Supplementary Materials

Three-Dimensional-Printed ZnO and TiO2 Polymer Composites for UV-A Photocatalysis: Effects of Filler Grade, Loading and Architecture

Nikolaos Raphael Vrithias [1,2], Maria Sevastaki [1], Zacharias Viskadourakis [1], Emmanuel Koudoumas [1,3] and George Kenanakis [1,*]

1 Institute of Electronic Structure and Laser, Foundation for Research and Technology—Hellas, N. Plastira 100, Vassilika Vouton, 70013 Heraklion, Greece

2 Department of Materials Science and Technology, University of Crete, 70013 Heraklion, Greece

3 National Institute for R&D in Microtechnologies-IMT Bucharest, 077190 Bucharest, Romania

* Correspondence: gkenanak@iesl.forth.gr

This document supports the main article. It contains the compositional arithmetic, the complete observed-against-predicted comparison for the surface-exposure model, the sensitivity of the fitted parameters to the assumed particle diameters, the line-width sensitivity calculation, the control measurements that limit what the electron micrographs can be used for, the durability arithmetic behind Section 3.8, and the alternative explanations for the grade ordering. Section numbers prefixed S refer to this document; unprefixed numbers refer to the main text.

S1. Composite compositions

Filler volume fraction φ is obtained from the nominal weight fraction w and the bulk densities as $\varphi = (w/\rho_f) / [(w/\rho_f) + ((1 - w)/\rho_m)]$, with ρ = 1.04 (ABS), 0.95 (HDPE), 3.90 (TiO_2 P25) and 5.61 g cm^{-3} (ZnO, both grades). The volume fractions reported here were calculated from the nominal weight fractions and the stated bulk densities. The two ZnO grades share a density and therefore a φ at equal loading; they differ in the model only through particle diameter.

Table S1. Filler volume fraction φ by matrix, oxide and nominal loading.

Loading (wt%)	ABS/TiO_2	ABS/ZnO	HDPE/TiO_2	HDPE/ZnO
2.5	0.0068	0.0047	0.0062	0.0043
5	0.0138	0.0097	0.0127	0.0088
10	0.0288	0.0202	0.0264	0.0185
20	0.0625	0.0443	0.0574	0.0406

At equal nominal weight fraction, the calculated filler volume fraction is lower in HDPE than in ABS because of the lower matrix density. This difference is included explicitly in the model through φ. Because the adopted rate expression is nonlinear in φ, the volume-fraction ratio should not be used directly as a linear correction to the measured rate ratio.

S2. Coupon geometry and the definition of the effective area

Both architectures were printed as 60 × 30 mm plates — eighteen unit tiles, the printed mesh carrying an 18 × 9 array of 162 apertures measured at 1.6–2.1 mm on a 3.33 mm pitch across the plate — and irradiated as single 10 × 10 × 1 mm tiles cut from them (Section 2.3, Fig. 1a). The mesh coupon carries a 3 × 3 array of 2 × 2 mm square apertures on 1 mm struts; the pillared coupon is a solid 1 mm plate carrying a 4 × 3 array of twelve cylinders of 1 mm diameter and 1 mm height. Four areas can be computed from those dimensions, and they differ by more than two-fold.

Table S2. Candidate definitions of the area a coupon presents, and the ratio each implies. The ideal pillared geometry gives $100 + 12\pi = 137.699$ mm². The model uses the rounded value of 138 mm², giving an adopted effective-area ratio of 138/64 = 2.15625, reported as 2.156. The measured architecture-swap factors are 2.163 (ABS) and 2.174 (HDPE), Table 3.

Definition	Mesh (mm²)	Pillared (mm²)	Ratio	Consistent with observed ratio under proportional-area scaling
Adopted effective-area convention, using rounded model areas: mesh top face; pillared top surfaces and pillar flanks	64	138	**2.156**	**yes**
+ side walls of the mesh apertures	136.0	137.7	1.01	no
Total immersed area (both faces, all edges)	240.0	277.7	1.16	no
Footprint	100.0	100.0	1.00	no

The adopted convention includes the solid top face of the mesh and the top surfaces and pillar flanks of the pillared coupon. It excludes the downward face, outer edges and mesh-aperture walls. The asymmetric treatment of pillar flanks and aperture walls is an approximation, not an independently measured irradiance distribution.

Using the ideal dimensions, the mesh area is $100 - 9 \times 4 = 64$ mm². The pillared area is $100 + 12\pi = 137.7$ mm², rounded to 138 mm² in the model. The resulting adopted effective-area ratio is 2.156; it is the ratio of the rounded model areas, not an exact geometric ratio. The measured architecture-swap ratios are consistent with this convention under proportional-area scaling, but do not establish that it is physically unique.

In the rate expression, the dimensionless effective-area factor A_{eff} is normalised to the mesh and takes values of 1 and 2.156 for the mesh and pillared architectures, respectively. The fitted coefficients therefore refer to this normalised convention rather than to an independently measured active area.

S3. Model equations and fitting procedure

The model represents the filler as spheres of diameter d at volume fraction φ. An effective exclusion depth δ is assumed to limit how closely particle centres approach the nominal polymer–water boundary. A particle contributes an exposed cap when its centre depth z satisfies $\delta \leq z < d/2$. Integrating over a uniform distribution of centre depths gives:

$$a = (3\varphi/4)(1 - 2\delta/d)^2, \text{ for } d > 2\delta, \text{ and } a = 0 \text{ otherwise.} \qquad \text{(S1)}$$

Here a is the modelled exposed oxide area per unit nominal polymer surface area. The parameter δ is an effective exclusion depth, not an independently measured thickness of a continuous polymer coating. At fixed φ, particle diameter enters through the exposure factor $(1-2\delta/d)^2$.

Defining the photolysis-corrected apparent increment as $\Delta k = k_{app} - k_{photolysis}$, the adopted rate expression is:

$$\Delta k = \kappa_{ox} \cdot A_{eff} \cdot a / (1 + a/a_0), \qquad \text{(S2)}$$

The coefficient κ_{ox} is an effective fitted coefficient, with one coefficient for TiO_2 and one shared by the two ZnO grades. Sharing a coefficient between the ZnO grades is a model assumption that assigns their contrast to accessibility rather than to intrinsic differences between commercial powders. The parameter a_0 controls saturation, and A_{eff} equals 1 for meshes and 2.156 for pillared plates.

The adopted fit uses 26 published mean observations and kph = 0.00341 min^{-1}. The grid has δ = 1–40 nm in 0.25 nm steps and 121 logarithmic a0 values from 0.001 to 1. Nonnegative oxide coefficients are solved analytically; fits are unweighted because SDs are unavailable for the full set. Exploratory continuous variants use bounded multistart SciPy $least_{squares}$ with bounds and starts specified in Code S1. Total-area saturation uses 121 logarithmic b0 values from 0.1 to 1000 mm^2. Intermediate values retain full precision.

Assumed structural diameters are 5000 nm for coarse ZnO, 100 nm for fine ZnO and 200 nm for TiO_2. Final parameters are δ = 18.25 nm, a0 = 0.0237137, κ_{ZnO} = 1.309059 min^{-1} and κ_{TiO_2} = 0.633108 min^{-1}, R^2 = 0.899661. The 200 nm value is an assumption, not a uniquely identified size. Retained Figures 6 and S1 illustrate reference parameters δ = 18.0 nm, a0 = 0.0251, κ_{ZnO} = 1.272 min^{-1} and κ_{TiO_2} = 0.615 min^{-1}, not the exact final solution. Final predictions are in the updated tables and code outputs.

S4. Observed and predicted rate increments

Table S3. Final observed/model-predicted operational increments for 26 composition–architecture combinations at k_{ph} = 3.41 × 10^{-3} min^{-1}. Exposure a uses δ = 18.25 nm and the assumed structural diameters. Aeff is dimensionless: mesh 1, pillars 2.156. The final column is predicted/observed increment, not an uncertainty. These predictions are distinguished from the retained reference-parameter graphics.

