Electronic Supplementary Material (ESI) for Journal of Materials Chemistry A.


# Insights from Experiment and Machine Learning for Enhanced $TiO_2$ Coated Glazing for Photocatalytic $NO_x$ Remediation

Zhi-Peng Lin[a], Yuan-Kai Li[a], Saif A. Haque[a], Alex M. Ganose[a,b,*], and Andreas Kafizas[a,c,*]

[a]*Department of Chemistry, Molecular Science Research Hub, Imperial College London, White City, London, W12 0BZ, U.K.*
[b]*Thomas Young Centre, The London Centre for the Theory and Simulation of Materials, Royal School of Mines, Imperial College London, South Kensington, London, U.K.*
[c]*London Centre for Nanotechnology, Imperial College London, SW7 2AZ, U.K.*

[*]*Corresponding author:*
*Alex Ganose: a.ganose@imperial.ac.uk;*
*Andreas Kafizas: a.kafizas@imperial.ac.uk*

# A Figures

## A.1 Introduction

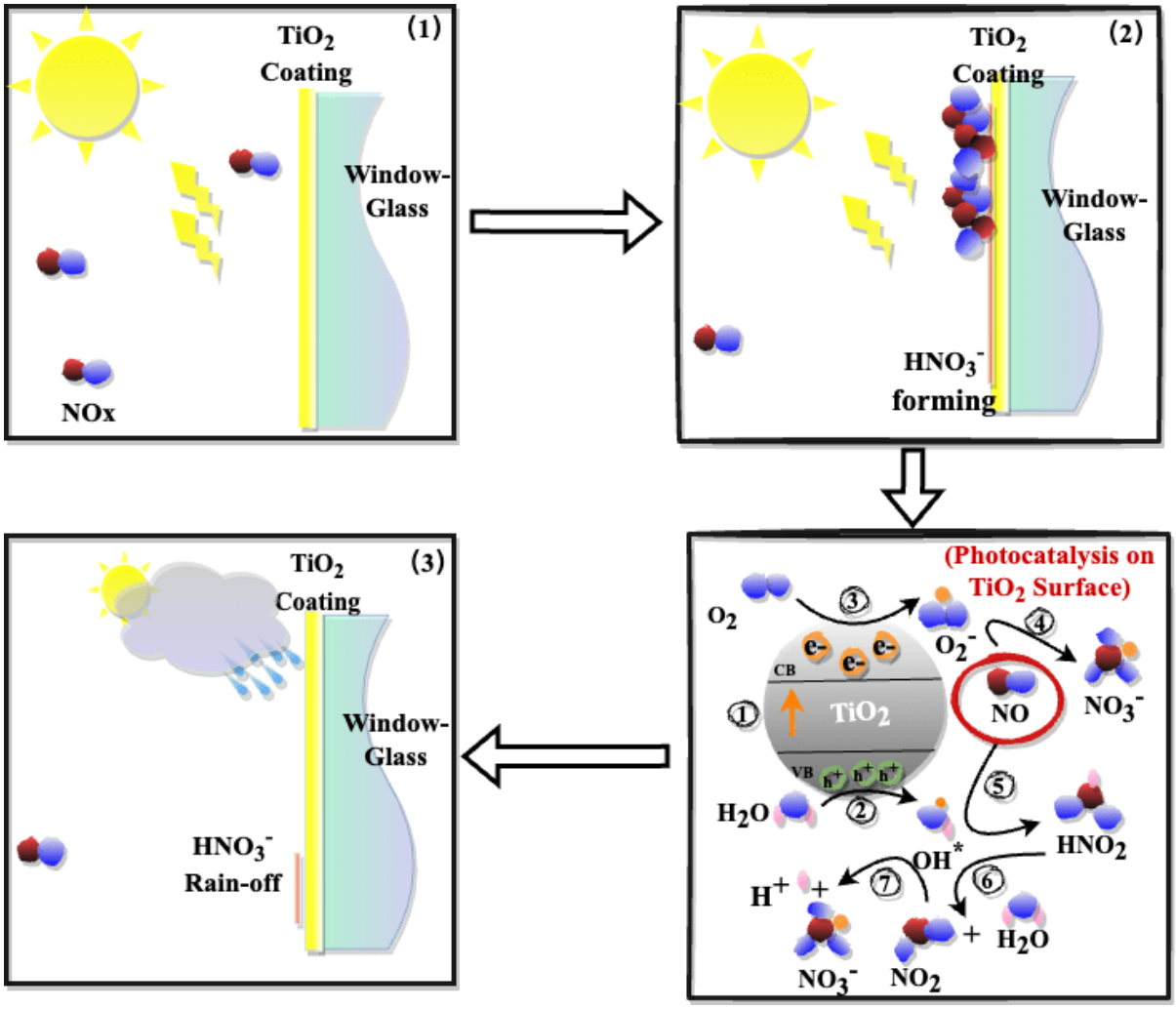



Figure S1: *Mechanism illustrating the various stages in the photocatalytic degradation of $NO_x$ in the air to $HNO_3$ on a $TiO_2$ coated window.*

## A.2 Experimental

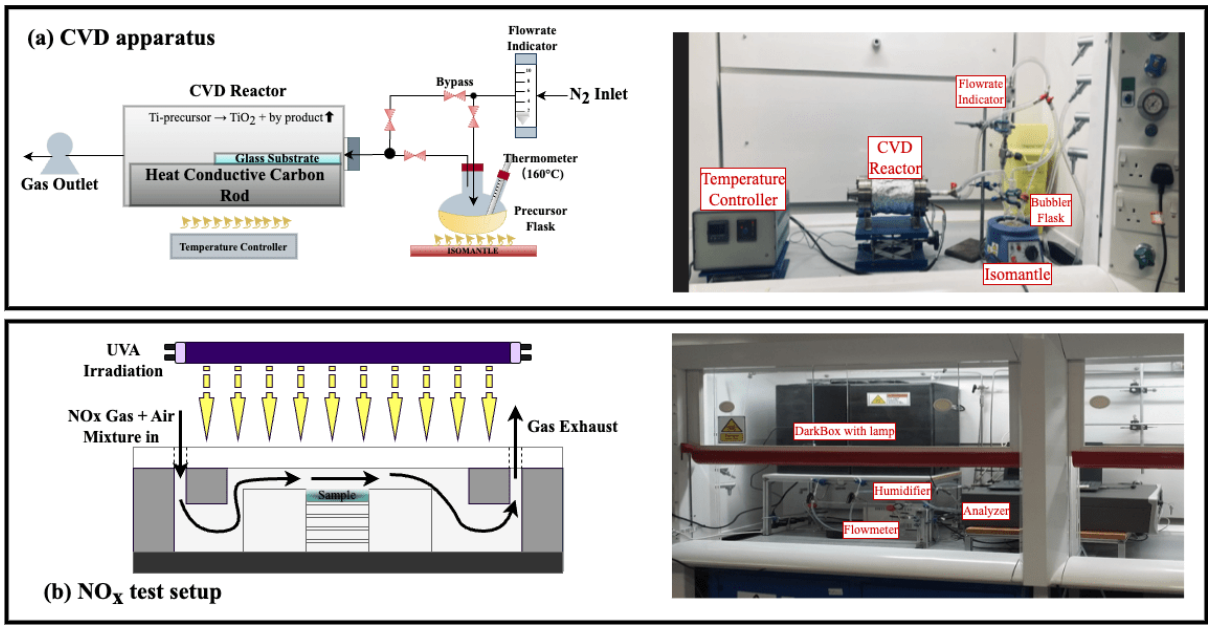



Figure S2: *(a) The APCVD apparatus used to produce the $TiO_2$-coated glass herein (which utilises a Model 3216 Temperature controller from EUROTHEM™; K-type therm°Couples from RS COMPONENTS™ and carbon rod from SENSEMASTER™); (b) Photocatalytic $NO_x$ test set-up built in accordance with ISO 22197-1:2016 (with an illustration of the gas-flow through the sample holder and photographs of the experimental set-up, which includes a chemi-luminescence analyser Model 40 from SERINUS®, Ecotech™).*

## A.3 Results and Discussion


Figure S3: *Photographs of all 58 unique $TiO_2$ coatings on glass synthesized using APCVD with differing synthesis parameters.*

### A.3.1 Effect of CVD Deposition Temperature

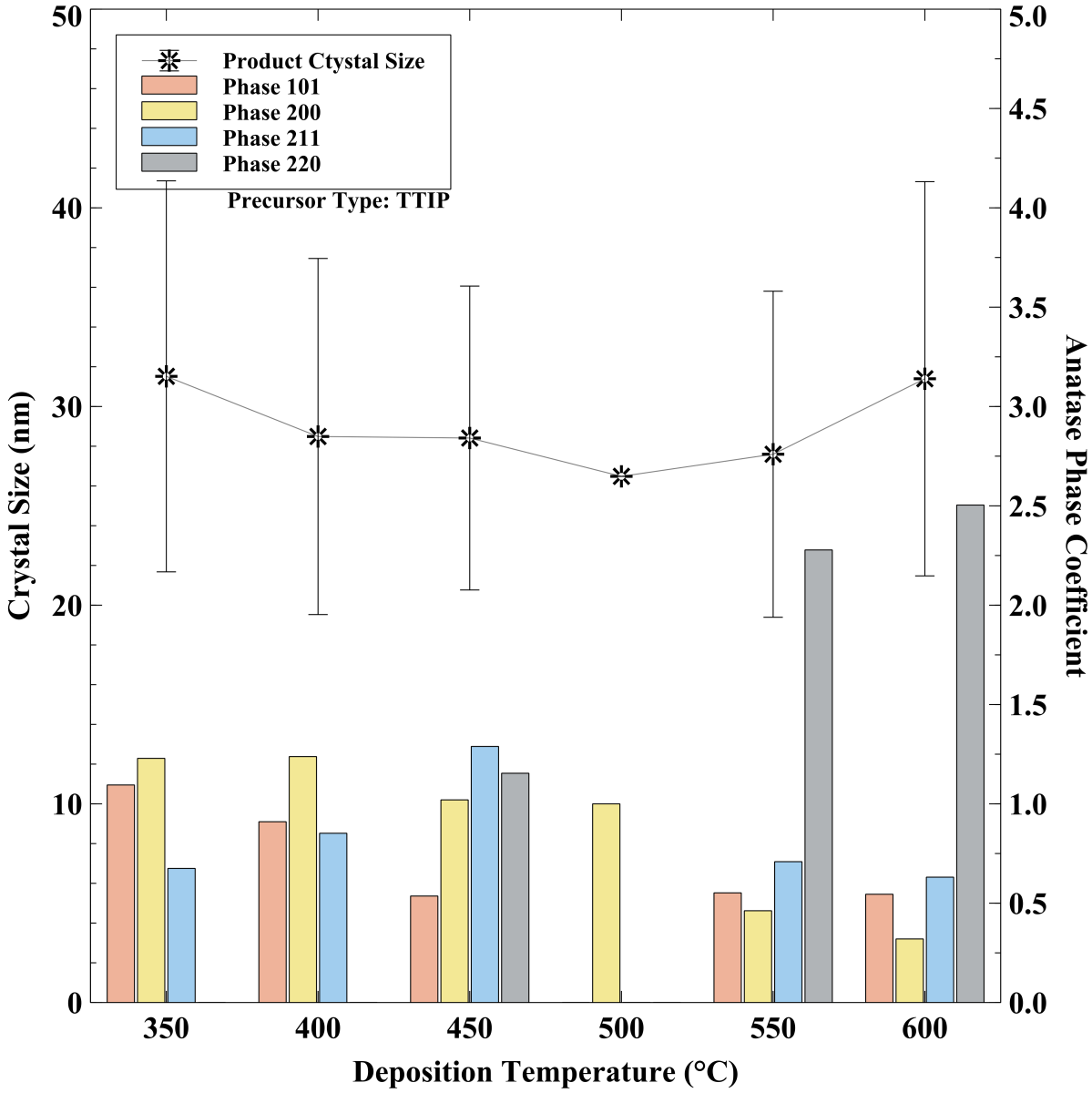



Figure S4: *Using TTIP as the precursor, here we show how average crystal size and preferred orientation of particular crystal facets are affected at different deposition temperatures. The errors on the average crystal size represent the range of average crystal sizes seen for samples produced at that temperature. The colors corresponding to each facet are: 101 facet: Pink; 200 facet: Yellow; 211 facet: SkyBlue; 220 facet: Grey.*

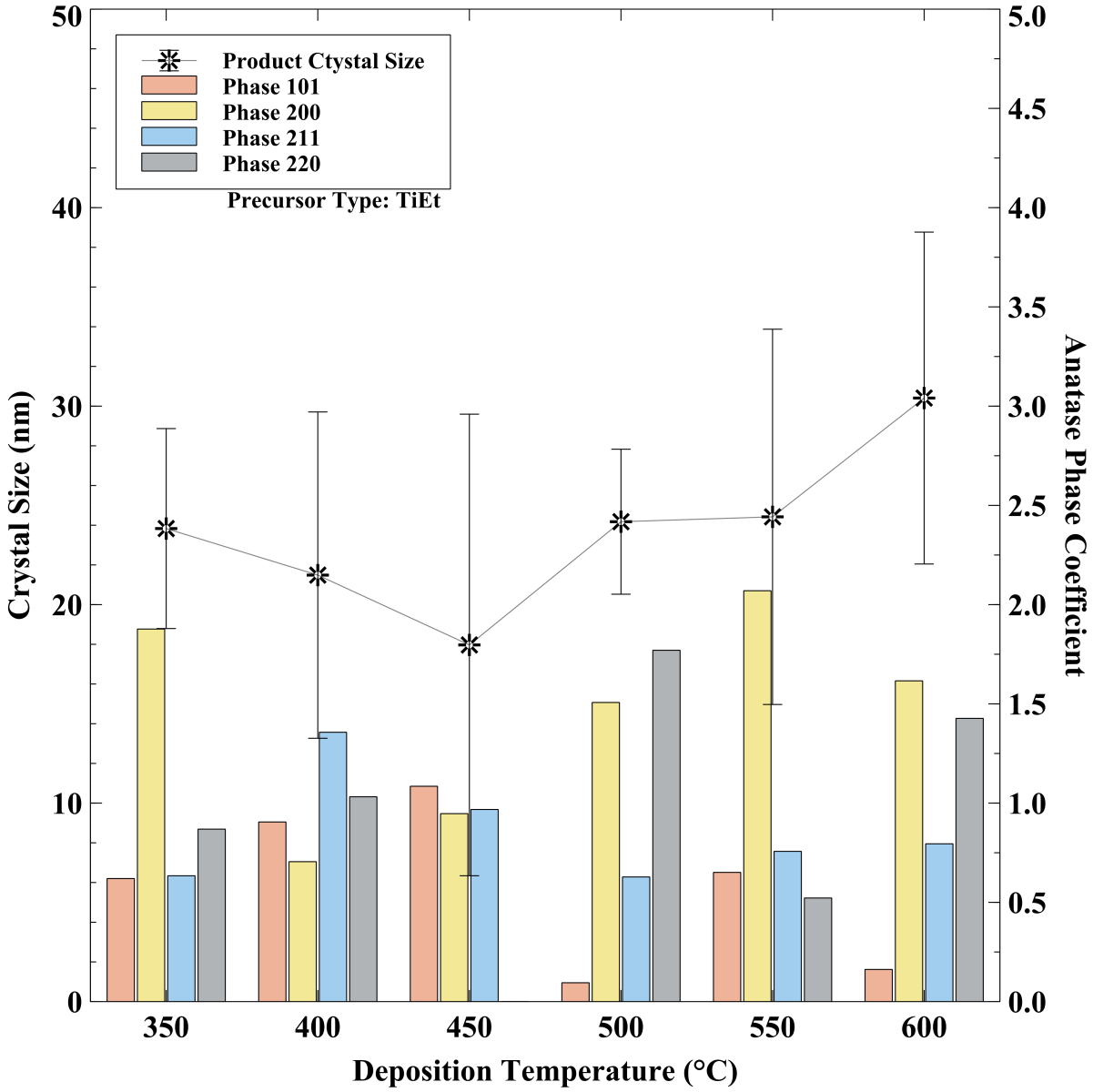



Figure S5: *Using TiEt as the precursor, here we show how average crystal size and preferred orientation of particular crystal facets are affected at different deposition temperatures. The errors on the average crystal size represent the range of average crystal sizes seen for samples produced at that temperature. The colors corresponding to each facet are: 101 facet: Pink; 200 facet: Yellow; 211 facet: SkyBlue; 220 facet: Grey.*

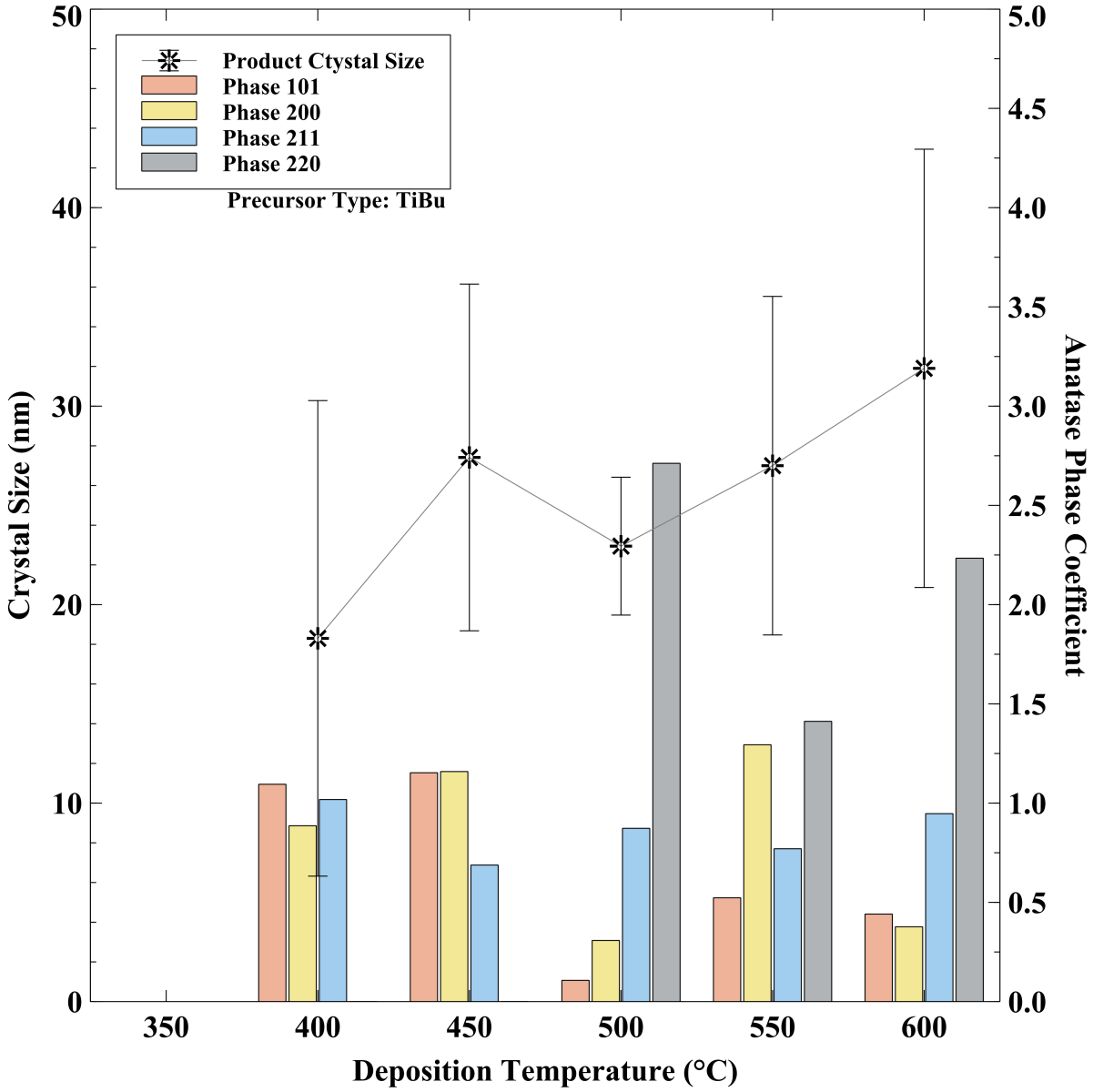



Figure S6: *Using TiBu as the precursor, here we show how average crystal size and preferred orientation of particular crystal facets are affected at different deposition temperatures. The errors on the average crystal size represent the range of average crystal sizes seen for samples produced at that temperature. The colors corresponding to each facet are: 101 facet: Pink; 200 facet: Yellow; 211 facet: SkyBlue; 220 facet: Grey.*

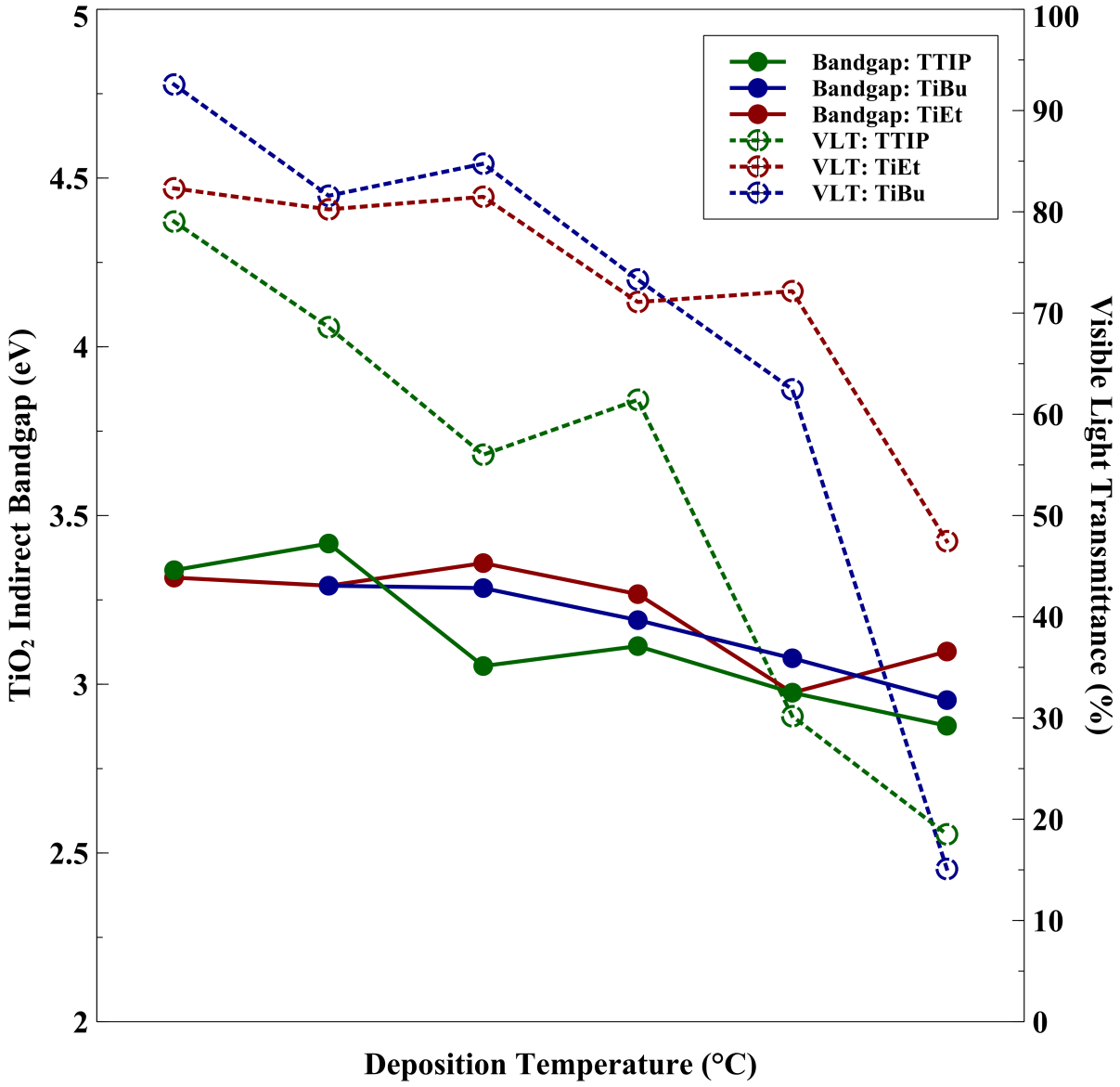



Figure S7: *Variations in Bandgap and Visible Light Transmittance with Deposition Temperature: Green lines for the precursor TTIP, red for TiEt, and blue for TiBu show the changes. Solid lines depict bandgap values and dashed lines represent VLT measurements.*

