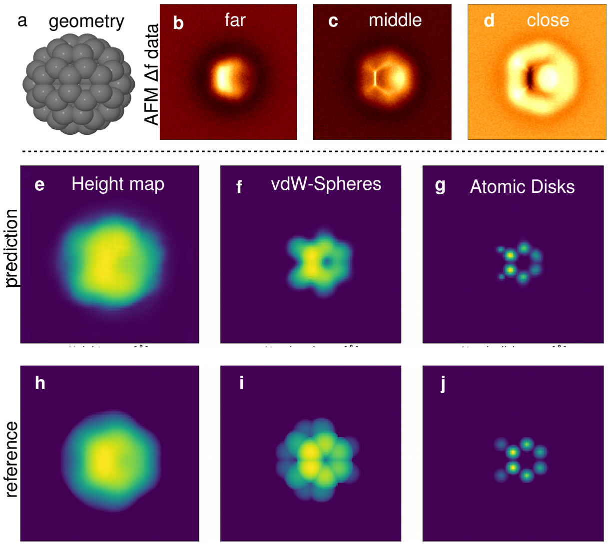

Fig. S2. **Different 2D image representations of the output geometry X for simulated AFM images of a C_{60} molecule. a** 3D render of molecular geometry; **b-d** simulated AFM Δf images with decreasing tip-sample distance; **e-g** 2D Image representation predicted by our CNN; **h-j** Reference 2D Image representation produced directly from the molecule geometry.

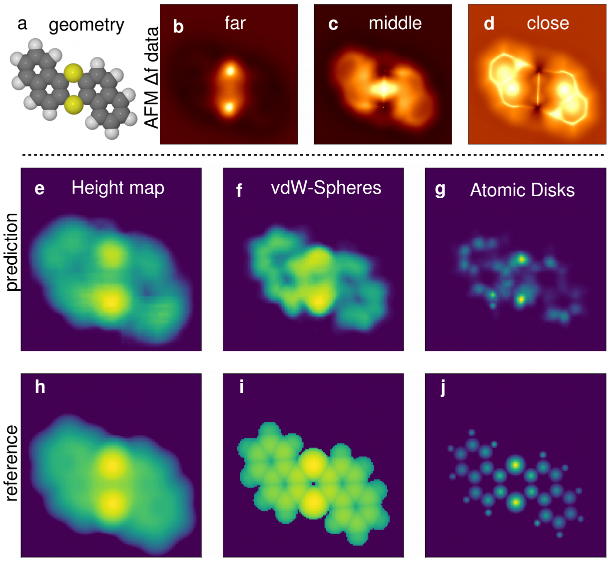

Fig. S3. **Different 2D image representations of the output geometry X for simulated AFM images of dibenzo[a,h]thianthrene molecule (*71*). a** 3D render of molecular geometry; **b-d** simulated AFM Δf images with decreasing tip-sample distance; **e-g** 2D Image representation predicted by our CNN; **h-j** Reference 2D Image representation produced directly from the molecule geometry. Note that the Atomic Disk representation fails to reliably predict the exact positions for atoms of the aromatic systems near to the sulphurs, but located *deeper* i.e. lower in contrast. Failure to predict deeper atoms is a typical problem of our NN. It follows up from the ”conservatism” enforced by regularization techniques (e.g. related to noise and dropouts). If the CNN does not accumulate enough evidence, it omits atoms completely instead of predicting them at wrong positions. In this particular case, the prediction of lower lying atoms is perhaps disturbed by the dominant sulphur atoms.