Matrix	Oxide	wt%	φ	Geometry	*a* (×10^{-3})	Δ*k* obs (10^{-3} min^{-1})	Δk model (10^{-3} min^{-1})	model/obs
ABS	TiO_2	2.5	0.00679	mesh	3.404	1.140	1.885	1.653
ABS	TiO_2	5	0.01384	mesh	6.937	2.400	3.398	1.416
ABS	TiO_2	10	0.02878	mesh	14.424	8.860	5.678	0.641
ABS	TiO_2	20	0.06250	mesh	31.327	9.870	8.545	0.866
ABS	ZnOmicro	2.5	0.00473	mesh	3.497	0.310	3.989	12.868
ABS	ZnOmicro	5	0.00966	mesh	7.142	0.730	7.185	9.842
ABS	ZnOmicro	10	0.02018	mesh	14.917	18.410	11.987	0.651
ABS	ZnOmicro	20	0.04429	mesh	32.737	19.830	18.002	0.908
ABS	ZnOmicro	20	0.04429	pillars	32.737	42.900	38.813	0.905
ABS	ZnOnano	2.5	0.00473	mesh	1.431	0.310	1.766	5.698
ABS	ZnOnano	5	0.00966	mesh	2.922	1.950	3.406	1.746
ABS	ZnOnano	10	0.02018	mesh	6.104	8.830	6.354	0.720
ABS	ZnOnano	20	0.04429	mesh	13.395	9.530	11.205	1.176
HDPE	TiO_2	2.5	0.00621	pillars	3.111	2.090	3.754	1.796
HDPE	TiO_2	5	0.01266	pillars	6.345	4.060	6.832	1.683
HDPE	TiO_2	10	0.02635	pillars	13.209	14.380	11.580	0.805
HDPE	TiO_2	20	0.05740	pillars	28.771	15.950	17.744	1.112
HDPE	ZnOmicro	2.5	0.00432	pillars	3.195	8.990	7.947	0.884
HDPE	ZnOmicro	5	0.00883	pillars	6.529	9.900	14.449	1.459
HDPE	ZnOmicro	10	0.01847	pillars	13.650	29.370	24.450	0.832
HDPE	ZnOmicro	20	0.04062	mesh	30.019	14.620	17.343	1.186
HDPE	ZnOmicro	20	0.04062	pillars	30.019	31.790	37.391	1.176
HDPE	ZnOnano	2.5	0.00432	pillars	1.307	5.100	3.497	0.686
HDPE	ZnOnano	5	0.00883	pillars	2.672	6.550	6.777	1.035
HDPE	ZnOnano	10	0.01847	pillars	5.585	17.810	12.758	0.716
HDPE	ZnOnano	20	0.04062	pillars	12.283	19.700	22.837	1.159

23 of 26 predictions lie within a factor of two of observations; median predicted/observed ratio is 1.136. This global summary does not validate local loading trends.

S5. Residual structure

The residuals are not random: they are organised by matrix and loading, and the organisation is what identifies the model's single systematic failure.

Table S4. Median, minimum and maximum of model/observed Δ*k* within each matrix–loading cell.

Matrix	Loading (wt%)	*n*	Median model/obs	Min	Max
ABS	2.5	3	5.698	1.653	12.868
ABS	5	3	1.746	1.416	9.842
ABS	10	3	0.651	0.641	0.720
ABS	20	4	0.906	0.866	1.176
HDPE	2.5	3	0.884	0.686	1.796
HDPE	5	3	1.459	1.035	1.683
HDPE	10	3	0.805	0.716	0.832
HDPE	20	4	1.168	1.112	1.186

Updated group statistics are in Table S4. Relative discrepancies are largest for low-loading ABS/ZnO near photolysis, where small observed increments make ratios sensitive. The retained Figure S1 is a reference-parameter illustration, not the exact plot of these updated statistics.

The largest relative discrepancies occur for low-loading ABS/ZnO composites. At 2.5 wt% coarse-grade ZnO, the observed increments are 0.31 × 10^{-3} min^{-1} for the ABS mesh and 8.99 × 10^{-3} min^{-1} for the HDPE pillared plate. This raw comparison differs in both matrix and architecture and should not be interpreted as an isolated matrix effect.

The model captures the overall response range but does not describe the onset of activity in low-loading ABS. Relative prediction errors near the photolysis baseline must also be interpreted cautiously because the observed increment is small. The absence of a retained matrix term in the reported fit does not demonstrate equality of the two polymer systems.

Figure S1 illustrates this loading-dependent pattern graphically, showing both the parity between predicted and observed Δk values and the relative residuals as a function of filler loading.

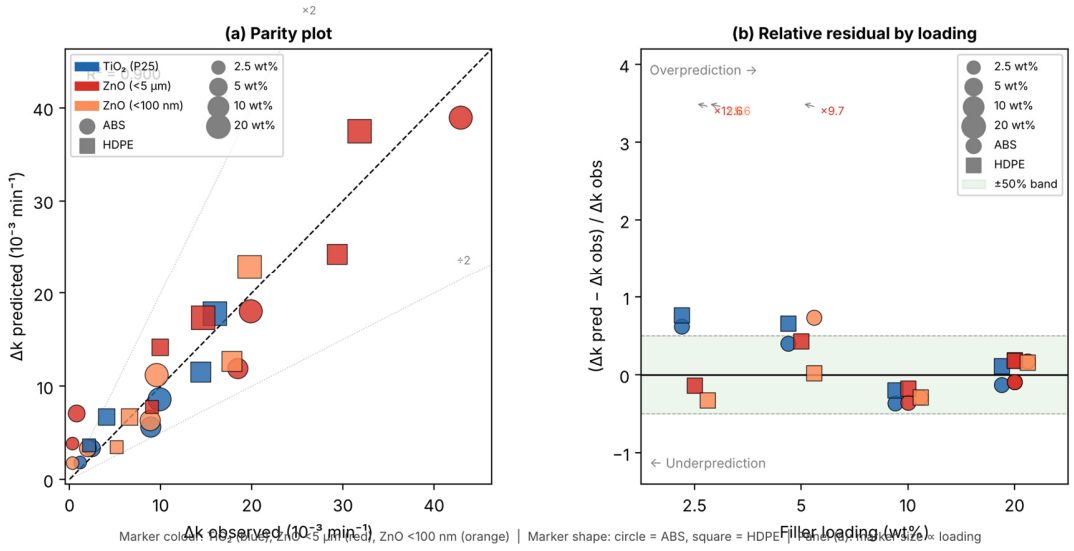

Figure S1. Reference-parameter residual illustration, retained in its original graphical form. (a) Parity and factor-of-two guides; (b) relative residual (predicted−observed)/observed versus loading with the original ±50% band. Reference parameters: δ = 18.0 nm, a0 = 0.0251, κ_{TiO_2} = 0.615 min^{-1}, κ_{ZnO} = 1.272 min^{-1}. Colours denote fillers and markers matrices. Final numerical predictions and summaries are in Tables S3–S4; this is not their exact residual plot. Reference and final predicted increments differ by at most approximately 2.2%. Structured errors do not establish a connectivity mechanism. *Data were plotted and annotated in OriginPro 2021 (OriginLab Corporation, Northampton, MA, USA); final panel layout was assembled in Affinity Photo 2 (v2.6.5; Serif Europe Ltd.).*

S6. Sensitivity of the fitted parameters to the assumed particle diameters

A conditional sensitivity analysis was performed using assumed coarse- and fine-grade ZnO diameters of 200–750 nm and 60–100 nm, respectively. These are exploratory model inputs, not experimentally established particle-size intervals. The TiO_2 diameter was held at 200 nm. The analysis evaluates how the fitted parameters change when the assumed ZnO diameters are varied.

Table S5. Complete refit at each combination of assumed diameters. The TiO_2 structural diameter is held at 200 nm throughout.

d_{micro} (nm)	d_{nano} (nm)	δ (nm)	δ/d_{nano}	a_0 (×10^{-2})	$\kappa_{TiO_2}/\kappa_{ZnO}$	R^2
200	60	13.00	0.2167	1.9953	0.3965	0.898600
200	80	18.75	0.2344	1.7783	0.3964	0.898584
200	100	26.00	0.2600	1.4125	0.3965	0.898587
350	60	12.00	0.2000	2.2387	0.4190	0.899043
350	80	16.50	0.2062	2.1135	0.4304	0.899221
350	100	22.00	0.2200	1.8836	0.4443	0.899371
500	60	11.50	0.1917	2.3714	0.4272	0.899167
500	80	15.75	0.1969	2.2387	0.4416	0.899363
500	100	20.50	0.2050	2.1135	0.4588	0.899521
750	60	11.25	0.1875	2.3714	0.4333	0.899262
750	80	15.25	0.1906	2.3714	0.4495	0.899419
750	100	19.50	0.1950	2.2387	0.4687	0.899596

Conditional diameter refits gave R^2 = 0.898584–0.899596, coefficient ratios 0.3964–0.4687, δ = 11.25–26.00 nm and δ/dnano = 0.1875–0.2600. Similar fit quality does not identify unique physical diameters.