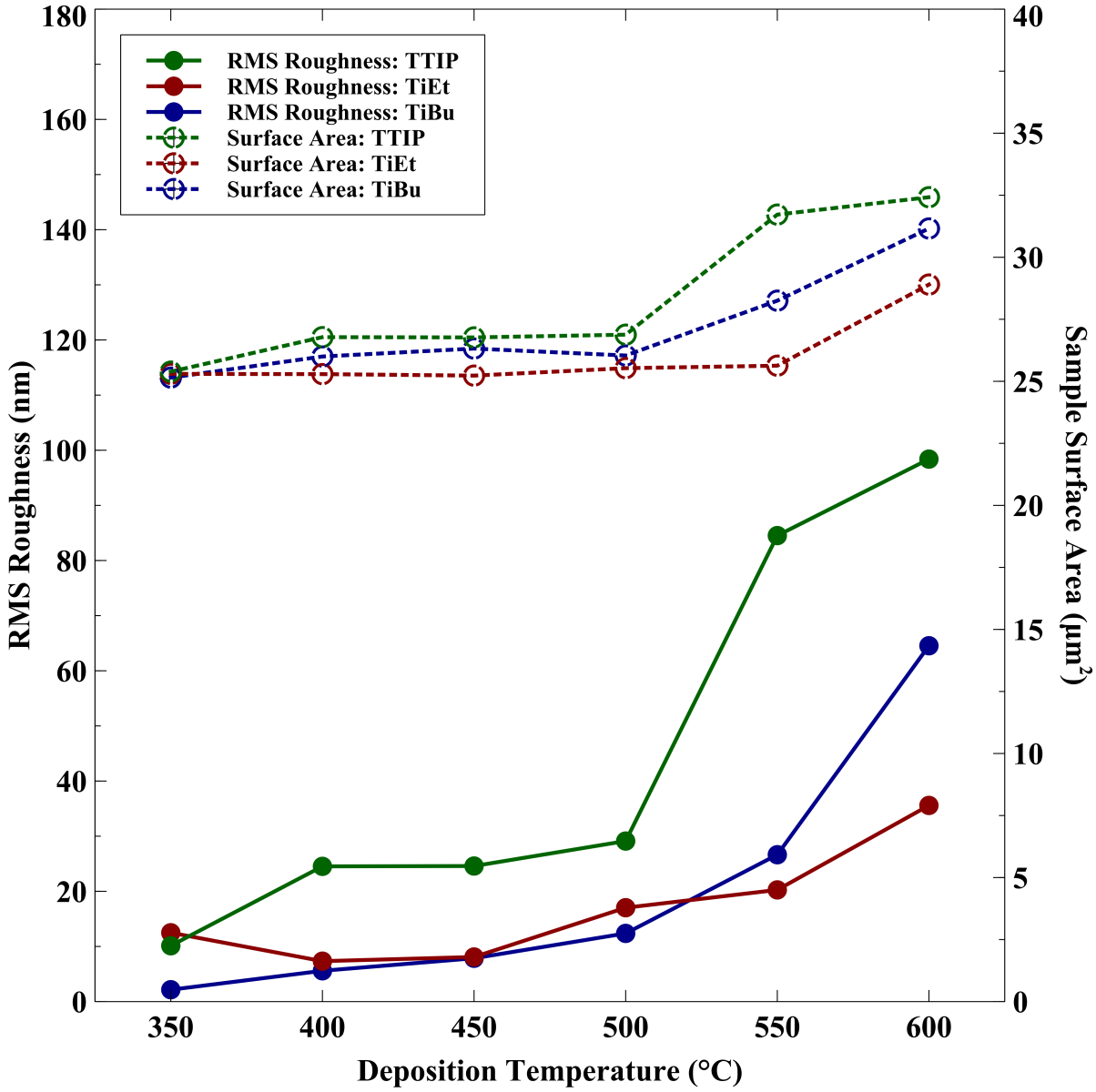



Figure S8: *Variations in Rq roughness and sample surface area with deposition temperature: Green lines for the precursor TTIP, red for TiEt, and blue for TiBu show the changes. Solid lines depict Rq roughness values and dashed lines represent sample surface area.*

### A.3.2 Effect of Synthesis Time

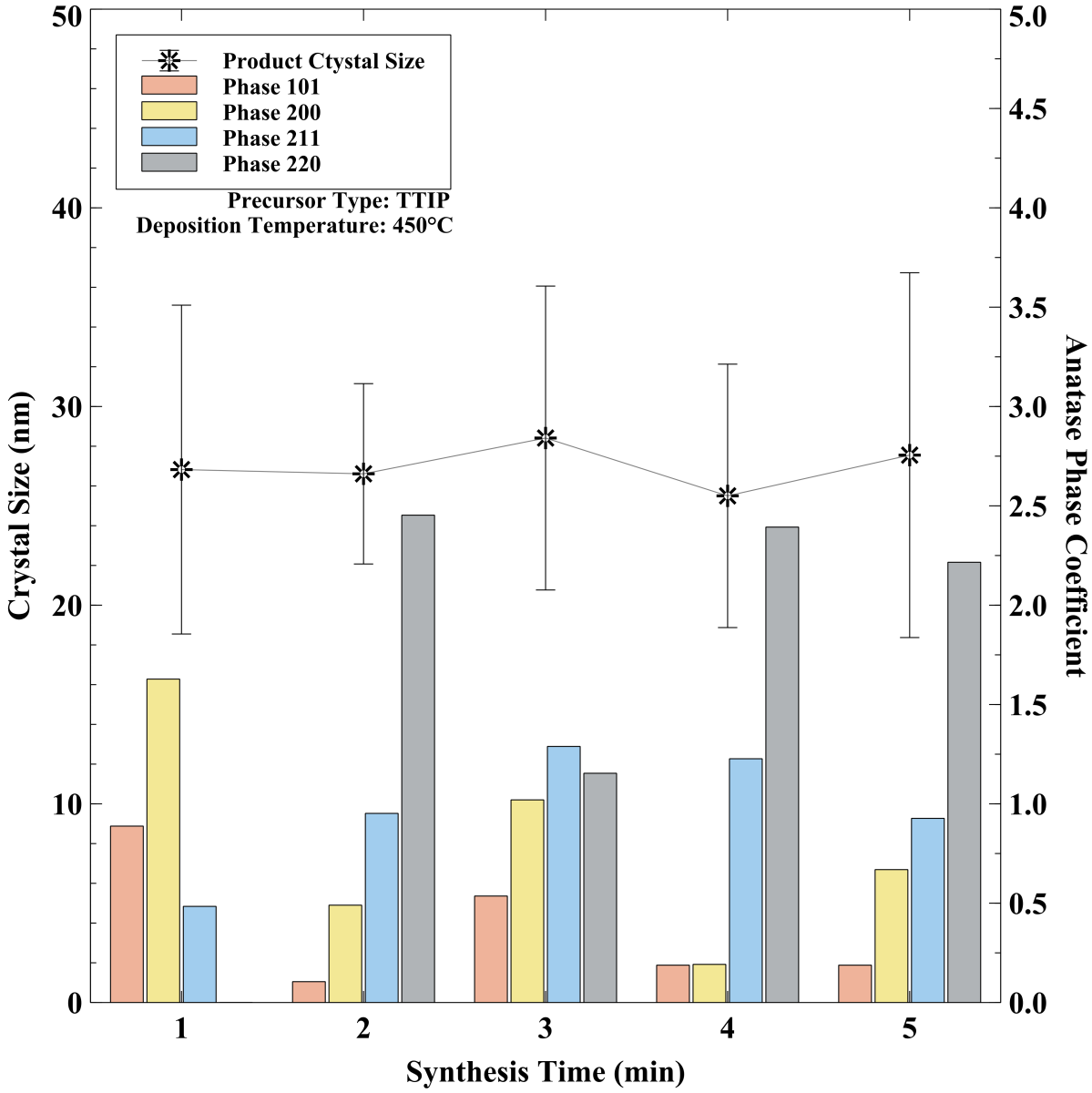



Figure S9: *Variations in crystal size and facet orientations at different synthesis time. Using TTIP as the precursor with a bubbler temperature of 160 °C and a deposition temperature of 450 °C. the scatter line graph represents the change in crystal size with varying deposition temperatures. the bar graph showcases the orientation distribution of different crystal facets. The colors corresponding to each facet are: 101 facet: Pink; 200 facet: Yellow; 211 facet: SkyBlue; 220 facet: Grey.*

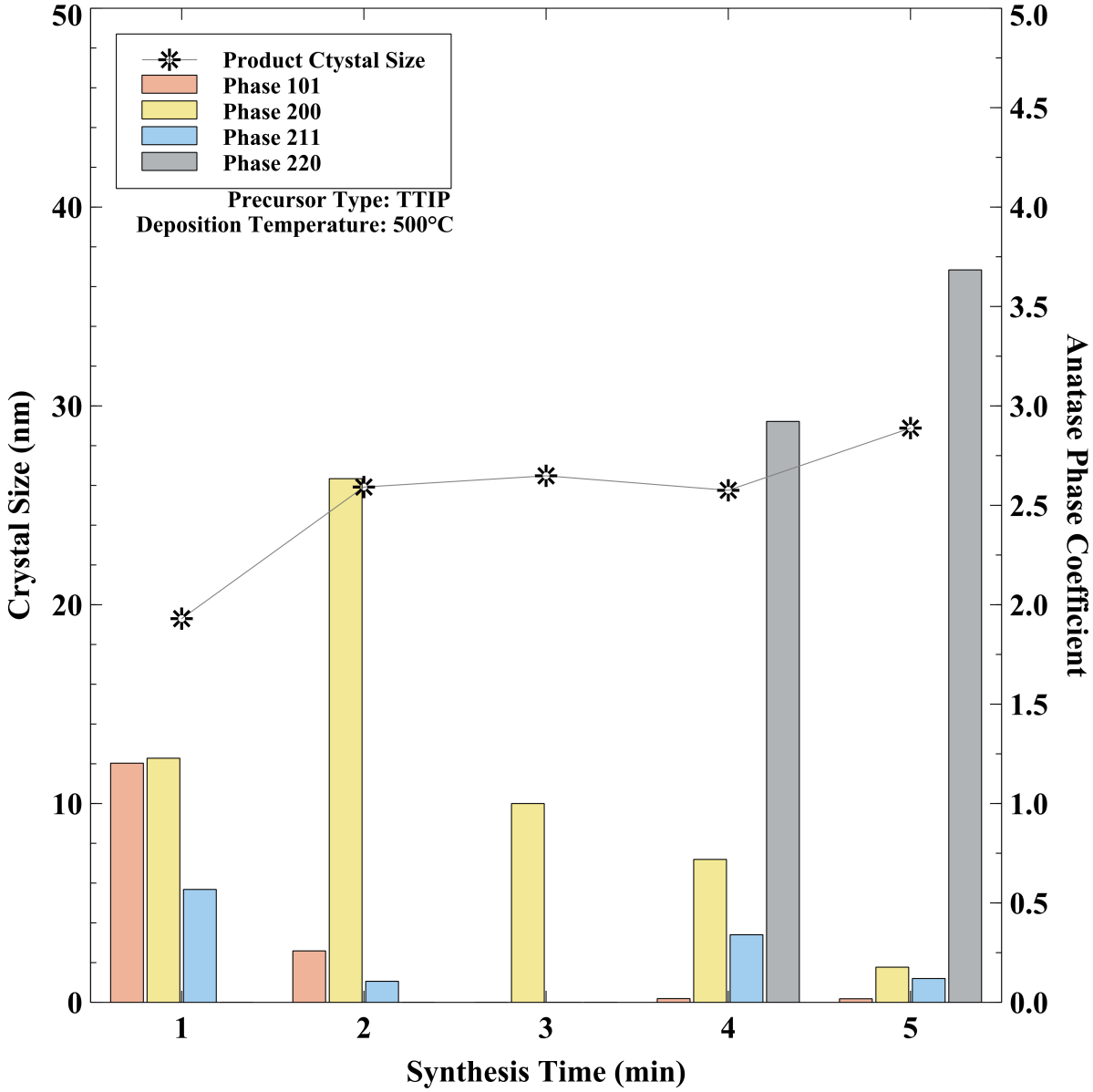



Figure S10: *Variations in crystal size and facet orientations at different synthesis time. Using TTIP as the precursor with a bubbler temperature of 160°C and a deposition temperature of 500 °C. the scatter line graph represents the change in crystal size with varying deposition temperatures. the bar graph showcases the orientation distribution of different crystal facets. The colors corresponding to each facet are: 101 facet: Pink; 200 facet: Yellow; 211 facet: SkyBlue; 220 facet: Grey.*

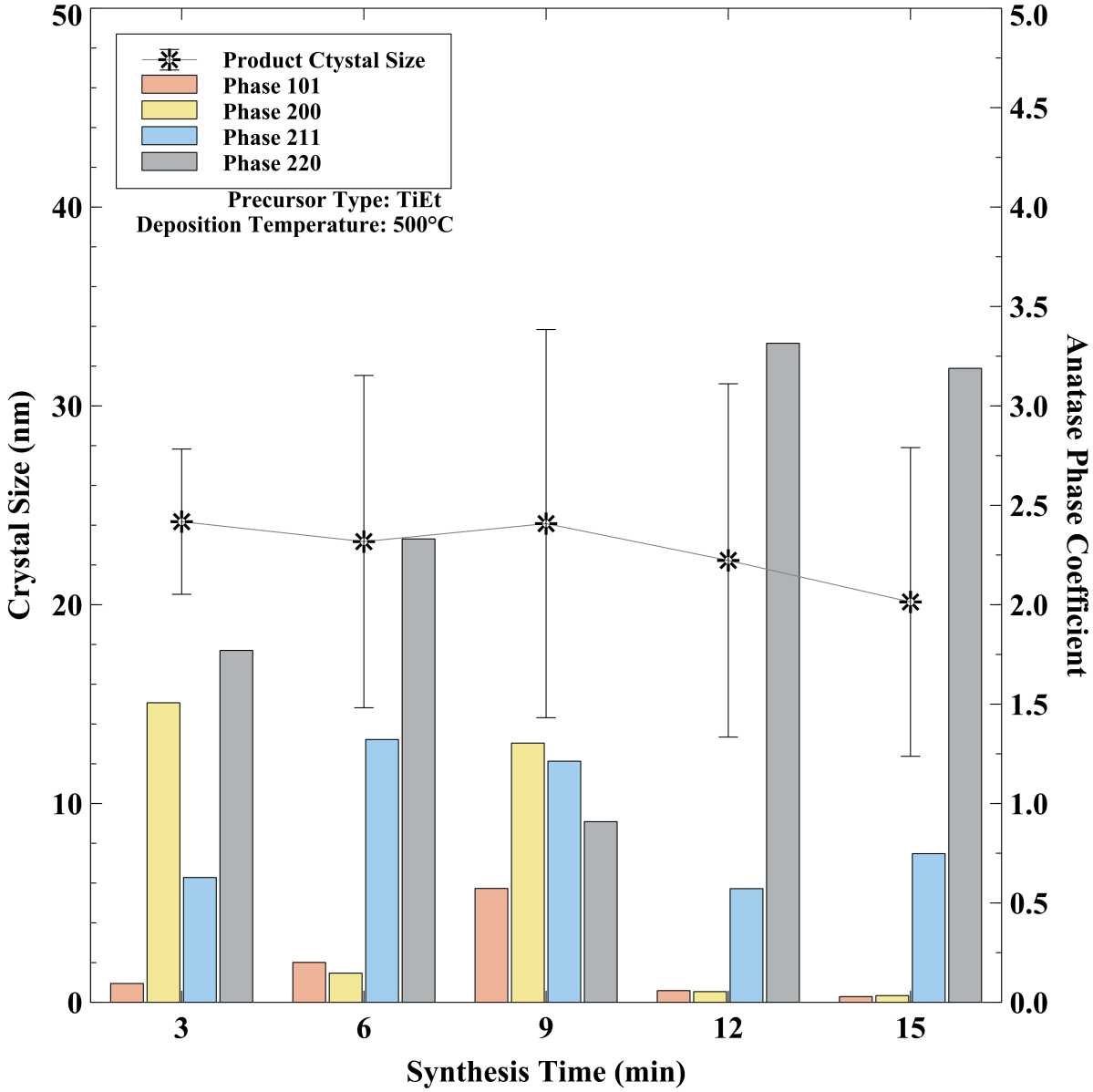



Figure S11: *Variations in crystal size and facet orientations at different synthesis time. Using TiEt as the precursor with a bubbler temperature of 160°C and a deposition temperature of 500 °C. the scatter line graph represents the change in crystal size with varying deposition temperatures. the bar graph showcases the orientation distribution of different crystal facets. The colors corresponding to each facet are: 101 facet: Pink; 200 facet: Yellow; 211 facet: SkyBlue; 220 facet: Grey.*

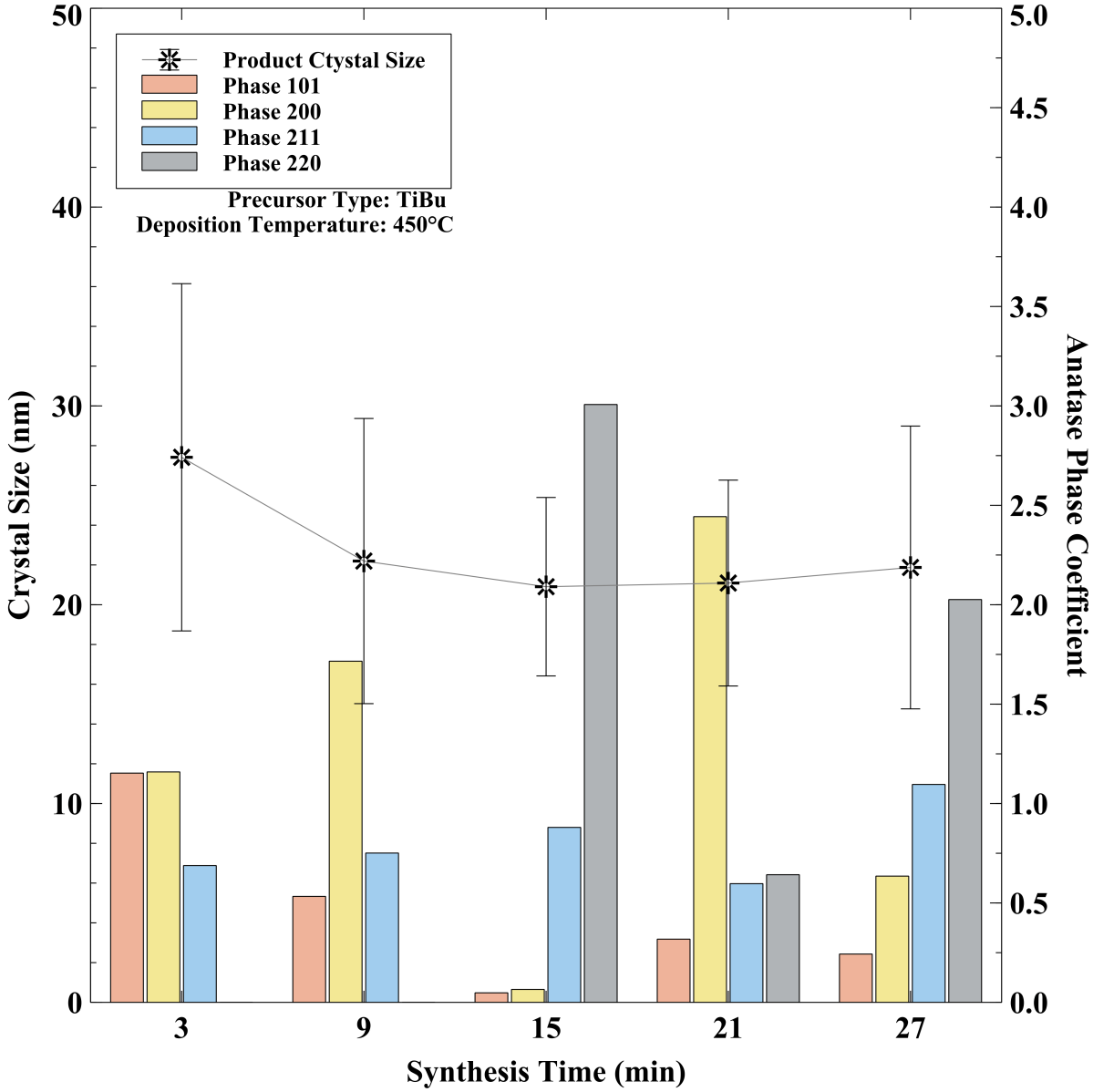



Figure S12: *Variations in crystal size and facet orientations at different synthesis time. Using TiBu as the precursor with a bubbler temperature of 160°C and a deposition temperature of 450 °C. the scatter line graph represents the change in crystal size with varying deposition temperatures. the bar graph showcases the orientation distribution of different crystal facets. The colors corresponding to each facet are: 101 facet: Pink; 200 facet: Yellow; 211 facet: SkyBlue; 220 facet: Grey.*

