SUPPORTING INFORMATION

Content

S_movie_1: a) movie showing fast growth of cancinite columns along the c-direcion of the crystal on the (100) face. Bridging of these cancritine colums results in the formation of the large 12-ring pores (see figure 14 for details)

b) movie showing overall crystal growth of zeolite L hexagonal prisms (see figure 14 for details)

c) movie showing fractal-like growth on the (001) face of zeolite L (see figure 14 for details)

S_movie_2: L-cystine growth via a screw dislocation exhibiting a pin-wheel structure on the (001) face with (right) and without (left) the presence of a growth modifier. The growth modifier is shown as the dark spheres (see figure 15 for details)

S_movie_3: Growth of MOF-5 starting under conditions of excess linker then switching to conditions of excess metal (see figure 15 for details)

S_movie_4: Growth of adipic acid illustrating both crystal habit and surface topography (see figure 15 for details)

S_movie_5: a) Screw dislocation on the (100) surface of SAV showing detailed structure of the interleaved spiral (see figure 16 for details)

b) Impurity intergrowth structure of AEI that occurs on the (100) facet of STA-7 at the latter stages of crystal growth when growth nutrient is exhausted in the reaction mixture (see figure 16 for details)

S_movie_6: a) Effect of increasing the supersaturation on the surface nucleation on different facets of zeolite A (see figure 19 for details)

b) Zeolite L grown with increasing supersaturation left to right. The left image is close to equilibrium and shows a similar morphology to the crystal grown at high supersaturation. The crystal habit changes at intermediate supersaturation

S1 Further Natural Tiling Information

No. of Closed Tiles	Structure Types
1	ACO, ASV, ATN, BEC, CAN, *CTH, EON, EZT, HEU, ISV, ITG, ITH, ITR, ITT, -ITV, ITW, IWR, IWS, IWV, IWW, MSE, MWW, POR, POS, SAO, SEW, SFF, SFG, SOF, SOR, SOS, SOV, SSF, STI, STW, *-SVY, SZR, UOS, UOV, USI, UTL, UWY, YFI **(43)**
2	GME, -IFT, -IFU, IFW, IRR, MAZ, MER, MSO, -SYT, *UOE, UOZ, -WEN **(12)**
3	-IRY, LTF, LTL, MWF, OFF, PAU, PWN, SBE, SBS, SBT **(10)**
4	-CLO, DFO, MOZ **(3)**

Table S1: *A statistical breakdown of the number of closed tile types found in frameworks composed of a mixture of open and closed tiles. The maximum number of closed tile types found in mixed cages is four. Three letter codes for all mixed zeolite frameworks are listed, along with a sum of the number of frameworks for each entry in parentheses. The total number of tile types in each framework varies substantially with a minimum of two and a maximum of sixteen tile types.*

An average taken over all frameworks displayed here results in a ratio of 2.31:1 for open:closed tiles in 68 mixed frameworks. This is slightly higher than (but comparable to) the 2.25:1 ratio seen when counting open and closed tiles across all zeolite frameworks. Two percentages were calculated for each zeolite structure to obtain these ratios: the percentage of the tiles in the structure that were open or closed tiles. As these are calculated for each structure individually by dividing the number of open or closed tiles by the total number of tile types in the system, structures with a large number of tile types do not skew the result. The ratio was then calculated between the averages of these percentages across all frameworks, and all mixed frameworks.

S2 Structure file use in CrystalGrower

```
249:LTA/lta/Systre;PPT 1

5

t-cub 1 (8,3) 6

5(-1,0,0) 5(-1,0,-1) 4(0,1,0) 5 5(0,0,-1) 4

Si 3[4] 4[1] 5(0,0,-1)[1] 5[1]
Si 3[4] 4[1] 5(0,0,-1)[1] 5(-1,0,-1)[1]
Si 3[4] 4[1] 5(-1,0,-1)[1] 5(-1,0,0)[1]
Si 3[4] 4[1] 5[1] 5(-1,0,0)[1]
Si 3[4] 4(0,1,0)[1] 5(0,0,-1)[1] 5[1]
Si 3[4] 4(0,1,0)[1] 5(0,0,-1)[1] 5(-1,0,-1)[1]
Si 3[4] 4(0,1,0)[1] 5[1] 5(-1,0,0)[1]
Si 3[4] 4(0,1,0)[1] 5(-1,0,-1)[1] 5(-1,0,0)[1]

t-cub 2 (8,3) 6

5(0,0,-1) 5(0,-1,-1) 4(1,0,0) 5 5(0,-1,0) 4

Si 3[4] 4[1] 5(0,-1,0)[1] 5[1]
Si 3[4] 4[1] 5(0,-1,-1)[1] 5(0,-1,0)[1]
Si 3[4] 4[1] 5(0,0,-1)[1] 5(0,-1,-1)[1]
Si 3[4] 4[1] 5(0,0,-1)[1] 5[1]
Si 3[4] 4(1,0,0)[1] 5(0,-1,0)[1] 5[1]
Si 3[4] 4(1,0,0)[1] 5(0,-1,-1)[1] 5(0,-1,0)[1]
Si 3[4] 4(1,0,0)[1] 5(0,0,-1)[1] 5[1]
Si 3[4] 4(1,0,0)[1] 5(0,-1,-1)[1] 5(0,0,-1)[1]

t-cub 3 (8,3) 6

5(0,-1,0) 5(-1,-1,0) 4(0,0,1) 5 5(-1,0,0) 4

Si 3[4] 4[1] 5[1] 5(-1,0,0)[1]
Si 3[4] 4[1] 5(-1,-1,0)[1] 5(-1,0,0)[1]
Si 3[4] 4[1] 5(0,-1,0)[1] 5(-1,-1,0)[1]
Si 3[4] 4[1] 5[1] 5(0,-1,0)[1]
Si 3[4] 4(0,0,1)[1] 5[1] 5(-1,0,0)[1]
Si 3[4] 4(0,0,1)[1] 5(-1,-1,0)[1] 5(-1,0,0)[1]
Si 3[4] 4(0,0,1)[1] 5[1] 5(0,-1,0)[1]
Si 3[4] 4(0,0,1)[1] 5(0,-1,0)[1] 5(-1,-1,0)[1]

t-toc 4 (24,3) 14

1(0,-1,0) 2 1 5(0,0,-1) 5(0,-1,-1) 5(-1,-1,-1) 3(0,0,-1) 5(-1,0,-1) 2(-1,0,0) 5(-
1,-1,0) 5(0,-1,0) 5 5(-1,0,0) 3

Si 3[4] 3[1] 5[1] 5(-1,0,0)[1]
Si 3[4] 3[1] 5(-1,-1,0)[1] 5(-1,0,0)[1]
Si 3[4] 3[1] 5(0,-1,0)[1] 5(-1,-1,0)[1]
Si 3[4] 3[1] 5[1] 5(0,-1,0)[1]
Si 3[4] 1[1] 5[1] 5(-1,0,0)[1]
Si 3[4] 1[1] 5(-1,0,-1)[1] 5(-1,0,0)[1]
Si 3[4] 2(-1,0,0)[1] 5(-1,0,-1)[1] 5(-1,0,0)[1]
```

Si 3[4] 2(-1,0,0)[1] 5(-1,-1,0)[1] 5(-1,0,0)[1]
Si 3[4] 2[1] 5(0,-1,0)[1] 5[1]
Si 3[4] 2[1] 5(0,0,-1)[1] 5[1]
Si 3[4] 1[1] 5(0,0,-1)[1] 5[1]
Si 3[4] 1(0,-1,0)[1] 5(0,-1,0)[1] 5(-1,-1,0)[1]
Si 3[4] 1(0,-1,0)[1] 5(0,-1,-1)[1] 5(0,-1,0)[1]
Si 3[4] 2[1] 5(0,-1,-1)[1] 5(0,-1,0)[1]
Si 3[4] 2(-1,0,0)[1] 5(-1,-1,-1)[1] 5(-1,-1,0)[1]
Si 3[4] 1(0,-1,0)[1] 5(-1,-1,-1)[1] 5(-1,-1,0)[1]
Si 3[4] 2(-1,0,0)[1] 5(-1,-1,-1)[1] 5(-1,0,-1)[1]
Si 3[4] 3(0,0,-1)[1] 5(0,0,-1)[1] 5(-1,0,-1)[1]
Si 3[4] 3(0,0,-1)[1] 5(-1,-1,-1)[1] 5(-1,0,-1)[1]
Si 3[4] 1[1] 5(0,0,-1)[1] 5(-1,0,-1)[1]
Si 3[4] 3(0,0,-1)[1] 5(0,-1,-1)[1] 5(-1,-1,-1)[1]
Si 3[4] 3(0,0,-1)[1] 5(0,0,-1)[1] 5(0,-1,-1)[1]
Si 3[4] 1(0,-1,0)[1] 5(0,-1,-1)[1] 5(-1,-1,-1)[1]
Si 3[4] 2[1] 5(0,0,-1)[1] 5(0,-1,-1)[1]

t-grc 5 (48,3) 26

4 2(0,1,1) 4(1,1,1) 1(1,0,1) 2(0,0,1) 4(1,0,1) 5(0,-1,0) 2 1(1,0,0) 4(1,0,0)
3(1,0,0) 5(1,0,0) 3(1,1,0) 4(1,1,0) 5(0,1,0) 2(0,1,0) 5(0,0,-1) 1 4(0,1,0)
3(0,1,0) 4(0,1,1) 5(0,0,1) 1(0,0,1) 4(0,0,1) 5(-1,0,0) 3