S1.2 Examples of molecules from the training set

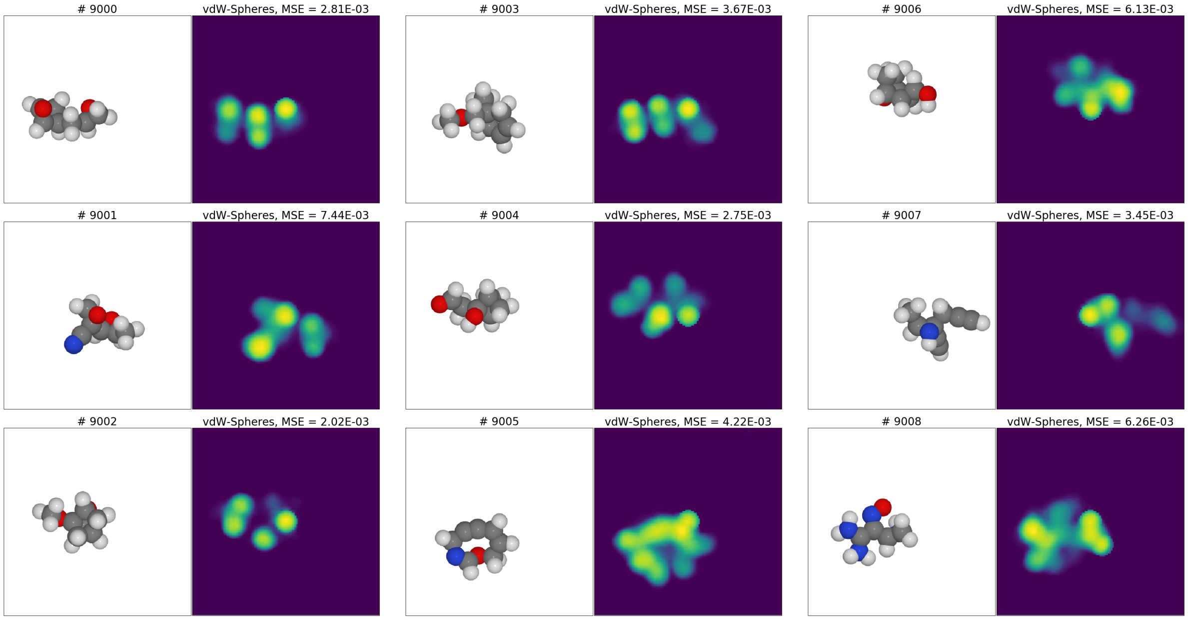

Fig. S4. **Molecules from the validation data set together with the vdW-Spheres representation predicted by the CNN.** Here are presented 9 simulated random molecules from our validation data set. Each pair of images contains: *left* – molecular geometry where color represents the type of atoms (white - **H**, gray - **C**, blue - **N**, red - **O**); *right* – vdW-Spheres representation of the structure predicted by the CNN. The orientation of the molecule is selected to maximize the number of visible atoms, as described in the main text.

Section S2. Matching experiment to relaxed on-surface simulated configurations

From a physical point of view, it is reasonable to match experimental AFM images only to those simulated orientations of a molecule which represent some local minimum on the surface. Nevertheless, such an approach is strongly dependent on the ability to reliably find all such local minima (possible configurations) by purely computational means, which is still generally an unsolved task. Although the development of novel global configuration search methodology is beyond the scope of this paper, we tentatively examined this idea. As an initial approximation, we used rigid-body molecular mechanics (*73*) with only a van der Waals force field to relax 500 uniformly distributed initial rotations of the camphor molecule into 7 distinct local minima. Then we further relaxed those configurations with the density functional package cp2k (*74*). These local energy minimum configurations were then compared to a set of experimental configurations using a linear correlation metric.