These ranges describe sensitivity to assumed structural diameters, not independently measured particle sizes or skin thicknesses. The threshold d ≈ 19δ corresponds to 80% exposure efficiency within Eq. (S1), but does not establish an experimentally validated size specification.

A nonparametric pairs bootstrap of the 26 published mean observations used 2000 resamples, seed 20261116, and the same full grid search for each resample. Conditional percentile 95% intervals were 6.50–24.50 nm for δ, 0.010593–0.118850 for a_0 and 0.3625–0.6255 for $\kappa_{TiO_2}/\kappa_{ZnO}$ (point estimate 0.4836). These intervals resample formulation-level means; they do not incorporate within-formulation measurement uncertainty, diameter uncertainty or uncertainty in the area convention, and do not independently establish a physical skin thickness.

S7. Line-width sensitivity at the ZnO (101) reflection

Scherrer broadening at the ZnO (101) reflection (2θ = 36.25°, λ = 1.54 Å, K = 0.9), convolved in quadrature with a plausible instrumental width for a sealed-tube laboratory diffractometer.

Table S6. Size broadening and total line width at ZnO (101).

Crystallite size (nm)	Scherrer β (°)	Total width at 0.08° instrumental	at 0.12°
20	0.418	0.425	0.435

50	0.167	0.185	0.206
62	0.135	0.157	0.180
93	0.090	0.120	0.150
100 (nominal <100 nm)	0.084	0.116	0.146
200	0.042	0.090	0.127
500	0.017	0.082	0.121
1000	0.008	0.080	0.120
5000 (nominal <5 μm)	0.002	0.080	0.120

These calculations illustrate the expected contribution of crystallite size to line broadening under assumed instrumental widths. They are not fits to the measured diffraction profiles. Because the instrumental broadening function was not measured, the present diffraction data are not used to establish crystallite-size bounds for either ZnO grade. In particular, the 60–100 nm interval explored in Table S5 is an assumed sensitivity range; the supplier specification provides an upper size limit but does not establish its lower bound.

S8. Electron microscopy: detector settings and the unfilled control

One local-contrast feature detector was applied to every micrograph with settings specified in nanometres — a 470, 700 or 1050 nm analysis window and a contrast margin of 14, 22 or 30 grey levels — and converted to pixels separately for each image, so that specimens recorded at different magnifications are compared on the same physical scale. An unfilled ABS specimen, imaged under the same conditions, was passed through the identical procedure.

Table S7. Detected area fraction and feature-size percentiles, ABS specimens. The unfilled reference contains no oxide, so everything it returns is fracture topography.

Specimen	Area fraction, margin 22	margin 14	p50 (nm)	p75	p90	Largest compact objects (nm)
Unfilled ABS (reference)	3.7%	7.4%	59	90	139	340, 176, 152, 152, 152
ZnO_{micro} 5 wt%	0.7%	2.0%	—	—	—	—
ZnO_{micro} 20 wt%	5.6%	10.0%	66	107	186	756, 395, 349, 302, 279
ZnO_{nano} 20 wt%	—	—	—	—	—	395, 349, 279, 267, 209

The unfilled ABS reference produces a detected area fraction and apparent feature-size distribution that overlap those of the filled specimens. These detector outputs therefore cannot be interpreted as quantitative oxide coverage or particle-size distributions. They also do not provide a reliable ratio of the particle sizes of the two ZnO grades.

The largest detected compact objects in the coarse-grade specimen extend approximately 400–750 nm, while the largest detected object in the fine-grade specimen is approximately 395 nm. These are apparent feature dimensions in the analysed images, not upper bounds on the full particle populations. Morphology and specimen composition support their assignment to filler-related features, but no compositional mapping was performed.

The images were acquired in secondary-electron mode from Au/Pd-coated fracture surfaces. Contrast is therefore not a direct quantitative indicator of oxide composition or exposure. The fracture surfaces also differ from the as-printed surfaces contacted by water during irradiation. These measurements do not determine the polymer exclusion depth or accessible oxide area.

S9. Alternative explanations for the grade ordering

The coarse commercial ZnO grade generally gave the larger apparent catalytic increment in the clearly active comparisons. The exposure model represents this ordering through size-dependent accessibility, but the experiments do not isolate particle size from other grade-dependent properties.

Differences in aggregation and dispersion during melt processing may alter the effective structural units and accessible surfaces of the powders. The single-screw processing route may also produce different dispersion states for the two grades. Aggregation was not quantified independently and cannot be excluded as a contributor to the ordering [60].

Differences in defect populations, surface chemistry and charge-carrier behaviour may also contribute. Both powders show the dominant wurtzite phase, but diffraction alone does not demonstrate identical interfacial or electronic properties.

A suspension comparison of the same ZnO grades at matched mass loading would help distinguish intrinsic grade-dependent performance from composite-specific effects. A reversal of the grade ordering between suspension and printed formats would support a role for composite accessibility, but would not by itself establish a unique burial mechanism or determine δ. Reference numbers correspond to the main article bibliography.

S10. Re-use results and incident-dose normalisation

Cycle-specific kapp values were fitted by ordinary least squares to $\ln(A_t/A_0)$ at seven times using the displayed one-decimal Table S9 profiles, not original unrounded UV–Vis exports. For re-use, ln(kn) was regressed against n−1 with a freely fitted intercept in $kn = K \exp[-\lambda(n-1)]$. ABS mesh: $\lambda = 0.1218$ cycle^{-1}, $R^2 = 0.933$, fitted loss = 11.5% per cycle and cycle-5 retention = 63.8%; HDPE pillars: $\lambda = 0.1291$ cycle^{-1}, $R^2 = 0.940$, fitted loss = 12.1% per cycle and cycle-5 retention = 63.6%. Retention is 100 kapp,5/kapp,1 without photolysis subtraction. Arithmetic means over cycles 1–5 are 19.705×10^{-3} min^{-1} for ABS and 28.329×10^{-3} min^{-1} for HDPE; sample SDs between cycles are 3.775 and 5.952×10^{-3} min^{-1}. These describe the five successive uses, including activity decline, and are not independent-replicate means or repeatability estimates.

$$H = It \quad \text{(S3)}$$

At reported irradiance 6.0 mW cm^{-2}, each 60 min cycle corresponds to nominal incident dose 21.6 J cm^{-2} and five cycles to 108 J cm^{-2}. λ/21.6 is descriptive. These doses refer to the reported radiometer-band irradiance, not calibrated full-spectrum Hg-lamp output.

Each re-use trajectory follows one coupon through five uses. Between-cycle SD includes declining activity and is not the SD of five independent fresh-coupon experiments. Re-use rates were recalculated from displayed rounded profiles, with HDPE cycle 2 corrected to the supplied seven values. Dose reciprocity and extrapolation beyond five cycles are not established.

Table S8. Updated five-cycle re-use results, using cycle 1 from S9-A and cycles 2–5 from S9-D. HDPE cycle 2 has been corrected. Retention = 100 kapp,5/kapp,1. Arithmetic cycle means and sample SDs are descriptive between-cycle quantities, not the independent mean ± SD values of Table 1.

Quantity	ABS/coarse-grade ZnO 20 wt%, mesh	HDPE/coarse-grade ZnO 20 wt%, pillars
Fitted λ (cycle^{-1})	0.1218	0.1291
Fitted loss per cycle (%)	11.5	12.1
Incident dose per cycle (J cm^{-2})	21.6	21.6
Total dose over five cycles (J cm^{-2})	108	108
Descriptive λ/dose (10^{-3} cm^2 J^{-1})	5.64	5.98
kapp retention at cycle 5 (%)	63.8	63.6
Re-use fit R^2 (log space)	0.933	0.940
Arithmetic mean over cycles 1–5 (10^{-3} min^{-1})	19.705	28.329
Sample SD between cycles (10^{-3} min^{-1})	3.775	5.952

S11. Raw kinetic profiles

S9-A contains individual first-use profiles (cycle 1), B the two architecture comparisons, C a mean photolysis profile from eight controls, and D four later uses (cycles 2–5) of the same selected coupons. Cycle 1 is not repeated in D. The corrected HDPE cycle-2 values are 100.0, 71.2, 50.4, 35.4, 25.7, 18.5 and 13.2 at 0–60 min. Table 1 means derive from five independent experiments, not the successive re-use cycles. Table S10 supplies independent constants for the two 20 wt% coarse-ZnO main-series formulations; full replicate profiles and row-specific SDs for the remaining formulations are not supplied.