Si 3[4] 3[1] 4[1] 5(-1,0,0)[1]
Si 3[4] 3[1] 4(0,0,1)[1] 5(-1,0,0)[1]
Si 3[4] 3[1] 4(0,0,1)[1] 5(0,-1,0)[1]
Si 3[4] 3[1] 4[1] 5(0,-1,0)[1]
Si 3[4] 1(0,0,1)[1] 4(0,0,1)[1] 5(-1,0,0)[1]
Si 3[4] 1(0,0,1)[1] 4(0,1,1)[1] 5(-1,0,0)[1]
Si 3[4] 3(0,1,0)[1] 4(0,1,1)[1] 5(-1,0,0)[1]
Si 3[4] 3(0,1,0)[1] 4(0,1,0)[1] 5(-1,0,0)[1]
Si 3[4] 1[1] 4(0,1,0)[1] 5(-1,0,0)[1]
Si 3[4] 1[1] 4[1] 5(-1,0,0)[1]
Si 3[4] 1(0,0,1)[1] 4(0,0,1)[1] 5(0,0,1)[1]
Si 3[4] 2(0,0,1)[1] 4(0,0,1)[1] 5(0,0,1)[1]
Si 3[4] 2(0,0,1)[1] 4(0,0,1)[1] 5(0,-1,0)[1]
Si 3[4] 1(0,0,1)[1] 4(0,1,1)[1] 5(0,0,1)[1]
Si 3[4] 2(0,1,1)[1] 4(0,1,1)[1] 5(0,0,1)[1]
Si 3[4] 2(0,1,1)[1] 4(1,1,1)[1] 5(0,0,1)[1]
Si 3[4] 1(1,0,1)[1] 4(1,1,1)[1] 5(0,0,1)[1]
Si 3[4] 1(1,0,1)[1] 4(1,0,1)[1] 5(0,0,1)[1]
Si 3[4] 2(0,0,1)[1] 4(1,0,1)[1] 5(0,0,1)[1]
Si 3[4] 3(0,1,0)[1] 4(0,1,1)[1] 5(0,1,0)[1]
Si 3[4] 2(0,1,1)[1] 4(0,1,1)[1] 5(0,1,0)[1]
Si 3[4] 3(0,1,0)[1] 4(0,1,0)[1] 5(0,1,0)[1]
Si 3[4] 1[1] 4(0,1,0)[1] 5(0,0,-1)[1]
Si 3[4] 2(0,1,0)[1] 4(0,1,0)[1] 5(0,0,-1)[1]
Si 3[4] 2(0,1,0)[1] 4(0,1,0)[1] 5(0,1,0)[1]
Si 3[4] 1[1] 4[1] 5(0,0,-1)[1]
Si 3[4] 2(0,1,0)[1] 4(1,1,0)[1] 5(0,0,-1)[1]
Si 3[4] 1(1,0,0)[1] 4(1,1,0)[1] 5(0,0,-1)[1]
Si 3[4] 1(1,0,0)[1] 4(1,0,0)[1] 5(0,0,-1)[1]
Si 3[4] 2[1] 4(1,0,0)[1] 5(0,0,-1)[1]
Si 3[4] 2[1] 4[1] 5(0,0,-1)[1]
Si 3[4] 2(0,1,0)[1] 4(1,1,0)[1] 5(0,1,0)[1]
Si 3[4] 3(1,1,0)[1] 4(1,1,0)[1] 5(0,1,0)[1]
Si 3[4] 3(1,1,0)[1] 4(1,1,1)[1] 5(0,1,0)[1]
Si 3[4] 2(0,1,1)[1] 4(1,1,1)[1] 5(0,1,0)[1]

```
Si 3[4] 3(1,1,0)[1] 4(1,1,0)[1] 5(1,0,0)[1]
Si 3[4] 1(1,0,0)[1] 4(1,1,0)[1] 5(1,0,0)[1]
Si 3[4] 3(1,1,0)[1] 4(1,1,1)[1] 5(1,0,0)[1]
Si 3[4] 3(1,0,0)[1] 4(1,0,0)[1] 5(1,0,0)[1]
Si 3[4] 3(1,0,0)[1] 4(1,0,1)[1] 5(1,0,0)[1]
Si 3[4] 1(1,0,1)[1] 4(1,0,1)[1] 5(1,0,0)[1]
Si 3[4] 1(1,0,1)[1] 4(1,1,1)[1] 5(1,0,0)[1]
Si 3[4] 1(1,0,0)[1] 4(1,0,0)[1] 5(1,0,0)[1]
Si 3[4] 3(1,0,0)[1] 4(1,0,0)[1] 5(0,-1,0)[1]
Si 3[4] 3(1,0,0)[1] 4(1,0,1)[1] 5(0,-1,0)[1]
Si 3[4] 2[1] 4(1,0,0)[1] 5(0,-1,0)[1]
Si 3[4] 2[1] 4[1] 5(0,-1,0)[1]
Si 3[4] 2(0,0,1)[1] 4(1,0,1)[1] 5(0,-1,0)[1]

 11.9190  11.9190  11.9190 P
 90.000  90.000  90.000

Non primitive data
 11.9190  11.9190  11.9190
 90.000  90.000  90.000

t-cub 1 8/6
1  0.18230  0.36840  0.00000
2  0.00000  0.36840 -0.18230
3 -0.18230  0.36840  0.00000
4  0.00000  0.36840  0.18230
5  0.18230  0.63160  0.00000
6  0.00000  0.63160 -0.18230
7  0.00000  0.63160  0.18230
8 -0.18230  0.63160  0.00000

 8 3 4 7
 3 8 6 2
 8 6 5 7
 1 5 7 4
 5 1 2 6
 1 2 3 4

t-cub 2 8/6
1  0.36840  0.00000  0.18230
2  0.36840 -0.18230  0.00000
3  0.36840  0.00000 -0.18230
4  0.36840  0.18230  0.00000
5  0.63160  0.00000  0.18230
6  0.63160 -0.18230  0.00000
7  0.63160  0.18230  0.00000
8  0.63160  0.00000 -0.18230

 8 3 4 7
 3 8 6 2
 8 6 5 7
 1 5 7 4
 5 1 2 6
 1 2 3 4

t-cub 3 8/6
1  0.00000  0.18230  0.36840
2 -0.18230  0.00000  0.36840
3  0.00000 -0.18230  0.36840
```

```
4  0.18230  0.00000  0.36840
5  0.00000  0.18230  0.63160
6 -0.18230  0.00000  0.63160
7  0.18230  0.00000  0.63160
8  0.00000 -0.18230  0.63160

 8 3 4 7
 3 8 6 2
 8 6 5 7
 1 5 7 4
 5 1 2 6
 1 2 3 4

t-toc 4 24/14
1  0.00000  0.18230  0.36840
2 -0.18230  0.00000  0.36840
3  0.00000 -0.18230  0.36840
4  0.18230  0.00000  0.36840
5  0.00000  0.36840  0.18230
6 -0.18230  0.36840  0.00000
7 -0.36840  0.18230  0.00000
8 -0.36840  0.00000  0.18230
9  0.36840  0.00000  0.18230
10  0.36840  0.18230  0.00000
11  0.18230  0.36840  0.00000
12  0.00000 -0.36840  0.18230
13  0.18230 -0.36840  0.00000
14  0.36840 -0.18230  0.00000
15 -0.36840 -0.18230  0.00000
16 -0.18230 -0.36840  0.00000
17 -0.36840  0.00000 -0.18230
18  0.00000  0.18230 -0.36840
19 -0.18230  0.00000 -0.36840
20  0.00000  0.36840 -0.18230
21  0.00000 -0.18230 -0.36840
22  0.18230  0.00000 -0.36840
23  0.00000 -0.36840 -0.18230
24  0.36840  0.00000 -0.18230

 16 23 13 12
 9 14 24 10
 11 20 6 5
 22 18 20 11 10 24
 21 22 24 14 13 23
 19 21 23 16 15 17
 21 19 18 22
 18 19 17 7 6 20
 8 7 17 15
 3 2 8 15 16 12
 4 3 12 13 14 9
 1 4 9 10 11 5
 2 1 5 6 7 8
 1 2 3 4

t-grc 5 48/26
1  0.00000  0.18230  0.36840
2  0.00000  0.18230  0.63160
3  0.18230  0.00000  0.63160
4  0.18230  0.00000  0.36840
```

```
5  0.00000  0.36840  0.81770
6  0.00000  0.63160  0.81770
7  0.00000  0.81770  0.63160
8  0.00000  0.81770  0.36840
9  0.00000  0.63160  0.18230
10  0.00000  0.36840  0.18230
11  0.18230  0.36840  1.00000
12  0.36840  0.18230  1.00000
13  0.36840  0.00000  0.81770
14  0.18230  0.63160  1.00000
15  0.36840  0.81770  1.00000
16  0.63160  0.81770  1.00000
17  0.81770  0.63160  1.00000
18  0.81770  0.36840  1.00000
19  0.63160  0.18230  1.00000
20  0.18230  1.00000  0.63160
21  0.36840  1.00000  0.81770
22  0.18230  1.00000  0.36840
23  0.18230  0.63160  0.00000
24  0.36840  0.81770  0.00000
25  0.36840  1.00000  0.18230
26  0.18230  0.36840  0.00000
27  0.63160  0.81770  0.00000
28  0.81770  0.63160  0.00000
29  0.81770  0.36840  0.00000
30  0.63160  0.18230  0.00000
31  0.36840  0.18230  0.00000
32  0.63160  1.00000  0.18230
33  0.81770  1.00000  0.36840
34  0.81770  1.00000  0.63160
35  0.63160  1.00000  0.81770
36  1.00000  0.81770  0.36840
37  1.00000  0.63160  0.18230
38  1.00000  0.81770  0.63160
39  1.00000  0.18230  0.36840
40  1.00000  0.18230  0.63160
41  1.00000  0.36840  0.81770
42  1.00000  0.63160  0.81770
43  1.00000  0.36840  0.18230
44  0.81770  0.00000  0.36840
45  0.81770  0.00000  0.63160
46  0.63160  0.00000  0.18230
47  0.36840  0.00000  0.18230
48  0.63160  0.00000  0.81770

 1 4 47 31 26 10
 21 35 16 15
 34 38 42 17 16 35
 18 17 42 41
 48 13 12 19
 40 45 48 19 18 41
 47 46 44 45 48 13 3 4
 47 46 30 31
 28 29 43 37
 44 39 43 29 30 46
 40 39 44 45
 39 40 41 42 38 36 37 43
 36 38 34 33
 36 33 32 27 28 37
```

```
25 32 33 34 35 21 20 22
32 25 24 27
26 23 24 27 28 29 30 31
26 23 9 10
22 8 9 23 24 25
7 8 22 20
7 20 21 15 14 6
11 14 15 16 17 18 19 12
14 11 5 6
3 2 5 11 12 13
1 2 5 6 7 8 9 10
1 2 3 4
```

Figure S1: *An example of a crystal structure file used in CrystalGrower for the LTA zeolite framework. The structure data are shown for only one tile of the five separate tiles in the LTA framework. Three tile types are present (**t-cub**, **t-toc** and **t-grc**) however, **t-cub** exists in three symmetry positions in the crystal structure, with each position shown as separate tiles in the structure file to give a total of five tiles.*