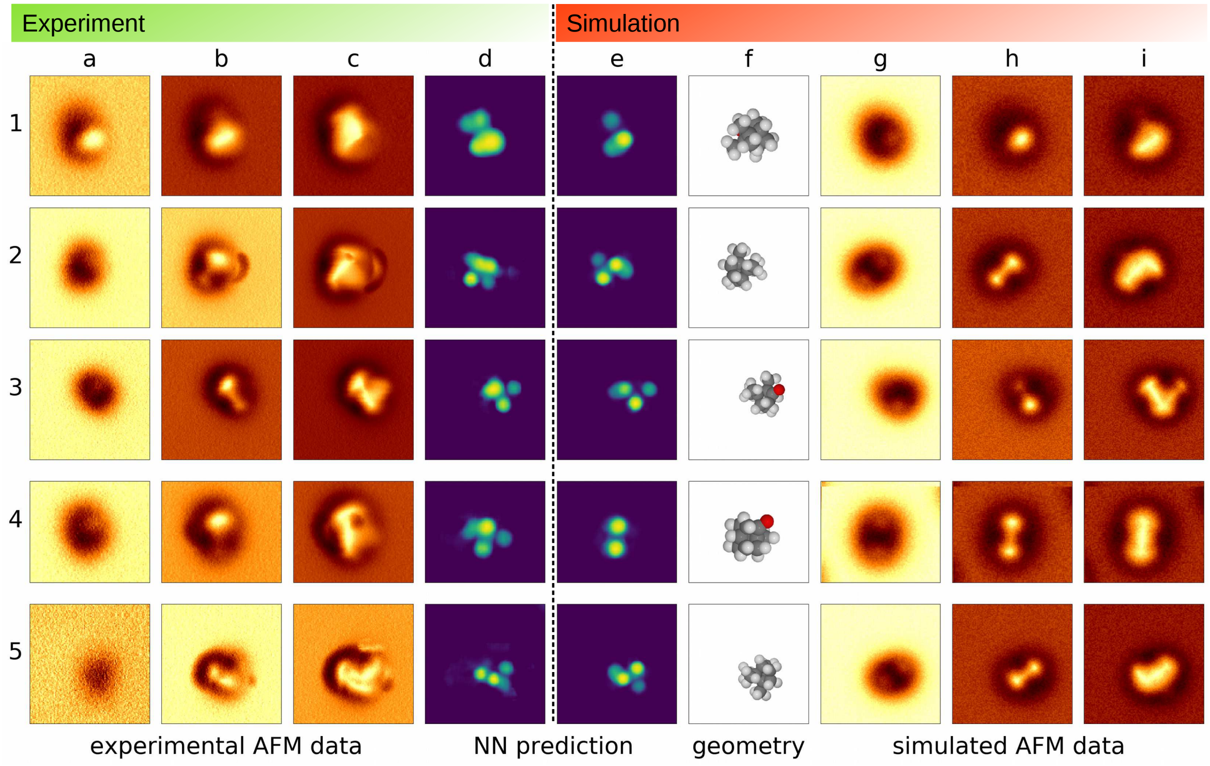

Fig. S5. **Matching between simulated relaxed configurations of 1*S*-camphor and experiment. 1-5** refer to distinct molecular configurations with experiments in columns **a-d** and simulations in columns **e-i**. Selected experimental AFM images (out of 10 slices used for input): **a** – far, **b** – middle, **c** – close tip-sample distances and NN prediction **d** for the vdW-Spheresrepresentation. The vdW-Spheresrepresentation **e** corresponds to the full molecular configuration **f** resulting in the best match with the experiments. The corresponding selected simulated AFM images are given in panels **g-i**.

The CNN model's predicted image representations for the molecular structure (vdW-Spheres) on Fig. S5 allow for the assignment of simulated configurations with known geometry (column (f)) for each experimental configuration. For the experiments considered in this work, we have only a few experimental and simulated configurations, allowing for easy validation by a human expert. We did not find better agreement between corresponding experimental and simulated configurations. The differences which we could see between experiment and simulation in this case are not connected with predictions of the CNN model and are more likely related to the accuracy of computationally relaxed configurations. Nevertheless, in the next sections we consider possible small adjustments to the accuracy of matching.

Section S3. Effect of small perturbations on AFM imaging and matching

To take account of possible tip- and surface-induced perturbations in the imaging process, we tried to address possible variations of molecular configuration for the case of our benchmark molecule 1S-camphor in two ways: 1) Tilting of the whole molecule as a rigid body and 2) rotation of $-CH_3$ groups. We found that both these perturbations quite significantly affect the simulated AFM images and associated configuration predictions.

S3.1 Effect of molecule tilting

At first we consider angular tilting of the molecule from model configurations by 0-5° from normal to the Cu(111) surface. The resultant images are presented in Fig. S6 – there are 5 different tilt variations of the same simulated molecule configuration. Even such small angular tilting of the molecule from normal to the surface significantly affects the AFM images. This follows from the nature of the AFM imaging process, where the tip ”scratches” the surface of the molecule and upper atoms dominate the interaction. Even a small height change of the top-most atoms, such as 50 pm, can have visible impact on the simulated AFM image. Image descriptors of simulated configuration also reflect these variations to a degree, but they are not as obvious as in the AFM data. For close configurations, the image descriptors are also similar and their features even allow estimates for the direction of molecular tilting. This feature emphasizes that image descriptors are a more suitable representation for machine learning than raw AFM images.