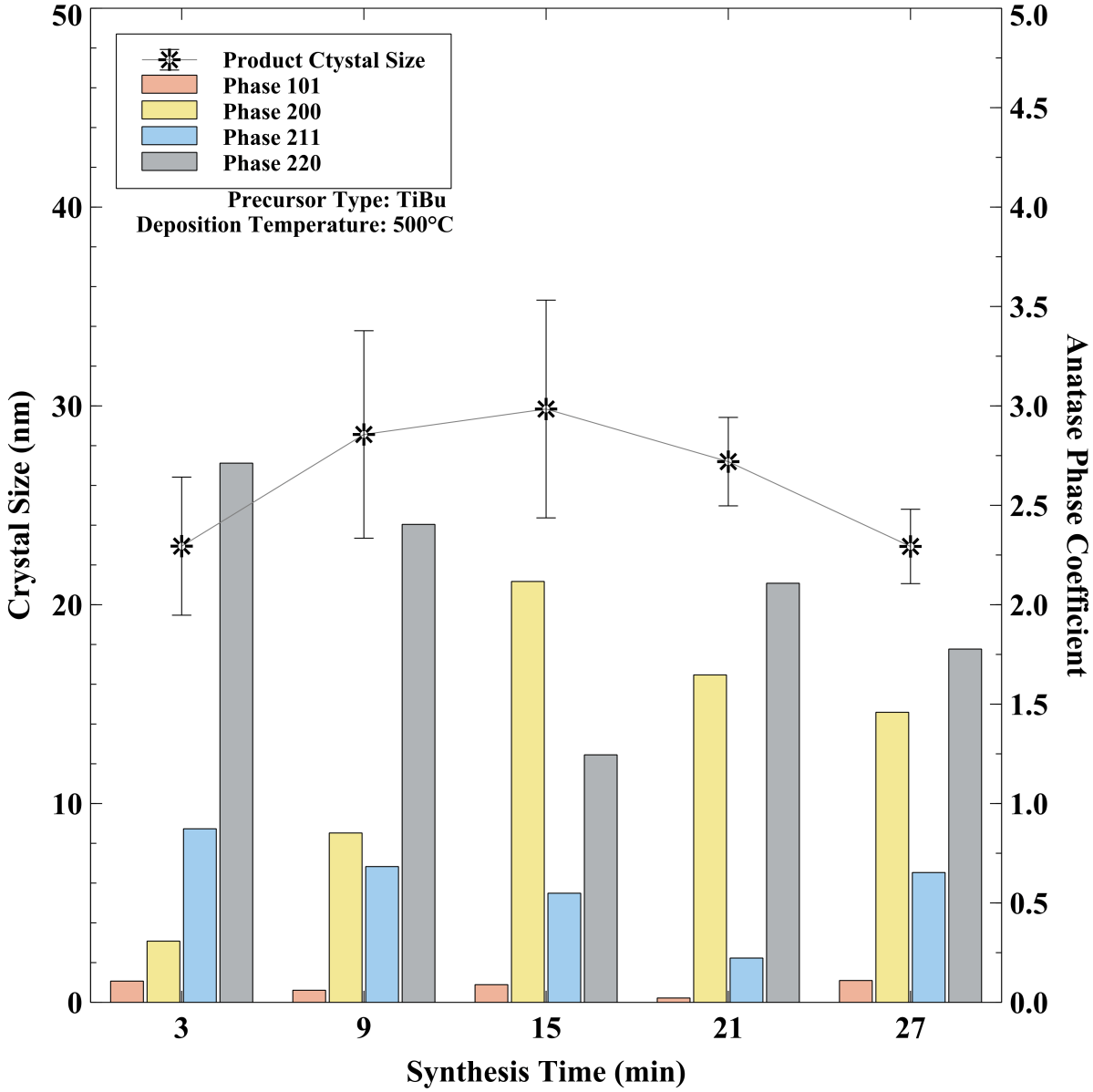



Figure S13: *Variations in crystal size and facet orientations at different synthesis time. Using TiBu as the precursor with a bubbler temperature of 160°C and a deposition temperature of 500 °C. the scatter line graph represents the change in crystal size with varying deposition temperatures. the bar graph showcases the orientation distribution of different crystal facets. The colors corresponding to each facet are: 101 facet: Pink; 200 facet: Yellow; 211 facet: SkyBlue; 220 facet: Grey.*

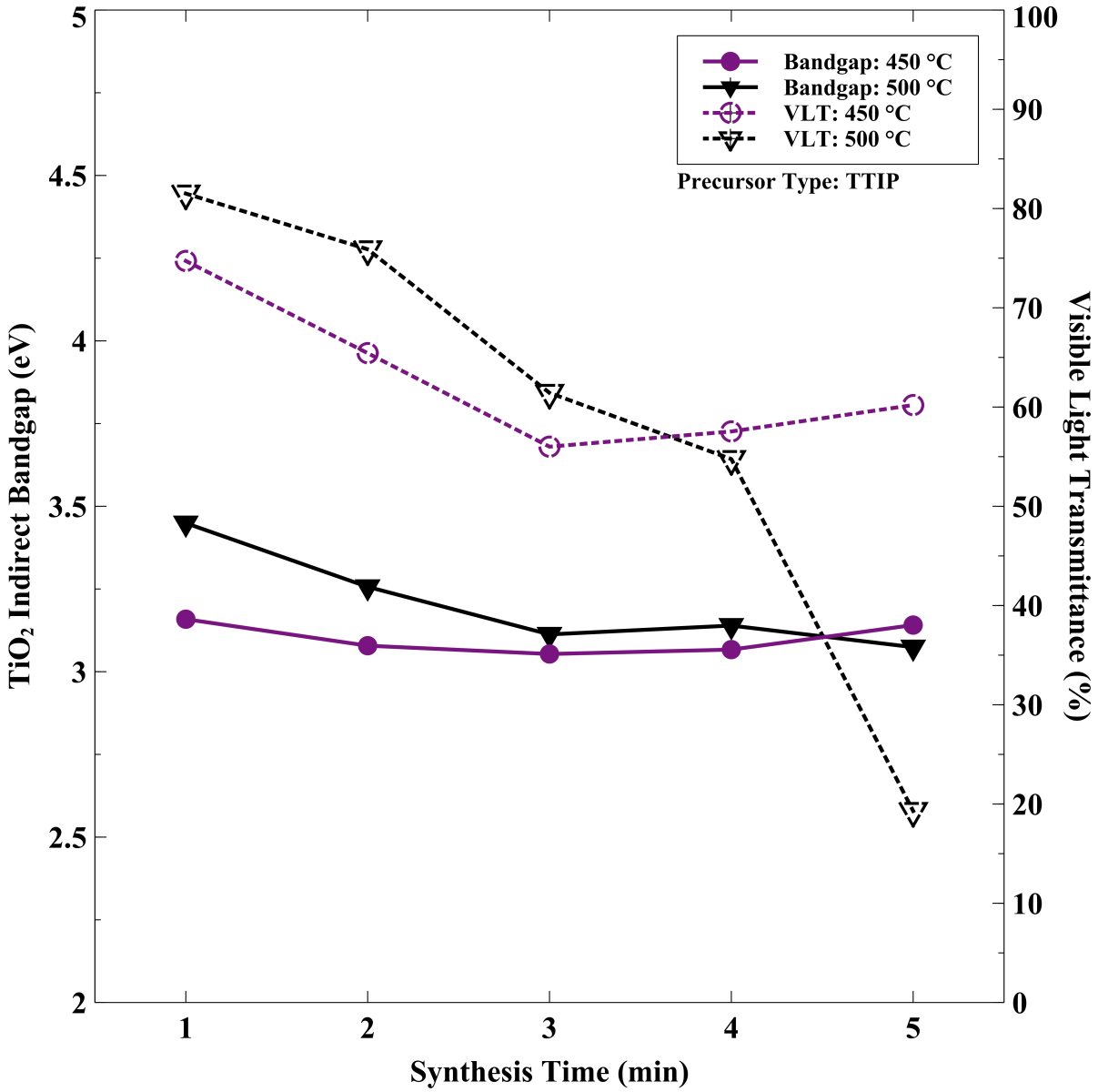



Figure S14: *Using TTIP as the precursor; variations in $TiO_2$ bandgap and visible light transmittance with synthesis time: Purple lines with circle symbol for the deposition temperature at 450 °C, Black lines with triangle symbol for the deposition temperature at 500 °C. Solid lines depict $TiO_2$ bandgap and dashed lines represent visible light transmittance.*

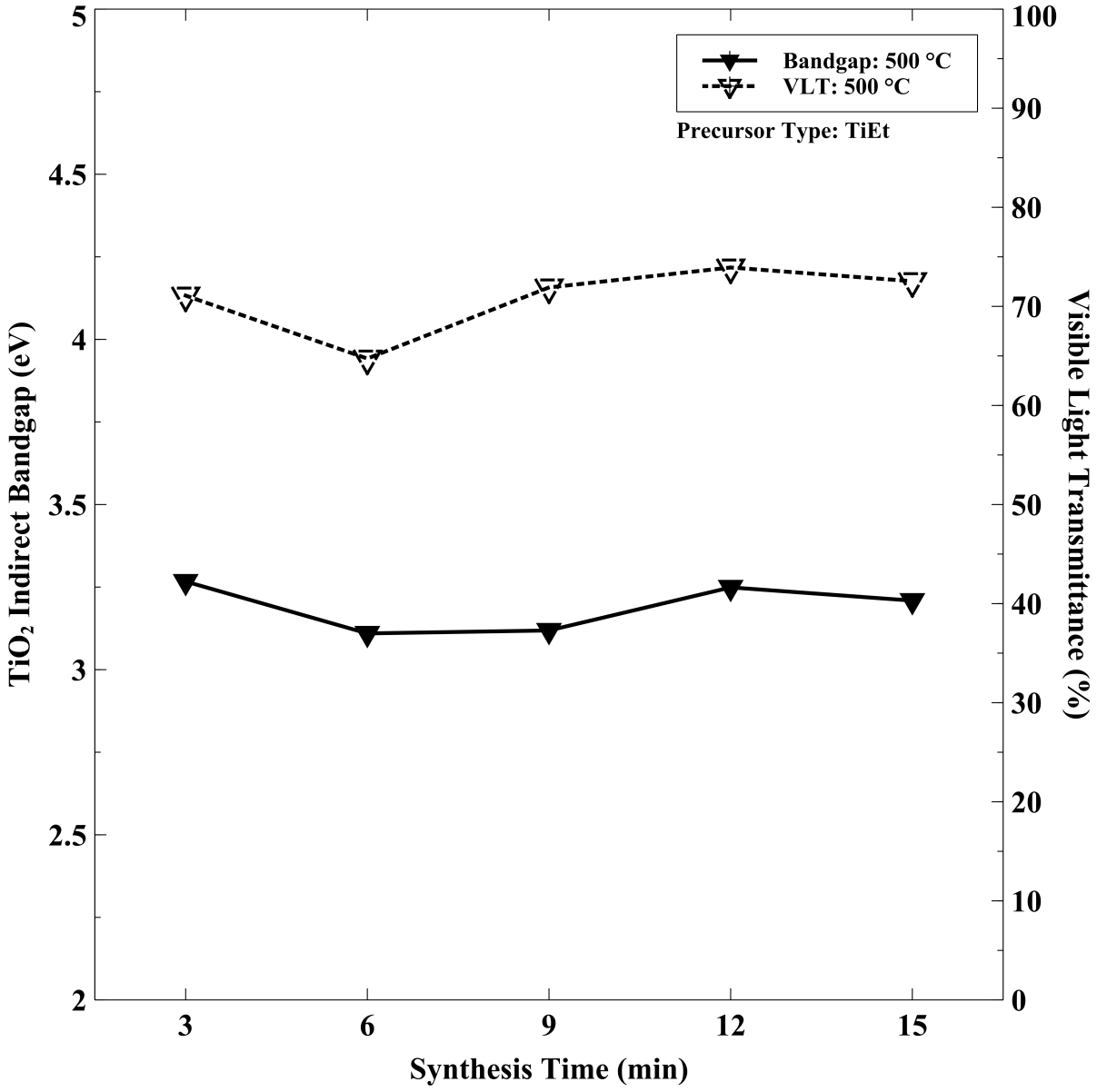



Figure S15: *Using TiEt as the precursor; variations in $TiO_2$ bandgap and visible light transmittance with synthesis time: Black lines with triangle symbol for the deposition temperature at 500 °C. Solid lines depict $TiO_2$ bandgap and dashed lines represent visible light transmittance.*

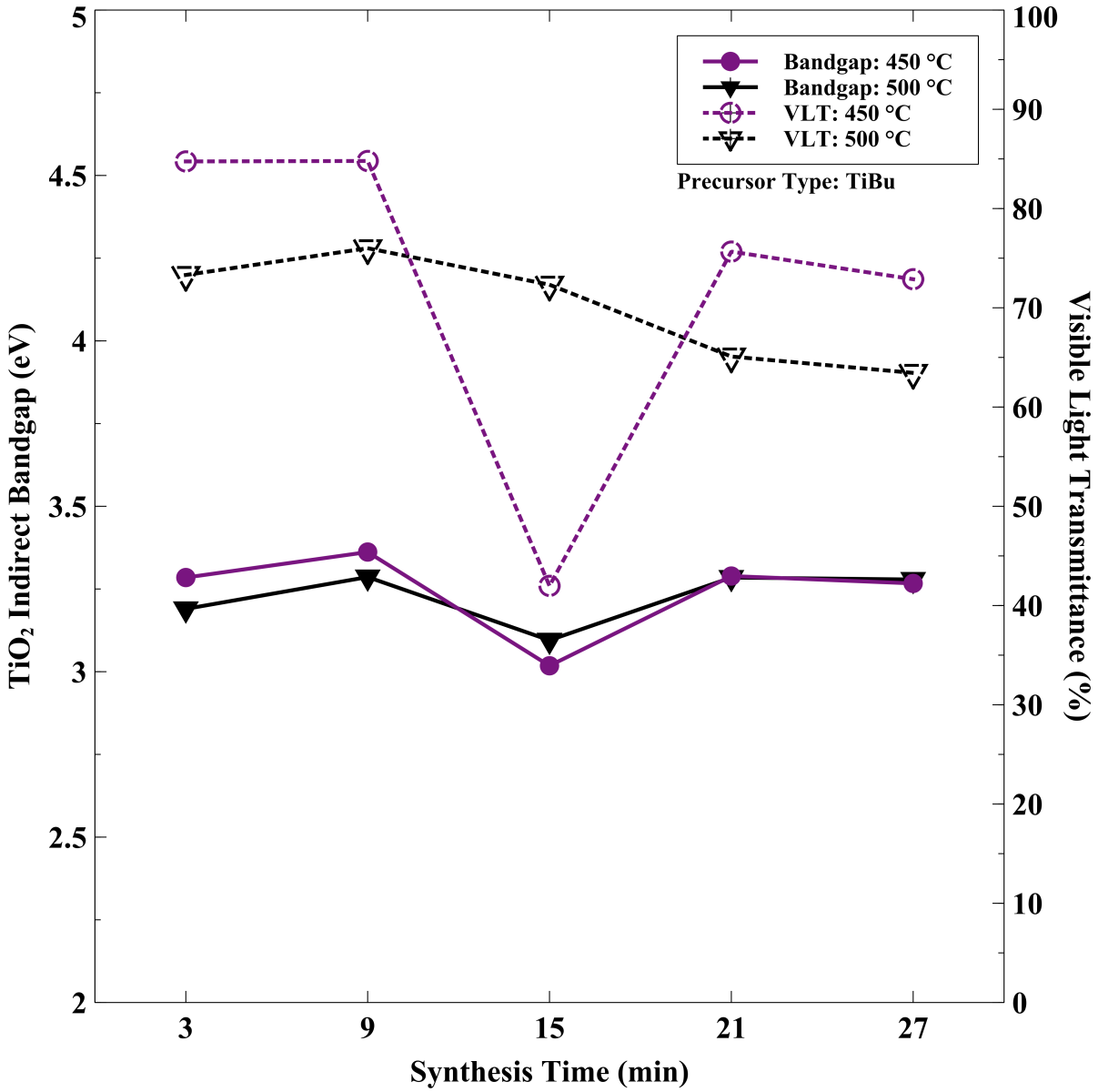



Figure S16: *Using TiBu as the precursor; variations in $TiO_2$ bandgap and visible light transmittance with synthesis time: Purple lines with circle symbol for the deposition temperature at 450 °C, Black lines with triangle symbol for the deposition temperature at 500 °C. Solid lines depict $TiO_2$ bandgap and dashed lines represent visible light transmittance.*

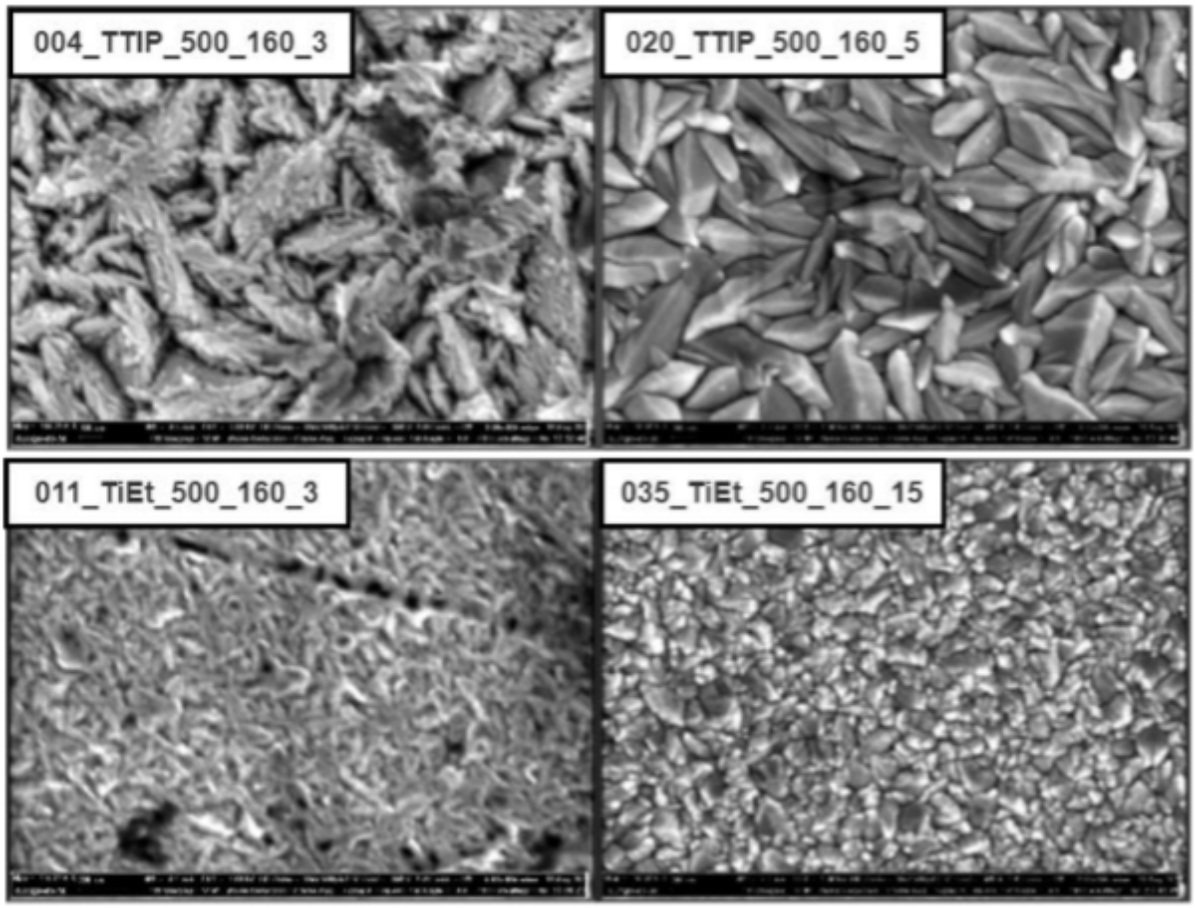



Figure S17: *SEM images of selected coatings produced using TTIP and TiEt at a constant deposition temperature of 500 °C and bubbler temperature of 160 °C.*

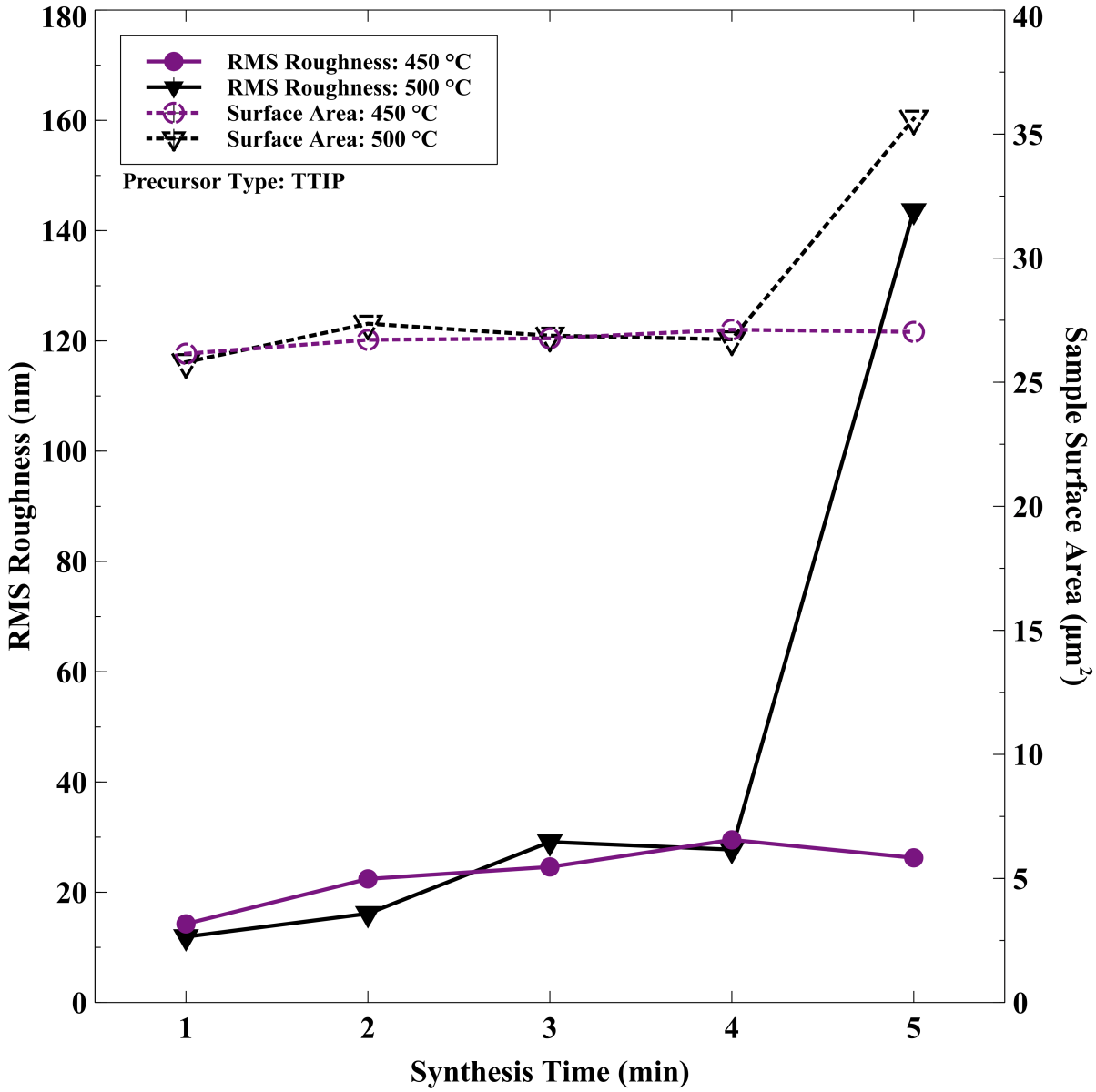



Figure S18: *Using TTIP as the precursor; variations in Rq roughness and sample surface area (per 25 $\mu m^2$ geometric area) with synthesis time: Purple lines with circle symbol for the deposition temperature at 450 °C, Black lines with triangle symbol for the deposition temperature at 500 °C. Solid lines depict Rq roughness and dashed lines represent sample surface area.*