Structure name

Number of separate tiles in structure (tile number, not type)

Tile type Tile number (Number of vertices with Q number, Q number of said vertices) -Repeat for all Q numbers present in tile- Number of neighbouring tiles through faces

Tile number of neighbouring tile through face (unit cell X, unit cell Y, unit cell Z) – position of unit cell containing neighbouring tile relative to current tile's unit cell, absent if within the same unit cell. –Repeat for all neighbours through faces in tile-

Tile vertex atom type Vertex Q number in isolated tile [Max Q number of vertex] Neighbouring tile number (unit cell X, unit cell Y, unit cell Z) – position of unit cell containing neighbouring tile relative to current tile's unit cell, absent if within same unit cell – [Missing condensation number filled if this neighbouring tile condensates, -Repeat for all missing bonds up to total max Q number-] -Repeat for all neighbouring tiles through faces-
–Repeat for all vertices in tile-

–Repeat for all tiles in unit cell–

Primitive unit cell a b c (4 decimal places) Unit cell type (e.g. I, F, P)
Primitive unit cell α β γ (3 decimal places)

Non primitive unit cell data
Unit cell a b c (4 decimal places)
Unit cell α β γ (3 decimal places)

Tile type Tile number Number of vertices in tile / Number of faces in tile
Vertex number Vertex X Y Z coordinates – scaled to unit cell a
-Repeat for all vertices in tile-

Vertex order in face listed by vertex number
-Repeat for all faces in tile-

Figure S2: *A general example of the crystal structure file used in CrystalGrower for a zeolite framework partitioned through natural tiling.*

25:Calcite

4

CO3 1 (3,0) 6

4(-1,0,0) 3(0,0,1) 4(0,-1,0) 3(1,0,0) 4(0,0,-1) 3(0,1,0)

O 0[2] 4(0,0,-1)[1] 3(0,1,0)[1]
O 0[2] 4(0,-1,0)[1] 3(1,0,0)[1]
O 0[2] 4(-1,0,0)[1] 3(0,0,1)[1]

CO3 2 (3,0) 6

4(1,0,0) 3(1,1,0) 4(0,1,0) 3(0,1,1) 4(0,0,1) 3(1,0,1)

O 0[2] 4(0,0,1)[1] 3(1,0,1)[1]
O 0[2] 4(0,1,0)[1] 3(0,1,1)[1]
O 0[2] 4(1,0,0)[1] 3(1,1,0)[1]

Ca 3 (1,0) 6

1(0,-1,0) 1(0,0,-1) 1(-1,0,0) 2(0,-1,-1) 2(-1,0,-1) 2(-1,-1,0)

Ca 0[6] 1(0,-1,0)[1] 1(0,0,-1)[1] 1(-1,0,0)[1] 2(0,-1,-1)[1] 2(-1,0,-1)[1] 2(-1,-1,0)[1]

Ca 4 (1,0) 6

1(1,0,0) 1(0,0,1) 1(0,1,0) 2(0,-1,0) 2(-1,0,0) 2(0,0,-1)

Ca 0[6] 1(1,0,0)[1] 1(0,0,1)[1] 1(0,1,0)[1] 2(0,-1,0)[1] 2(-1,0,0)[1] 2(0,0,-1)[1]

  6.3753   6.3753   6.3753 R
 46.078 46.078 46.078

Non primitive data
  4.9900   4.9900 17.0615
 90.000 90.000 120.000

```
CO3 1 4/3
1 C  0.25000  0.25000  0.25000
2 O  0.25000  0.50780 -0.00780
3 O  0.50780 -0.00780  0.25000
4 O -0.00780  0.25000  0.50780

 1 4
 1 3
 1 2

CO3 2 4/3
1 C  0.75000  0.75000  0.75000
2 O  0.75000  0.49220  1.00780
3 O  0.49220  1.00780  0.75000
4 O  1.00780  0.75000  0.49220

 1 4
 1 3
 1 2

Ca 3 1/0
1 Ca  0.00000  0.00000  0.00000


Ca 4 1/0
1 Ca  0.50000  0.50000  0.50000
```

Figure S3: *An example of a crystal structure file used in CrystalGrower for calcite. In the primitive unit cell for calcite, two Ca positions are present, along with two CO_3 positions. All are listed as separate species in the structure file. If a structure file is calculated in ToposPro, additional local bonding information is displayed.*

Structure name

Number of separate molecules in structure

Molecule type Molecule number (Number of atoms in molecule that can form connections to neighbours, Starting coordination number of atoms excluding intramolecular coordination -always 0-) Number of neighbouring molecules

Molecule number of neighbouring molecule (unit cell X, unit cell Y, unit cell Z) – position of unit cell containing neighbouring molecule relative to current molecule's unit cell, absent if within the same unit cell.

–Repeat for all neighbours to molecule-

Atom type Atom starting coordination number excluding intramolecular coordination -always 0- [Max coordination number of atom excluding intramolecular coordination] Neighbouring molecule number (unit cell X, unit cell Y, unit cell Z) – position of unit cell containing neighbouring molecule relative to current molecule's unit cell [Number of coordination sites filled if this neighbouring molecule is grown] -Repeat for all neighbouring molecules connected to this atom-

–Repeat for all intermolecularly bonding atoms in molecule-

THIS LOCALISED BONDING INFORMATION ONLY APPEARS IN STRUCTURE FILES GENERATED THROUGH TOPOSPRO, IT IS ABSENT IN STRUCTURE FILES GENERATED THROUGH THE CG INTERACTION PROGRAM, IT WILL BE REPLACED WITH:

C 0[0]

–Repeat for all molecules in unit cell–

Primitive unit cell a b c (4 decimal places) Unit cell type (e.g. I, F, P)

Primitive unit cell α β γ (3 decimal places)

Non primitive unit cell data

Unit cell a b c (4 decimal places)

Unit cell α β γ (3 decimal places)

Molecule type Molecule number Total number of atoms in molecule / Number of bonds in molecule

```
Atom number Atom type Atom X Y Z coordinates – scaled to unit cell a

-Repeat for all atoms in molecule-


Atom order in bond listed by atom number

-Repeat for all bonds in molecule-

----------------------------------------------------------------

For files generated with the CG interaction program, the atomic structure of
molecules is absent. It is replaced with central coordinates for each molecule to
place a sphere at the desired position.

----------------------------------------------------------------
```

Figure S4: *A general example of the crystal structure file used in CrystalGrower for an ionic / molecular framework partitioned using a simplified net / Voronoi-Dirichlet polyhedra.*

Discussion

Figures S1-S4 show examples along with general examples for crystal structure files read into CG during the initialisation stage. The values stored in the structure file allow CrystalGrower (CG) to construct a replica of the unit cell for a crystal structure and track the connectivity of all species in the structure. Numerous revisions were made to the structure file to ensure the data included could truly be applied to all crystal structures as part of a general model.

For a structure partitioned by natural tiles – beginning at the top of the file, the name of the framework is shown, matching the name of the CIF file imported into *ToposPro* to generate the structure file. The number of tiles in the system informs CG of the number of separate lists of tile vertices to follow. It is important to note here that two systems of distinguishing tile types exist: the *tile type* refers to the shape of the tile and the *tile number* refers to tiles of the same shape in different symmetry positions. This total tile number value corresponds to the total *tile number* values in the system rather than *tile type*.

Following these header values, the structure file proceeds into an individual list for each separate tile. The lists always follow the same format, although their length will differ due to each tile type possessing a different number of vertices and neighbours. Each tile list begins with two header lines. The first contains information on the tile as a whole: the tile type and tile number. This numbering system is integral to CG and the CrystalGrower Visualiser (CGV), allowing neighbours to be assigned types for connectivity in CG, and to assign separate colours in the CGV. Following these values on the same header line, the vertices within a tile are separated based on their condensation values, with a total number of vertices presented adjacent to their current condensation value (i.e. their Q^n values). The first header line ends with the number of neighbouring tiles through faces, informing CG of the maximum number of neighbours each tile can possess. The second header line for the list contains information on all the neighbouring tiles for the current tile through faces. The tile number for each neighbour is provided, along with the position of the unit cell it occupies, relative to the unit cell of the origin tile. This is absent for neighbours within the same unit cell.

Following the header files, a list for each tile vertex is provided. The atom type of each vertex is provided, allowing the future addition of energy scaling based on the atom type occupying each tile vertex (e.g. Si vs Al in zeolites). Current and maximum condensation values for each vertex are then provided, ensuring the condensation of a tile vertex cannot increase above its

maximum Q^n. The rest of the line is composed of a list of all neighbouring tiles that possess edges linking to the origin vertex. Unit cell positions relative to the unit cell containing the origin tile are once again used, ensuring connectivity between species in separate unit cells is accounted for when tiles grow.

For each neighbour connection to a tile vertex, the structure file also contains a value in square brackets. This value corresponds to the number of condensation steps that would be added to the origin tile vertex if the corresponding neighbouring tile were to grow. In the example in figure S1, the LTA structure is composed entirely of closed cages i.e. all vertices are Q^3 with a maximum condensation of Q^4. Therefore, the maximum number of condensation steps a neighbour could add would be one. For open tile systems this could extend to two, and for other framework types with higher coordination numbers (e.g. octahedral frameworks like titanosilicates), this may be higher. Following the list, the unit cell parameters for both the primitive unit and non-primitive unit cells are listed, as CG and *ToposPro* both perform calculations in the primitive unit cell to improve computational efficiency.

The entries in this list are crucial to a CG run as they are used to construct all site types in the structure, which leads to the construction of all energy ladders when combined with the energy parameters input by the user. Each list entry is also crucial to the main loop of CG, as the growth or dissolution of any tile necessitates a checking routine to assess the effect on the condensation for all neighbouring tiles.

The format of this file for crystals partitioned using a simplified net / Voronoi-Dirichlet polyhedra (VDP) is similar (figures S3 and S4), with some minor wording changes. All mention of tiles would instead correspond to an individual growth unit (e.g. a molecule or an ion), all mention of tile vertices would instead correspond to individual atoms assembling the unit and tile faces would instead refer to interactions between molecules. All mentions of the degree of condensation of a vertex would simply correspond to the coordination number of each atom assembling a molecule (ignoring intramolecular interactions)

The remaining data in the structure file are used for visualisation purposes. For natural tiles, each face is constructed by placing linked vertices at defined fractional coordinates in the order listed at the bottom of the structure file.