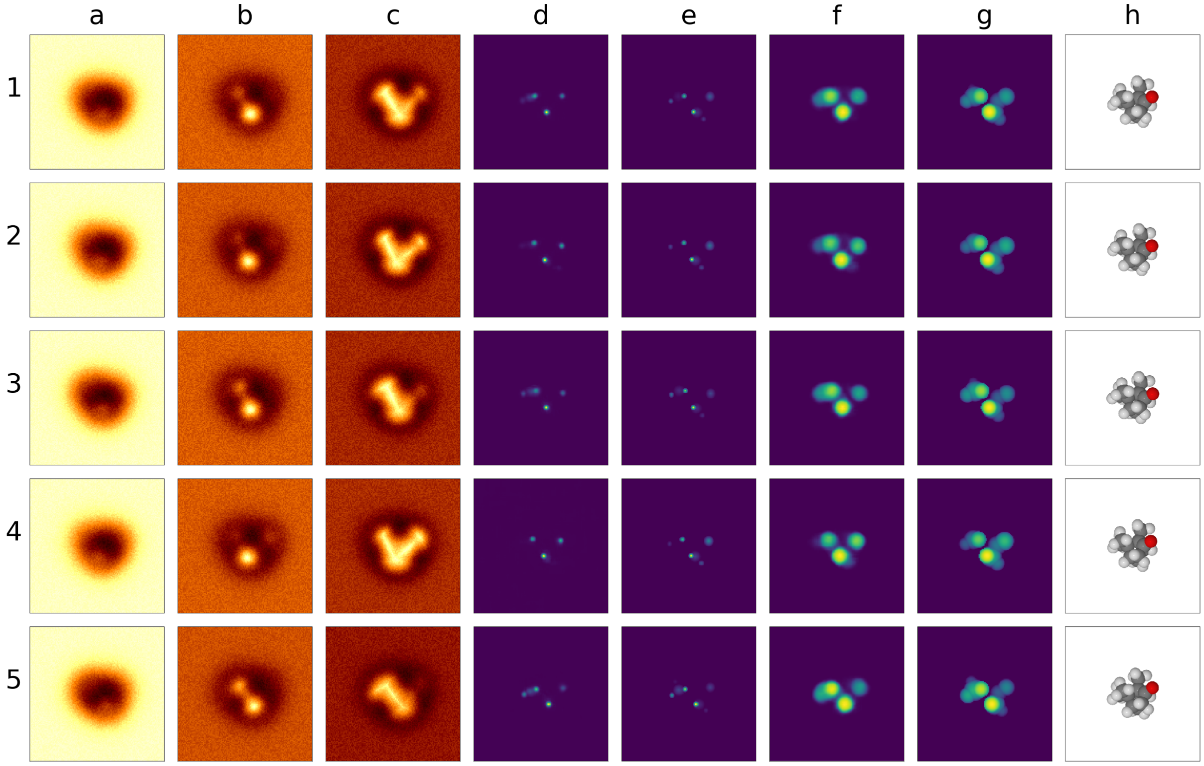

Fig. S6. **Effect of tilt of molecules on simulated AFM images 1 to 5.** Simulated configuration tilted with a small random angle from the normal to the Cu(111) surface; **a-c** 3 selected simulated AFM images: **a** – far, **b** – middle, **c** – close tip-sample distance; **d** - atomic disks representation as predicted by the CNN from a given set of simulated AFM images; **e** - atomic disks representation as a reference; **f** - vdW-spheres representation as predicted by the CNN; **g** - vdW-spheres representation as a reference. **h** Full molecular geometry corresponding to simulated AFM images.

S3.2 Effect of -CH_3 rotations

For simplicity we assumed that the geometry of the molecule is rigid, i.e. that internal degrees of freedom are frozen. While this is typically true for flat aromatic molecules, in the case of 3D aliphatic molecules, the barriers for rotation around a single-bond between two carbon atoms is of the order of just 10 kJ/mol, therefore it can be induced during deposition or scanning. Hence, we considered the possible impact of such rotations in the case of 1S-Camphor. In this molecule there are 3 -CH_3 groups capable of such rotations. AFM images are very sensitive to changes of atomic positions of the closest atoms. Therefore, despite not affecting the global minimum energy configuration of the molecule, the effects of rotating a -CH_3 can be crucial. Examples of changes are presented in Fig. S7, where we considered one simulated Camphor configuration. This configuration has 3 different -CH_3 groups sticking up (see upper right corner on Fig. S7) that could be rotated independently, and the result of the rotations on images and descriptors is shown in rows (1-5).

