1. Derivation of integration error due to *k*-point sampling for generalized grids

Here we derive equation (13) in the main text. Monkhorst and Pack have derived a similar result for diagonal grids [1], and we take advantage of the relationship between generalized Monkhorst-Pack grids and real-space supercells to extend their result to the general case.

Let $p_n(\mathbf{k})$ represent a property of the nth Bloch wave at wave vector $\mathbf{k}$ with respect to the real-space primitive cell of a crystal. For example, we may define

$$p_n(\mathbf{k}) = \varepsilon_n(\mathbf{k}) f(\varepsilon), \tag{1}$$

where $\varepsilon_n(\mathbf{k})$ is an eigenvalue of the Hamiltonian eigenvalue and $f(\varepsilon)$ is an occupancy function. We define

$$P(\mathbf{k}) = \sum_1^\infty p_n(\mathbf{k}), \tag{2}$$

where we will assume that the sum in equation (2) converges (e.g. due to the occupancy function). The integrated value of $p_n(\mathbf{k})$ over all states is therefore

$$P = \int_{BZ} P(\mathbf{k}) d\mathbf{k}. \tag{3}$$

Because P is an integral over all states, it is independent of the choice of real-space unit cell. Thus for a generalized *k*-point grid with grid points given by equation (10) in the main text, it is convenient to work with the corresponding real-space supercell with lattice vectors $\mathbf{c}_1$, $\mathbf{c}_2$, and $\mathbf{c}_3$ given by

$$(\mathbf{c}_1 \quad \mathbf{c}_2 \quad \mathbf{c}_3) = (\mathbf{a}_1 \quad \mathbf{a}_2 \quad \mathbf{a}_3)\mathbf{M} \tag{4}$$

where $\mathbf{M}$ is a non-singular integer matrix and $\mathbf{a}_1$, $\mathbf{a}_2$, and $\mathbf{a}_3$ are real-space primitive lattice vectors. For clarity, we will use $P_S(\mathbf{k})$ to represent the value of $P(\mathbf{k})$ in the supercell representation, BZS to represent the Brillouin zone of the supercell, and Ω_S to represent the volume of the Brillouin zone of the supercell. We will use $P(\mathbf{k})$, BZ, and Ω as the corresponding variables for the primitive cell representation. We will let N_S represent the ratio of the volume of the supercell to the volume of the primitive cell, or equivalently the number of total *k*-points in the Brillouin zone of the primitive cell.

By Bloch's theorem, the function $P_S(\mathbf{k})$ must be periodic with respect to the reciprocal lattice of the superlattice. Following Baldereschi [2] and Monkhorst and Pack [1], we expand $\dfrac{P_S(\mathbf{k})}{N_S}$ in a Fourier series:

$$\frac{P_S(\mathbf{k})}{N_S} = \sum_m f_m e^{i\mathbf{R_m}\cdot\mathbf{k}} . \tag{5}$$

where f_m are Fourier coefficients and $\mathbf{R_m}$ are real-space lattice vectors. The inverse Fourier transformation is

$$f_m = \frac{1}{\Omega_S} \int_{BZS} \frac{P_S(\mathbf{k})}{N_k} e^{-i\mathbf{R_m}\cdot\mathbf{k}} d\mathbf{k} = \frac{1}{\Omega} \int_{BZS} P_S(\mathbf{k}) e^{-i\mathbf{R_m}\cdot\mathbf{k}} d\mathbf{k} . \tag{6}$$

where the integral is over the Brillouin zone of the supercell (BZS) with volume Ω_S. We note that by the equivalence between the primitive cell and supercell representations of the Bloch waves, and because $e^{-i\mathbf{R_m}\cdot\mathbf{k}}$ must be periodic with respect to the Brillouin zone of the supercell, equation (6) can be re-written in terms of an integral over the Brillouin zone of the primitive cell:

$$f_m = \frac{1}{\Omega} \int_{BZ} P(\mathbf{k}) e^{-i\mathbf{R_m}\cdot\mathbf{k}} d\mathbf{k} . \tag{7}$$