For molecular species, the fractional coordinates for each atom are listed as for tile vertices, however, the tile face list shown at the bottom of the structure file instead will list atoms in pairs that share a bond. Bonds will be drawn along the vector running between each set of fractional coordinates to produce a fully constructed molecule.

S3 Additional input files for CrystalGrower

```
Line 1:   Directory to store data (./Path)
Line 2:   File root to store data (File_Name)
Line 3:   Do you want to load a checkpoint to start your simulation? (yes/no)
Line 4:   File path for checkpoint load (./Path/File_Name_checkpoint.txt)
Line 5:   Do you want to save a checkpoint at the end of your simulation?
          (yes/no)
Line 6:   File path to structure file: (./Path/Structure_Name.txt)
Line 7:   Simulation operation mode? (normal/growth_modifier/ordered)
Line 8:   Screw dislocations? (yes/no)
Line 9:   How many checking sweeps, 1 is normal, 2 to clear internal
          defects?(1/2)
Line      Method for determining multipliers for combinations? (1/2)?
10:
Line      Is this a tile or net crystal? (tile / net)
11:
Line      Do you want to calculate required memory for your calculation?
12:       (yes/no)
Line      What is the maximum memory in MBytes that you wish to use? Recommended
13:       default 10000 Mbytes (integer)
Line      Temperature in Celsius? (decimal value)
14:
Line      Number of iterations? (Must be an integer)?
15:
Line      Delta mu mode? (number from 1-7)
16:
Line      Do you want an excess supersaturation of any component (always yes for
17:       a MOF)? (yes/no)
Line      Starting delta mu [kcal/mol]? (decimal value)
18:
Line      MOVIE: The number of frames? (must be an integer)
19:
Line      MOVIE: Iteration at movie capture start? (default is 1) (must be an
20:       integer)
Line      MOVIE: Iteration at movie capture end? (default is number of
21:       iterations) (must be an integer)
Line      Delta mu DATA: The number of outputs? (must be an integer)
22:
Line      Delta mu DATA: Iteration at initial output? (default is 1) (must be an
23:       integer)
Line      Delta mu DATA: Iteration at final output? (default is number of
24:       iterations) (must be an integer)
Line      Do you want to visualise crystal terraces? (yes/no)
25:
Line      File path for crystal terrace colouring (./Path/File_Name.txt)
26:
```

Figure S5: *A general example of the input file ("input.txt") file used in CG. This file contains the user inputs to CG that are required across all simulations, and always has the same format. Follow up questions will be asked at runtime based on the responses in this file. The prompt shown by CG for each user input is shown, along with acceptable responses in parentheses. This file is generated automatically by the CG GUI and fed into CG.*

Variables belonging to different categories are colour coded in the following format:

- ***Red (bold) – File paths for data input.*** *Non-bold – File paths for data output.*
- *Green – Simulation defining parameters, length, size, temperature etc.*
- *Blue – Parameters related to addition crystallisation features.*
- *Purple – Parameters related to Δμ / solution behaviour.*
- *Orange – Parameters related to data output for visualisation.*
- *Black – Parameters related to numerical data output.*

```
1:[1A][CH4N2O(1)-CH4N2O(1)](1,0,0) R=4.349
1:[1A][CH4N2O(1)-CH4N2O(1)](0,-1,0) R=4.349
2:[1A][CH4N2O(1)-CH4N2O(1)] R=4.349
2:[1A][CH4N2O(1)-CH4N2O(1)](1,-1,0) R=4.349
3:[1A][CH4N2O(1)-CH4N2O(1)](1/2+x,1/2-y,-z)(0,0,1) R=4.712
4:[1A][CH4N2O(1)-CH4N2O(1)](1/2+x,1/2-y,-z)(0,0,-1) R=4.712
5:[1A][CH4N2O(1)-CH4N2O(1)](1,0,1) R=5.010
5:[1A][CH4N2O(1)-CH4N2O(1)](0,-1,1) R=5.010
6:[1A][CH4N2O(1)-CH4N2O(1)](0,0,1) R=5.010
6:[1A][CH4N2O(1)-CH4N2O(1)](1,-1,1) R=5.010
4.425
4.425
5.650
5.650
3.250
3.250
```

Figure S6: *A truncated example of the net interaction energy file (net.txt) used in CG for urea. The excerpt is shown for a single molecule from the urea unit cell, with ten neighbours and six interaction types.*

```
Interaction type : [Number identifying symmetry position of origin species Origin species molecule / ion type] [Origin species chemical formula (Origin species number for species present in multiple symmetry positions) – Destination species chemical formula (Destination species number for species with multiple symmetry positions)] (Space group symmetry information) (unit cell translation information) R = Interaction length in angstroms.
-Repeat for all interactions with neighbours-
Free energy of crystallisation for interaction type (kcal/mol)
-Repeat for all interaction types from molecule / ion-
```

```
-Repeat for all molecules in unit cell-
```

Figure S7: *A general example of the net interaction energy input file (net.txt) used in CG.*

Discussion

Simulation parameters are fed into CG via a text file “input.txt” located in the same folder as the CG executable. The graphical user interface (GUI) for CG creates this file automatically based on user responses to questions displayed on the screen and feeds this information into CG at runtime. The responses to the prompts in this file will decide how a crystal is treated withing the CG simulation e.g. if it is partitioned as a tile structure, the energy level spacing, if a screw dislocation is present, the simulation temperature etc. Figure S5 shows a general example of this file with the user prompt for each response given. Prompts are colour coded based on the area of the simulation they control e.g. visualisation data output or additional simulation features such as screw dislocations or internal defect sweeps. With the implementation of the GUI it is unlikely that users will manually manipulate this file, however users of the terminal version of CG will still be required to assemble and manipulate this file.

Based on certain responses to these questions / prompts, follow-up questions will appear requiring user input to control aspects of a CG simulation that only appear under certain conditions. This, for example, could be defining the length of supersaturation steps, tile scaling energies or information related to a screw dislocation. As these responses are context-dependent, they have no regular format and are therefore input in at runtime by the user, or fed in through a text file if the order of responses is known. The GUI automatically populates the responses to these follow-up questions in the correct order in a text file (normally referred to as “addinput.txt”) and feeds this into CG at runtime.

Figure S6 shows another input file to CG: “net.txt”. This is used exclusively for crystals partitioned by a simplified net / VDP (e.g. molecular and ionic crystals) and is used to generate the energy levels based on intermolecular (or interionic) interactions. This file is explicitly tied to the order of species in the structure file, with species and neighbours appearing in the same order.

In figure S6 an example of a single urea molecule is shown, with ten interaction entries displayed. Each of these entries corresponds to an interaction with a neighbour and are assigned a sequential number denoting the interaction type. If an interactions from the molecule are related by symmetry they will be assigned the same type, otherwise they are assigned a different type. Of the ten interactions in urea, there are six different interaction types based on point group symmetry.

Paired structure and net interaction energy files can be generated simultaneously through *ToposPro* by partitioning structures as a simplified net or with VDP. Care must be taken to decide what strength of interaction to consider when partitioning a structure, as the generated adjacency matrix will decide which neighbours and interactions are considered in CG for growing a crystal. Interaction strength in *ToposPro* is governed by the solid angle of a calculated VDP face, which is essentially a measurement of how large a face is relative to the rest of the faces in the VDP. Generally, a VDP face denotes an interaction to a neighbour with the estimated strength of an interaction correlated to the VDP face size / solid angle.

It must be noted here that the free energies of crystallisation calculated automatically by *ToposPro* are solely based on the VDP faces. This is strictly related to the bond strength and not the free energy of crystallisation which is required by CG, as this requires consideration of the desolvation and crystal attachment energies for a unit in a particular solvent. These will likely require changing, but it is a good first approximation to begin from.

One other input file is used in some CG simulations: the facet colouring file. This file is used to colour a crystal based on the facets it displays, along with the number of crystal terrace layers displayed on each of these facets. This file contains information on the Miller indices of the facets shown, the number of layers to colour per facet, the d-spacing and other values to adjust the colouring. Users can edit this file manually to define the colouring required and read a completed file back into CG to recolour the crystal.

S4 Δμ modes

To model the solution phase during crystallisation, a number of modes have been programmed into CG to replicate typical experiment solution profiles. Modes are divided into phases, where each phase contains a single type of behaviour for $\Delta\mu$. The possible behaviours for $\Delta\mu$ are: remaining fixed at a constant value, behaving as if an unlimited, fixed concentration solution surrounds the growing crystal (e.g. a constant flow experiment); or allowed to equilibrate, behaving as if a finite solution surrounds the growing crystal (e.g. crystallisation in a beaker or autoclave with no new nutrient added). An equilibration phase is, in actuality, two phases. The first phase is a decrease from the starting $\Delta\mu$ value towards the equilibrium point (at a user-defined rate), followed by the actual equilibrium phase where $\Delta\mu$ fluctuates around an average value equal to the equilibrium point (ΔG^{kink} – the free energy at the crystal kink site). For simplicity when discussing the available $\Delta\mu$ modes, equilibration is referred to as a single phase.

Seven modes currently exist in the CG program that match most crystal growth experiments. Two modes contain single phases of $\Delta\mu$ control (modes one and two), two contain two phases (modes three and five) and three contain three phases (modes four, six and seven). Modes one and two are the simplest available modes, modelling $\Delta\mu$ as a finite or infinite solution, respectively. Combining modes one and two together in different orders lead to the more complex modes, which are more useful for producing simulation results that can be compared directly with experimental surface topology data measured *via* AFM.