Table S9. Measured normalised integrated absorbance band areas (220–320 nm) for the profiles presented. Values are 100 At/A0, each irradiation normalised to its own initial band area. Section A contains individual first-use runs, not averages of five runs. For the selected ABS/coarse-ZnO and HDPE/coarse-ZnO 20 wt% coupons, cycle 1 is in A and cycles 2–5 in D. Section C is a mean profile, not eight separately listed control profiles.

Formulation	Architecture	Note	0 min	10 min	20 min	30 min	40 min	50 min	60 min
A. Main composition series (24 formulations; ABS mesh / HDPE pillared plate)									
ABS / TiO_2 P25 / 2.5 wt %	Mesh	kapp = 4.55 × 10^{-3} min^{-1}	100.0	95.6	91.3	87.2	83.4	79.7	76.1
ABS / TiO_2 P25 /	Mesh	kapp = 5.81 × 10^{-3} min^{-1}	100.0	94.4	89.0	84.0	79.3	74.8	70.6

5 wt %									
ABS / TiO_2 P25 / 10 wt %	Mesh	kapp = 12.27×10^{-3} min^{-1}	100.0	88.5	78.2	69.2	61.2	54.1	47.9
ABS / TiO_2 P25 / 20 wt %	Mesh	kapp = 13.28×10^{-3} min^{-1}	100.0	87.6	76.7	67.1	58.8	51.5	45.1
ABS / ZnO (<5 µm) / 2.5 wt %	Mesh	kapp = 3.72×10^{-3} min^{-1}	100.0	96.3	92.8	89.4	86.2	83.0	80.0
ABS / ZnO (<5 µm) / 5 wt %	Mesh	kapp = 4.14×10^{-3} min^{-1}	100.0	95.9	92.1	88.3	84.7	81.3	78.0
ABS / ZnO (<5 µm) / 10 wt %	Mesh	kapp = 21.82×10^{-3} min^{-1}	100.0	80.4	64.6	52.0	41.8	33.6	27.0
ABS / ZnO (<5 µm) / 20 wt %	Mesh	kapp = 23.24×10^{-3} min^{-1}	100.0	79.3	62.8	49.8	39.5	31.3	24.8
ABS / ZnO (<100 nm) / 2.5 wt %	Mesh	kapp = 3.72×10^{-3} min^{-1}	100.0	96.3	92.8	89.4	86.2	83.0	80.0
ABS / ZnO (<100 nm) / 5 wt %	Mesh	kapp = 5.36×10^{-3} min^{-1}	100.0	94.8	89.8	85.1	80.7	76.5	72.5
ABS / ZnO (<100 nm) / 10 wt %	Mesh	kapp = 12.24×10^{-3} min^{-1}	100.0	88.5	78.3	69.3	61.3	54.2	48.0
ABS / ZnO (<100 nm) / 20 wt %	Mesh	kapp = 12.94×10^{-3} min^{-1}	100.0	87.9	77.2	67.8	59.6	52.4	46.0
HDPE / TiO_2 P25 / 2.5 wt %	Pillared plate	kapp = 5.5×10^{-3} min^{-1}	100.0	94.6	89.6	84.8	80.3	76.0	71.9
HDPE / TiO_2 P25 / 5 wt %	Pillared plate	kapp = 7.47×10^{-3} min^{-1}	100.0	92.8	86.1	79.9	74.2	68.8	63.9
HDPE / TiO_2 P25 / 10 wt %	Pillared plate	kapp = 17.79×10^{-3} min^{-1}	100.0	83.7	70.1	58.6	49.1	41.1	34.4
HDPE / TiO_2 P25 / 20 wt %	Pillared plate	kapp = 19.36×10^{-3} min^{-1}	100.0	82.4	67.9	55.9	46.1	38.0	31.3
HDPE / ZnO (<5 µm) / 2.5 wt %	Pillared plate	kapp = 12.4×10^{-3} min^{-1}	100.0	88.3	78.0	68.9	60.9	53.8	47.5
HDPE / ZnO (<5 µm) / 5 wt %	Pillared plate	kapp = 13.31×10^{-3} min^{-1}	100.0	87.5	76.6	67.1	58.7	51.4	45.0
HDPE / ZnO (<5 µm) / 10 wt %	Pillared plate	kapp = 32.78×10^{-3} min^{-1}	100.0	72.1	51.9	37.4	26.9	19.4	14.0
HDPE / ZnO (<5 µm) / 20 wt %	Pillared plate	kapp = 35.2×10^{-3} min^{-1}	100.0	70.3	49.5	34.8	24.5	17.2	12.1
HDPE / ZnO (<100 nm) / 2.5 wt %	Pillared plate	kapp = 8.51×10^{-3} min^{-1}	100.0	91.8	84.3	77.5	71.1	65.3	60.0
HDPE / ZnO (<100 nm) / 5 wt %	Pillared plate	kapp = 9.96×10^{-3} min^{-1}	100.0	90.5	81.9	74.2	67.1	60.8	55.0
HDPE / ZnO (<100 nm) / 10 wt %	Pillared plate	kapp = 21.22×10^{-3} min^{-1}	100.0	80.9	65.4	52.9	42.8	34.6	28.0
HDPE / ZnO (<100 nm) / 20 wt %	Pillared plate	kapp = 23.11×10^{-3} min^{-1}	100.0	79.4	63.0	50.0	39.7	31.5	25.0
B. Architecture-comparison runs (ZnO < 5 µm, 20 wt %, swapped architecture)									
ABS / ZnO (<5 µm) / 20 wt % → Pillared plate	Pillared plate	kapp = 46.31×10^{-3} min^{-1}	100.0	62.9	39.6	24.9	15.7	9.9	6.2
HDPE / ZnO (<5 µm) / 20 wt % → Mesh	Mesh	kapp = 18.03×10^{-3} min^{-1}	100.0	83.5	69.7	58.2	48.6	40.6	33.9
C. Photolysis controls (no specimen; kapp = 3.41×10^{-3} min^{-1}, mean of 8 runs)									
No specimen (photolysis control)	—	Mean of 8 control runs (individual range: 3.2–3.5)	100.0	96.6	93.4	90.3	87.2	84.3	81.5
D. Re-use series: cycles 2–5 for ABS/ZnO <5 µm and HDPE/ZnO <5 µm, 20 wt%. Each specimen underwent four further irradiation cycles following its first use in Section A.									
ABS / ZnO < 5 µm / 20 wt %	Mesh	Cycle 2	100.0	79.3	62.9	49.9	39.5	31.3	24.9
ABS / ZnO < 5 µm / 20 wt %	Mesh	Cycle 3	100.0	81.5	66.5	54.2	44.2	36.1	29.4
ABS / ZnO < 5 µm /	Mesh	Cycle 4	100.0	84.5	71.3	60.2	50.9	43.0	36.3

20 wt %									
ABS / ZnO < 5 µm / 20 wt %	Mesh	Cycle 5	100.0	86.2	74.4	64.1	55.3	47.7	41.1
HDPE / ZnO < 5 µm / 20 wt %	Pillared plate	Cycle 2	100.0	71.2	50.4	35.4	25.7	18.5	13.2
HDPE / ZnO < 5 µm / 20 wt %	Pillared plate	Cycle 3	100.0	76.0	57.8	44.0	33.4	25.4	19.3
HDPE / ZnO < 5 µm / 20 wt %	Pillared plate	Cycle 4	100.0	79.5	63.3	50.3	40.0	31.8	25.3
HDPE / ZnO < 5 µm / 20 wt %	Pillared plate	Cycle 5	100.0	79.9	63.9	51.1	40.8	32.6	26.1

Table S10. Independent kapp values for the two 20 wt% coarse-ZnO main-series formulations. Constants, means and sample SDs are in 10^{-3} min^{-1}. These are five independent experiments, not the five re-use cycles; original replicate spectra are not included.