Without loss of generality we can define $\mathbf{R}_1$ to be the origin, giving us

$$f_1 = \frac{1}{\Omega} \int_{BZS} P_S(\mathbf{k}) d\mathbf{k} = \frac{1}{\Omega} P . \tag{8}$$

The integral in equation (8) can be approximated by averaging over k-points. We consider a generalized k-point grid, defined with respect to the real-space primitive cell, with k-points given by

$$\mathbf{k} = \begin{pmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 \end{pmatrix} \left(\mathbf{M}^{-1}\right)^{\mathbf{T}} \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix} + \mathbf{s} \tag{9}$$

where $\mathbf{b}_1$, $\mathbf{b}_2$ and $\mathbf{b}_3$ are reciprocal lattice vectors with respect to the real-space primitive lattice, n_1, n_2, and n_3 are integers, and $\mathbf{s}$ is a shift vector. As in the main text, we define the average integration error as

$$\Delta = \frac{1}{N_S} \sum_{\mathbf{k}} P(\mathbf{k}) - \frac{1}{\Omega} P, \tag{10}$$

where Ω is the volume of the primitive cell Brillouin zone and the sum is over all N_S k-points in the Brillouin zone of the primitive cell. As demonstrated in the main text, a sum over Bloch states on a generalized k-point grid given by equation (9) in the primitive cell representation is the same as the sum over Bloch states at $\mathbf{s}$ in the supercell representation. Therefore

$$\sum_{\mathbf{k}} P(\mathbf{k}) = P_S(\mathbf{s}), \tag{11}$$

where the sum on the left is over all k-points in the Brillouin zone of the primitive cell. Substituting equation (11) into equation (10) yields

$$\Delta = \frac{P_S(\mathbf{s})}{N_S} - \frac{1}{\Omega} P, \tag{12}$$

and substituting equations (5) and (8) into equation (12) yields

$$\Delta = \sum_m f_m e^{i\mathbf{R_m} \cdot \mathbf{s}} - f_1. \tag{13}$$

Using the fact that we have defined $\mathbf{R}_1$ to be the origin, we arrive at the final result:

$$\Delta = f_1 + \sum_{m>1} f_m e^{i\mathbf{R_m} \cdot \mathbf{s}} - f_1 = \sum_{m>1} f_m e^{i\mathbf{R_m} \cdot \mathbf{s}}. \tag{14}$$

2. Materials used for benchmarking and analysis

For benchmarking, we randomly selected entries from the inorganic crystal structure database [3] with fully occupied sites. Each structure was relaxed prior to the static calculations using a k-point grid with $r_{lattice}$ of at least 62 Å, which as seen in the main text is more than enough to achieve convergence. Structures that took more than 500 atomic steps to relax were discarded, as were structures for which VASP threw an error on unreasonable input geometry (i.e. "Sub-Space-Matrix is not hermitian" error). All relaxations were run spin-polarized. For materials that relaxed to a state with no net spin, the static benchmark calculations were run with no spin polarization. For materials that relaxed to a spin-polarized structure, spin polarization was used in all static benchmark calculations. To reduce the amount of noise due to structures relaxing to different spin states, in all benchmark calculations the net spin of each

material was fixed to a constant value calculated by rounding the net spin of the final relaxed state to the nearest integer.

For the static benchmarking calculations, energies were converged to a difference of less than 10^{-5} eV per unit cell. Across all grid generation methods, some calculations produced anomalous energies at even fairly dense k-point grids, introducing substantial noise to the results. All of these calculations had both of the following warnings in their output files:

```
WARNING: DENTET: can't reach specified precision
```

```
BRMIX: very serious problems
 the old and the new charge density differ
```

The problem was fixed by re-running the calculations using Gaussian smearing (with a width of 0.05 eV) instead of the tetrahedron method. This fix was used on all 496 calculations that had both of the above warnings.

Even after the fix described above, there were two structures that failed to reach electronic self-consistency after 150 electronic steps at grid densities that could have affected the convergence results. This behavior was observed for all grid generation methods. These materials were discarded, and the remaining 102 structures, shown in the below table, were used for benchmarking.