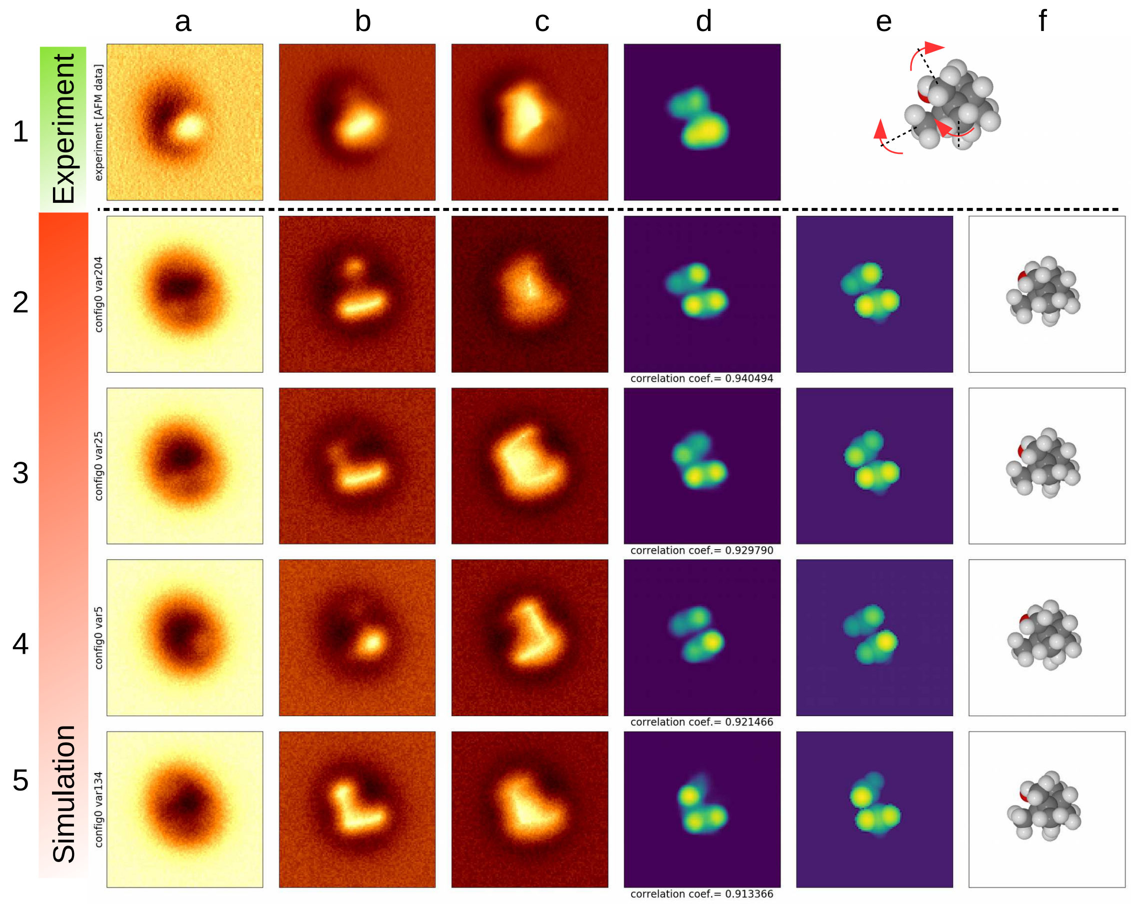

Fig. S7. **Adjustment of simulated configuration by -CH_3 group rotations. 1** - Experimental configuration # 1; **2-5** - Simulated configuration with different rotations of the -CH_3 groups. **a-c** selected AFM images: **a** – far, **b** – middle, **c** – close tip-sample distance; **d** - vdW-Spheres representation as predicted by the CNN and **e** - references. **f** Full molecular geometry corresponding to simulated AFM data.

S3.3 Ambiguity of molecular orientation

For a 3D molecule such as 1S-Camphor, the AFM tip is able to directly probe just 2-3 surface atoms. This limited information is not sufficient to always reliably discriminate a single orientation of the molecule. In particular, it may easily happen that a similar doublet or triplet of atoms is present on several places of the molecular surface. In

Fig. S8 we present 5 best matches from 500 uniformly distributed rotations to a single experimental stack. We found that two very distinct configurations (row 3,4,5 vs. 2,6) both match experiment with a linear correlation coefficient of the vdW-Spheres$>$0.92. Note that configuration in row 3 is close to a symmetric rotation of 180 degrees of the configurations in rows 4 and 5. For more specific molecular identification, either further experimental configurations are required (as in this work) or further refinement requires input from multiple descriptors in parallel. However, even if this was the only data available, the possible configurations have still been greatly reduced with very little computational effort.