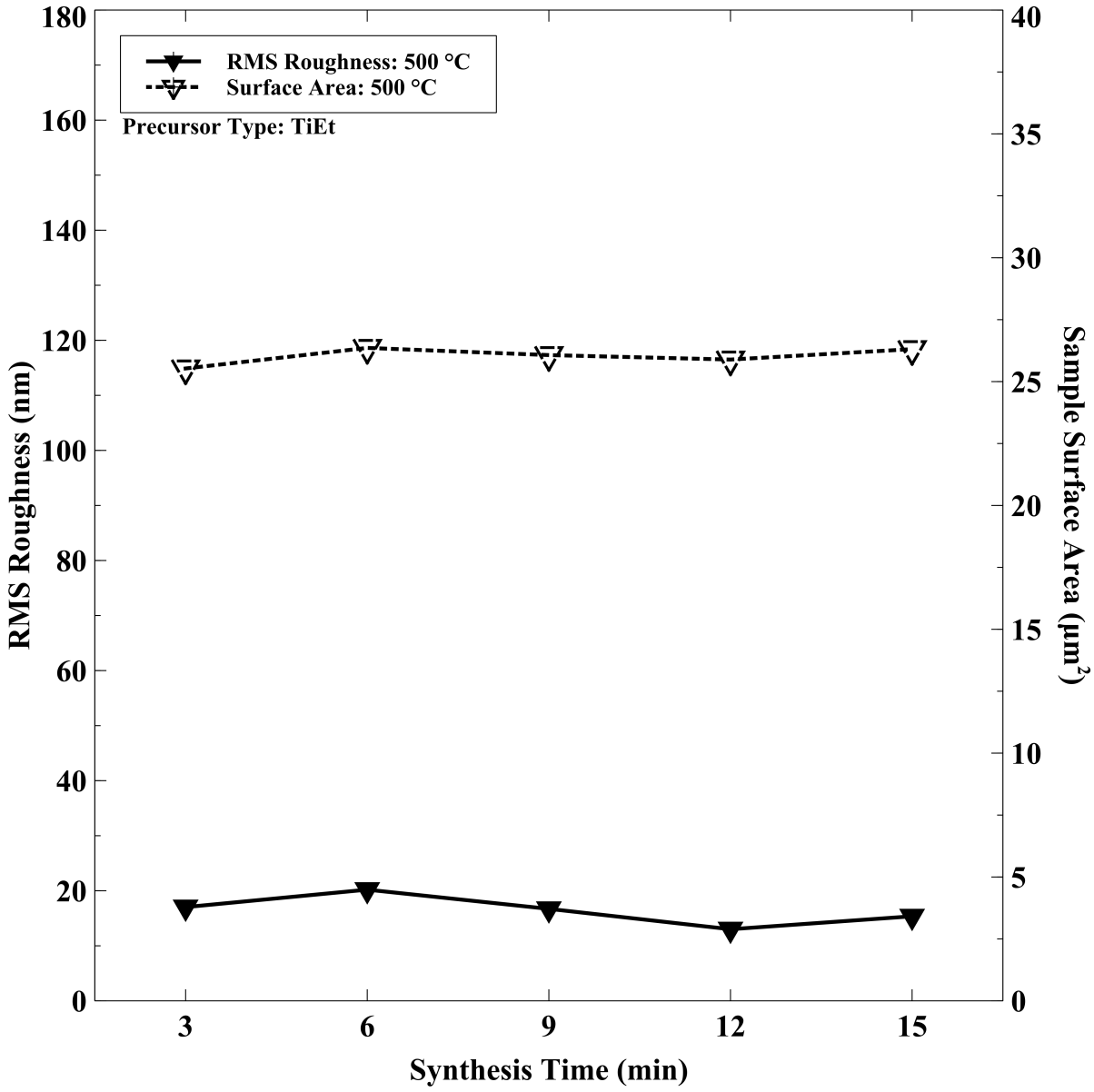



Figure S19: *Using TiEt as the precursor; variations in Rq roughness and sample surface area (per 25 $\mu m^2$ geometric area) with synthesis time: Black lines with triangle symbol for the deposition temperature at 500 °C. Solid lines depict Rq roughness and dashed lines represent sample surface area.*

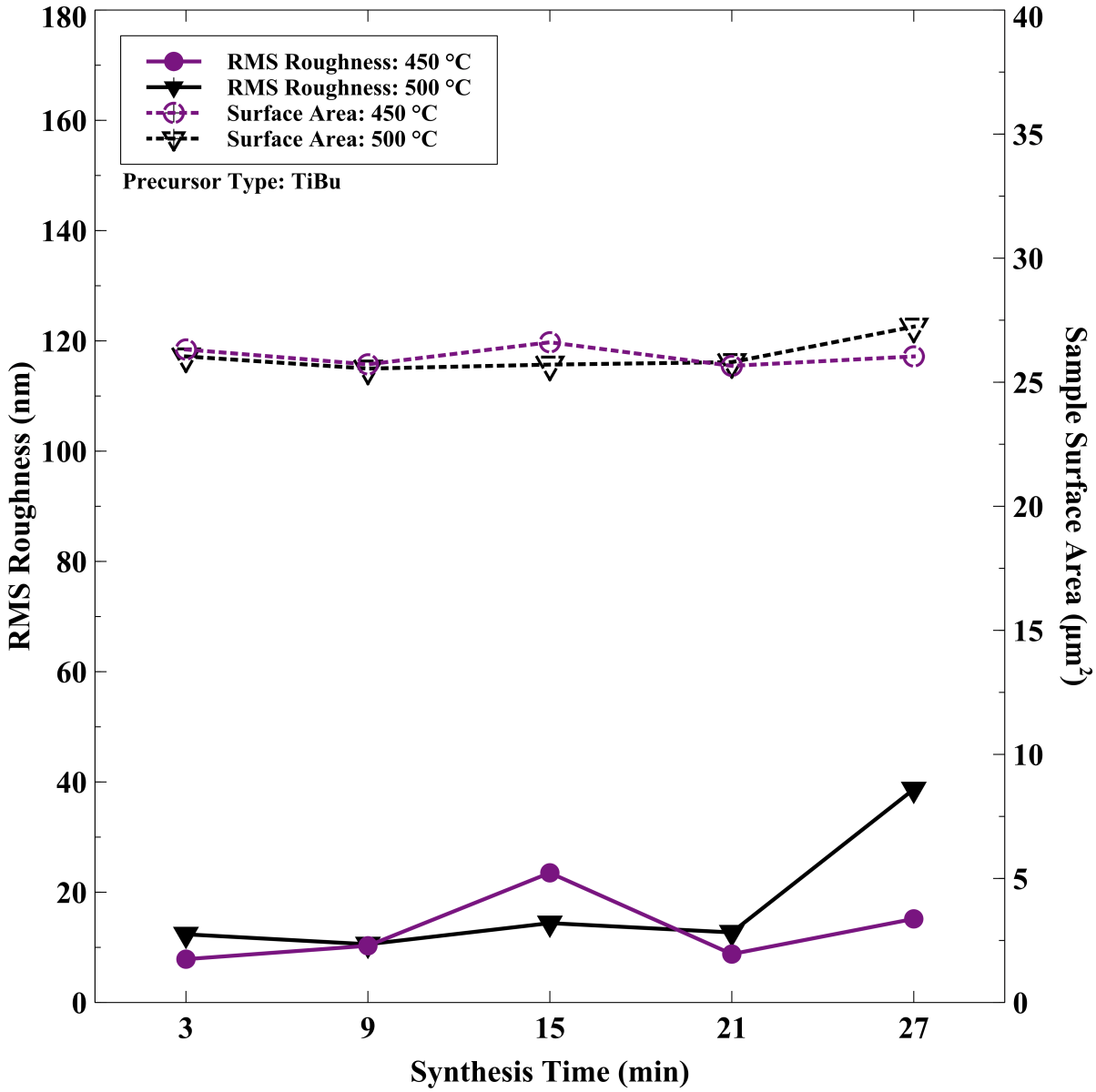



Figure S20: *Using TiBu as the precursor; variations in Rq roughness and sample surface area (per 25 $\mu m^2$ geometric area) with synthesis time: Purple lines with circle symbol for the deposition temperature at 450 °C, Black lines with triangle symbol for the deposition temperature at 500 °C. Solid lines depict Rq roughness and dashed lines represent sample surface area.*

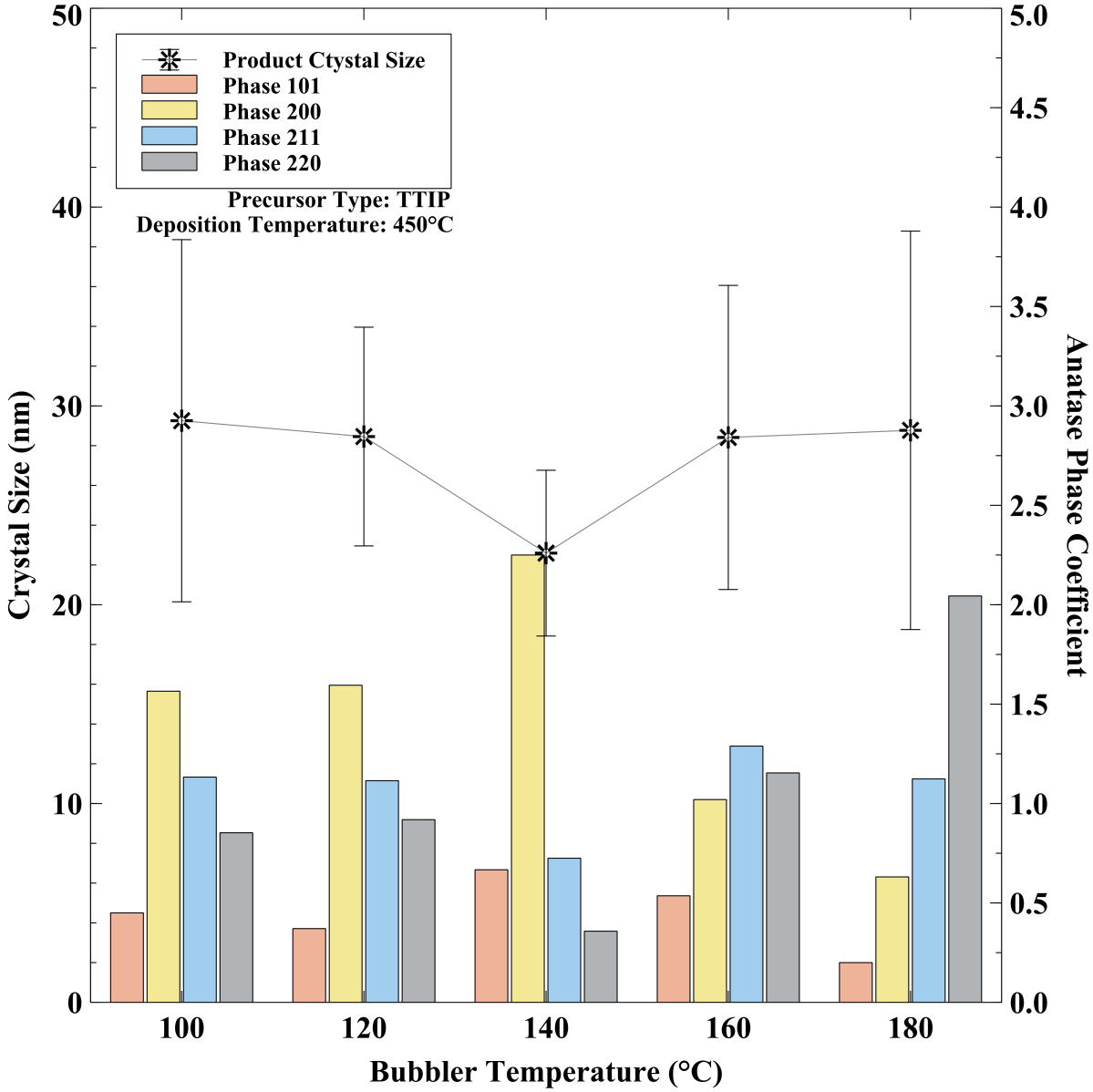



Figure S21: *Variations in crystal size and facet orientations at different bubbler temperature using TTIP as the precursor with a bubbler temperature of 160°C and a deposition temperature of 450°C. The scatter line graph represents the change in crystal size with varying deposition temperatures. The bar graph showcases the orientation distribution of different crystal facets. The colors corresponding to each facet are: 101 facet: Pink; 200 facet: Yellow; 211 facet: SkyBlue; 220 facet: Grey.*

### A.3.3 Effect of Bubbler Temperature

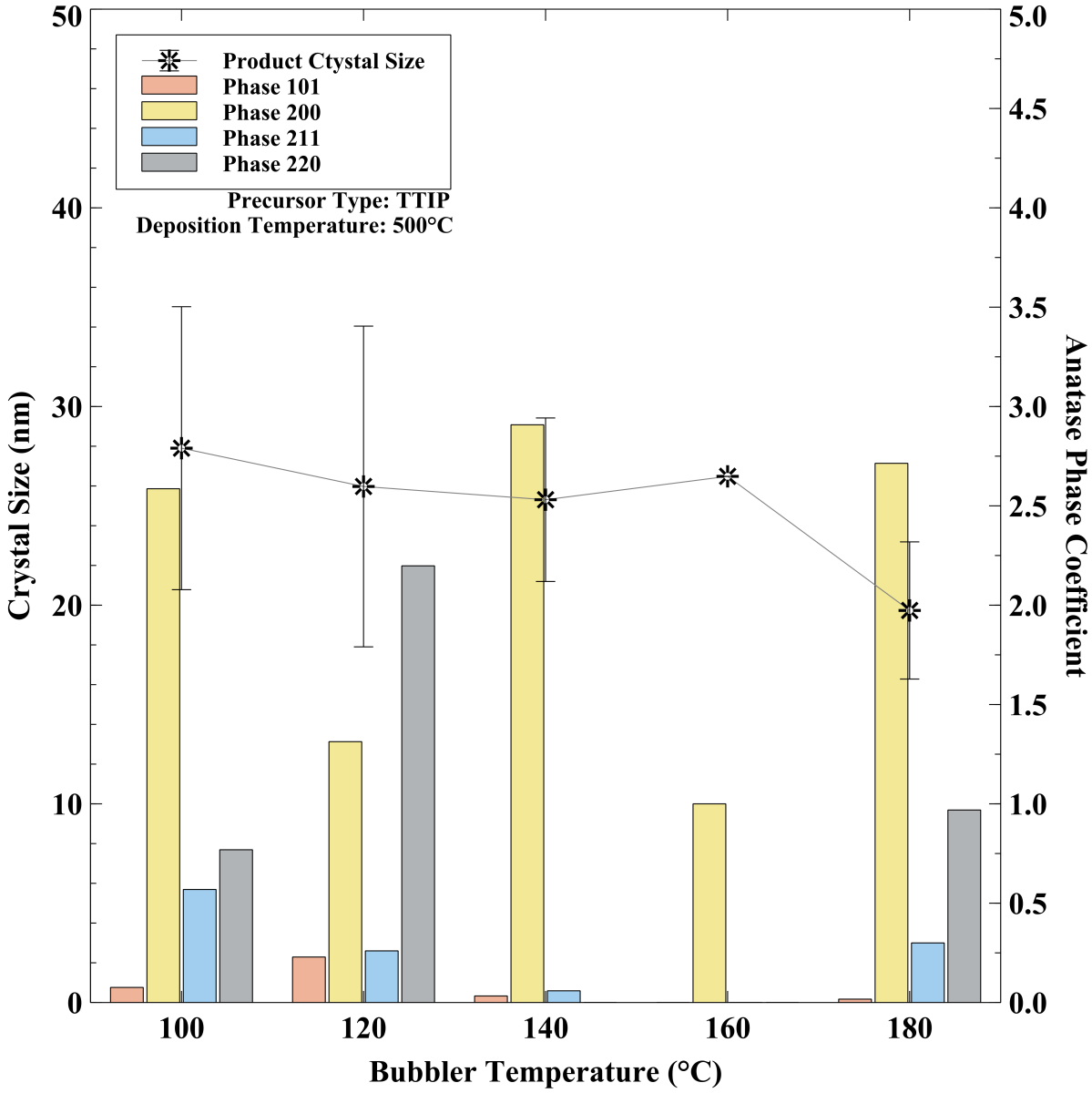



Figure S22: *Variations in crystal size and facet orientations at different bubbler temperature using TTIP as the precursor with a bubbler temperature of 160 °C and a deposition temperature of 500 °C. The scatter line graph represents the change in crystal size with varying deposition temperatures. The bar graph showcases the orientation distribution of different crystal facets. The colors corresponding to each facet are: 101 facet: Pink; 200 facet: Yellow; 211 facet: SkyBlue; 220 facet: Grey.*

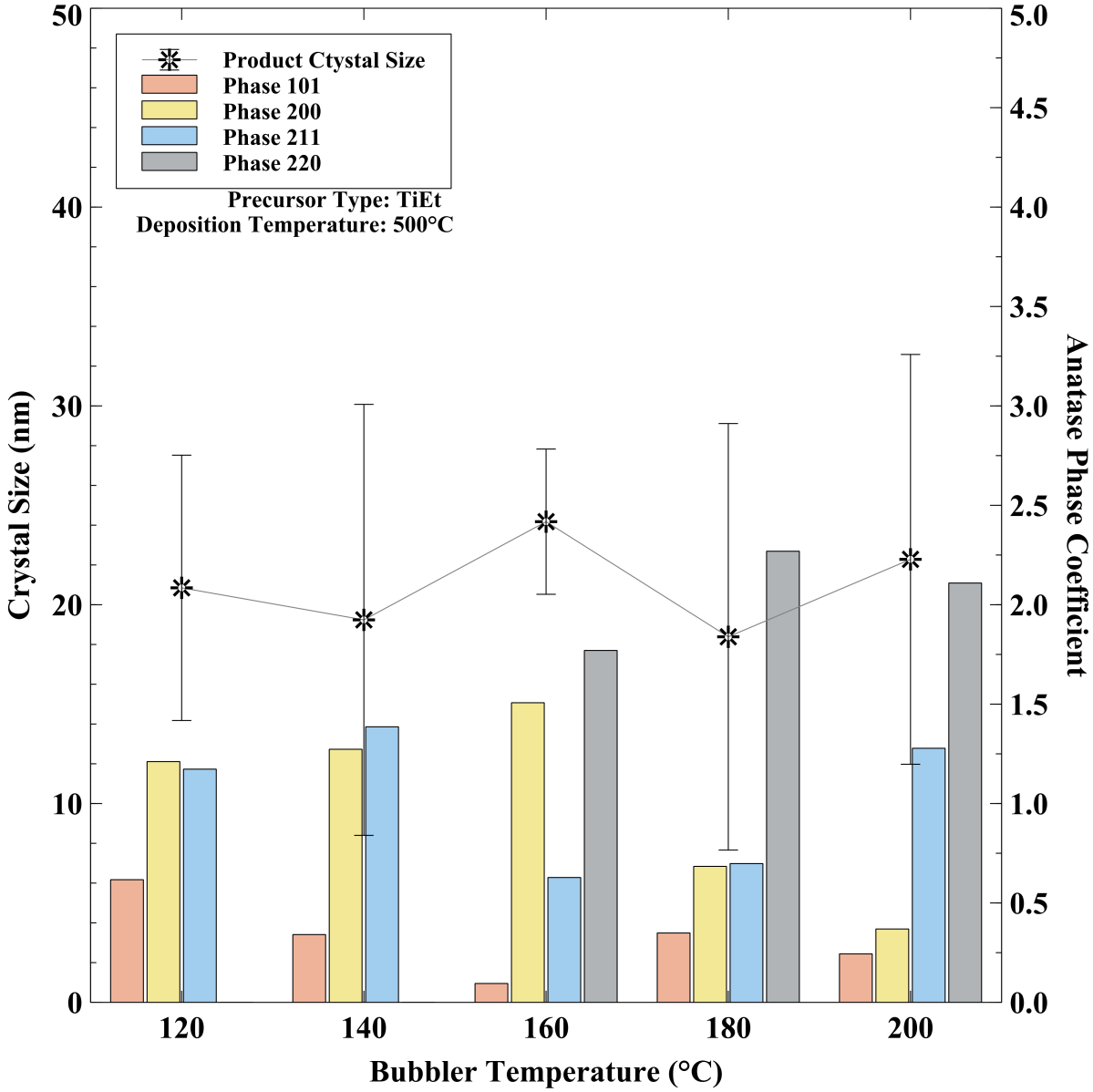



Figure S23: *Variations in crystal size and facet orientations at different bubbler temperature using TiEt as the precursor with a bubbler temperature of 160 °C and a deposition temperature of 500 °C. The scatter line graph represents the change in crystal size with varying deposition temperatures. The bar graph showcases the orientation distribution of different crystal facets. The colors corresponding to each facet are: 101 facet: Pink; 200 facet: Yellow; 211 facet: SkyBlue; 220 facet: Grey.*