Modes three and five are the two most well used modes in CG. Mode three begins at a constant $\Delta\mu$ value, set by the user for a defined length in the simulation, then allows $\Delta\mu$ to equilibrate for the second part of the simulation. This mode not only ensures the growth of a large enough crystal to study (provided $\Delta\mu$ is set high enough to overcome the energy barrier to nucleation), but by allowing the solution and crystal phases to equilibrate, allows the interrogation of the energetics of the crystal growth process. The information gathered from this phase, primarily the equilibrium point (or ΔG^{kink}), can then be used for other modes where users can set $\Delta\mu$ values relative to this point. Mode five is one mode that benefits from this information, containing two phases where $\Delta\mu$ is fixed with different constant values. The switch between values happens instantaneously, contrasting with equilibrium phases that occur over a number of iterations in the simulation. The use of this mode is intended for

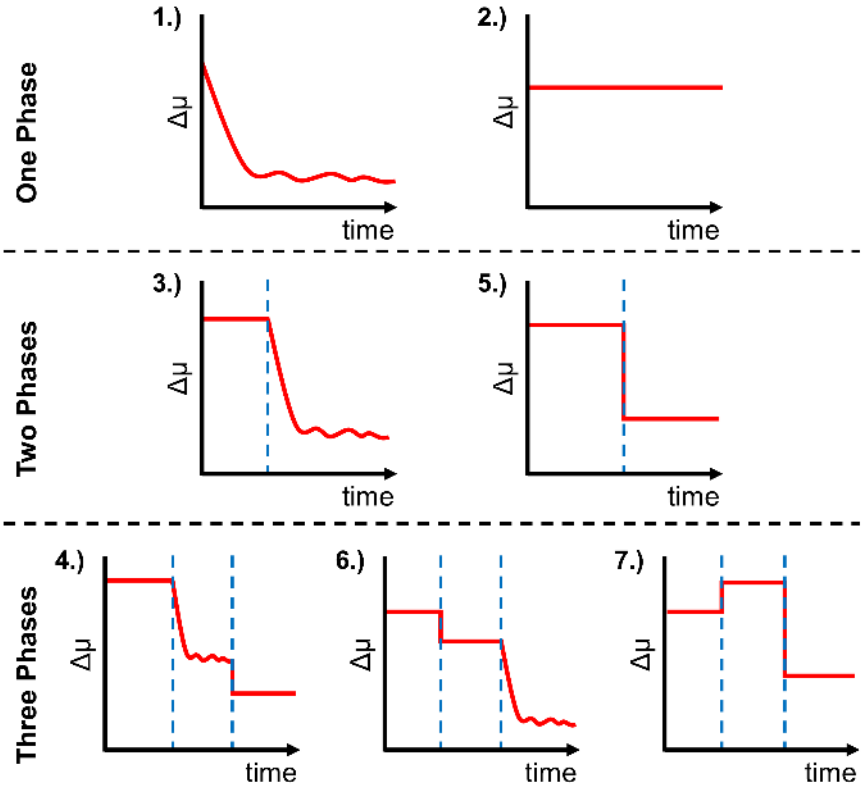

Figure S8: *The seven modes available for the modelling of the driving force for crystallisation (Δμ) in CG. Modes are separated by the number of different Δμ behaviour phases are included. The value of Δμ with respect to time is plotted by the red line, and the boundaries between separate Δμ phases are denoted by dashed blue lines.*

comparison with *in-situ* AFM results, where a crystal can be grown at high $\Delta\mu$, then switched to a value slightly above or below the equilibrium point to observe slow growth or dissolution, dependent on the experimental conditions to be matched to.

The remaining three modes are more complex three-phase $\Delta\mu$ profiles to be used in conjunction with three-phase experiments in crystal growth. Modes four and six are combinations of modes three and five, with the order of the equilibration phase differing between them, whereas mode seven is an extension of mode five with a third $\Delta\mu$ value switch added. These complex modes are very system specific and will ideally be replaced with a custom mode setup in later versions of CG. All modes are displayed graphically in figure S8.

An additional feature available for improving the modelling of $\Delta\mu$ in CG is the supersaturation excess parameter. In numerous experiments, the stoichiometry of the species in solution will not match the stoichiometry of species in the crystal structure. Considering calcite ($CaCO_3$, a

1:1 crystal structure) for example, a large amount of Ca^{2+} ions could be added to a 1:1 mixture of Ca^{2+} and CO_3^{2-}, leading to an excess of Ca^{2+} in solution compared to the crystal structure. This would result in the $\Delta\mu$ value for Ca^{2+} being higher than CO_3^{2-} as there is an abundance of Ca^{2+} in solution to add to the crystal. Simply adding an offset to the $\Delta\mu$ value for this species would account for this effect. Both species would still be consumed in the same ratio as the stoichiometry of the crystal is still 1:1, but the likelihood of Ca^{2+} being added to the crystal would increase due to the inclusion of $\Delta\mu$ in the calculation of P_s^{growth} and $P_s^{\text{dissolution}}$. This additional solution behaviour can be add on top of any of the $\Delta\mu$ modes discussed. Users can also declare any number of different excess periods over the simulation, allowing the creation of a custom mode containing any number of $\Delta\mu$ switches when used with mode two.

S5 Computational methodology

S5.1 CrystalGrower

The initialisation stage for CG begins with the input of several parameters from the user through a text file ("input.txt" shown in figure S5). These parameters dictate the method applied to the crystal structure loaded into the program. Users define whether the crystal is treated using the natural tile method, or the simplified net crystal method (including VDP), which will affect the data required for the construction of the energy ladders and how neighbouring species are considered. Users input the parameters discussed in the main manuscript when presenting the energy level diagrams (figure 8); the unknown parameters such as tile condensation energies or free energy of crystallisation values for replacing interactions to neighbouring species with interactions to solvent molecules. All parameters related to the $\Delta\mu$ are input following these values: the starting value, the selected mode, and parameters defining the simulation time lengths of the phases in the chosen $\Delta\mu$ mode. Users also assign a simulation temperature, total simulation length (in the number of program iterations) and parameters relating to the output of data to files. It will also be specified whether a screw dislocation is present in the crystal structure, along with its position and direction (discussed in greater detail later). Labelled example of a input files for CG are provided in figures S5-S7.

A crucial step in the initialisation stage is the loading of a crystal structure file. This file contains all information necessary to construct the unit cell of the crystal structure to be simulated. This information allows the effect of the growth or dissolution of any species in the unit cell on the energetics of its neighbouring sites to be calculated by CG, when combined with the energy ladders created from user-defined energy parameters. All structure files for CG are produced with the topology software: *ToposPro* and can be produced for any obtainable CIF (Crystallographic Information File). Pictorial representations of how crystal structures are treated in CG are displayed in figure S9. A real and general example of a structure file is displayed in figures S1-S4, with a full discussion of the file's contents.

Upon assembling a model of the unit cell for the loaded structure, CG proceeds to the next initialisation step: using the energy parameters input by the user to construct the required energy ladders for each species. For natural tiles, this involves finding all permutations of condensation value for all tile vertices, including lower Q^n values if open tiles are present.

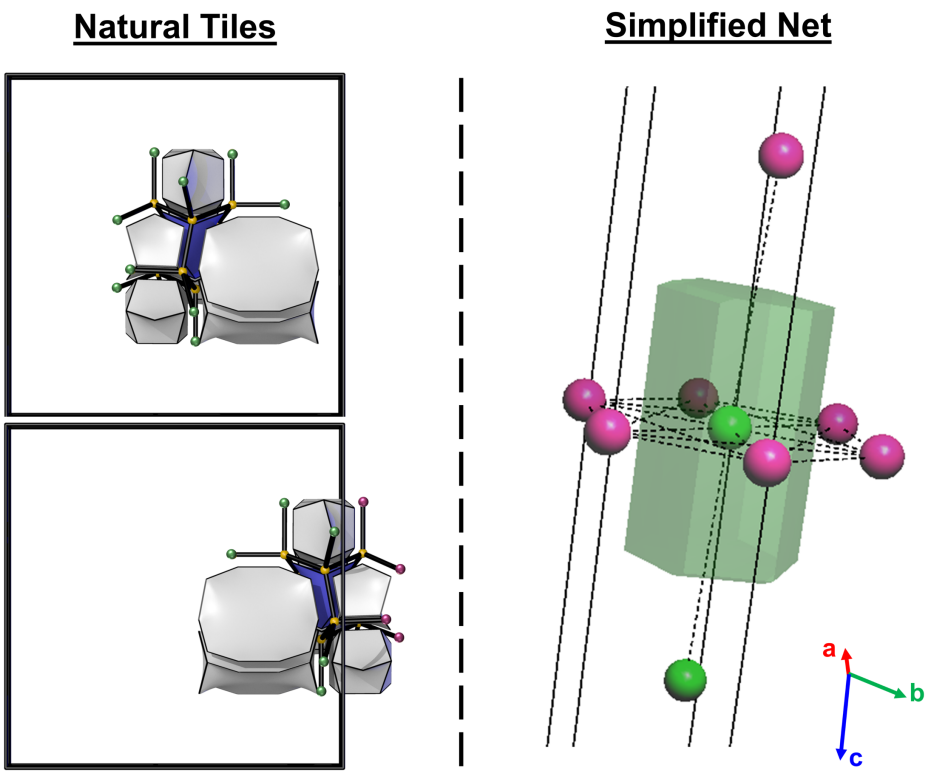

Figure S9 Left: *A zeolite structure (MFI) as treated in CG. The highlighted tile shown in blue with yellow vertices, will have a set of neighbouring tiles shown in white. Neighbours will possess connections to the T-sites connected to the highlighted tile and will affect their condensation values as they grow. The top figure shows a tile in the centre of a unit cell (denoted by black lines), where all neighbours are in the same unit cell, with connected T-sites shown in green. The bottom figure shows a tile at the edge of a unit cell, where some neighbouring tiles are in an adjacent unit cell. The connected T-sites to the original tile are shown as green within the same unit cell but pink in an adjacent cell. Both cases are treated identically, but CG must know the location of the neighbours relative to the highlighted tile.* ***Right:*** *A similar example for a molecular structure (L-cystine, with each molecule represented by a single sphere). The central molecule is shown in green with a neighbour in the same unit cell also shown in green. Neighbours outside the unit cell are shown in pink and all connections to molecules are shown by dashed lines. The VDP for the central molecule is shown as a green polyhedron and the unit cell edges are shown as solid black lines.*

Each Q^{max-1} to Q^{max} condensation is assigned an energy equal to the tile free energy input by the user in kcal/mol, which is then multiplied by a scaling value specified for lower Q^{n} vertices.

The free energy for every site type is calculated by counting the number of T-sites present with each condensation value, multiplying these values by the assigned energy values for the condensation type, then summed together for a total free energy value for the tile. Finally, all energy rung values are multiplied by a baseline scaling parameter which scales the spacing between all rungs simultaneously while retaining the overall ladder structure. This allows users to quickly change the spacing on all ladders whilst maintaining the overall structure of each ladder. By altering the baseline scaling, energy levels are separated further but the relative spacing across all energy ladders is maintained, retaining the different energy scaling values assigned to different tile types.