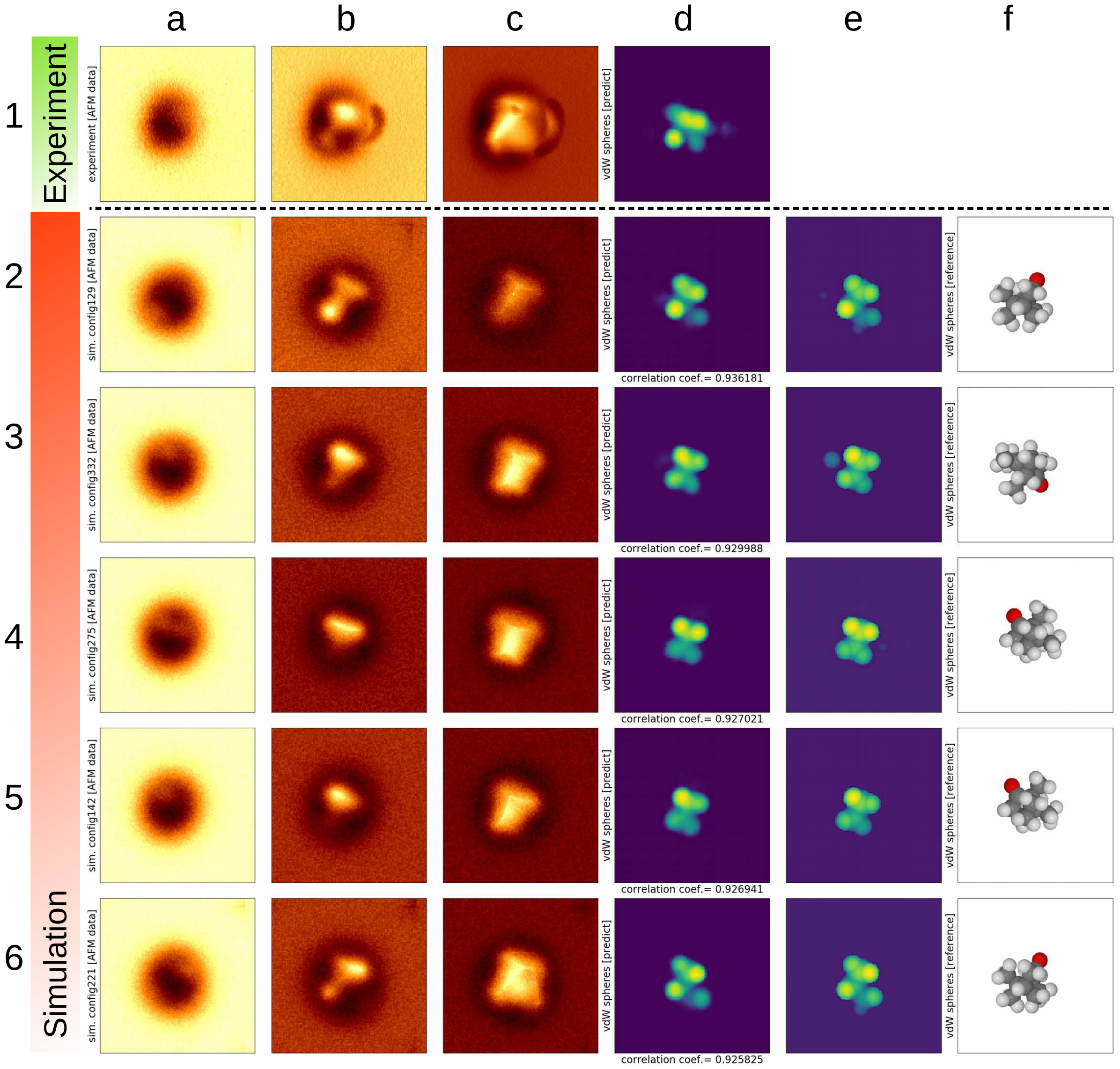

Fig. S8. **Matching experimental configuration 2 of 1*S*-camphor with the closest simulated configurations.** 1 - Experimental configuration # 3; **2-6** Distinct simulated configurations ordered by descending similarity of vdW-Spheres; **a-c** selected AFM images: **a** – far, **b** – middle, **c** – close tip-sample distance; **d** – vdW-Spheres representation as predicted by the CNN and **e** references. **f** Full molecular geometry corresponding to simulated AFM images.

Section S4. Neural network architecture

To predict 2D image representations from a stack of AFM images we used a convolutional neural network (CNN) approach. Our model is implemented in Keras (*75*) with a Tensorflow (*76*) backend. The network is similar to encoder-decoder type networks (*77*) and equipped with 12 convolutional layers. The input AFM data size was (128x128x10) and the output image representation was (128x128). The range in z is limited by the induced instabilities at close approach and the usefulness of contrast features at longer distance, and we found that we have a "useful" actual distance range of 1-1.5 Å and the contrast evolves rather smoothly over this range in 10 steps. A schematic view of the network is presented in Fig. S9. Initially the AFM data stack contains features that have lateral (x,y) and depth (z) dependencies, so we use 3D convolutional layers for data processing ('3D Conv' on Fig. S9). The model has three 3D convolution layers each followed by an average pooling layer that reduces the size of the feature maps by a factor of 2 in each dimension, except for the last pooling layer which does not reduce the size in the z-direction. The motivation for this is to avoid reducing the size in the z-direction too much before moving to 2D feature maps. We also tried max pooling layers, but found average pooling to work better, possibly due to max pooling losing too much information. The transition to 2D happens by reshaping the 3D image slices into channels of a 2D feature map. Following this, a pair of 3x3 2D convolutions were applied.

The feature maps are upsampled in three stages, as in the downsampling. The upsampling is done by simply using nearest neighbour interpolation followed by two 2D convolutions. The final convolution serves to reduce the number of feature maps to one. In earlier attempts we tried using transposed convolutions for the upsampling, but found that output density maps contained patterned artifacts. This has been known to be a problem with transposed convolutions (*78*). Another related method is the subpixel convolution (*79*), which could mitigate the effect of the artifacts (*80*).

All the convolutional layers use a $\mathrm{LeakyReLU}_{0.1}$ activation, except the last one which has a ReLU activation. We tried using ReLU in all layers, but this resulted in "dead" filters that gave zero or nearly zero output everywhere, and we found that $\mathrm{LeakyReLU}$ yielded better results in practice. Using regular ReLU for the last layer is a natural choice since the output has a clear zero point at the cut-off height that we choose. The ReLU and $\mathrm{LeakyReLU}$

functions are defined as

$$\mathrm{ReLU}(z) = \max(0, z) \tag{S1}$$

$$\mathrm{LeakyReLU}_{\alpha}(z) = \begin{cases} z & \text{if } z \geq 0 \\ \alpha z & \text{if } z < 0 \end{cases} \tag{S2}$$

The loss function was the mean squared error. The optimizer for the gradient descent was the Adaptive Moment Estimation (Adam) (*81*) optimizer. We set the learning rate to 0.001 and the decay to 10^{-5}, and otherwise we use the default parameters as defined in Keras.

The model was trained separately for the three different representations on two different data sets. The first data set contains the elements H, C, N, O, and F (see Fig. S4), and the second data set was extended to additionally contain Si, P, S, Cl, and Br. Table S1 lists the final losses on the trained models and Fig. S10 shows the losses as functions of training epochs. The loss has the interpretation of being the square of the average error in height in Å. The loss on the training and test sets are almost the same in every case except on the second data set with atomic disks. From this we can conclude that the model is not badly overfitting to the training data. The losses on the second data set are greater than on the first one, as would be expected since the data has larger variance.

Table S1. Losses on the training and test sets for the trained models.