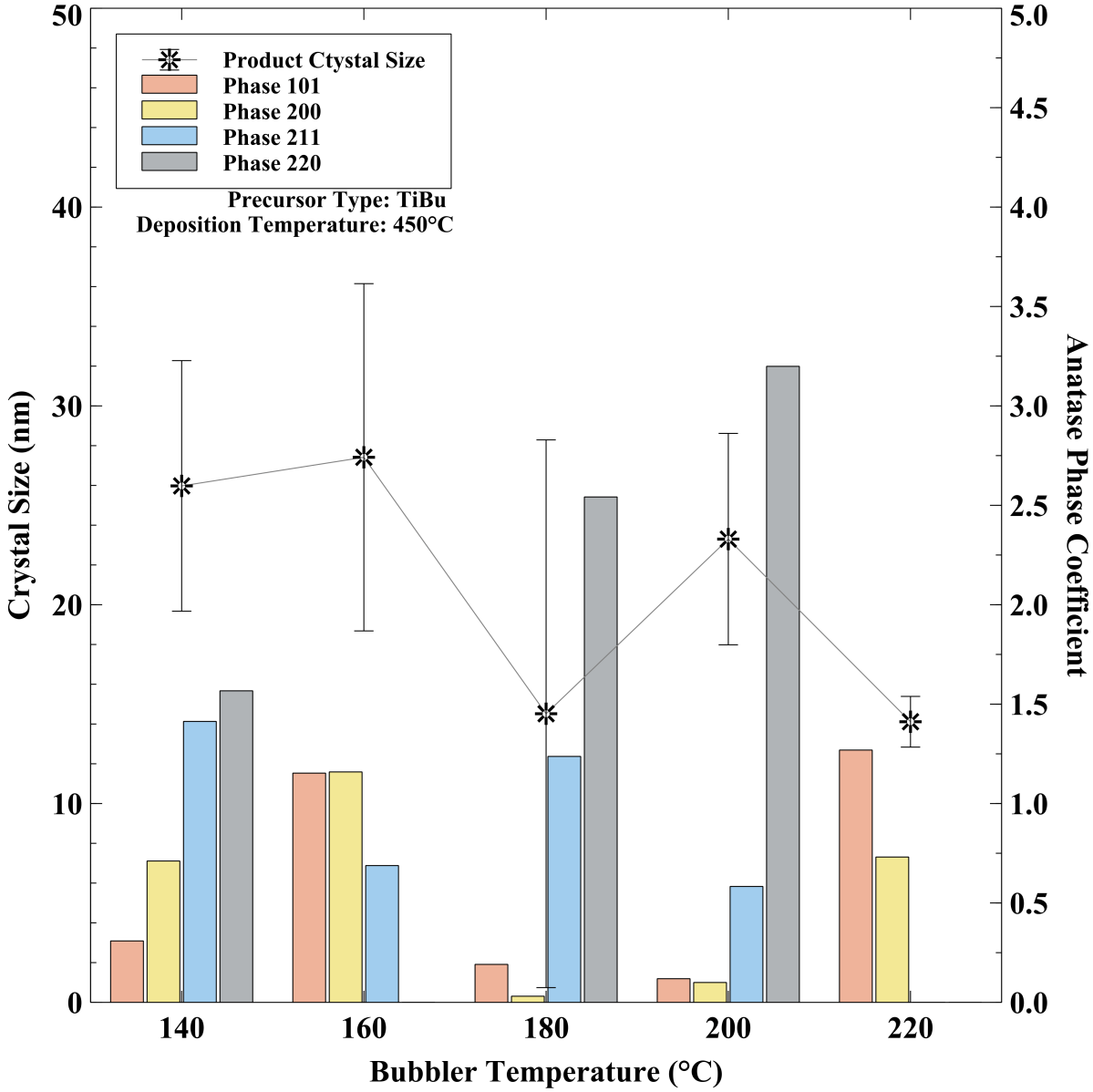



Figure S24: *Variations in crystal size and facet orientations at different bubbler temperature using TiBu as the precursor with a bubbler temperature of 160 °C and a deposition temperature of 450 °C. The scatter line graph represents the change in crystal size with varying deposition temperatures. The bar graph showcases the orientation distribution of different crystal facets. The colors corresponding to each facet are: 101 facet: Pink; 200 facet: Yellow; 211 facet: SkyBlue; 220 facet: Grey.*

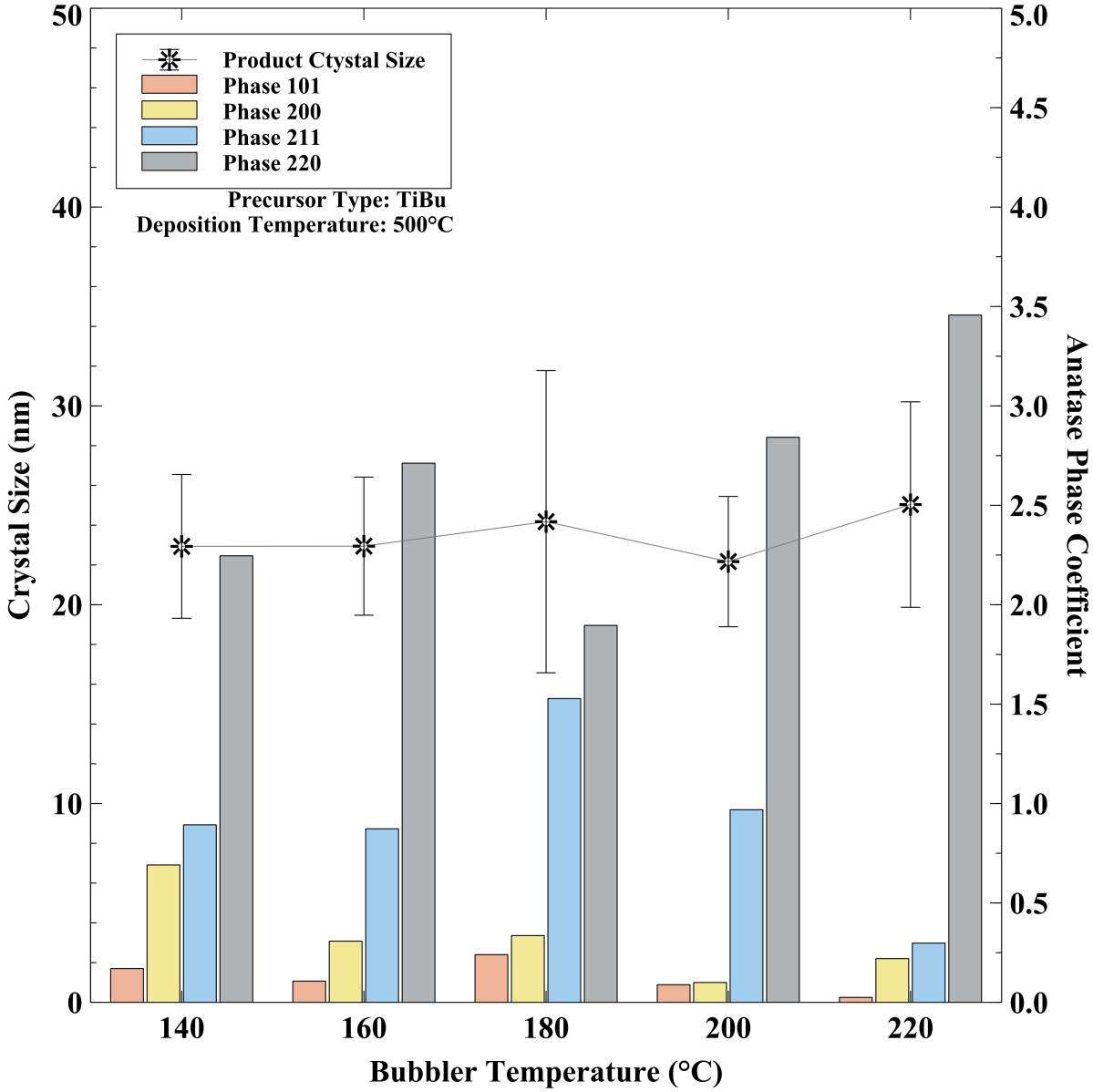



Figure S25: *Variations in crystal size and facet orientations at different bubbler temperature using TiBu as the precursor with a bubbler temperature of 160 °C and a deposition temperature of 500 °C. The scatter line graph represents the change in crystal size with varying deposition temperatures. The bar graph showcases the orientation distribution of different crystal facets. The colors corresponding to each facet are: 101 facet: Pink; 200 facet: Yellow; 211 facet: SkyBlue; 220 facet: Grey.*

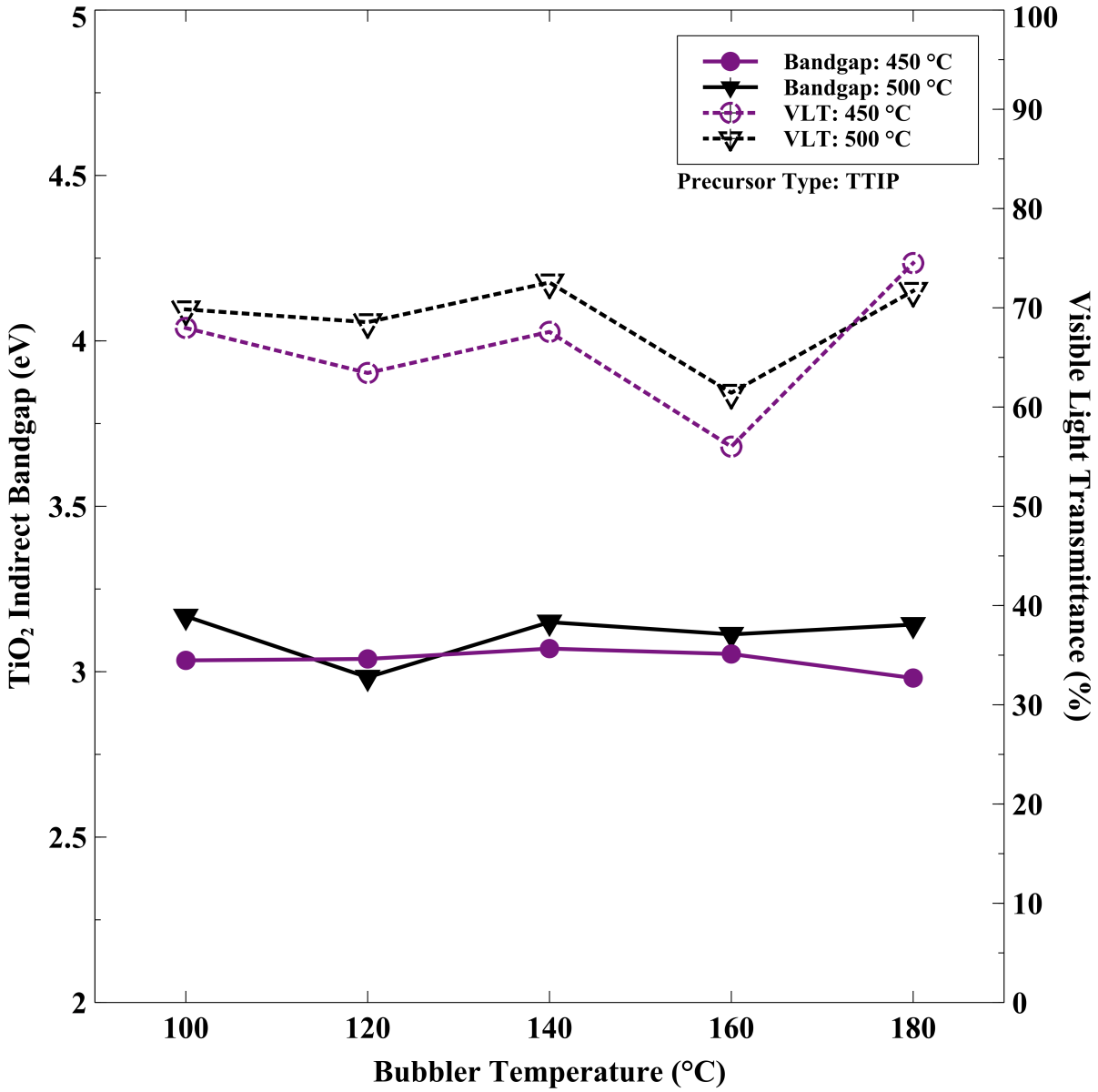



Figure S26: *Using TTIP as the precursor; variations in $TiO_2$ bandgap and visible light transmittance with bubbler temperature: Purple lines with circle symbol for the deposition temperature at 450 °C, Black lines with triangle symbol for the deposition temperature at 500 °C. Solid lines depict $TiO_2$ bandgap and dashed lines represent visible light transmittance.*

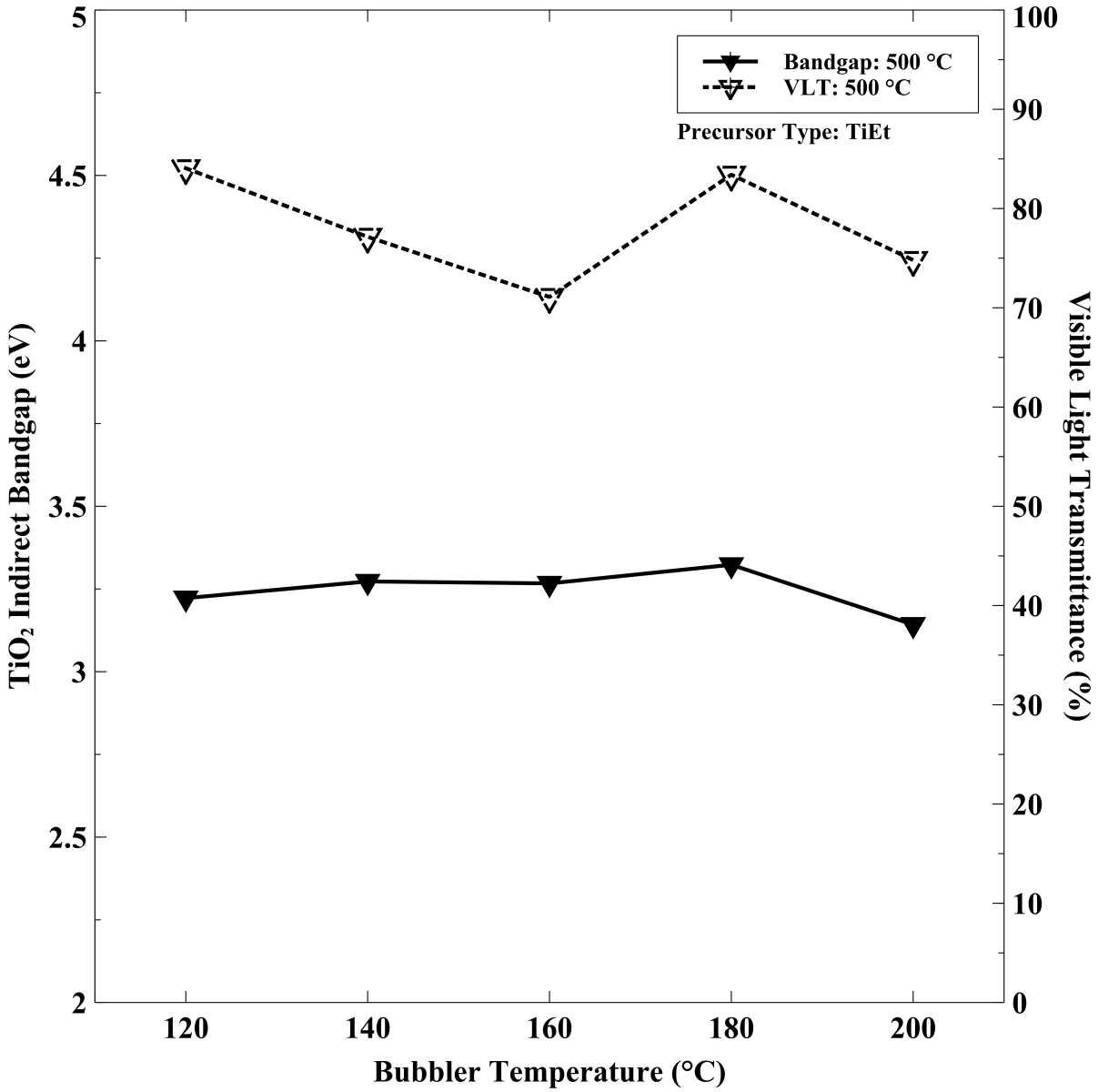



Figure S27: *Using TiEt as the precursor; variations in $TiO_2$ bandgap and visible light transmittance with bubbler temperature: Black lines with triangle symbol for the deposition temperature at 500 °C. Solid lines depict $TiO_2$ bandgap and dashed lines represent visible light transmittance.*

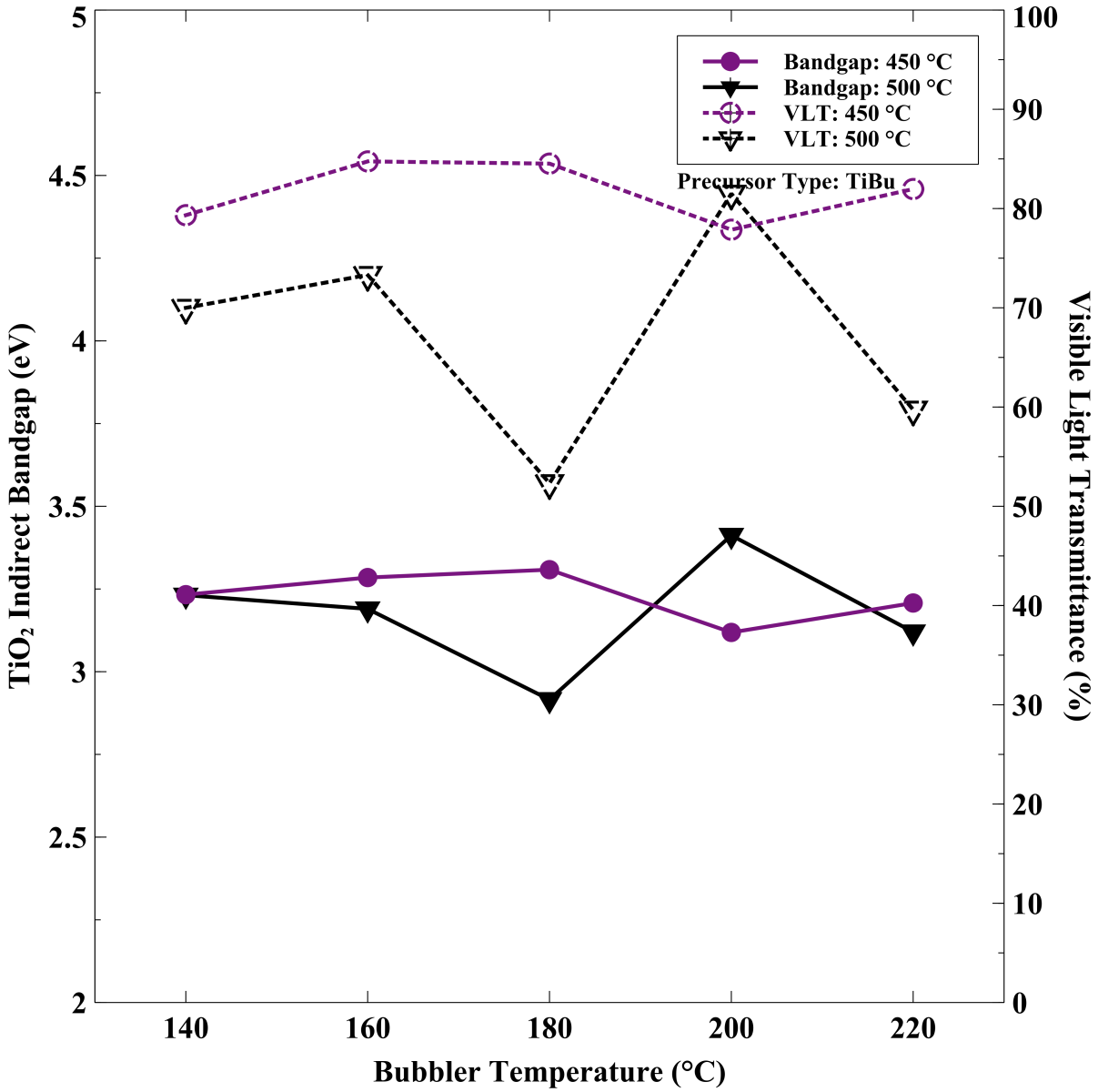



Figure S28: *Using TiBu as the precursor; variations in $TiO_2$ bandgap and visible light transmittance with bubbler temperature: Purple lines with circle symbol for the deposition temperature at 450 °C, Black lines with triangle symbol for the deposition temperature at 500 °C. Solid lines depict $TiO_2$ bandgap and dashed lines represent visible light transmittance.*