Independent run/statistic	ABS / coarse ZnO / 20 wt% / mesh	HDPE / coarse ZnO / 20 wt% / pillars
1	22.56	34.20
2	22.98	34.80
3	23.24	35.20
4	23.52	35.60
5	23.90	36.20
Mean	23.24	35.20
Sample SD (n−1)	0.5109	0.7616
CV (%)	2.1983	2.1636
n	5	5

Code S1. Final analysis including corrected HDPE cycle 2

Code S1 provides a self-contained Python implementation of the reported numerical analyses. All required numerical inputs are embedded in the code as CSV-formatted text strings; no separately supplied input files are required. The inputs comprise 26 published mean composition-architecture observations, 35 displayed kinetic profiles from Table S9 and ten independent rate constants from Table S10. These datasets are analysed separately. The HDPE cycle-2 profile is 100.0, 71.2, 50.4, 35.4, 25.7, 18.5 and 13.2 at 0, 10, 20, 30, 40, 50 and 60 min.

Copy the entire code below into a plain-text file named code_s1.py. Install NumPy, pandas, SciPy and threadpoolctl. Execute: python code_s1.py --out results --stage all. The script generates its output files locally; no input CSV files need to be supplied. The stages base, bootstrap and sensitivity may also be executed separately.

The embedded inputs reproduce analyses of the published means, displayed rounded profiles and the two supplied sets of independent constants. They are not the original UV-Vis spectral exports or a complete replicate-level dataset for all formulations. No missing SDs or measurements are generated. Bootstrap intervals are conditional on structural diameters and the area convention; nominal nonlinear matrix F tests remain exploratory.

Tested package versions: numpy 1.26.4, pandas 2.2.3, scipy 1.13.1, threadpoolctl 3.6.0.