ICSD ID	Arithmetic Crystal Class	Composition	Is Metal	Number of Atoms in Unit Cell	Unit Cell Volume (Å^3)
56545	19	$AgCl$	FALSE	4	75.72
159858	41	Ag_3IS	FALSE	5	118.29
8117	18	$AlCl_4Na$	FALSE	24	615.01
92522	50	$AlLaO_3$	FALSE	10	108.65
70109	49	AlO_4P	FALSE	18	223.19
14270	8	Al_4CaO_7	FALSE	24	294.81
36478	18	$AsCuPbS_3$	FALSE	24	553.31
16145	18	As_4S_3	FALSE	28	781.53
98792	7	As_8S_9	FALSE	34	893.45
412235	7	$Au_3F_{18}Sb_2$	FALSE	46	594.25
613405	58	BCo_4Y	TRUE	12	145.31
613529	36	B_2CrNb_2	TRUE	10	114.06
44431	58	B_2LaRh_3	TRUE	6	82.44

ICSD ID	Arithmetic Crystal Class	Composition	Is Metal	Number of Atoms in Unit Cell	Unit Cell Volume (Å^3)
615671	21	B_4V_3	TRUE	7	59.26
172338	8	$BaC_2F_6O_6S_2$	FALSE	34	479.50
108831	18	$BaCuO_5Y_2$	FALSE	36	488.57
25311	37	$BaMg_2Si_2$	TRUE	5	118.10
74006	8	$BaMn_2O_{14}Si_4Sr_2$	FALSE	23	289.90
155512	27	BaO_4W	FALSE	12	190.90
108097	71	$BaTe$	FALSE	2	72.82
162894	49	$Ba_3Fe_3NbO4Si_2$	FALSE	23	327.92
75963	19	$Ba_5ClO_9Ru_2$	FALSE	34	563.21
180893	37	$BiNi_2O_{10}Sr_5$	TRUE	18	246.33
155011	36	$BrFPb$	FALSE	6	132.58
29114	36	$Br_2CO_3Pb_2$	FALSE	32	626.01
618234	58	CLu_5Si_3	TRUE	18	362.99
420223	8	CS_3Tl_2	FALSE	12	301.71
85953	71	C_7V_8	TRUE	60	572.70
170824	8	$CaCuGe_2O_6$	TRUE	20	239.51
159035	19	$CaIrO_3$	TRUE	10	112.23
168903	71	$CaMnO_3$	TRUE	5	51.72
182052	49	Ca_2O_4Si	FALSE	14	179.38
96484	50	Ca_3FeO_6Rh	FALSE	22	257.54
183509	72	CdS	FALSE	2	50.96
81658	18	$Cd_3K_2Te_4$	FALSE	36	1253.50
82547	18	$CeGeRh$	TRUE	12	234.21
622339	58	$CeZn_5$	TRUE	6	104.05
420548	18	ClO_3PbV	FALSE	24	374.94
90123	37	$Cl_2Co_2O_5Sr_3$	TRUE	12	178.03
65731	72	Cl_6Cs_2LiY	FALSE	10	283.07
64613	72	Cl_6Cs_2Re	FALSE	9	268.20
27180	7	$Cl_6H_{22}O_{10}Sn$	FALSE	78	820.07
26692	72	Cl_6Rb_2Se	FALSE	9	247.82
624863	18	$CoSSb$	FALSE	24	403.92
249347	18	Cs_2S_4W	FALSE	28	958.93