For simplified nets / VDP, ladders are constructed directly from the energy values for replacement of a connected neighbour with solvent, assigned through an additional input file ("net.txt" – figures S6 and S7). These values are used in conjunction with all permutations of neighbours for a species, to generate a total free energy value for each permutation.

Before proceeding to the main growth / dissolution loop, memory required for growing the simulated crystal is allocated, either calculated automatically or based on values input by the user at the simulation input stage. The size of this memory block is defined by a, b and c values, which correspond to the number of primitive unit cells available for growth along each axis. Thus, the more complex the unit cell and the larger the simulation memory size requested, the larger the memory requirement for the simulation. The starting $\Delta\mu$ value is assigned by CG, together with the behaviour for the first phase of the simulation (dependent on the $\Delta\mu$ mode chosen), which changes whenever a new phase of the $\Delta\mu$ mode is triggered.

A CG run then proceeds into the main growth/dissolution loop, which repeats according to the total number of iterations assigned to the simulation. Each iteration of the main loop begins by calculating the probability of growth (P_s^{growth}) and the probability of dissolution ($P_s^{\text{dissolution}}$) for all site types, using the value of the energy ladder rung, the current $\Delta\mu$ value and the system temperature, according to equations 1 and 2. A record of the population of each site type is also kept in CG, for both potential growth and dissolution sites.

$$P_s^{\text{growth}} = \exp\left(-0.5\left(\frac{\Delta G_s}{kT}\right) + 0.5\left(\frac{\Delta\mu}{kT}\right)\right) \qquad \text{Equation 1}$$

$$P_s^{\text{dissolution}} = \exp\left(0.5\left(\frac{\Delta G_s}{kT}\right) - 0.5\left(\frac{\Delta \mu}{kT}\right)\right) \quad \text{Equation 2}$$

The following step of the main loop is where the Monte Carlo selection algorithm is employed. First, a random number is generated using the built-in Fortran random number generator, then the "true" probability of selection for each site type is calculated (P_s^{true}):

$$P_s^{\text{true growth}} = C_s P_s^{\text{growth}} \quad \text{Equation 3}$$

$$P_s^{\text{true dissolution}} = C_s P_s^{\text{dissolution}} \quad \text{Equation 4}$$

where C_s denotes the "count" or population of site type s at the current iteration of the program. A sum of these values is then calculated:

$$\text{Total Probability} = \sum_{s=1}^{n} P_s^{\text{true growth}} + \sum_{s=n+1}^{m} P_s^{\text{true dissolution}} \quad \text{Equation 5}$$

where n and m denote the total number of growth and dissolution site types, respectively, which are equal in value. By cycling through all site types one-by-one, equation 5 results in a single value known in the CG method as the total probability. This totalling method will account for both probability weighing by free energy value and statistical chance by population value when calculating P_s^{true} of a site type being chosen. If the population of a growth site type is zero, then it has no chance of growth; it is not a potential transition and therefore contributes nothing to the total probability.

A second pass is then performed sequentially of each site type. Each time a site type true probability is calculated and added to the running total, the running total is checked against the random number generated at the beginning of the main loop. This running total is referred to as the "sum probability" in CG. Each calculated P_s^{true} can be represented pictorially as a "probability block" (figure S10). The larger the P_s^{true} value for a site type, the larger its probability block size is, and the likelihood of its selection increases. The sum probability value will always denote the end of a site type probability block, therefore if the value of the random number falls below the sum probability, it indicates the current site type being processed has been chosen through the Monte Carlo method.

Once a site type has been chosen, a second random number is generated. This number is used

to select one site from the chosen site type at random to grow or dissolve (dependent on the site type chosen). All sites of the same type will have the same P_s^{true} value, thus a truly random method of selection is suitable. The chosen site from the chosen site type is marked as grown or dissolved and be is updated to become a potential dissolution or growth site type, depending on the site type chosen during the selection process.

Following this update, the main loop will progress to the neighbour-checking routine. Each neighbour for the selected site (as defined in the structure file) has their site type reassessed based on the change in their coordination or condensation caused by the Monte Carlo selection. Most neighbour free energy values are different upon changing site type, as are their probabilities of growth or dissolution. After all site types are updated for the neighbours, the site type count values also change, affecting the value of the true probability of selection for each site type. Also affected by the selected growth or dissolution is the value of $\Delta\mu$, dependent on its set behaviour at the current point in the simulation. If the $\Delta\mu$ is allowed to change, a growth event will cause it to lower in value – removing a species from solution,

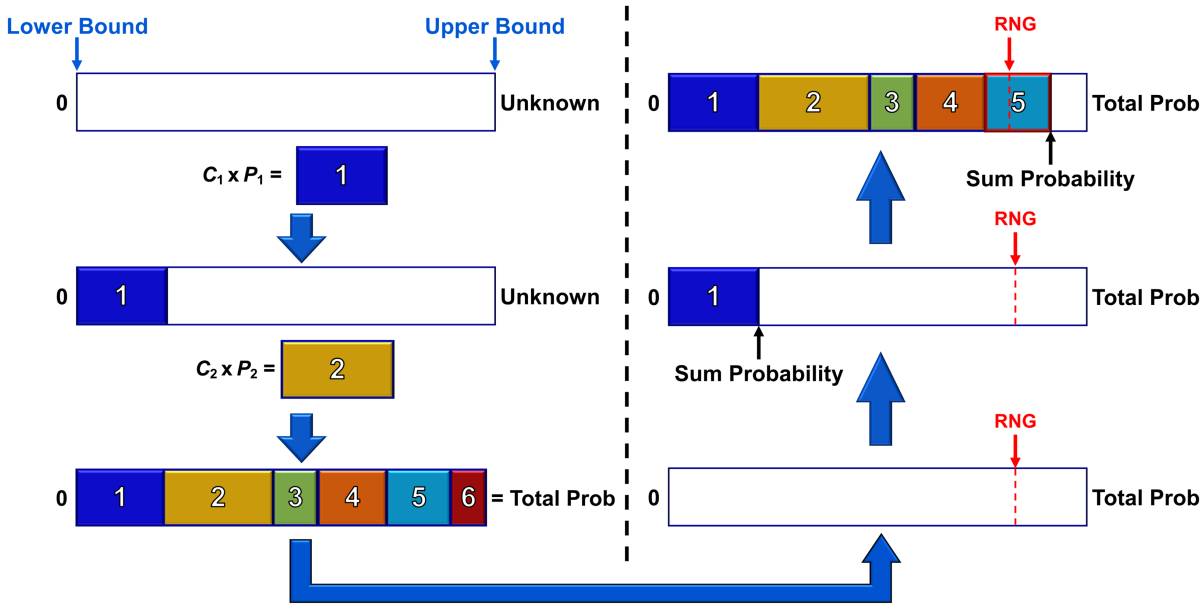

Figure S10: *A pictorial representation of the Monte Carlo selection process in CG. The left side of the image demonstrates the "first pass" through the site types 1-6, calculating the total probability of growth or dissolution for each site type – C_s multiplied by P_s – then summing them together. Site types with larger C_s or P_s values (e.g. site types 1 and 2) will have a higher chance of being chosen, shown as larger blocks. This step is necessary to know how to scale the generated random number which by default is between 0 and 0.99999.*

The right side shows the "second pass" where the Monte Carlo selection actually takes place. As each site type's probability is calculated and added to a running total, the value is checked against a random number generated at each iteration (RNG, red dashed line) scaled to the total probability. As soon as the running total exceeds RNG, it means the last site type to be processed must be the chosen site type, as RNG did not fall into the site processed prior to this site, but is less than the maximum value for this site type.

whereas a dissolution event will raise its value – returning a species to solution. This is ignored if the $\Delta\mu$ is currently set to be held at a constant value. These values are all recalculated at this point in the iteration.

Before restarting the main loop, the total probability and sum probability are both reset to zero to be recalculated in the next cycle, due to the aforementioned changes in the count of site types, free energies and $\Delta\mu$. Another possible change prior to restarting the loop is the progression to a new phase in the chosen $\Delta\mu$ mode. The number of main loop iterations is tracked by CG at each Monte Carlo event and checked against the phase lengths chosen by the user in the initialisation phase. Once these values match, the $\Delta\mu$ progresses to the next

phase of the mode and the loop repeats as before. The number of iterations is also checked relative to the user selections for data output frequency and outputs data to the correct files at defined intervals in the simulation. The main loop will repeat a number of times equal to the number of program iterations specified by the user in the initialisation phase of a CG run. A flow diagram for the main loop of CG is presented on the final page of the supplementary information (figure S14).

Numerous streams of data are output at varying simulation times in a typical CG run but the output of data can be considered as the "final" phase of a CG run. Data output by CG can be split into three categories: numerical values for crystal growth parameters, data for visualisation and facet identification data. CG outputs numerous useful parameters versus global time (the reciprocal of the sum of all probabilities in the system) at set intervals in a CG run, controlled by user input at the initialisation stage. Users can graph $\Delta\mu$ and the number of growth events versus global time using this information. Also output is a file containing the population of all site types at each iteration a visualisation frame is output. Sites of particular interest are separated into categories for easy viewing, such as surface sites, bulk sites and sites containing Q^2 species for natural tiles, along with possible internal defect sites within a grown crystal.

Visualisation data is formatted for use in combination with two pieces of software: the CG Visualiser (CGV), a companion visualisation program developed alongside CG; and Ovito, a third-party visualisation package used by many computational research groups. These data consist of scaled Cartesian coordinates, along with numbers assigning species type to distinguish different species in grown crystals. As with numerical parameters, visualisation data can be output at user-specified intervals within a CG run. The default behaviour is to save a single "frame" at the final simulation step of a fully-grown crystal that can be mapped into 3D space. This can be changed to output frames at regular intervals in time, allowing users to create movies of their crystals growing versus simulation time. Visualisation using the CGV is discussed in greater detail in the following section.

Many additional features are available in CG to improve the simulation of crystal growth for a wide range of materials or to make features clearer for analysis. The files containing facet identification data are related to one such additional feature in CG, used to locate and partition simulated crystal facets for easy viewing. This function attempts to identify existing

facets in a grown crystal and separate them into layers using the d-spacing for Miller planes for the corresponding facet. In most cases observed with AFM, the height of terraces on facets usually coincide with the d-spacing or fractions of the d-spacing. By partitioning the surface in this manner, layers can be coloured separately to be distinguished from each other. When visualised, this results in a similar appearance to that of an AFM micrograph where layers are coloured by a gradient according to their relative heights.