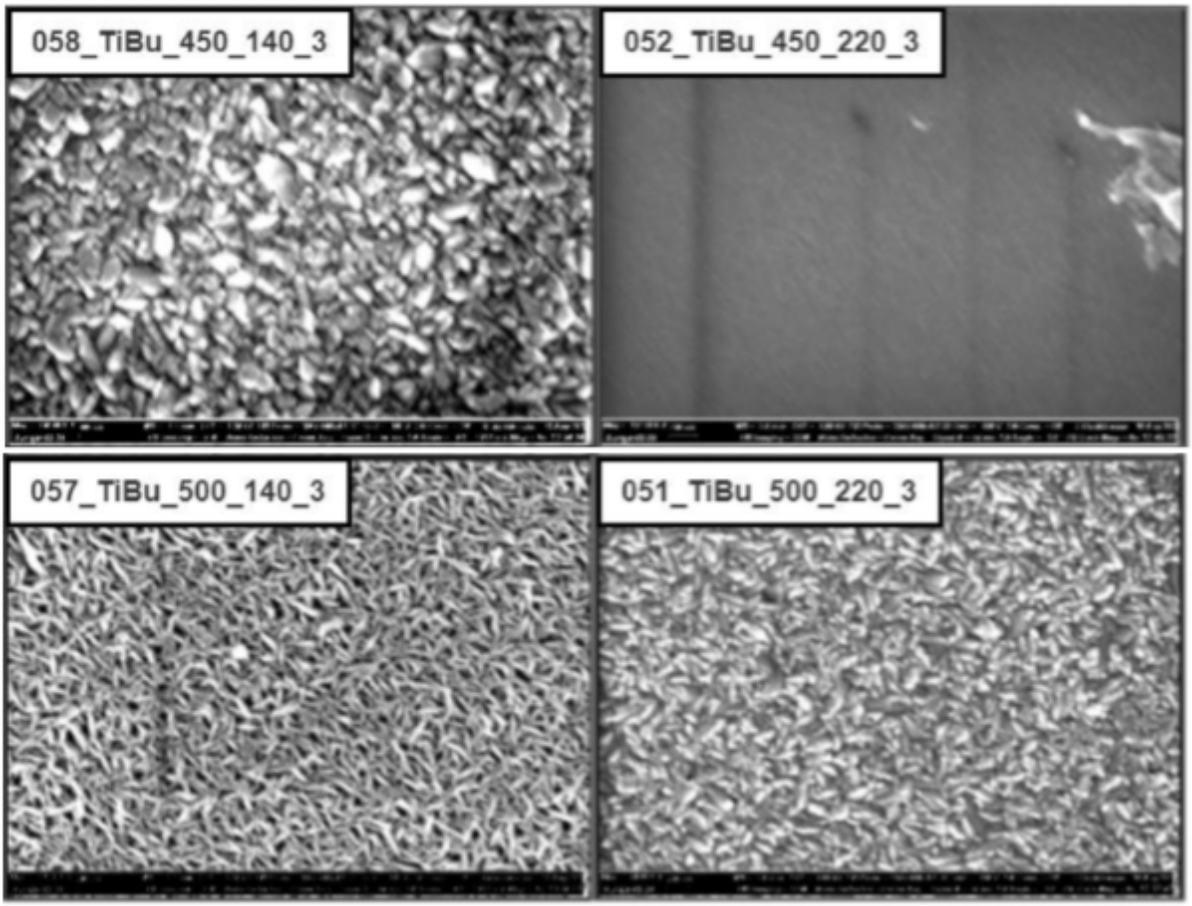



Figure S29: *SEM images of selected coatings produced using TiBu at a constant reaction time of 3 mins and either a deposition temperature of 450 or 500 °C and bubbler temperature of 140 and 220 °C.*

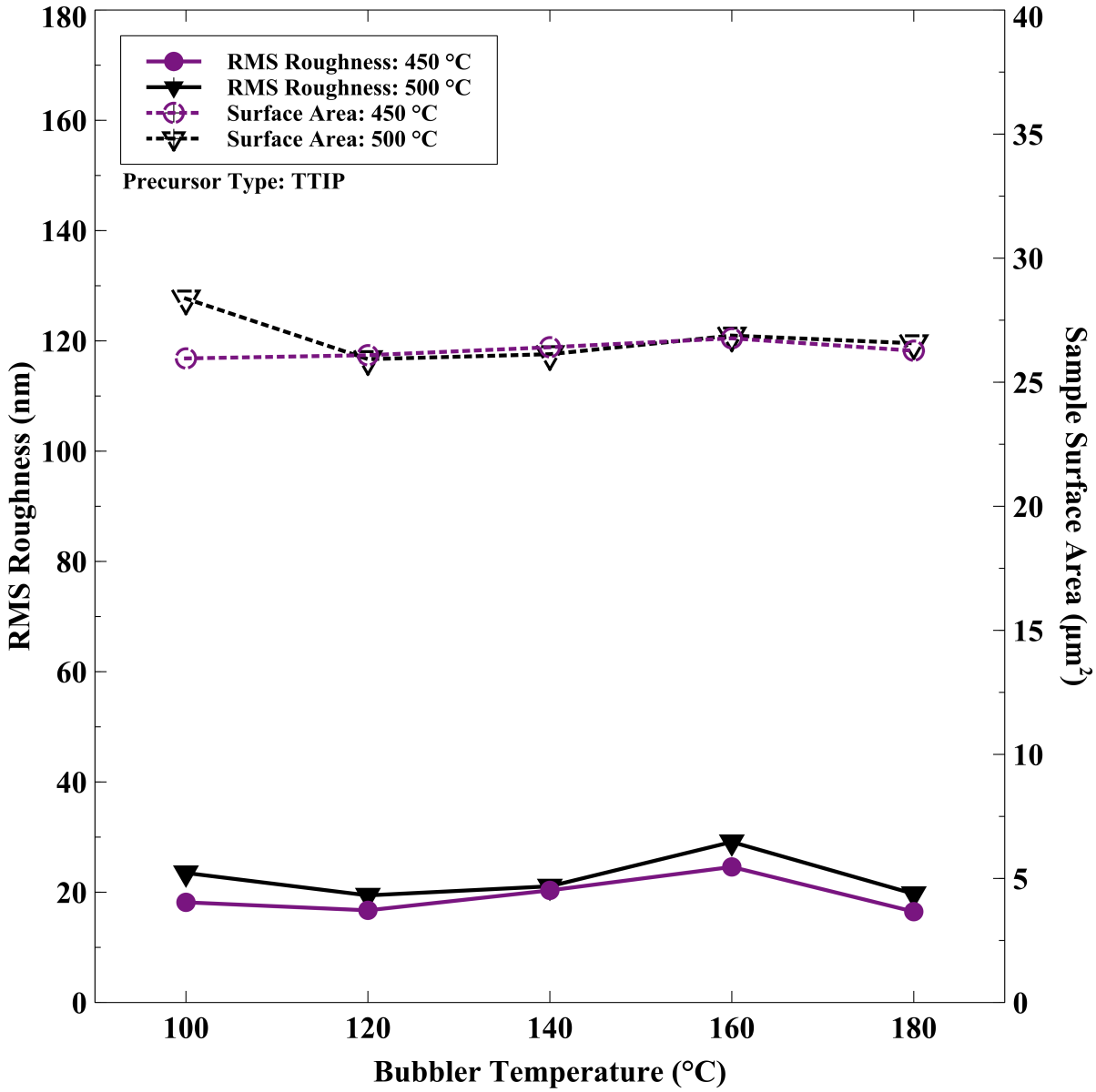



Figure S30: *Using TTIP as the precursor; variations in Rq roughness and sample surface area with bubbler temperature: Purple lines with circle symbol for the deposition temperature at 450 °C, Black lines with triangle symbol for the deposition temperature at 500 °C. Solid lines depict Rq roughness and dashed lines represent sample surface area.*

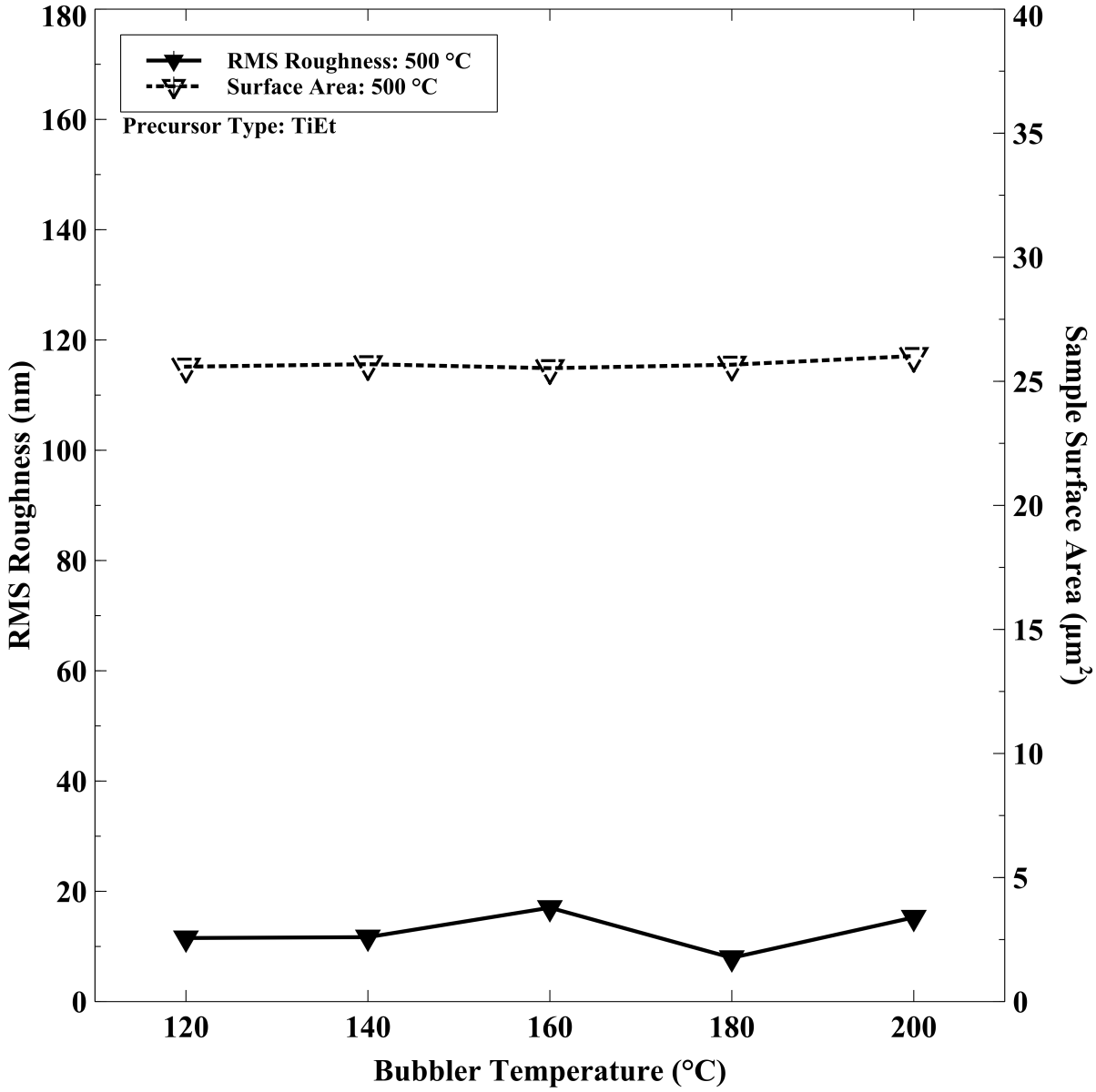



Figure S31: *Using TiEt as the precursor; variations in Rq roughness and sample surface area with bubbler temperature: Black lines with triangle symbol for the deposition temperature at 500 °C. Solid lines depict Rq roughness and dashed lines represent sample surface area.*

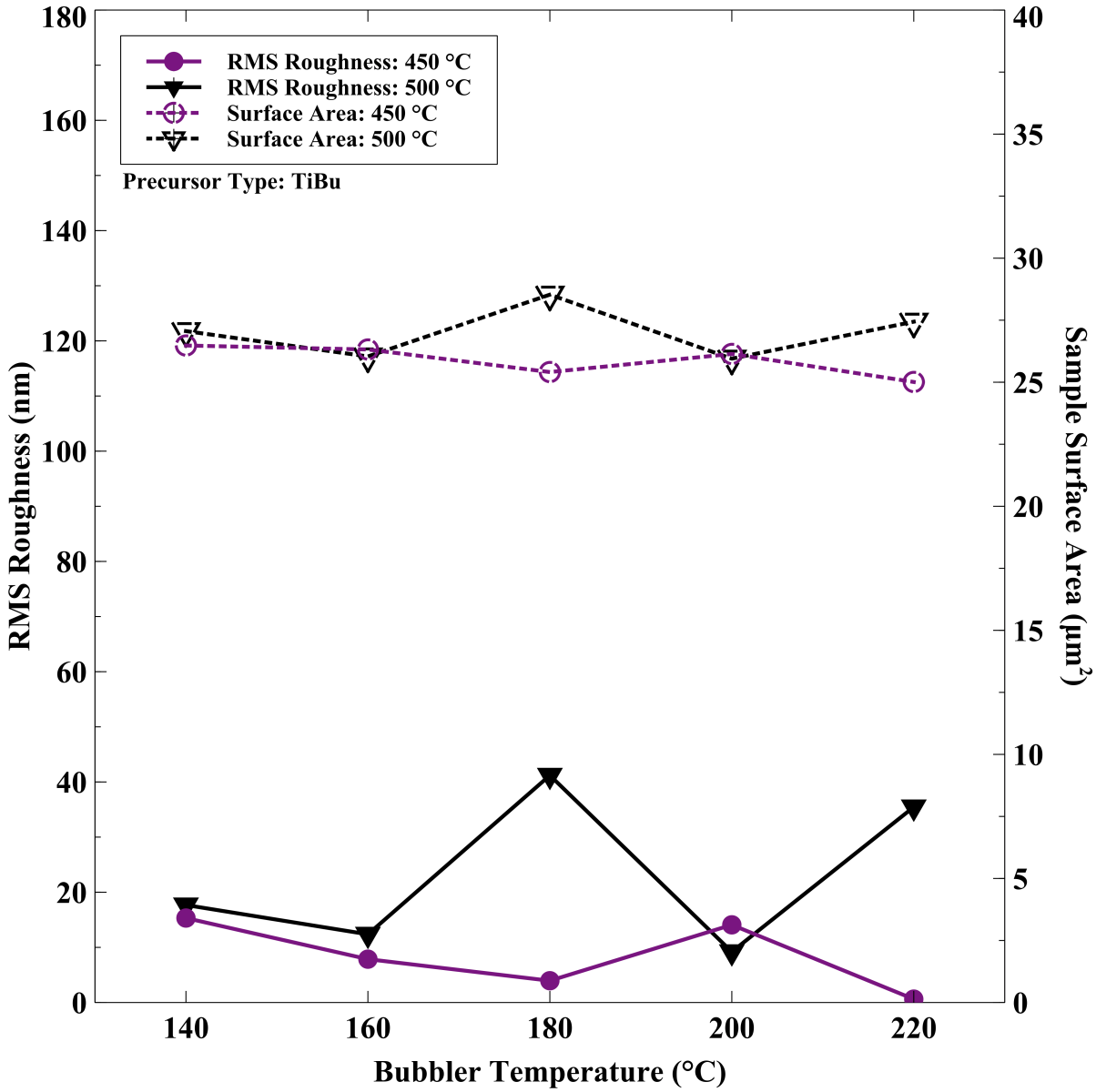



Figure S32: *Using TiBu as the precursor; variations in Rq roughness and sample surface area with bubbler temperature: Purple lines with circle symbol for the deposition temperature at 450 °C, Black lines with triangle symbol for the deposition temperature at 500 °C. Solid lines depict Rq roughness and dashed lines represent sample surface area.*

## A.4 ML-Assisted Insights on Photocatalytic $NO_x$ Activity Prediction

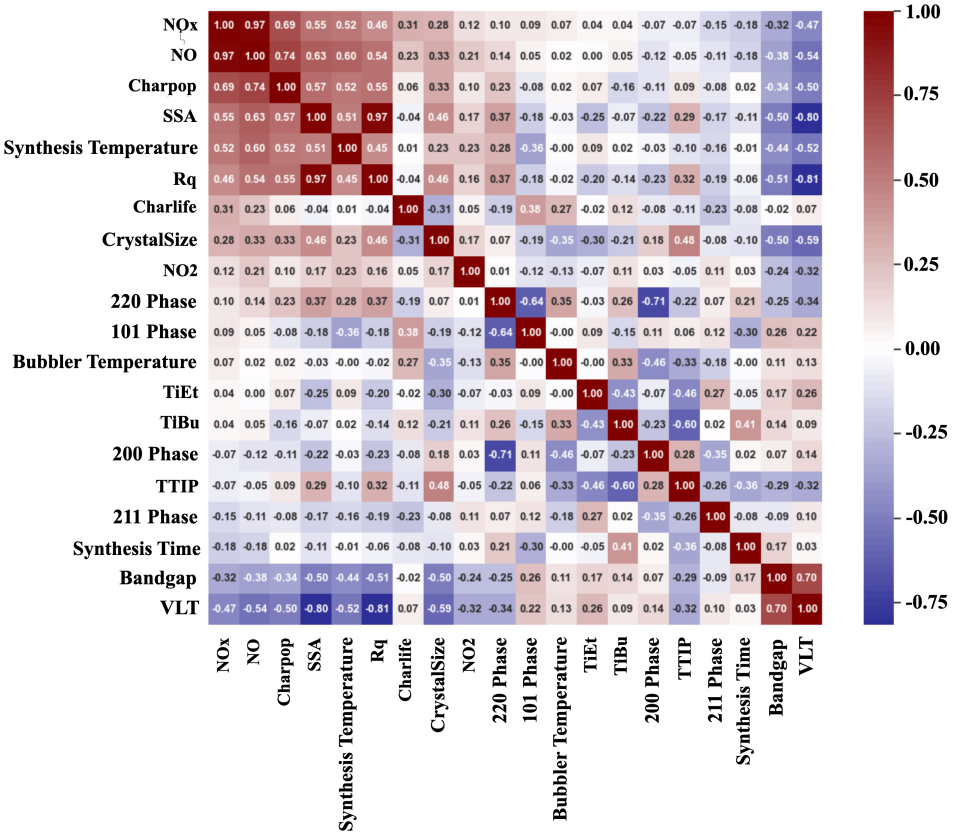



Figure S33: *Pearson correlation coefficients between synthetic parameters (precursor used, deposition temperature, bubbler temperature and synthesis time), physicochemical properties and photocatalytic $NO_x$ activity for experiment data*

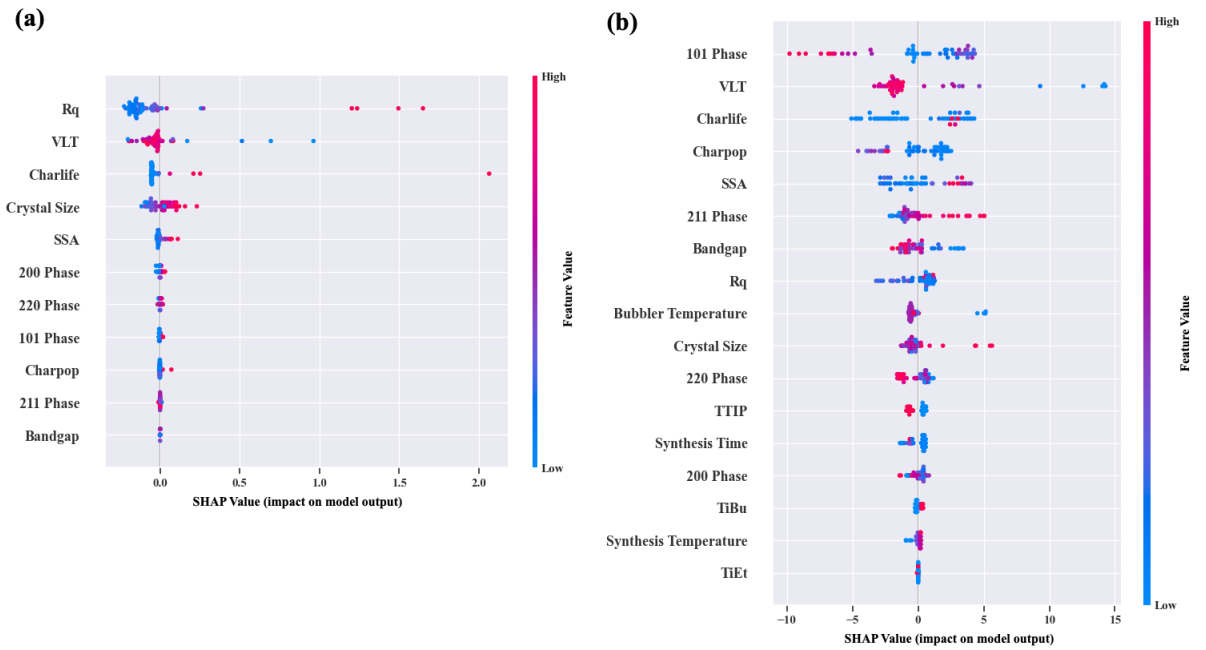
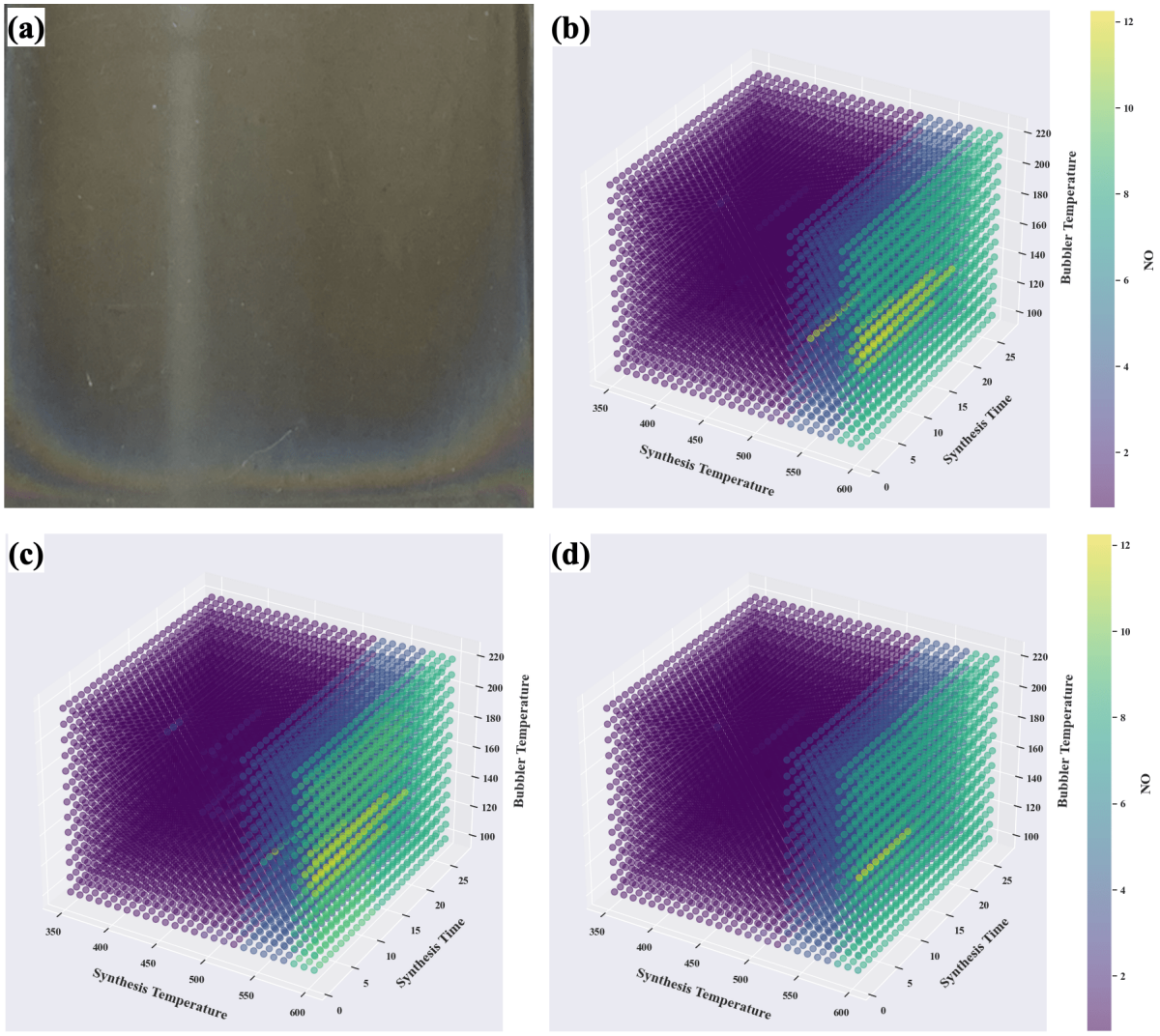



Figure S34: *SHAP value for predicting (a) photocatalytic NO conversion; (b) $NO_2$ selectivity when applying the "Modified Property-Prior Prediction" Strategy.*

The SHAP plot on the left relates to $NO_x$ conversion. It is visible that the features 'Surface Roughness', 'VLT', and 'Charlife' are positioned at the top, indicating higher SHAP values and consequently a more substantial influence on the model's predictive capability. The photocatalytic activity is positively correlated with these properties, as illustrated by the color gradient from blue to red. Similar to NO prediction, this plot implies that increasing these factors could improve the effectiveness of $NO_x$ conversion. The VLT, however, contributes the most negative influence on product activity of $NO_x$ conversion.