ICSD ID	Arithmetic Crystal Class	Composition	Is Metal	Number of Atoms in Unit Cell	Unit Cell Volume ($\mathring{A}^3$)
261952	7	$Cs_4Ge_8O_{20}U$	TRUE	66	1112.96
28737	37	$CuGaSe_2$	FALSE	8	177.97
657666	37	$CuInSe_2$	FALSE	8	201.40
75531	72	$CuIr_2S_4$	TRUE	14	243.76
65677	21	$CuLiO_4V$	FALSE	14	141.18
86752	37	$CuNd_2O_4$	TRUE	7	90.95
247127	41	CuO_3Si	TRUE	30	462.08
280597	72	Cu_6O_8Pb	TRUE	15	156.56
631453	8	Eu_2O_3	TRUE	15	220.41
37448	41	F_6NiPt	FALSE	8	91.25
20029	18	F_8K_2Re	FALSE	44	582.58
169262	58	$FeSe$	TRUE	4	61.53
37124	18	Fe_2MoP_{12}	FALSE	60	868.85
15840	50	Fe_2O_3	TRUE	10	86.23
420235	21	$Fe_2O_5Sr_3$	TRUE	10	138.39
1118	2	$Fe_3Na_7O_8$	FALSE	36	474.65
106712	72	Ga_2LiPd	TRUE	4	57.77
100052	50	Ga_2Li_3	TRUE	5	76.39
167567	36	$Ga_4Mn_3Ni_9$	TRUE	16	190.40
39788	18	$GeNaO_5Sb$	FALSE	64	878.91
60943	18	H_2O_4Se	FALSE	28	298.03
169603	37	H_2Ti	TRUE	3	21.22
240915	18	H_4I_3NPb	FALSE	36	926.89
418251	7	$H_4Li_2O_6P$	FALSE	52	444.94
2226	7	H_5NaO_4S	FALSE	88	866.11
170776	62	$H_7NO_{13}P_3Ti_2$	TRUE	104	957.40
174108	18	HfO_3Pb	FALSE	40	555.23
640117	58	$InNi$	TRUE	6	95.24
174222	7	IrP_2	FALSE	12	185.17
104572	72	Ir_2Th	TRUE	6	114.65
641029	21	$Ir_3Si_5Y_2$	TRUE	20	332.47
602269	36	$KMnSb$	TRUE	6	175.29
82227	18	$LaMnO_3$	TRUE	20	236.69

ICSD ID	Arithmetic Crystal Class	Composition	Is Metal	Number of Atoms in Unit Cell	Unit Cell Volume (Å^3)
641459	41	$LaMo_6Se_8$	TRUE	15	314.31
641499	19	LaNi	TRUE	4	90.31
420918	8	$La_5NO_2S_4$	FALSE	24	527.60
642364	49	$LiSe_2Zr$	TRUE	4	79.09
83791	18	Mg_2O_4Si	FALSE	28	275.45
159441	58	MoN	TRUE	2	20.23
167863	72	NOs	TRUE	2	20.58
35198	27	NaO_3P	FALSE	40	492.77
92128	50	NiO	TRUE	2	18.00
657414	50	Ni_3S_2	TRUE	5	66.76
50942	18	$O_{12}Sc_2W_3$	FALSE	68	1156.49
41474	49	O_2Si	FALSE	9	93.89
169629	36	O_2Ti	FALSE	6	62.90
79914	18	O_2Zr	FALSE	12	120.33
33803	37	$O_7Ru_2Sr_3$	TRUE	12	156.13
36630	19	$P_5Pd_5Sr_4$	TRUE	28	566.04
105719	58	Pd_5Th_3	TRUE	8	172.54
167651	36	Pd_5Ti_3	TRUE	8	124.90
162013	37	$PrRh_2Si_2$	TRUE	5	84.51
650389	71	$Rh_4Sn_{13}Sr_3$	TRUE	40	967.62
38354	71	S_3U_4	TRUE	7	155.61
651499	71	Sb	TRUE	1	29.84
167903	73	Ta	TRUE	1	18.05
653446	72	W_2Zr	TRUE	6	109.68

3. Additional details about DFT calculations

We used the projector augmented wave (PAW) [4] potentials provided with VASP. For all elements, we used the GW version of the potentials. All VASP calculations were run with PREC=Accurate, avoiding wrap-around errors in the Fourier transform.

For the electronic self-consistency loop, we chose ALGO = FAST. This setting uses a blocked Davidson iteration scheme for the first five electronic minimization steps, followed by a residual minimization

scheme, direct minimization in the iterative subspace (RMM-DIIS). The RMM-DIIS algorithm caused an error in 1138 out of our total of 20196 calculations. For these 1138, we switched to a pure blocked Davidson iteration scheme (i.e. ALGO = NORMAL).