This function works by progressing along a crystallographic direction (specified by h, k, l values using Miller index notation) until reaching the furthest grown unit from the crystal origin. Progressing back towards the origin at steps equal to the d-spacing for the plane in the crystal structure, the routine checks if crystal surface units are present. All species in the same plane "layer" are assigned a layer number. This stepping inwards towards the origin will repeat until reaching the maximum number of layers to be coloured, designated by the user. Facets are located by first completing a CG run, and invoking the automatic routine to produce facet identification data. This can then be manually adjusted to the user's preference and used as an input for a following CG run. Both observed facets and all allowed facets are produced in

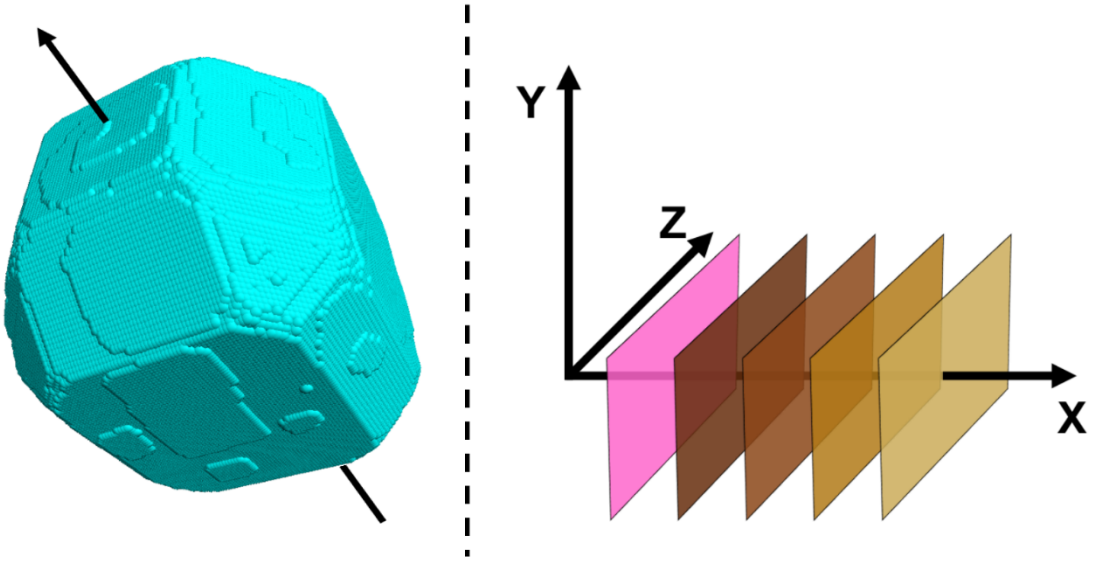

Figure S11, Left: *a depiction of a screw dislocation added in CG.* ***Right:*** *the default colouring method employed by the facet identification routine in CG. Each plane denotes a step towards the origin in units of d-spacing.*

separate files, allowing customisation if the automatic routine does not locate all present facets.

Another additional feature in CG, briefly mentioned during the initialisation phase, is the incorporation of screw dislocations. These defects occur in several frameworks and are one of the main mechanisms through which crystals can grow. Screw dislocations were suggested by Frank as an explanation as to why many crystals still formed in mixtures where the $\Delta\mu$ values were measured to be lower than the activation energy required to form a critical nucleus size.[1] They form where a slip occurs along a plane in a crystal structure (figure S12, right), causing a shift where one section of the crystal structure is displaced above another section. This results in a step edge protruding from the crystal surface which units of growth from solution can add to.

Using the Terrace Ledge Kink (TLK) model (figure S12, left)[2,3] of a primitive cubic crystal composed of cubic units, surface nucleation sites offer only one pre-formed neighbour connection for adding growth units, whereas step edges will offer two, or three neighbour connections (depending on whether units add directly to the step or add to protruding units on the step – a kink site). Sites along this step edge will therefore have a lower energy barrier to growth compared to surface nucleation sites on the defect-free crystal surface. Additionally, as the step grows another step is formed perpendicular to its direction of growth where more units can add. This process repeats whenever the length of a step reaches two

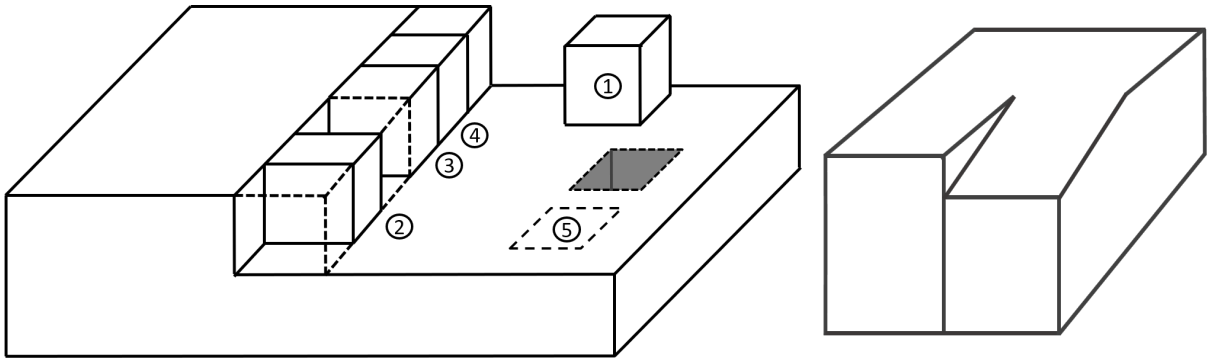

Figure S12, Left: *An example of the TLK mechanism when applied to a cubic system. Crystal sites are labelled with numbers from 1 to 5. (1) an adatom, (2) a step adatom, (3) a kink site, (4) a step adatom and (5) a surface site. Bulk sites are not shown but would be fully surrounded cubes in an underlying layer of the crystal. The sites outlined by dashed lines are the two types of vacancies, a surface vacancy (grey) and a step vacancy* ***Right:*** *An illustration of a screw dislocation on a flat crystal surface provided low energy step edges to attach to (such as units 2-4 shown in the TLK diagram).*

times the radius of the critical nucleus,[4] resulting in crystal growth in a spiralling pattern around the core of the screw dislocation.

In the CG model, screw dislocations function in the same way; the structure is displaced along a crystallographic direction by a multiple or fraction of non-primitive unit cells, a value known as the Burgers vector. The choice of the non-primitive unit cell for the screw dislocation is due to this being the more convenient cell commonly used when visualising and discussing screw dislocations. This results in the aforementioned step edge for units of growth to add to, with sites along this step possessing higher P_s^{growth} values, due to their reduced free energy values from their existing neighbour connections. Adding a screw dislocation invariably increases the rate of crystal growth in the direction of the screw core, resulting in the elongation of the crystal along this direction. Screw dislocations can be required in some cases to match both experimental morphologies and surface topologies and are frequently observed by AFM in combination with other crystal growth mechanisms (e.g. the birth and spread mechanism). CG currently supports the addition of a single screw dislocation with plans to add groups of screw dislocations in the future, once a general algorithm can be developed for their incorporation.

Growth modifiers can also be added to a simulation in CG, where users can target specific site types to be slowed during crystal growth. This can be used to mimic the selective binding of modifiers / poisons to step edges, kink sites or particular intermolecular interaction types. The strength of the growth modifier attachment is controlled by the number of iterations a modifier stays attached before being removed, while the concentration level of the modifier can be modelled by adjusting the number of iterations between each modifier attachment (i.e. adjusting the attachment frequency).

The final additional feature CG can perform currently is the identification of internal defects within a simulated crystal. As a crystal grows, some units can be left incomplete as it becomes more energetically favourable to add to the crystal surface. These defects can appear in segregated zones which match experimental observations (such as the optical birefringence seen in the zeolite framework MFI). As the visualisation process for CG removes all completely grown (bulk) units, removing the front face of a simulated crystal exposes the incomplete species within to be interrogated. Taking cross sections of simulated crystals can give clear depictions of the zoning of defects, features that could be important for catalysis. By using

the aforementioned facet identification tool to define the outer surface of the crystal correctly, users can also quantitatively compare the number of internal defects between simulations with different parameters. This could again be important for catalytic applications, by highlighting energetic conditions that lead to high or low defect densities.

S5.2 CrystalGrower Visualiser

The CGV algorithm can construct three object types: polygons, spheres and cylinders. For natural tile structures, CG outputs scaled coordinates for tile vertices, along with the tile faces they appear in and the order they appear in each face. The CGV processes the natural tile face-by-face, calculating whether faces are flat or curved. Flat faces are constructed as simple polygons, whereas non-flat faces are subdivided into a number of triangles to give the appearance of smooth curvature. These faces are combined to create a three-dimensional natural tile which is saved as a constructed object and assigned an identifying tile type and tile number (for tiles that are the same type but in different symmetry positions). This is repeated for all tiles in the unit cell. Once all objects in the unit cell are constructed, the CGV processes the cartesian coordinate output from CG and begins translating the appropriate tiles to the listed positions by reading its tile type, tile number and XYZ position.

For structures built from individual growth units such as molecules or ions, two visualisation options exist. The first, and most simplistic depiction, places spheres with an average radius at the unit's central fractional coordinates in the unit cell. These spheres are then assigned types and numbers as for natural tiles and translated to the correct cartesian coordinates output by CG.

The second visualisation option constructs the structure of the entire growth unit. For molecules, this includes atoms and bonds, where atoms are represented by spheres with incrementally increasing radii across the periodic table. This is achieved by reading in atom types, fractional coordinates, information on which atoms form bonded pairs and the bond types between atom pairs from CG. The CGV begins by reading the atom types and coordinates, using a database for each element type (H to Og) to place a sized sphere at the correct unit cell position. To create bonds, the CGV calculates vectors between atoms of bonded pairs, rotates to align this axis with the viewing direction (the Z axis of the screen), and places a cylinder or multiple offset cylinders along this axis, dependent on the bond type.

The CGV currently supports single, double, triple, dashed and dotted bonds. Upon construction of a single molecular unit, each atom type and bond type composing the unit are saved as separate objects. Following the same procedure as natural tiles, these objects are translated to the cartesian coordinates output by CG, with the type and number assigned to the unit ensuring the correct species is called at the required position.