In terms of $NO_2$ , it is notable that features such as '101 Phase', 'VLT', and 'Charlife' contribute diverse impacts on the model's output for $NO_2$ selectivity. Surprisely, The '101 Phase' feature demonstrates mostly positive SHAP values, positioning it at the highest point on the plot. This suggests a beneficial correlation between the growth of the (101) plane and the $NO_2$ selectivity, implied the facilitating of (101) facet growth may suppres the NO gas convert to $NO_2$ . On the other hand, 'VLT' exhibits a combination of positive and negative SHAP values, suggesting a complex connection between transparency and $NO_2$ selectivity in our case, and suggesting that the effect of 'VLT' on $NO_2$ selectivity could vary depending on the interplay with other features.




Machine Learning predictions of the photocatalytic $NO_x$ activity for a range of synthetic parameters. (a) A digital image of a coating produced with the following synthetic parameters - Precursor Selection: TiEt; Deposition Temperature: 430 °C; Bubbler Temperature: 210 °C; Synthesis Time: 1 minute. 4D scatter plots of ML simulated results of the photocatalytic NO removal activity against synthetic parameters (deposition temperature ( °C), bubbler temperature ( °C) and synthesis time (mins)) for (b) TTIP (c),TiBu and (d) TiEt as precursors.

# B Tables

Table S1: *Sample numbers, experimental synthesis conditions, and photocatalytic $NO_x$ performance of all 58 $TiO_2$ coatings produced herein uing atmopsheric pressure chemical vapour deposition (APCVD) on glass.*

| **Sample No.** | **Precursor Selection** | **Synthesis Temperature (°C)** | **Precursor Preheating Temperature (°C)** | **Synthesis Time (min)** | **NO Conversion (%)** | **NO error** | **$NO_x$ Conversion(%)** | **$NO_x$ Error** | **$NO_2$ Selectivity (%)** | **$NO_2$ Error** |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | TTIP | 350 | 160 | 3 | 0.606 | 0.172 | 0.232 | 0.162 | 61.098 | 19.934 |
| 2 | TTIP | 400 | 160 | 3 | 1.427 | 0.335 | 0.798 | 0.1 | 44.226 | 21.447 |
| 3 | TTIP | 450 | 160 | 3 | 0.302 | 0.165 | 0.038 | 0.087 | 87.174 | 57.297 |
| 4 | TTIP | 500 | 160 | 3 | 1.941 | 0.228 | 0.417 | 0.117 | 78.405 | 12.687 |
| 5 | TTIP | 550 | 160 | 3 | 11.43 | 1.079 | 2.342 | 0.375 | 79.345 | 10.473 |
| 6 | TTIP | 600 | 160 | 3 | 14.977 | 1.687 | 3.278 | 0.593 | 77.87 | 11.976 |
| 7 | TiEt | 350 | 160 | 3 | 1.568 | 0.342 | 1.056 | 0.176 | 31.037 | 15.614 |
| 8 | TiEt | 400 | 160 | 3 | 0.773 | 0.458 | 0.319 | 0.11 | 57.894 | 64.21 |
| 9 | TiEt | 450 | 160 | 3 | 0.865 | 0.234 | 0.318 | 0.147 | 62.733 | 22.295 |
| 10 | TiEt | 500 | 160 | 3 | 0.63 | 0.271 | 0.206 | 0.083 | 66.932 | 41.99 |
| 11 | TiEt | 550 | 160 | 3 | 3.65 | 0.65 | 1.334 | 0.258 | 62.689 | 16.049 |
| 12 | TiEt | 600 | 160 | 3 | 22.675 | 1.961 | 6.627 | 0.957 | 70.171 | 8.908 |
| 13 | TiBu | 350 | 160 | 3 | 0.489 | 0.164 | 0.37 | 0.119 | 23.255 | 22.345 |
| 14 | TiBu | 400 | 160 | 3 | 0.256 | 0.154 | 0.229 | 0.074 | 9.184 | 32.733 |
| 15 | TiBu | 450 | 160 | 3 | 0.655 | 0.217 | 0.017 | 0.414 | 54.303 | 28.547 |
| 16 | TiBu | 500 | 160 | 3 | 5.416 | 0.754 | 1.133 | 0.219 | 78.91 | 15.532 |
| 17 | TiBu | 550 | 160 | 3 | 10.854 | 1.025 | 2.653 | 0.368 | 75.373 | 10.23 |
| 18 | TiBu | 600 | 160 | 3 | 12.685 | 1.141 | 3.203 | 0.434 | 74.556 | 9.398 |
| 19 | TTIP | 450 | 160 | 5 | 0.359 | 0.247 | 0.258 | 0.183 | 27.658 | 26.371 |
| 20 | TTIP | 500 | 160 | 5 | 7.72 | 0.844 | 2.069 | 0.347 | 73.15 | 11.127 |
| 21 | TTIP | 450 | 160 | 2 | 0.205 | 0.115 | 0.171 | 0.099 | 16.138 | 13.309 |
| 22 | TTIP | 500 | 160 | 2 | 0.145 | 0.081 | 0.061 | 0.035 | 58.009 | 49.97 |
| 23 | TTIP | 450 | 160 | 4 | 0.103 | 0.086 | 0.002 | 0.032 | 98.062 | 110.451 |
| 24 | TTIP | 500 | 160 | 4 | 0.362 | 0.154 | 0.405 | 0.105 | 12.307 | 17.047 |
| 25 | TTIP | 450 | 160 | 1 | 0.025 | 0.123 | 0.013 | 0.057 | 48.366 | 365.092 |
| 26 | TTIP | 500 | 160 | 1 | 0.074 | 0.042 | 0.051 | 0.028 | 31.272 | 60.697 |
| 27 | TTIP | 450 | 180 | 3 | 1.884 | 0.398 | 0.781 | 0.182 | 57.982 | 17.751 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 28 | TTIP | 500 | 180 | 3 | 0.308 | 0.123 | 0.146 | 0.081 | 52.317 | 34.635 |
| 29 | TTIP | 450 | 140 | 3 | 0.571 | 0.18 | 0.359 | 0.115 | 36.589 | 18.846 |
| 30 | TTIP | 500 | 140 | 3 | 0.154 | 0.126 | 0.18 | 0.119 | 16.99 | 24.789 |
| 31 | TTIP | 450 | 120 | 3 | 0.595 | 0.181 | 0.3 | 0.1 | 49.134 | 21.662 |
| 32 | TTIP | 500 | 120 | 3 | 0.015 | 0.127 | 0.009 | 0.06 | 161.474 | 1460.097 |
| 33 | TTIP | 450 | 100 | 3 | 2.355 | 0.482 | 0.456 | 0.122 | 80.485 | 22.952 |
| 34 | TTIP | 500 | 100 | 3 | 2.237 | 0.4 | 0.548 | 0.12 | 75.342 | 19.036 |
| 35 | TiEt | 500 | 160 | 15 | 0.494 | 0.191 | 0.151 | 0.091 | 69.208 | 33.999 |
| 36 | TiEt | 500 | 160 | 12 | 0.331 | 0.18 | 0.145 | 0.086 | 56.045 | 42.477 |
| 37 | TiEt | 500 | 160 | 9 | 0.045 | 0.054 | 0.049 | 0.028 | 9.742 | 70.311 |
| 38 | TiEt | 500 | 160 | 6 | 0.146 | 0.096 | 0.124 | 0.06 | 14.69 | 55.586 |
| 39 | TiEt | 500 | 200 | 3 | 0.572 | 0.194 | 0.095 | 0.045 | 83.225 | 40.113 |
| 40 | TiEt | 500 | 180 | 3 | 0.233 | 0.181 | 0.064 | 0.06 | 72.777 | 77.031 |
| 41 | TiEt | 500 | 140 | 3 | 0.961 | 0.308 | 0.157 | 0.111 | 83.512 | 34.072 |
| 42 | TiEt | 500 | 120 | 3 | 1.307 | 0.37 | 0.23 | 0.084 | 82.211 | 32.12 |
| 43 | TiBu | 500 | 160 | 27 | 0.627 | 0.23 | 0.2 | 0.08 | 67.916 | 36.497 |
| 44 | TiBu | 450 | 160 | 27 | 0.553 | 0.218 | 0.114 | 0.096 | 79.126 | 43.741 |
| 45 | TiBu | 500 | 160 | 21 | 0.638 | 0.223 | 0.269 | 0.114 | 57.383 | 28.49 |
| 46 | TiBu | 450 | 160 | 21 | 0.75 | 0.223 | 0.21 | 0.116 | 71.785 | 28.866 |
| 47 | TiBu | 500 | 160 | 15 | 0.893 | 0.259 | 0.26 | 0.07 | 70.69 | 30.967 |
| 48 | TiBu | 450 | 160 | 15 | 0.719 | 0.227 | 0.356 | 0.085 | 50.145 | 27.049 |
| 49 | TiBu | 500 | 160 | 9 | 0.628 | 0.152 | 0.38 | 0.07 | 39.174 | 19.107 |
| 50 | TiBu | 450 | 160 | 9 | 0.861 | 0.208 | 0.129 | 0.085 | 84.988 | 27.159 |
| 51 | TiBu | 500 | 220 | 3 | 0.612 | 0.24 | 0.157 | 0.078 | 74.121 | 40.414 |
| 52 | TiBu | 450 | 220 | 3 | 7.479 | 0.837 | 2.977 | 0.407 | 59.886 | 9.396 |
| 53 | TiBu | 500 | 200 | 3 | 0.324 | 0.149 | 0.201 | 0.057 | 37.471 | 34.453 |
| 54 | TiBu | 450 | 200 | 3 | 1.047 | 0.329 | 0.283 | 0.113 | 72.845 | 30.969 |
| 55 | TiBu | 500 | 180 | 3 | 1.379 | 0.333 | 0.232 | 0.104 | 83.12 | 26.738 |
| 56 | TiBu | 450 | 180 | 3 | 0.695 | 0.263 | 0.201 | 0.119 | 70.926 | 34.851 |
| 57 | TiBu | 500 | 140 | 3 | 7.954 | 0.711 | 1.781 | 0.255 | 77.505 | 9.826 |
| 58 | TiBu | 450 | 140 | 3 | 0.476 | 0.232 | 0.115 | 0.107 | 75.654 | 45.712 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Activ™ Glass | / | / | / | 0.606 | 0.172 | 0.232 | 0.162 | 61.098 | 19.934 |

Table S2: *Optical Properties for coatings with different synthetic parameters.*

| **Variables** | **Sample No.** | **Direct Bandgap (eV)** | **Indirect Bandgap (eV)** | **Visible Light (380-750) Transmittance** | **Average Thickness (nm)** |
|---|---|---|---|---|---|
| Temperature | 001_TTIP_350_160_3 | 3.83 | 3.34 | 79.03 | 165.63 |
| | 002_TTIP_400_160_3 | 3.82 | 3.42 | 68.60 | 416.84 |
| | 003_TTIP_450_160_3 | 3.64 | 3.05 | 56.01 | / |
| | 004_TTIP_500_160_3 | 3.44 | 3.11 | 61.43 | / |
| | 005_TTIP_550_160_3 | 3.35 | 2.97 | 30.17 | / |
| | 006_TTIP_600_160_3 | 3.23 | 2.88 | 18.51 | / |
| | 007_TiEt_350_160_3 | 3.74 | 3.32 | 82.32 | 293.45 |
| | 008_TiEt_400_160_3 | 3.80 | 3.29 | 80.24 | 339.66 |
| | 009_TiEt_450_160_3 | 3.83 | 3.36 | 81.48 | 162.13 |
| | 010_TiEt_500_160_3 | 3.62 | 3.27 | 71.09 | 711.78 |
| | 011_TiEt_550_160_3 | 3.69 | 2.97 | 72.16 | 756.75 |
| | 012_TiEt_600_160_3 | 3.42 | 3.10 | 47.46 | 1744.02 |
| | 013_TiBu_350_160_3 | / | / | 92.57 | / |
| | 014_TiBu_400_160_3 | 3.84 | 3.29 | 81.59 | 157.40 |
| | 015_TiBu_450_160_3 | 3.80 | 3.28 | 84.75 | 212.47 |
| | 016_TiBu_500_160_3 | 3.54 | 3.19 | 73.31 | 943.57 |
| | 017_TiBu_550_160_3 | 3.46 | 3.08 | 62.45 | 1042.06 |
| | 018_TiBu_600_160_3 | 3.37 | 2.95 | 15.08 | / |
| Preheating Temperature | 033_TTIP_450_100_3 | 3.6818 | 3.03 | 67.97 | 857.73 |
| | 031_TTIP_450_120_3 | 3.7292 | 3.04 | 63.43 | / |
| | 029_TTIP_450_140_3 | 3.7604 | 3.07 | 67.57 | 776.97 |
| | 003_TTIP_450_160_3 | 3.6359 | 3.05 | 56.01 | / |
| | 027_TTIP_450_180_3 | 3.5959 | 2.98 | 74.50 | / |
| | 034_TTIP_500_100_3 | 3.5455 | 3.17 | 69.84 | / |
| | 032_TTIP_500_120_3 | / | 2.98 | 22.10 | / |
| | 030_TTIP_500_140_3 | 3.5040 | 3.15 | 72.55 | / |
| | 004_TTIP_500_160_3 | 3.4373 | 3.11 | 61.43 | / |

| | | | | | |
|---|---|---|---|---|---|
| | 028_TTIP_500_180_3 | 3.4447 | 3.14 | 71.70 | / |
| | 042_TiEt_500_120_3 | 3.8433 | 3.22 | 84.06 | 191.59 |
| | 041_TiEt_500_140_3 | 3.7367 | 3.27 | 77.15 | 408.20 |
| | 010_TiEt_500_160_3 | 3.6196 | 3.27 | 71.09 | 711.78 |
| | 040_TiEt_500_180_3 | 3.6789 | 3.32 | 83.41 | 570.92 |
| | 039_TiEt_500_200_3 | 3.6640 | 3.14 | 74.81 | 794.75 |
| | 058_TiBu_450_140_3 | 3.6418 | 3.23 | 79.33 | 561.75 |
| | 015_TiBu_450_160_3 | 3.8004 | 3.28 | 84.75 | 212.47 |
| | 056_TiBu_450_180_3 | 3.7752 | 3.31 | 84.53 | / |
| | 054_TiBu_450_200_3 | 3.3884 | 3.12 | 77.87 | / |
| | 052_TiBu_450_220_3 | 3.4625 | 3.21 | 81.96 | 268.47 |
| | 057_TiBu_500_140_3 | 3.5810 | 3.23 | 69.99 | 1317.50 |
| | 016_TiBu_500_160_3 | 3.5396 | 3.19 | 73.31 | 943.57 |
| | 055_TiBu_500_180_3 | 3.6344 | 2.92 | 52.33 | / |
| | 053_TiBu_500_200_3 | 3.7218 | 3.41 | 81.50 | 513.35 |
| | 051_TiBu_500_220_3 | 3.4432 | 3.12 | 59.74 | / |
| Synthesis Time | 025_TTIP_450_160_1 | 3.8241 | 3.16 | 74.72 | 488.13 |
| | 021_TTIP_450_160_2 | 3.5129 | 3.08 | 65.43 | 1431.94 |
| | 003_TTIP_450_160_3 | 3.6359 | 3.05 | 56.01 | / |
| | 023_TTIP_450_160_4 | 3.3988 | 3.07 | 57.54 | 1973.66 |
| | 019_TTIP_450_160_5 | 3.4106 | 3.14 | 60.20 | / |
| | 026_TTIP_500_160_1 | 3.8611 | 3.45 | 81.51 | 411.47 |
| | 022_TTIP_500_160_2 | 3.5588 | 3.26 | 75.89 | 1492.11 |
| | 004_TTIP_500_160_3 | 3.4373 | 3.11 | 61.43 | / |
| | 024_TTIP_500_160_4 | 3.3810 | 3.14 | 54.76 | / |
| | 020_TTIP_500_160_5 | 3.3528 | 3.07 | 19.30 | / |
| | 010_TiEt_500_160_3 | 3.6196 | 3.27 | 71.09 | 711.78 |
| | 038_TiEt_500_160_6 | 3.6803 | 3.11 | 64.73 | 816.42 |
| | 037_TiEt_500_160_9 | 3.7678 | 3.12 | 71.91 | 718.10 |
| | 036_TiEt_500_160_12 | 3.5825 | 3.25 | 73.92 | 992.82 |
| | 035_TiEt_500_160_15 | 3.5084 | 3.21 | 72.54 | 1079.47 |
| | 015_TiBu_450_160_3 | 3.8004 | 3.28 | 84.75 | 212.47 |

| | | | | |
|---|---|---|---|---|
| 050_TiBu_450_160_9 | 3.8626 | 3.36 | 84.79 | 199.59 |
| 048_TiBu_450_160_15 | 3.4566 | 3.02 | 41.98 | / |
| 046_TiBu_450_160_21 | 3.8330 | 3.29 | 75.66 | 342.74 |
| 044_TiBu_450_160_27 | 3.6729 | 3.27 | 72.88 | 633.40 |
| 016_TiBu_500_160_3 | 3.5396 | 3.19 | 73.31 | 943.57 |
| 049_TiBu_500_160_9 | 3.6285 | 3.29 | 76.00 | 798.74 |
| 047_TiBu_500_160_15 | 3.3943 | 3.10 | 72.33 | 703.47 |
| 045_TiBu_500_160_21 | 3.5929 | 3.28 | 65.09 | 941.65 |
| 043_TiBu_500_160_27 | 3.6833 | 3.28 | 63.44 | 453.29 |

Table S3: *Photogenerated charge carrier kinetics for different synthetic parameters.*

| **Sample No.** | **Precursor Selection** | **Synthesis Temperature (°C)** | **Precursor Preheating Temperature(°C)** | **Synthesis Time (min)** | **Population @ 10 $\mu$s(m$\triangle$O.D.)** | **$t_{50\%}$ from 10$\mu$s (ms)** |
|---|---|---|---|---|---|---|
| 1 | TTIP | 350 | 160 | 3 | 0.0032 | 0.142 |
| 2 | TTIP | 400 | 160 | 3 | 0.0304 | 0.102 |
| 3 | TTIP | 450 | 160 | 3 | 0.2364 | 0.309 |
| 4 | TTIP | 500 | 160 | 3 | 0.0779 | 0.0424 |
| 5 | TTIP | 550 | 160 | 3 | 3.2088 | 5.86 |
| 6 | TTIP | 600 | 160 | 3 | 2.9154 | 14.1 |
| 7 | TiEt | 350 | 160 | 3 | 0.1024 | 0.225 |
| 8 | TiEt | 400 | 160 | 3 | 0.0652 | 1.64 |
| 9 | TiEt | 450 | 160 | 3 | 0.0128 | 0.148 |
| 10 | TiEt | 500 | 160 | 3 | 0.1524 | 0.234 |
| 11 | TiEt | 550 | 160 | 3 | 0.1018 | 0.672 |
| 12 | TiEt | 600 | 160 | 3 | 3.423 | 0.0204 |
| 13 | TiBu | 350 | 160 | 3 | 0.01 | 0.0236 |
| 14 | TiBu | 400 | 160 | 3 | 0.0159 | 0.14 |
| 15 | TiBu | 450 | 160 | 3 | -0.0011 | / |
| 16 | TiBu | 500 | 160 | 3 | 0.1871 | 0.256 |
| 17 | TiBu | 550 | 160 | 3 | 0.113 | 2.02 |
| 18 | TiBu | 600 | 160 | 3 | 0.0443 | 0.626 |
| 19 | TTIP | 450 | 160 | 5 | 0.507 | 0.115 |
| 20 | TTIP | 500 | 160 | 5 | 0.7328 | 0.262 |
| 21 | TTIP | 450 | 160 | 2 | 0.1794 | 0.16 |
| 22 | TTIP | 500 | 160 | 2 | 0.1031 | 0.044 |
| 23 | TTIP | 450 | 160 | 4 | 0.2203 | 0.112 |
| 24 | TTIP | 500 | 160 | 4 | 0.0359 | 0.157 |
| 25 | TTIP | 450 | 160 | 1 | 0.076 | 0.11 |
| 26 | TTIP | 500 | 160 | 1 | 0.0121 | 0.0688 |
| 27 | TTIP | 450 | 180 | 3 | 0.2166 | 0.134 |
| 28 | TTIP | 500 | 180 | 3 | 0.0476 | 0.0308 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 29 | TTIP | 450 | 140 | 3 | 0.2042 | 0.201 |
| 30 | TTIP | 500 | 140 | 3 | 0.1936 | 0.134 |
| 31 | TTIP | 450 | 120 | 3 | 0.1513 | 0.126 |
| 32 | TTIP | 500 | 120 | 3 | 0.0929 | 0.12 |
| 33 | TTIP | 450 | 100 | 3 | 0.0814 | 0.0672 |
| 34 | TTIP | 500 | 100 | 3 | 0.1551 | 0.112 |
| 35 | TiEt | 500 | 160 | 15 | 0.4213 | 0.438 |
| 36 | TiEt | 500 | 160 | 12 | 0.6032 | 0.419 |
| 37 | TiEt | 500 | 160 | 9 | 0.5099 | 0.3 |
| 38 | TiEt | 500 | 160 | 6 | 0.384 | 0.224 |
| 39 | TiEt | 500 | 200 | 3 | 0.186 | 0.153 |
| 40 | TiEt | 500 | 180 | 3 | 0.1666 | 0.24 |
| 41 | TiEt | 500 | 140 | 3 | 0.0471 | 0.137 |
| 42 | TiEt | 500 | 120 | 3 | 0.0096 | 23.9 |
| 43 | TiBu | 500 | 160 | 27 | 0.4595 | 0.654 |
| 44 | TiBu | 450 | 160 | 27 | 0.2153 | 0.328 |
| 45 | TiBu | 500 | 160 | 21 | 0.423 | 0.836 |
| 46 | TiBu | 450 | 160 | 21 | 0.2545 | 0.448 |
| 47 | TiBu | 500 | 160 | 15 | 0.5633 | 0.836 |
| 48 | TiBu | 450 | 160 | 15 | 0.1633 | 0.328 |
| 49 | TiBu | 500 | 160 | 9 | 0.3307 | 0.32 |
| 50 | TiBu | 450 | 160 | 9 | 0.1196 | 0.245 |
| 51 | TiBu | 500 | 220 | 3 | 0.2671 | 0.234 |
| 52 | TiBu | 450 | 220 | 3 | 0.1046 | 71.1 |
| 53 | TiBu | 500 | 200 | 3 | 0.2784 | 0.201 |
| 54 | TiBu | 450 | 200 | 3 | 0.1016 | 0.419 |
| 55 | TiBu | 500 | 180 | 3 | 0.1488 | 0.096 |
| 56 | TiBu | 450 | 180 | 3 | 0.0863 | 0.205 |
| 57 | TiBu | 500 | 140 | 3 | 0.1096 | 0.6 |
| 58 | TiBu | 450 | 140 | 3 | 0.0655 | 0.1 |
| Activ™ Glass | / | / | / | / | 0.0085 | 0.205 |