4. Errors that prevented VASP from returning energies.

The following error was observed in two calculations using a SD Γ-centered grid with the highest grid density ($r_{min} = 100$ Å).

Error Text	Number of Occurrences
`VERY BAD NEWS! internal error in subroutine TETIRR:` ` NTET > NTETD   107879`	2

In an additional 2 calculations using a SD Γ-centered grid with the highest grid density, VASP stalled every time it was run. These errors are likely due to the exceptionally high (and arguably unrealistic) grid densities used. In all four of these cases, the average of the energies for the other five methods at $r_{min} = 100$ Å was used in place of the energy for the SD Γ-centered grid at $r_{min} = 100$ Å.

The following errors were observed only in calculations using SD shifted grids. When these errors occurred, the SD shifted grid was replaced with a SD Γ-centered grid that was generated using the same value for *kspacing*.

Error Text	Number of Occurrences

`VERY BAD NEWS! internal error in subroutine IBZKPT:`
` Routine TETIRR needs special values for the k-mesh`
`shifts!     -3`
	400

`VERY BAD NEWS! internal error in subroutine SGRCON:`
` Found some non-integer element in rotation matrix`
`3`
	440

`VERY BAD NEWS! internal error in subroutine IBZKPT:` ` Fatal error: unable to match k-point!      0`	454
`VERY BAD NEWS! internal error in subroutine IBZKPT:` ` Could not get correct shifts      0`	33
`VERY BAD NEWS! internal error in subroutine IBZKPT:` ` Fatal error detecting k-mesh type!   60001`	7

5. How to use the server

In order to easily use the method presented on this paper, we have created a public servlet to generate k-point grids that can be called from any internet connected device. To further simplify the process, a script written in Bash, named "getKPoints", which should work on all UNIX and Unix-like systems, was made to automate the process of requesting, retrieving, and writing the k-point grids into disk. The getKPoints script is provided as a separate file in the supplemental material. As of the writing of this paper, the format that is currently supported is only for grids used by VASP. However, we intend to support additional software packages over time.

Our servlet uses a file named PRECALC that contains the input parameters for the generation of the k-point grids. The currently available parameters are as shown in the table below.

Parameter	Allowed Values	Explanation
INCLUDEGAMMA	TRUE / FALSE / AUTO	Determines whether the grid will be Γ-centered or not. AUTO selects the grid with the smallest number of irreducible k-points. The default is AUTO.
MINDISTANCE*	Numeric (Angstroms)	The value of r_{min} in Angstroms. The default is 0 Å.
HEADER	VERBOSE / SIMPLE	Set whether additional grid information will be written to the header of the file. The default is SIMPLE.
MINTOTALKPOINTS*	Numeric	The minimum value of the desired total k-points. The default is 1.
KPPRA*	Numeric	The minimum allowed number of k-points per reciprocal atom. The use of this parameter for systems with less than three periodic dimensions is not recommended.
GAPDISTANCE	Numeric (Angstroms)	This parameter is used to auto-detect slabs, nanowires, and nanoparticles. If there is a gap (vacuum) that is at least as GAPDISTANCE wide in the provided structure, the k-point density in the corresponding direction will be reduced accordingly. The default value is 7 Å.

*If the PRECALC file does not include at least one of `MINDISTANCE`, `MINTOTALKPOINTS`, or `KPPRA`, then `MINDISTANCE=28.1` will be used to determine grid density.

Below is an example of a PRECALC file:

```
INCLUDEGAMMA = FALSE
MINDISTANCE = 25.7
```

To generate the file containing the *k*-point grids, place the PRECALC file in the same location as the structure (e.g. POSCAR) file and call "getKPoints", which will automatically generate the file containing the *k*-point grid.

6. Values in Figure 6

The table below lists the values shown in Figure 6 in the main paper.