	Data Set 1			Data Set 2		
	Atomic Disks	vdW-Spheres	Height Map	Atomic Disks	vdW-Spheres	Height Map
Train	3.47×10^{-4}	3.79×10^{-3}	4.01×10^{-3}	3.83×10^{-4}	3.98×10^{-3}	4.49×10^{-3}
Test	3.50×10^{-4}	3.79×10^{-3}	3.97×10^{-3}	4.41×10^{-4}	4.16×10^{-3}	4.57×10^{-3}

Table S2. **Model architecture**. The factors in parentheses denote parallel layers for separate output branches. The total number of parameters is 122,811.

	Layer type	Output dimension	Kernel size	Activation	Parameters
0	Input	$128 \times 128 \times 10 \times 1$	-	-	-
1	3D conv	$128 \times 128 \times 10 \times 4$	$3 \times 3 \times 3$	LeakyReLU$_{0.1}$	112
2	Avg pool	$64 \times 64 \times 5 \times 4$	$2 \times 2 \times 2$	-	-
3	3D conv	$64 \times 64 \times 2 \times 8$	$3 \times 3 \times 3$	LeakyReLU$_{0.1}$	872
4	Avg pool	$32 \times 32 \times 2 \times 8$	$2 \times 2 \times 2$	-	-
5	3D conv	$32 \times 32 \times 2 \times 16$	$3 \times 3 \times 3$	LeakyReLU$_{0.1}$	3472
6	Avg pool	$16 \times 16 \times 2 \times 16$	$2 \times 2 \times 1$	-	-
7	Reshape to 2D	$16 \times 16 \times 32$	-	-	-
8	2D conv	$16 \times 16 \times 64$	3×3	LeakyReLU$_{0.1}$	18496
9	2D conv	$16 \times 16 \times 64$	3×3	LeakyReLU$_{0.1}$	36928
10	NN-upsample	$32 \times 32 \times 64(\times 3)$	-	-	-
11	2D conv	$32 \times 32 \times 16(\times 3)$	3×3	LeakyReLU$_{0.1}$	$9232(\times 3)$
12	2D conv	$32 \times 32 \times 16(\times 3)$	3×3	LeakyReLU$_{0.1}$	$2320(\times 3)$
13	NN-upsample	$64 \times 64 \times 16(\times 3)$	-	-	-
14	2D conv	$64 \times 64 \times 16(\times 3)$	3×3	LeakyReLU$_{0.1}$	$2320(\times 3)$
15	2D conv	$64 \times 64 \times 16(\times 3)$	3×3	LeakyReLU$_{0.1}$	$2320(\times 3)$
16	NN-upsample	$128 \times 128 \times 16(\times 3)$	-	-	-
17	2D conv	$128 \times 128 \times 16(\times 3)$	3×3	LeakyReLU$_{0.1}$	$2320(\times 3)$
18	2D conv	$128 \times 128 \times 16(\times 3)$	3×3	LeakyReLU$_{0.1}$	$2320(\times 3)$
19	2D conv	$128 \times 128 \times 1(\times 3)$	3×3	None/ReLU/ReLU	$145(\times 3)$

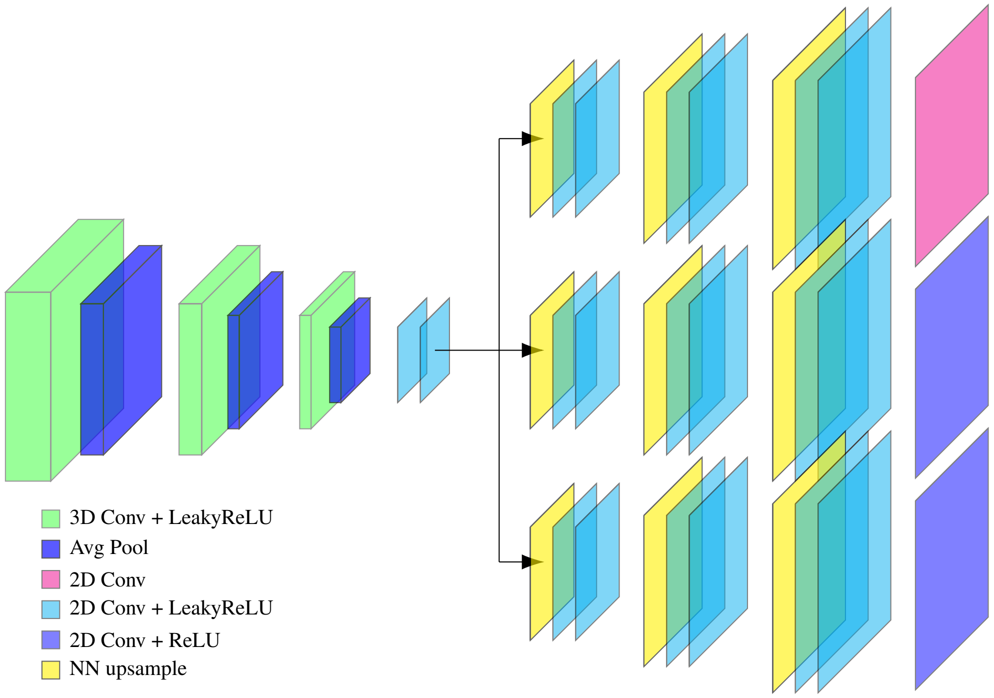

Fig. S9. **Illustration of the layers of the CNN model**. The forward direction is from left to right. The sizes of the layers represent the relative size of the feature maps. Not to scale.