For all visualisation methods, once the required units have been drawn according to the number of grown units in a frame output by CG, all drawn units for each species type are saved as large objects, made up of thousands of smaller objects. Thus, improving processing times when manipulating the entire crystal during visualisation. This procedure is repeated for the number of frames output by CG during its run, which allows transitioning between simulation frames in the CGV with a single button press. This can be used to view the crystal growth process evolving with time.

Using the CGV allows interrogation of the simulated crystal output from CG with image processing for the presentation of data. A grown crystal can be manipulated in three-dimensions, with real-time rotation and translation possible for reasonably sized frames on standard computer hardware. Manipulation speed is entirely dependent on the number of species in a single frame, rather than the number of frames loaded into the CGV (this number is however, limited by the RAM available on the machine). Crystals can be displayed using a variety of colouring methods, utilising the facet identification tool in CG or coloured according to species type. All colouring is fully customisable, and the visibility of any object type can be toggled on or off at will. All objects aside from natural tiles are also resizable, and curved object resolution can be increased for rendering improved images. A set of CGV images showcasing various display settings is presented in S13.

Additional features have been added to the visualiser to improve its use for the presentation of simulation results. Aside from being able to produce movies of crystal growth versus time, the CGV is also capable of rendering movies of rotation and translation to highlight areas of interest in simulations. Users can create longer movies composed of several smaller movies stitched together to transition between important features seen on simulated crystals.

Several common movie presets are built into the CGV, producing simple rotation movies displaying a crystal with one click. Custom states can also be saved, with all user customised settings along with the viewing position retained. Simulation states can be loaded at will for

the same structure types to quickly return to highlighted areas within a simulation with minimal interruption. The CGV is also capable of high-throughput processing of simulations with the same crystal structure. By loading the crystal structure output from CG, then loading a single simulation file, users can tune image settings to their liking before invoking the high-throughput command for a set of simulations. This mode can be tweaked to either produce a single image of each simulation, or rotate along any of the principle axes, taking a series of images to view the simulated crystals from different angles. An additional feature in the high-throughput function is to cycle through displaying each species type individually, saving an image for each species type. This feature is useful for locating internal defects frequently obscured by the crystal surface when all species are displayed.

Currently, the CGV is available only on Windows operating system, whereas Ovito can be used by Windows, Mac and Linux users. We hope to develop the CGV into a cross-platform format in the future.

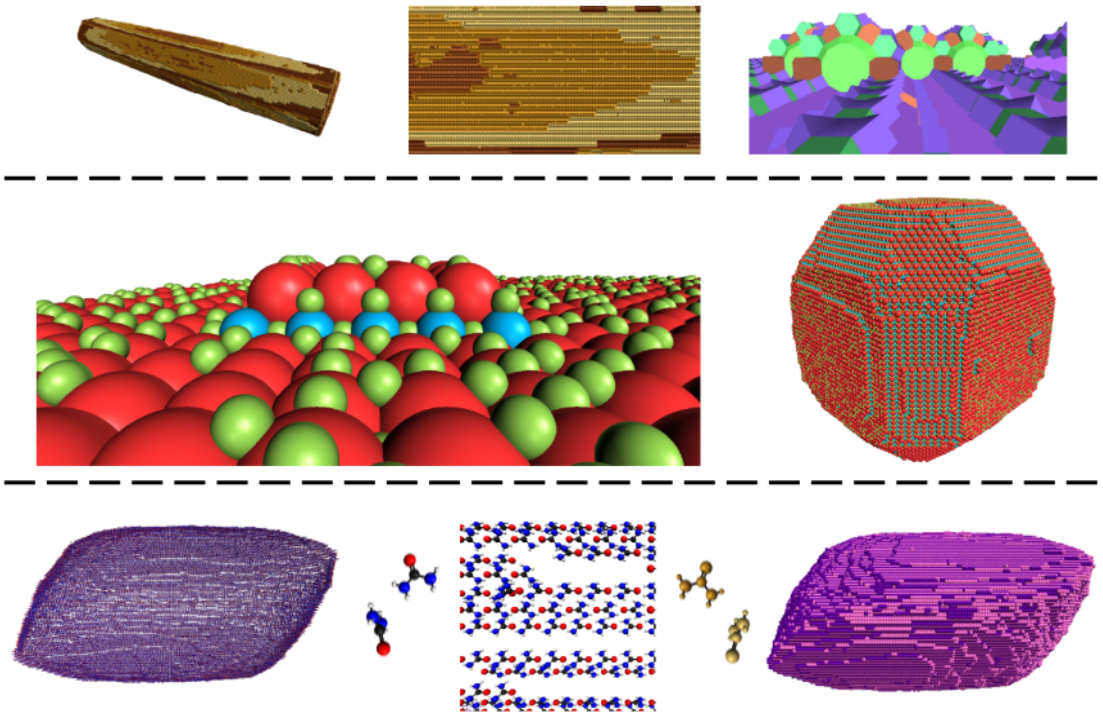

Figure S13: *A set of images taken using the CGV, with various settings.*
Top: *Images of the zeolite L (LTL) framework with facet / layer colouring (left and centre) and tile type colouring (right).*
Centre: *Images of the zeolite A (LTA) framework. The leftmost image uses high resolution spheres with radii calculated by the root mean square of each tile's fractional coordinates, whereas the rightmost image uses standard natural tile polyhedra.*
Bottom: *Images of a urea crystal. The leftmost and central images use atoms and bonds to construct the crystal, whereas the rightmost image uses average-sized spheres placed at the centre of each molecule to construct the crystal. The small images flanking the central image show the colouring schemes available for structures built from atoms and bonds. Molecules can be coloured by element type (oxygen in red, nitrogen in blue, carbon in black and hydrogen in white) or by layers using the facet detection tool. The layer / facet colouring is also available when displaying molecules as simple spheres, as used for the rightmost crystal image.*

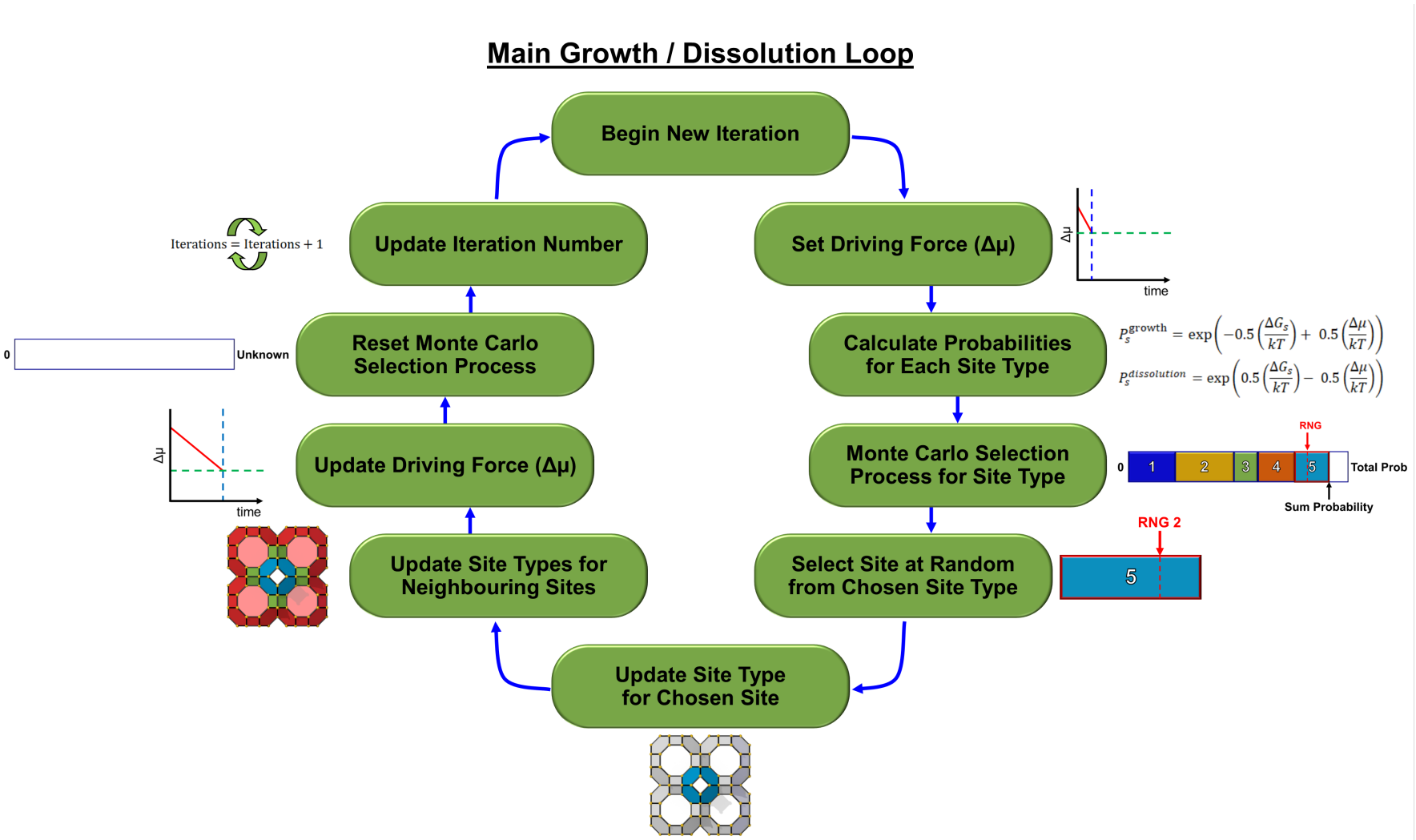

Figure S14: *A flow diagram for the main growth /dissolution loop within CG.All main steps are shown, along with pictorial representations of each step.*

References

1 F. C. Frank, The influence of dislocations on crystal growth, *Discuss. Faraday Soc.*, 1949, **5**, 48.

2 W. Kossel, Zur Energetik von Oberflächenvorgängen, *Ann. Phys.*, 1934, **413**, 457–480.

3 I. N. Stranski, Zur Theorie des Kristallwachstums, *Z. Phys. Chem.*, 1928, **136U**, 259–278.

4 P. Cubillas and M. W. Anderson, in *Zeolites and Catalysis: Synthesis, Reactions and Applications*, eds. Čejka Jiři, Corma Avelino and Zones Stacey, Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, Germany, 2010, vol. 1, pp. 1–55.