Code begins on the next page. Copy only the code, starting with #!/usr/bin/env python3 and ending with main(). Do not add line numbers or replace ordinary spaces with non-breaking spaces.

```
#!/usr/bin/env python3
"""Code S1: self-contained numerical analysis.

Run: python code_s1.py --out results --stage all
Dependencies: numpy, pandas, scipy, threadpoolctl.
No separate input files or network access are required.
All three input datasets are embedded below as CSV-formatted strings.

Scope: 26 published mean composition-architecture observations;
35 displayed Table S9 profiles, including the corrected HDPE cycle 2;
and 10 independent replicate constants from Table S10.
Published means, individual profiles and independent replicates are distinct.
Original UV-Vis spectra and missing replicate SDs are not reconstructed.
Outputs are created locally; they are not required as pre-existing inputs.
"""
import argparse
import json
from io import StringIO
from pathlib import Path

import numpy as np
import pandas as pd
from scipy.stats import linregress, f as fdist
from scipy.optimize import least_squares
from threadpoolctl import threadpool_limits

KPH = 0.00341
RM = {"ABS": 1.04, "HDPE": 0.95}
RF = {"TiO2": 3.90, "ZnO-micro": 5.61, "ZnO-nano": 5.61}
DIAM = {"TiO2": 200.0, "ZnO-micro": 5000.0, "ZnO-nano": 100.0}
AREA = {"mesh": 1.0, "pillars": 2.156}
DELTA = np.arange(1.0, 40.0001, 0.25)
A0 = np.logspace(-3, 0, 121)
TIMES = np.arange(0, 61, 10, dtype=float)

MEANS_CSV = """
Formulation,Matrix,Filler,Loading_wt,Architecture,Series,kapp_mean,n,Source
ABS_TiO2_2.5,ABS,TiO2,2.5,mesh,main,0.00455,5,Table 1 published mean
ABS_TiO2_5,ABS,TiO2,5.0,mesh,main,0.00581,5,Table 1 published mean
ABS_TiO2_10,ABS,TiO2,10.0,mesh,main,0.01227,5,Table 1 published mean
ABS_TiO2_20,ABS,TiO2,20.0,mesh,main,0.01328,5,Table 1 published mean
ABS_ZnO-micro_2.5,ABS,ZnO-micro,2.5,mesh,main,0.00372,5,Table 1 published mean
ABS_ZnO-micro_5,ABS,ZnO-micro,5.0,mesh,main,0.00414,5,Table 1 published mean
ABS_ZnO-micro_10,ABS,ZnO-micro,10.0,mesh,main,0.02182,5,Table 1 published mean
ABS_ZnO-micro_20,ABS,ZnO-micro,20.0,mesh,main,0.02324,5,Table 1 published mean
ABS_ZnO-nano_2.5,ABS,ZnO-nano,2.5,mesh,main,0.00372,5,Table 1 published mean
```

```
ABS_ZnO-nano_5,ABS,ZnO-nano,5.0,mesh,main,0.00536,5,Table 1 published mean
ABS_ZnO-nano_10,ABS,ZnO-nano,10.0,mesh,main,0.012240000000000001,5,Table 1 published mean
ABS_ZnO-nano_20,ABS,ZnO-nano,20.0,mesh,main,0.01294,5,Table 1 published mean
HDPE_TiO2_2.5,HDPE,TiO2,2.5,pillars,main,0.0055,5,Table 1 published mean
HDPE_TiO2_5,HDPE,TiO2,5.0,pillars,main,0.00747,5,Table 1 published mean
HDPE_TiO2_10,HDPE,TiO2,10.0,pillars,main,0.01779,5,Table 1 published mean
HDPE_TiO2_20,HDPE,TiO2,20.0,pillars,main,0.01936,5,Table 1 published mean
HDPE_ZnO-micro_2.5,HDPE,ZnO-micro,2.5,pillars,main,0.012400000000000001,5,Table 1 published mean
HDPE_ZnO-micro_5,HDPE,ZnO-micro,5.0,pillars,main,0.01331,5,Table 1 published mean
HDPE_ZnO-micro_10,HDPE,ZnO-micro,10.0,pillars,main,0.032780000000000004,5,Table 1 published mean
HDPE_ZnO-micro_20,HDPE,ZnO-micro,20.0,pillars,main,0.0352,5,Table 1 published mean
HDPE_ZnO-nano_2.5,HDPE,ZnO-nano,2.5,pillars,main,0.00851,5,Table 1 published mean
HDPE_ZnO-nano_5,HDPE,ZnO-nano,5.0,pillars,main,0.009960000000000002,5,Table 1 published mean
HDPE_ZnO-nano_10,HDPE,ZnO-nano,10.0,pillars,main,0.02122,5,Table 1 published mean
HDPE_ZnO-nano_20,HDPE,ZnO-nano,20.0,pillars,main,0.02311,5,Table 1 published mean
ABS_ZnO-micro_20_swap,ABS,ZnO-micro,20.0,pillars,arch-swap,0.046310000000000004,,Table 3 Delta k plus kph; replicate
count not established
HDPE_ZnO-micro_20_swap,HDPE,ZnO-micro,20.0,mesh,arch-swap,0.018029999999999997,,Table 3 Delta k plus kph; replicate
count not established
"""

PROFILES_CSV = """
Formulation,Matrix,Filler,Loading_wt,Architecture,Series,Coupon_ID,Run_ID,Cycle,Profile_kind,t0,t10,t20,t30,t40,t50,t6
0
ABS_TiO2_2.5,ABS,TiO2,2.5,mesh,main,firstuse_2,S9_row_2,1,individual_displayed_profile,100.0,95.6,91.3,87.2,83.4,79.7,
76.1
ABS_TiO2_5,ABS,TiO2,5.0,mesh,main,firstuse_3,S9_row_3,1,individual_displayed_profile,100.0,94.4,89.0,84.0,79.3,74.8,70
.6
ABS_TiO2_10,ABS,TiO2,10.0,mesh,main,firstuse_4,S9_row_4,1,individual_displayed_profile,100.0,88.5,78.2,69.2,61.2,54.1,
47.9
ABS_TiO2_20,ABS,TiO2,20.0,mesh,main,firstuse_5,S9_row_5,1,individual_displayed_profile,100.0,87.6,76.7,67.1,58.8,51.5,
45.1
ABS_ZnO-micro_2.5,ABS,ZnO-
micro,2.5,mesh,main,firstuse_6,S9_row_6,1,individual_displayed_profile,100.0,96.3,92.8,89.4,86.2,83.0,80.0
ABS_ZnO-micro_5,ABS,ZnO-
micro,5.0,mesh,main,firstuse_7,S9_row_7,1,individual_displayed_profile,100.0,95.9,92.1,88.3,84.7,81.3,78.0
ABS_ZnO-micro_10,ABS,ZnO-
micro,10.0,mesh,main,firstuse_8,S9_row_8,1,individual_displayed_profile,100.0,80.4,64.6,52.0,41.8,33.6,27.0
ABS_ZnO-micro_20,ABS,ZnO-
micro,20.0,mesh,main,ABS_reuse,S9_row_9,1,individual_displayed_profile,100.0,79.3,62.8,49.8,39.5,31.3,24.8
ABS_ZnO-nano_2.5,ABS,ZnO-
nano,2.5,mesh,main,firstuse_10,S9_row_10,1,individual_displayed_profile,100.0,96.3,92.8,89.4,86.2,83.0,80.0
ABS_ZnO-nano_5,ABS,ZnO-
nano,5.0,mesh,main,firstuse_11,S9_row_11,1,individual_displayed_profile,100.0,94.8,89.8,85.1,80.7,76.5,72.5
ABS_ZnO-nano_10,ABS,ZnO-
nano,10.0,mesh,main,firstuse_12,S9_row_12,1,individual_displayed_profile,100.0,88.5,78.3,69.3,61.3,54.2,48.0
ABS_ZnO-nano_20,ABS,ZnO-
nano,20.0,mesh,main,firstuse_13,S9_row_13,1,individual_displayed_profile,100.0,87.9,77.2,67.8,59.6,52.4,46.0
HDPE_TiO2_2.5,HDPE,TiO2,2.5,pillars,main,firstuse_14,S9_row_14,1,individual_displayed_profile,100.0,94.6,89.6,84.8,80.
3,76.0,71.9
HDPE_TiO2_5,HDPE,TiO2,5.0,pillars,main,firstuse_15,S9_row_15,1,individual_displayed_profile,100.0,92.8,86.1,79.9,74.2,
68.8,63.9
HDPE_TiO2_10,HDPE,TiO2,10.0,pillars,main,firstuse_16,S9_row_16,1,individual_displayed_profile,100.0,83.7,70.1,58.6,49.
1,41.1,34.4
```

```
HDPE_TiO2_20,HDPE,TiO2,20.0,pillars,main,firstuse_17,S9_row_17,1,individual_displayed_profile,100.0,82.4,67.9,55.9,46.
1,38.0,31.3
HDPE_ZnO-micro_2.5,HDPE,ZnO-
micro,2.5,pillars,main,firstuse_18,S9_row_18,1,individual_displayed_profile,100.0,88.3,78.0,68.9,60.9,53.8,47.5
HDPE_ZnO-micro_5,HDPE,ZnO-
micro,5.0,pillars,main,firstuse_19,S9_row_19,1,individual_displayed_profile,100.0,87.5,76.6,67.1,58.7,51.4,45.0
HDPE_ZnO-micro_10,HDPE,ZnO-
micro,10.0,pillars,main,firstuse_20,S9_row_20,1,individual_displayed_profile,100.0,72.1,51.9,37.4,26.9,19.4,14.0
HDPE_ZnO-micro_20,HDPE,ZnO-
micro,20.0,pillars,main,HDPE_reuse,S9_row_21,1,individual_displayed_profile,100.0,70.3,49.5,34.8,24.5,17.2,12.1
HDPE_ZnO-nano_2.5,HDPE,ZnO-
nano,2.5,pillars,main,firstuse_22,S9_row_22,1,individual_displayed_profile,100.0,91.8,84.3,77.5,71.1,65.3,60.0
HDPE_ZnO-nano_5,HDPE,ZnO-
nano,5.0,pillars,main,firstuse_23,S9_row_23,1,individual_displayed_profile,100.0,90.5,81.9,74.2,67.1,60.8,55.0
HDPE_ZnO-nano_10,HDPE,ZnO-
nano,10.0,pillars,main,firstuse_24,S9_row_24,1,individual_displayed_profile,100.0,80.9,65.4,52.9,42.8,34.6,28.0
HDPE_ZnO-nano_20,HDPE,ZnO-
nano,20.0,pillars,main,firstuse_25,S9_row_25,1,individual_displayed_profile,100.0,79.4,63.0,50.0,39.7,31.5,25.0
ABS_ZnO-micro_20_swap,ABS,ZnO-micro,20.0,pillars,arch-
swap,swap_27,S9_row_27,1,individual_displayed_profile,100.0,62.9,39.6,24.9,15.7,9.9,6.2
HDPE_ZnO-micro_20_swap,HDPE,ZnO-micro,20.0,mesh,arch-
swap,swap_28,S9_row_28,1,individual_displayed_profile,100.0,83.5,69.7,58.2,48.6,40.6,33.9
photolysis_mean,,,,,photolysis-mean,,S9_row_30,,mean_of_eight_controls,100.0,96.6,93.4,90.3,87.2,84.3,81.5
ABS_ZnO-micro_20,ABS,ZnO-micro,20.0,mesh,re-
use,ABS_reuse,S9_row_32,2,individual_displayed_profile,100.0,79.3,62.9,49.9,39.5,31.3,24.9
ABS_ZnO-micro_20,ABS,ZnO-micro,20.0,mesh,re-
use,ABS_reuse,S9_row_33,3,individual_displayed_profile,100.0,81.5,66.5,54.2,44.2,36.1,29.4
ABS_ZnO-micro_20,ABS,ZnO-micro,20.0,mesh,re-
use,ABS_reuse,S9_row_34,4,individual_displayed_profile,100.0,84.5,71.3,60.2,50.9,43.0,36.3
ABS_ZnO-micro_20,ABS,ZnO-micro,20.0,mesh,re-
use,ABS_reuse,S9_row_35,5,individual_displayed_profile,100.0,86.2,74.4,64.1,55.3,47.7,41.1