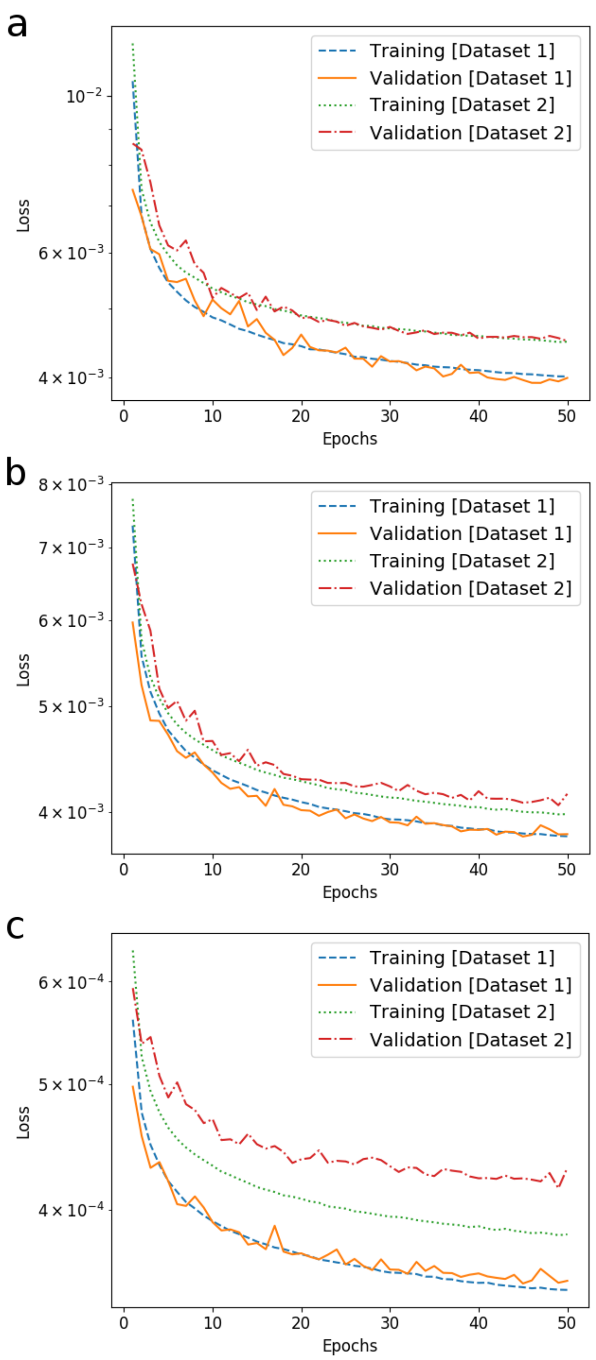

Fig. S10. **The mean squared loss for height maps, vdW-Spheres, and atomic disks.** The mean squared loss on the training and validation sets as functions of the number of training epochs **a** for the height map prediction, **b** for the vdW-Spheresprediction and **c** for the atomic disks prediction.

Section S5. PP simulations

For each molecule, the Lennard-Jones and Electrostatic force field was pre-calculated on a regular cubic real-space grid of size $30 \times 30 \times 30$ Å with spacing 0.1 Å in each direction and stored in a floating point 3D texture on a GPU. The Lennard-Jones field is calculated using standard OPLS parameters listed in table S3 (there is no re-fitting of this force field within our approach). The electrostatic field is calculated from Mulliken charges taken from the quantum chemistry simulations (note that the use of high-level couple cluster calculations is not important for getting reasonable charges, but offers high accuracy charge densities for benchmarking and future applications).

The weighting of molecular orientations in the database mentioned in the main text is done automatically by sorting orientations with respect to distance-weighted counting of atoms using the function $S = \sum_i \exp(-\beta(z_i - z_{close}))$. z_i is the z-coordinate of atom i in the coordinate system of the current scan, z_{close} corresponds to the closest atom and the decay factor β is currently set to 1.0 [Å^{-1}]. For a constant number of atoms, the function is clearly maximal when all z_i are similar.

In the process of the neural network training, we vary some simulation parameters in order to regularize the training and to make it less dependent on a particular setup, therefore more robust with respect to uncertain experimental conditions. Predominantly we vary the equilibrium tilt of the probe particle in the range ± 1.0Å, reflecting the asymmetric absorption of CO typical in experiment. We also vary the charge as well as Lennard-Jones radius and binding energy by 0.1e, 0.2 Å and 5% respectively.

Table S3. Lennard-Jones parameters in PP simulation and rigid body relaxation of surface.

type	R_{ii}[Å]	E_{ii}[eV]
H	1.4870	0.000681
C	1.9080	0.003729
N	1.7800	0.007372
O	1.6612	0.009106
S	2.0000	0.010841
Cu	2.2300	0.010000