	Non-Metals				Metals		
r_{min}(Å)	Maximum absolute error (left axis, meV/atom)	Average absolute error (left axis, meV/atom)	Average CPU time (right axis, seconds)	r_{min}(Å)	Maximum absolute error (left axis, meV/atom)	Average absolute error (left axis, meV/atom)	Average CPU Time (right axis, seconds)
0.480314	607.1235	55.39885	263.3733	2.480314	20658.95	939.7631	318.1854
2.784058	607.1235	55.39885	263.3733	2.784058	28831.55	1103.229	318.1854
3.125	607.1235	55.39885	263.3733	3.125	20659.09	878.5713	319.0936
3.507694	607.1235	55.39885	263.3733	3.507694	20659.08	792.7009	319.3835
3.937253	607.1235	55.39885	263.3733	3.937253	20658.94	705.4542	319.7326
4.419417	607.1235	54.18077	263.3733	4.419417	20658.92	665.1168	319.8036
4.960628	607.1235	52.06037	271.3574	4.960628	20659.03	658.5679	320.6245
5.568117	452.0103	35.36554	272.8324	5.568117	20659.04	604.1967	323.3993
6.25	336.744	18.28436	273.3797	6.25	20659.08	491.5249	328.2533
7.015388	336.744	12.71835	274.7932	7.015388	20658.95	479.5333	333.4692
7.874507	336.744	12.55861	292.5946	7.874507	20679.45	466.7066	354.0252
8.838835	43.14975	6.169579	361.2262	8.838835	20675.6	438.9394	370.2843
9.921257	43.14975	4.660473	389.4166	9.921257	219.575	24.14219	390.9785
11.13623	43.14975	4.253414	422.5317	11.13623	24927.87	514.7415	468.7653
12.5	20.552	1.484548	529.5233	12.5	114.532	8.53006	573.7892
14.03078	20.552	1.470214	597.8778	14.03078	54.271	6.155463	610.1744
15.74901	20.552	1.409616	750.6962	15.74901	40.116	4.06124	823.1187
17.67767	6.841111	0.409946	1034.541	17.67767	14.5496	2.82389	1030.079
19.84251	2.424	0.295545	1318.216	19.84251	23.3743	2.673938	1590.915
22.27247	2.424	0.25427	1595.621	22.27247	17.6574	1.498976	1856.348
25	2.424	0.150721	2354.969	25	14.394	1.089108	2275.455
28.06155	2.051111	0.11999	3077.215	28.06155	6.759	0.679233	2495.456
31.49803	0.769722	0.083473	3779.091	31.49803	5.1485	0.532098	3224.721
35.35534	0.769722	0.056097	5243.48	35.35534	1.453875	0.311824	4389.24

39.68503	0.367778	0.045734	6898.464	39.68503	1.26625	0.25544	6517.739
44.54494	0.336944	0.028036	9040.797	44.54494	1.051667	0.187206	8298.198
50	0.155	0.025388	12374.01	50	0.569333	0.13113	10581.01
56.1231	0.155	0.019587	17171.98	56.1231	0.474667	0.086801	13700.88
62.99605	0.07175	0.019293	23388.21	62.99605	0.3814	0.059642	19428.73
70.71068	0.063	0.014798	31491.99	70.71068	0.2765	0.045961	26364.53
79.37005	0.057667	0.010922	44350.71	79.37005	0.326	0.037771	37311.22
89.08987	0.04345	0.009393	59764.12	89.08987	0.342667	0.031017	51417.87
100	0	0	82445.44	100	0	0	71123.04

7. References cited.

[1]   H. J. Monkhorst and J. D. Pack, *Special Points for Brillouin-Zone Integrations*, Phys. Rev. B **13**, 5188 (1976).
[2]   A. Baldereschi, *Mean-Value Point in the Brillouin Zone*, Phys. Rev. B **7**, 5212 (1973).
[3]   Inorganic Crystal Structure Database, (Fiz Karlsruhe) http://www.fiz-karlsruhe.de/icsd.html.
[4]   P. E. Blochl, *Projector Augmented-Wave Method*, Phys. Rev. B **50**, 17953 (1994).