HDPE_ZnO-micro_20,HDPE,ZnO-micro,20.0,pillars,re-
use,HDPE_reuse,S9_row_36,2,individual_displayed_profile,100.0,71.2,50.4,35.4,25.7,18.5,13.2
HDPE_ZnO-micro_20,HDPE,ZnO-micro,20.0,pillars,re-
use,HDPE_reuse,S9_row_37,3,individual_displayed_profile,100.0,76.0,57.8,44.0,33.4,25.4,19.3
HDPE_ZnO-micro_20,HDPE,ZnO-micro,20.0,pillars,re-
use,HDPE_reuse,S9_row_38,4,individual_displayed_profile,100.0,79.5,63.3,50.3,40.0,31.8,25.3
HDPE_ZnO-micro_20,HDPE,ZnO-micro,20.0,pillars,re-
use,HDPE_reuse,S9_row_39,5,individual_displayed_profile,100.0,79.9,63.9,51.1,40.8,32.6,26.1
"""

REPLICATES_CSV = """
Formulation,Independent_run,kapp
ABS_ZnO-micro_20,1,0.02256
ABS_ZnO-micro_20,2,0.02298
ABS_ZnO-micro_20,3,0.02324
ABS_ZnO-micro_20,4,0.02352
ABS_ZnO-micro_20,5,0.023899999999999998
HDPE_ZnO-micro_20,1,0.0342
HDPE_ZnO-micro_20,2,0.0348
HDPE_ZnO-micro_20,3,0.0352
HDPE_ZnO-micro_20,4,0.0356
HDPE_ZnO-micro_20,5,0.0362
"""
```

```
def write_json(path, obj):
    Path(path).write_text(json.dumps(obj, indent=2), encoding="utf-8")


def readmeans():
    d = pd.read_csv(StringIO(MEANS_CSV))
    keys = ["Matrix", "Filler", "Loading_wt", "Architecture", "Series"]
    if len(d) != 26 or d.duplicated(keys).any():
        raise ValueError("Expected 26 unique composition-architecture means")
    if not d.Series.isin(["main", "arch-swap"]).all():
        raise ValueError("Invalid model-input series")
    for col, valid in [("Matrix", RM), ("Filler", RF), ("Architecture", AREA)]:
        if not d[col].isin(valid).all():
            raise ValueError("Invalid metadata: " + col)
    w = d.Loading_wt.to_numpy(float) / 100
    if not np.isfinite(w).all() or ((w <= 0) | (w >= 1)).any():
        raise ValueError("Invalid loading")
    rm = d.Matrix.map(RM).to_numpy(float)
    rf = d.Filler.map(RF).to_numpy(float)
    d["phi"] = (w / rf) / (w / rf + (1 - w) / rm)
    d["observed"] = d.kapp_mean - KPH
    if not np.isfinite(d.observed).all():
        raise ValueError("Nonfinite rate")
    return d


def grid(d, diam=None, mode="coverage"):
    if mode not in ["coverage", "none", "total"]:
        raise ValueError("Unknown saturation mode")
    dia = d.Filler.map(DIAM if diam is None else diam).to_numpy(float)
    if not np.isfinite(dia).all() or (dia <= 0).any():
        raise ValueError("Invalid structural diameter")
    a = 0.75 * d.phi.to_numpy()[None, :] * np.maximum(
        1 - 2 * DELTA[:, None] / dia[None, :], 0
    ) ** 2
    af = d.Architecture.map(AREA).to_numpy(float)
    if mode == "none":
        return af[None, :] * a, DELTA, np.full(len(DELTA), np.nan)
    factors = (
        d.Architecture.map({"mesh": 240.0, "pillars": 277.7}).to_numpy()
        if mode == "total" else np.ones(len(d))
    )
    scales = np.logspace(-1, 3, 121) if mode == "total" else A0
    x = af[None, None, :] * a[:, None, :] / (
        1 + a[:, None, :] * factors[None, None, :] / scales[None, :, None]
    )
    return (
        x.reshape(-1, len(d)),
        np.repeat(DELTA, len(scales)),
        np.tile(scales, len(DELTA)),
    )
```

```
def linear(x, y, ti, w=None, one=False):
    w = np.ones(len(y)) if w is None else np.asarray(w, float)
    masks = [np.ones(len(y), bool)] if one else [~ti, ti]
    pred = np.zeros_like(x)
    bs = []
    for mask in masks:
        den = (x[:, mask] ** 2) @ w[mask]
        num = x[:, mask] @ (w[mask] * y[mask])
        b = np.maximum(
            np.divide(num, den, out=np.zeros_like(num), where=den > 0), 0
        )
        bs.append(b)
        pred[:, mask] = b[:, None] * x[:, mask]
    err = ((pred - y) ** 2) @ w
    j = int(np.argmin(err))
    return j, float(err[j]), [float(b[j]) for b in bs], pred[j]


def fitgrid(d, diam=None, mode="coverage"):
    x, de, a0 = grid(d, diam, mode)
    y = d.observed.to_numpy()
    ti = d.Filler.eq("TiO2").to_numpy()
    j, sse, b, pred = linear(x, y, ti)
    sst = float(np.sum((y - y.mean()) ** 2))
    if sst <= 0 or b[0] <= 0:
        raise ValueError("Degenerate model input or ZnO coefficient")
    return dict(
        delta_nm=float(de[j]), a0=float(a0[j]),
        kappa_ZnO=b[0], kappa_TiO2=b[1], ratio=b[1] / b[0],
        SSE=sse, R2=1 - sse / sst,
    ), pred


def fitprofile(a):
    a = np.asarray(a, float)
    if a.shape != (7,) or not np.isfinite(a).all() or (a <= 0).any():
        raise ValueError("Seven positive finite measurements required")
    z = linregress(TIMES, np.log(a / a[0]))
    return -z.slope, z.rvalue ** 2, z.intercept, z.stderr


def kinetics(out):
    p = pd.read_csv(StringIO(PROFILES_CSV), keep_default_na=False)
    if len(p) != 35 or p.Run_ID.duplicated().any():
        raise ValueError("Expected 35 profiles with unique run IDs")
    fits = [
        fitprofile(row)
        for row in p[[f"t{int(t)}" for t in TIMES]].to_numpy(float)
    ]
    for j, key in enumerate([
        "kapp", "R2", "log_intercept", "slope_standard_error"
    ]):
        p[key] = [v[j] for v in fits]
    p.to_csv(out / "per_profile_kinetics.csv", index=False)
```

```
    rows = []
    coupons = p.loc[p.Series.eq("re-use"), "Coupon_ID"].unique()
    for coupon in coupons:
        v = p[p.Coupon_ID.eq(coupon)].copy()
        v.Cycle = pd.to_numeric(v.Cycle)
        v = v.sort_values("Cycle")
        if v.Cycle.tolist() != [1, 2, 3, 4, 5]:
            raise ValueError("Expected exactly cycles 1-5")
        meta = ["Matrix", "Filler", "Loading_wt", "Architecture"]
        if len(v[meta].drop_duplicates()) != 1:
            raise ValueError("Inconsistent coupon metadata")
        if (v.kapp <= 0).any():
            raise ValueError("Nonpositive rate in exponential re-use fit")
        z = linregress(v.Cycle - 1, np.log(v.kapp))
        lam = -z.slope
        rows.append(dict(
            Coupon_ID=coupon, lambda_cycle_1=lam, R2=z.rvalue ** 2,
            loss_per_cycle_pct=100 * (1 - np.exp(-lam)),
            retention_cycle5_pct=100 * v.kapp.iloc[-1] / v.kapp.iloc[0],
            mean_cycles1_5=v.kapp.mean(),
            sample_SD_between_cycles=v.kapp.std(ddof=1),
            dose_per_cycle_J_cm2=21.6, total_dose_J_cm2=108,
            lambda_per_dose=lam / 21.6,
        ))
    if len(rows) != 2:
        raise ValueError("Expected two re-use trajectories")
    pd.DataFrame(rows).to_csv(out / "reuse_summary.csv", index=False)


def replicates(d, out):
    p = pd.read_csv(StringIO(REPLICATES_CSV))
    if p.duplicated(["Formulation", "Independent_run"]).any():
        raise ValueError("Duplicate independent run")
    rows = []
    for label, v in p.groupby("Formulation", sort=False):
        a = v.kapp.to_numpy(float)
        if len(a) != 5 or not np.isfinite(a).all() or (a <= 0).any():
            raise ValueError("Expected five positive independent constants")
        if sorted(v.Independent_run.tolist()) != [1, 2, 3, 4, 5]:
            raise ValueError("Invalid independent-run identifiers")
        mean, sd = a.mean(), a.std(ddof=1)
        ref = d[d.Formulation.eq(label)]
        if len(ref) != 1 or not np.isclose(
            mean, ref.kapp_mean.iloc[0], atol=1e-12, rtol=0
        ):
            raise ValueError("Replicate mean mismatch")
        rows.append(dict(
            Formulation=label, n=len(a), mean=mean,
            sample_SD=sd, CV_percent=100 * sd / mean,
        ))
    if len(rows) != 2:
        raise ValueError("Expected two independent-constant datasets")
    pd.DataFrame(rows).to_csv(out / "independent_statistics.csv", index=False)
```

```
def variants(d, base, out):
    y = d.observed.to_numpy()
    ti = d.Filler.eq("TiO2").to_numpy()
    hd = d.Matrix.eq("HDPE").to_numpy()
    phi = d.phi.to_numpy()
    af = d.Architecture.map(AREA).to_numpy()
    diam = d.Filler.map(DIAM).to_numpy()
    sst = np.sum((y - y.mean()) ** 2)
    rows = []

    def add(name, n, sse, b, delta=np.nan, a0=np.nan, extra=np.nan):
        rows.append(dict(
            Model=name, parameters=n, delta_nm=delta, a0=a0,
            ratio=b[-1] / b[0], extra=extra, SSE=sse, R2=1 - sse / sst,
        ))

    for one in [True, False]:
        _, ss, b, _ = linear((af * 0.75 * phi)[None, :], y, ti, one=one)
        add("loading_one" if one else "loading_two", 1 if one else 2, ss, b)
    for mode in ["none", "total"]:
        f, _ = fitgrid(d, mode=mode)
        add(mode, 3 if mode == "none" else 4, f["SSE"],
            [f["kappa_ZnO"], f["kappa_TiO2"]], f["delta_nm"], f["a0"])
    add("adopted_grid", 4, base["SSE"],
        [base["kappa_ZnO"], base["kappa_TiO2"]],
        base["delta_nm"], base["a0"])
    for mode in [
        "continuous_shared", "matrix_multiplier", "matrix_delta",
        "free_TiO2", "TiO2_21nm"
    ]:
        def predict(z):
            delta = np.where(hd, z[4], z[0]) if mode == "matrix_delta" else z[0]
            dd = diam.copy()
            if mode == "free_TiO2":
                dd[ti] = np.exp(z[4])
            if mode == "TiO2_21nm":
                dd[ti] = 21.0
            a = 0.75 * phi * np.maximum(1 - 2 * delta / dd, 0) ** 2
            pred = af * a / (1 + a / np.exp(z[1])) * np.where(
                ti, np.exp(z[3]), np.exp(z[2])
            )
            if mode == "matrix_multiplier":
                pred *= np.where(hd, np.exp(z[4]), 1.0)
            return pred

        best = None
        starts = [5.0, 9.0, 10.25] if mode == "TiO2_21nm" else [8.0, 18.0, 28.0]
        for de in starts:
            z = [de, np.log(0.025), np.log(1.3), np.log(0.63)]
            lo, hi = [1.0, np.log(0.001), -20.0, -20.0], [40.0, 0.0, 20.0, 20.0]
            if mode == "TiO2_21nm":
                hi[0] = 10.4999
            if mode == "matrix_multiplier":
```

```
                z += [0.0]; lo += [np.log(0.1)]; hi += [np.log(10.0)]
            if mode == "matrix_delta":
                z += [de]; lo += [1.0]; hi += [40.0]
            if mode == "free_TiO2":
                z += [np.log(200.0)]; lo += [np.log(21.0)]; hi += [np.log(2000.0)]
            fitted = least_squares(
                lambda q: predict(q) - y, z, bounds=(lo, hi),
                xtol=1e-12, ftol=1e-12, gtol=1e-12, max_nfev=3000,
            )
            ss = float(np.sum(fitted.fun ** 2))
            if best is None or ss < best[0]:
                best = ss, fitted.x
        ss, z = best
        extra = (z[4] if mode == "matrix_delta" else np.exp(z[4])) if len(z) > 4 else np.nan
        add(mode, len(z), ss, [np.exp(z[2]), np.exp(z[3])],
            z[0], np.exp(z[1]), extra)
    tab = pd.DataFrame(rows)
    ref = tab[tab.Model.eq("continuous_shared")].iloc[0]
    for mode in ["matrix_multiplier", "matrix_delta"]:
        i = tab.index[tab.Model.eq(mode)][0]
        v = tab.loc[i]
        F = max(0.0, ((ref.SSE - v.SSE) / (v.parameters - ref.parameters)) /
                (v.SSE / (len(d) - v.parameters)))
        tab.loc[i, "nominal_F"] = F
        tab.loc[i, "nominal_p"] = fdist.sf(
            F, v.parameters - ref.parameters, len(d) - v.parameters
        )
    tab.to_csv(out / "model_ladder.csv", index=False)


def bootstrap(d, out, n=2000):
    x, de, a0 = grid(d)
    y = d.observed.to_numpy()
    ti = d.Filler.eq("TiO2").to_numpy()
    rng = np.random.default_rng(20261116)
    weights = rng.multinomial(len(y), np.ones(len(y)) / len(y), size=n).T.astype(float)
    if (weights[ti].sum(axis=0) == 0).any() or (weights[~ti].sum(axis=0) == 0).any():
        raise ValueError("Bootstrap draw omits an oxide group; coefficient ratio undefined")
    rows = []
    for start in range(0, n, 100):
        w = weights[:, start:start + 100]
        score = np.zeros((len(x), w.shape[1]))
        bs = []
        for mask in [~ti, ti]:
            den = (x[:, mask] ** 2) @ w[mask]
            num = x[:, mask] @ (y[mask, None] * w[mask])
            b = np.maximum(np.divide(num, den, out=np.zeros_like(num), where=den > 0), 0)
            score += den * b * b - 2 * num * b
            bs.append(b)
        for q, j in enumerate(np.argmin(score, axis=0)):
            if bs[0][j, q] <= 0:
                raise ValueError("Nonpositive bootstrap ZnO coefficient")
            rows.append(dict(
                delta_nm=de[j], a0=a0[j], kappa_ZnO=bs[0][j, q],
```

```
                kappa_TiO2=bs[1][j, q], ratio=bs[1][j, q] / bs[0][j, q],
            ))
    b = pd.DataFrame(rows)
    b.to_csv(out / "bootstrap_samples.csv", index=False)
    ci = b.quantile([0.025, 0.5, 0.975]).T
    ci.columns = ["lower_2.5pct", "median", "upper_97.5pct"]
    ci.to_csv(out / "bootstrap_intervals.csv")
    write_json(out / "bootstrap_settings.json", dict(
        n=n, seed=20261116, method="pairs bootstrap of 26 published means",
        delta_at_boundary=int(b.delta_nm.isin([DELTA[0], DELTA[-1]]).sum()),
        a0_at_boundary=int(b.a0.isin([A0[0], A0[-1]]).sum()),
    ))


def sensitivity(d, out):
    rows = []
    for dm in [200.0, 350.0, 500.0, 750.0]:
        for dn in [60.0, 80.0, 100.0]:
            f, _ = fitgrid(d, {"TiO2": 200.0, "ZnO-micro": dm, "ZnO-nano": dn})
            rows.append(dict(d_micro=dm, d_nano=dn,
                             delta_over_dnano=f["delta_nm"] / dn, **f))
    pd.DataFrame(rows).to_csv(out / "diameter_sensitivity.csv", index=False)
    rows = []
    for fill, vals in [
        ("ZnO-nano", [50.0, 70.0, 100.0, 150.0]),
        ("TiO2", [21.0, 80.0, 100.0, 200.0, 400.0, 1200.0]),
        ("ZnO-micro", [500.0, 5000.0]),
    ]:
        for diameter in vals:
            dia = DIAM.copy()
            dia[fill] = diameter
            f, _ = fitgrid(d, dia)
            rows.append(dict(Filler=fill, diameter=diameter, **f))
    pd.DataFrame(rows).to_csv(out / "additional_sensitivity.csv", index=False)
    old = AREA.copy()
    rows = []
    try:
        for name, ratio in [
            ("adopted", 2.156),
            ("all_vertical", (100 + 12 * np.pi) / 136.0),
            ("wetted", 277.7 / 240.0),
        ]:
            AREA["pillars"] = ratio
            f, _ = fitgrid(d)
            rows.append(dict(convention=name, area_ratio=ratio, **f))
    finally:
        AREA.clear()
        AREA.update(old)
    pd.DataFrame(rows).to_csv(out / "area_sensitivity.csv", index=False)


def main():
    parser = argparse.ArgumentParser(description="Self-contained Supplementary Code S1")
    parser.add_argument("--out", default="results", help="Directory for generated outputs")
```

```
    parser.add_argument("--stage", choices=["base", "bootstrap", "sensitivity", "all"], default="all")
    args = parser.parse_args()
    out = Path(args.out)
    out.mkdir(parents=True, exist_ok=True)
    d = readmeans()
    with threadpool_limits(limits=1):
        if args.stage in ["base", "all"]:
            f, pred = fitgrid(d)
            write_json(out / "model_parameters.json", f)
            d["predicted"] = pred
            d["residual"] = d.observed - pred
            d["pred_obs_ratio"] = pred / d.observed
            d.to_csv(out / "model_predictions.csv", index=False)
            d.groupby(["Matrix", "Loading_wt"]).pred_obs_ratio.agg(
                ["count", "median", "min", "max"]
            ).to_csv(out / "residual_groups.csv")
            variants(d, f, out)
            kinetics(out)
            replicates(d, out)
        if args.stage in ["bootstrap", "all"]:
            bootstrap(d, out)
        if args.stage in ["sensitivity", "all"]:
            sensitivity(d, out)
    write_json(out / "analysis_scope.json", dict(
        input_mode="embedded datasets; no external input files",
        model_observations=26, displayed_profiles=35,
        supplied_independent_replicate_constants=10,
        HDPE_cycle2=[100.0, 71.2, 50.4, 35.4, 25.7, 18.5, 13.2],
        rate_units="min^-1", stages_requested=args.stage,
        replicate_profiles_for_all_formulations_included=False,
        original_spectral_provenance_independently_verified=False,
        statistical_note="Bootstrap is conditional on diameters and area convention; nominal nonlinear F tests are
exploratory.",
        dose_note="Dose uses reported radiometer-band irradiance, not calibrated full-spectrum output.",
    ))


if __name__ == "__main__":
    main()
```