Table S4: *Calculated total $\xi NO$ and $\xi NO_x$and $\xi NO_2$ and quantities of $NO_2$ selectivity and De-$NO_x$index ($\xi DeNO_x$)*

| **Sample NO.** | $\xi$**DeNO**$_x^*$ | $\xi$**NO/ppm** | $\xi$**NO**$_x$**/ppm** | $\xi$**NO$_2$/ppm** | **NO$_2$ selectivity/%** |
|---|---|---|---|---|---|
| 1 | -0.065 | 0.078 | 0.031 | 0.048 | 61.098 |
| 2 | -0.060 | 0.184 | 0.105 | 0.082 | 44.226 |
| 3 | -0.063 | 0.039 | 0.005 | 0.034 | 87.174 |
| 4 | -0.342 | 0.253 | 0.055 | 0.198 | 78.405 |
| 5 | -2.051 | 1.486 | 0.307 | 1.179 | 79.345 |
| 6 | -2.598 | 1.945 | 0.430 | 1.514 | 77.870 |
| 7 | 0.014 | 0.202 | 0.139 | 0.063 | 31.037 |
| 8 | -0.073 | 0.099 | 0.042 | 0.057 | 57.894 |
| 9 | -0.099 | 0.112 | 0.042 | 0.070 | 62.733 |
| 10 | -0.082 | 0.081 | 0.027 | 0.054 | 66.932 |
| 11 | -0.416 | 0.472 | 0.176 | 0.296 | 62.689 |
| 12 | -3.240 | 2.932 | 0.875 | 2.058 | 70.171 |
| 13 | 0.019 | 0.064 | 0.049 | 0.015 | 23.255 |
| 14 | 0.024 | 0.033 | 0.030 | 0.003 | 9.184 |
| 15 | -0.053 | 0.085 | 0.002 | 0.046 | 54.303 |
| 16 | -0.962 | 0.704 | 0.148 | 0.555 | 78.910 |
| 17 | -1.777 | 1.409 | 0.347 | 1.062 | 75.373 |
| 18 | -2.039 | 1.648 | 0.419 | 1.229 | 74.556 |
| 19 | 0.008 | 0.046 | 0.034 | 0.013 | 27.658 |
| 20 | -1.194 | 1.000 | 0.268 | 0.731 | 73.150 |
| 21 | 0.014 | 0.026 | 0.022 | 0.004 | 16.138 |
| 22 | -0.014 | 0.019 | 0.008 | 0.011 | 58.009 |
| 23 | -0.026 | 0.013 | 0.000 | 0.013 | 98.062 |
| 24 | 0.030 | 0.047 | 0.053 | 0.006 | 12.307 |
| 25 | -0.001 | 0.003 | 0.002 | 0.002 | 48.366 |
| 26 | 0.001 | 0.010 | 0.007 | 0.003 | 31.272 |
| 27 | -0.179 | 0.242 | 0.102 | 0.140 | 57.982 |
| 28 | -0.023 | 0.040 | 0.019 | 0.021 | 52.317 |
| 29 | -0.007 | 0.074 | 0.047 | 0.027 | 36.589 |
| 30 | 0.010 | 0.020 | 0.023 | 0.003 | 16.990 |
| 31 | -0.036 | 0.077 | 0.039 | 0.038 | 49.134 |
| 32 | -1.329 | 0.731 | 0.044 | 0.687 | 93.948 |
| 33 | -0.429 | 0.303 | 0.059 | 0.244 | 80.485 |
| 34 | -0.363 | 0.288 | 0.071 | 0.217 | 75.342 |
| 35 | -0.069 | 0.064 | 0.020 | 0.044 | 69.208 |
| 36 | -0.029 | 0.043 | 0.019 | 0.024 | 56.045 |
| 37 | 0.004 | 0.006 | 0.006 | 0.001 | 9.742 |
| 38 | 0.011 | 0.019 | 0.016 | 0.003 | 14.690 |
| 39 | -0.111 | 0.074 | 0.012 | 0.062 | 83.225 |
| 40 | -0.036 | 0.030 | 0.008 | 0.022 | 72.777 |
| 41 | -0.186 | 0.124 | 0.020 | 0.103 | 83.512 |
| 42 | -0.247 | 0.168 | 0.030 | 0.138 | 82.211 |
| 43 | -0.085 | 0.082 | 0.026 | 0.056 | 67.916 |

| | | | | | |
|---|---|---|---|---|---|
| 44 | -0.099 | 0.072 | 0.015 | 0.057 | 79.126 |
| 45 | -0.060 | 0.083 | 0.035 | 0.048 | 57.383 |
| 46 | -0.113 | 0.098 | 0.028 | 0.070 | 71.785 |
| 47 | -0.130 | 0.116 | 0.034 | 0.082 | 70.690 |
| 48 | -0.047 | 0.093 | 0.047 | 0.047 | 50.145 |
| 49 | -0.014 | 0.082 | 0.050 | 0.032 | 39.174 |
| 50 | -0.174 | 0.112 | 0.017 | 0.095 | 84.988 |
| 51 | -0.097 | 0.079 | 0.021 | 0.059 | 74.121 |
| 52 | -0.769 | 0.966 | 0.387 | 0.578 | 59.886 |
| 53 | -0.005 | 0.042 | 0.026 | 0.016 | 37.471 |
| 54 | -0.160 | 0.135 | 0.037 | 0.099 | 72.845 |
| 55 | -0.267 | 0.179 | 0.030 | 0.149 | 83.120 |
| 56 | -0.101 | 0.090 | 0.026 | 0.064 | 70.926 |
| 57 | -1.367 | 1.032 | 0.232 | 0.800 | 77.505 |
| 58 | -0.078 | 0.062 | 0.015 | 0.047 | 75.654 |

*$\xi\text{DeNO}_\text{x} = \xi\text{NO} - 3 \times \xi\text{NO}_2$

$\xi$DeNO$_\text{x}$ is defined by assigning toxicity values to NO and $NO_2$ separately, and expressing changes in total toxicity rather than changes in the concentration of nitrogen oxides alone.[1] The calculation is determined by following **Equation B**:

$$\xi_{[x]} = \frac{(c_d - c_i) \cdot V \cdot p}{A \cdot R \cdot T \cdot \Phi}$$

In which $c_d$ is dark concentration of gas [X] in ppm, $c_i$ is light concentration of gas [X] in ppm. $V$,$p$,$A$,$R$ and $T$ represents the volumeric flowrate, pressure, irradiated area, the gas constant, and absolute temperature, respectively. $\Phi$, indicates the photon flux illuminate on the sample surface.

A negative $\xi$DeNO$_\text{x}$ leads a increasing toxicity in photocatalytic oxidation of NO, and *vice versa*.

Table S5: *data set Used for ML Modelling.*

| TTIP | TiBu | TiEt | Synthesis Temperature(°C)) | Bubbler Temperature(°C) | Synthesis Time(mins) | Crystal Size(nm) | 101 Phase | 200 Phase | 211 Phase | 220 Phase | Bandgap(eV) | VLT(%) | Rq(nm) | SSA($\mu m^2$) | Charpopm$\triangle$O.D. | Charlife(ms) | NO(%) | Nox(%) | NO2(%) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 350 | 160 | 3 | 31.5176 | 1.0954 | 1.2293 | 0.6754 | 0 | 3.338 | 79.034 | 10.13 | 25.39 | 0.0032 | 0.142 | 0.6064 | 0.2324 | 61.1 |
| 1 | 0 | 0 | 400 | 160 | 3 | 28.4896 | 0.9099 | 1.2376 | 0.8524 | 0 | 3.4165 | 68.5988 | 24.53 | 26.78 | 0.0304 | 0.102 | 1.4267 | 0.7979 | 44.23 |
| 1 | 0 | 0 | 450 | 160 | 3 | 28.4128 | 0.5363 | 1.02 | 1.2892 | 1.1544 | 3.0535 | 56.0072 | 24.59 | 26.77 | 0.2364 | 0.309 | 0.3019 | 0.038 | 87.17 |
| 1 | 0 | 0 | 500 | 160 | 3 | 26.4843 | 0 | 1 | 0 | 0 | 3.1127 | 61.428 | 29.12 | 26.88 | 0.0779 | 0.0424 | 1.941 | 0.4175 | 78.4 |
| 1 | 0 | 0 | 550 | 160 | 3 | 27.5976 | 0.5517 | 0.4619 | 0.7088 | 2.2776 | 2.9749 | 30.1676 | 84.52 | 31.72 | 3.2088 | 5.86 | 11.4302 | 2.3422 | 79.34 |
| 1 | 0 | 0 | 600 | 160 | 3 | 31.3955 | 0.5449 | 0.32 | 0.6311 | 2.504 | 2.8771 | 18.5055 | 98.39 | 32.42 | 2.9154 | 14.1 | 14.9768 | 3.2779 | 77.87 |
| 0 | 0 | 1 | 350 | 160 | 3 | 23.832 | 0.62 | 1.877 | 0.6342 | 0.8688 | 3.3158 | 82.3167 | 12.49 | 25.3 | 0.1024 | 0.225 | 1.5676 | 1.0558 | 31.04 |
| 0 | 0 | 1 | 400 | 160 | 3 | 21.4866 | 0.9054 | 0.7048 | 1.3575 | 1.0323 | 3.2921 | 80.2398 | 7.336 | 25.29 | 0.0652 | 1.64 | 0.7726 | 0.3188 | 57.89 |
| 0 | 0 | 1 | 450 | 160 | 3 | 17.9681 | 1.0855 | 0.947 | 0.9676 | 0 | 3.3587 | 81.4821 | 8.091 | 25.23 | 0.0128 | 0.148 | 0.8647 | 0.3177 | 62.73 |
| 0 | 0 | 1 | 500 | 160 | 3 | 24.1769 | 0.0952 | 1.5075 | 0.6278 | 1.7696 | 3.2669 | 71.0893 | 17.03 | 25.53 | 0.1524 | 0.234 | 0.6298 | 0.2057 | 66.93 |
| 0 | 0 | 1 | 550 | 160 | 3 | 24.422 | 0.6507 | 2.0701 | 0.7569 | 0.5223 | 2.9749 | 72.1592 | 20.26 | 25.63 | 0.1018 | 0.672 | 3.6499 | 1.3343 | 62.69 |
| 0 | 0 | 1 | 600 | 160 | 3 | 30.4073 | 0.1622 | 1.6164 | 0.7946 | 1.4268 | 3.0965 | 47.459 | 35.58 | 28.9 | 3.423 | 0.0204 | 22.6746 | 6.6269 | 70.17 |
| 0 | 1 | 0 | 400 | 160 | 3 | 18.3004 | 1.0954 | 0.8864 | 1.0182 | 0 | 3.2921 | 81.5864 | 5.588 | 26 | 0.0159 | 0.14 | 0.2555 | 0.2291 | 9.18 |
| 0 | 1 | 0 | 500 | 160 | 3 | 22.9459 | 0.1067 | 0.3078 | 0.8731 | 2.7124 | 3.1898 | 73.3079 | 12.37 | 26.04 | 0.1871 | 0.256 | 5.4158 | 1.1325 | 78.91 |
| 0 | 1 | 0 | 550 | 160 | 3 | 26.9996 | 0.5231 | 1.2943 | 0.7705 | 1.4122 | 3.0772 | 62.4537 | 26.64 | 28.25 | 0.113 | 2.02 | 10.8541 | 2.6529 | 75.37 |
| 0 | 1 | 0 | 600 | 160 | 3 | 31.9029 | 0.4412 | 0.3774 | 0.9473 | 2.2342 | 2.9527 | 15.0838 | 64.54 | 31.16 | 0.0443 | 0.626 | 12.6852 | 3.2027 | 74.56 |
| 1 | 0 | 0 | 450 | 160 | 5 | 27.5518 | 0.1878 | 0.6691 | 0.9269 | 2.2162 | 3.1409 | 60.1976 | 26.24 | 27.03 | 0.507 | 0.115 | 0.3591 | 0.2577 | 27.66 |
| 1 | 0 | 0 | 500 | 160 | 5 | 28.8829 | 0.0185 | 0.1772 | 0.1204 | 3.684 | 3.0742 | 19.3009 | 143.6 | 35.62 | 0.7328 | 0.262 | 7.7195 | 2.0686 | 73.15 |
| 1 | 0 | 0 | 450 | 160 | 2 | 26.6123 | 0.1045 | 0.4902 | 0.9519 | 2.4534 | 3.0786 | 65.4312 | 22.42 | 26.71 | 0.1794 | 0.16 | 0.2046 | 0.1707 | 16.14 |
| 1 | 0 | 0 | 500 | 160 | 2 | 25.9187 | 0.2595 | 2.6342 | 0.1064 | 0 | 3.2565 | 75.8937 | 16.12 | 27.36 | 0.1031 | 0.044 | 0.1452 | 0.0607 | 58.01 |
| 1 | 0 | 0 | 450 | 160 | 4 | 25.5019 | 0.1876 | 0.192 | 1.227 | 2.3933 | 3.0668 | 57.5399 | 29.5 | 27.12 | 0.2203 | 0.112 | 0.103 | 0.0021 | 98.06 |
| 1 | 0 | 0 | 500 | 160 | 4 | 25.754 | 0.0194 | 0.7191 | 0.3398 | 2.9217 | 3.1394 | 54.759 | 27.73 | 26.73 | 0.0359 | 0.157 | 0.3625 | 0.405 | 12.31 |
| 1 | 0 | 0 | 450 | 160 | 1 | 26.8252 | 0.8883 | 1.6277 | 0.484 | 0 | 3.1587 | 74.7212 | 14.26 | 26.15 | 0.076 | 0.11 | 0.025 | 0.0129 | 48.37 |
| 1 | 0 | 0 | 500 | 160 | 1 | 19.3002 | 1.2031 | 1.2284 | 0.5685 | 0 | 3.4491 | 81.5103 | 11.92 | 25.81 | 0.0121 | 0.0688 | 0.0741 | 0.0506 | 31.27 |
| 1 | 0 | 0 | 450 | 180 | 3 | 28.7711 | 0.2005 | 0.6312 | 1.1242 | 2.0441 | 2.9808 | 74.5017 | 16.48 | 26.27 | 0.2166 | 0.134 | 1.8842 | 0.7811 | 57.98 |
| 1 | 0 | 0 | 500 | 180 | 3 | 19.7342 | 0.0171 | 2.7143 | 0.3 | 0.9686 | 3.1424 | 71.7007 | 19.71 | 26.57 | 0.0476 | 0.0308 | 0.3081 | 0.1459 | 52.32 |
| 1 | 0 | 0 | 450 | 140 | 3 | 22.5945 | 0.667 | 2.2503 | 0.7252 | 0.3575 | 3.0698 | 67.5738 | 20.34 | 26.41 | 0.2042 | 0.201 | 0.5711 | 0.3592 | 36.59 |
| 1 | 0 | 0 | 500 | 140 | 3 | 25.3076 | 0.0326 | 2.9084 | 0.059 | 0 | 3.1498 | 72.5488 | 21.08 | 26.13 | 0.1936 | 0.134 | 0.1545 | 0.1797 | 16.99 |
| 1 | 0 | 0 | 450 | 120 | 3 | 28.4577 | 0.371 | 1.5952 | 1.1149 | 0.9189 | 3.0386 | 63.4309 | 16.72 | 26.09 | 0.1513 | 0.126 | 0.5948 | 0.3002 | 49.13 |
| 1 | 0 | 0 | 500 | 120 | 3 | 31.2943 | 0.3657 | 2.6702 | 0.764 | 0.2001 | 2.9833 | 22.1049 | 19.43 | 25.93 | 0.0929 | 0.12 | 0.0149 | 0.0087 | 161.47 |
| 1 | 0 | 0 | 450 | 100 | 3 | 29.254 | 0.4497 | 1.565 | 1.1327 | 0.8527 | 3.0342 | 67.9671 | 18.17 | 25.96 | 0.0814 | 0.0672 | 2.3547 | 0.456 | 80.49 |
| 1 | 0 | 0 | 500 | 100 | 3 | 27.9019 | 0.076 | 2.5862 | 0.5688 | 0.769 | 3.1676 | 69.8413 | 23.5 | 28.37 | 0.1551 | 0.112 | 2.2372 | 0.5478 | 75.34 |
| 0 | 0 | 1 | 500 | 160 | 15 | 20.1386 | 0.0289 | 0.034 | 0.7484 | 3.1887 | 3.2091 | 72.5424 | 15.36 | 26.31 | 0.4213 | 0.438 | 0.4937 | 0.1507 | 69.21 |
| 0 | 0 | 1 | 500 | 160 | 12 | 22.2277 | 0.0591 | 0.0535 | 0.5724 | 3.315 | 3.2491 | 73.9234 | 13.01 | 25.89 | 0.6032 | 0.419 | 0.3313 | 0.1448 | 56.04 |
| 0 | 0 | 1 | 500 | 160 | 9 | 24.0789 | 0.5734 | 1.3042 | 1.2132 | 0.9092 | 3.1187 | 71.9104 | 16.71 | 26.07 | 0.5099 | 0.3 | 0.0452 | 0.0488 | 9.74 |
| 0 | 0 | 1 | 500 | 160 | 6 | 23.1755 | 0.2009 | 0.1468 | 1.3217 | 2.3306 | 3.1098 | 64.7298 | 20.2 | 26.36 | 0.384 | 0.224 | 0.1461 | 0.1243 | 14.69 |
| 0 | 0 | 1 | 500 | 200 | 3 | 22.2821 | 0.2442 | 0.3689 | 1.2781 | 2.1087 | 3.1424 | 74.8134 | 15.27 | 26.02 | 0.186 | 0.153 | 0.5722 | 0.0954 | 83.23 |
| 0 | 0 | 1 | 500 | 180 | 3 | 18.3855 | 0.3489 | 0.6844 | 0.6976 | 2.2692 | 3.3232 | 83.4086 | 7.943 | 25.67 | 0.1666 | 0.24 | 0.2331 | 0.0637 | 72.78 |
| 0 | 0 | 1 | 500 | 140 | 3 | 19.2376 | 0.3405 | 1.273 | 1.3865 | 0 | 3.2728 | 77.1512 | 11.69 | 25.69 | 0.0471 | 0.137 | 0.9611 | 0.1569 | 83.51 |
| 0 | 0 | 1 | 500 | 120 | 3 | 20.8493 | 0.6168 | 1.2106 | 1.1726 | 0 | 3.2224 | 84.0608 | 11.53 | 25.59 | 0.0096 | 23.9 | 1.3068 | 0.2305 | 82.21 |
| 0 | 1 | 0 | 500 | 160 | 27 | 22.9315 | 0.1098 | 1.4594 | 0.6535 | 1.7773 | 3.2787 | 63.441 | 38.68 | 27.24 | 0.4595 | 0.654 | 0.627 | 0.2004 | 67.92 |
| 0 | 1 | 0 | 450 | 160 | 27 | 21.8744 | 0.2431 | 0.6352 | 1.0958 | 2.0259 | 3.2669 | 72.8794 | 15.17 | 26.04 | 0.2153 | 0.328 | 0.5532 | 0.1139 | 79.13 |
| 0 | 1 | 0 | 500 | 160 | 21 | 27.1974 | 0.022 | 1.6468 | 0.2229 | 2.1083 | 3.2846 | 65.0884 | 12.71 | 25.81 | 0.423 | 0.836 | 0.6376 | 0.2686 | 57.38 |
| 0 | 1 | 0 | 450 | 160 | 21 | 21.0895 | 0.3181 | 2.4427 | 0.5974 | 0.6418 | 3.2891 | 75.6596 | 8.786 | 25.66 | 0.2545 | 0.448 | 0.7496 | 0.2101 | 71.79 |
| 0 | 1 | 0 | 500 | 160 | 15 | 29.8427 | 0.0892 | 2.1167 | 0.5487 | 1.2453 | 3.095 | 72.3259 | 14.4 | 25.71 | 0.5633 | 0.836 | 0.8933 | 0.2595 | 70.69 |
| 0 | 1 | 0 | 450 | 160 | 15 | 20.9057 | 0.0485 | 0.0651 | 0.8797 | 3.0067 | 3.0179 | 41.9822 | 23.52 | 26.61 | 0.1633 | 0.328 | 0.719 | 0.3558 | 50.15 |
| 0 | 1 | 0 | 500 | 160 | 9 | 28.5631 | 0.0612 | 0.8521 | 0.683 | 2.4037 | 3.2861 | 76.0024 | 10.57 | 25.55 | 0.3307 | 0.32 | 0.6284 | 0.38 | 39.17 |
| 0 | 1 | 0 | 450 | 160 | 9 | 22.1958 | 0.5329 | 1.7162 | 0.751 | 0 | 3.3617 | 84.7942 | 10.29 | 25.72 | 0.1196 | 0.245 | 0.8614 | 0.129 | 84.99 |
| 0 | 1 | 0 | 500 | 220 | 3 | 25.0382 | 0.0251 | 0.2198 | 0.2983 | 3.4567 | 3.1201 | 59.7387 | 35.46 | 27.44 | 0.2671 | 0.234 | 0.6116 | 0.1569 | 74.12 |
| 0 | 1 | 0 | 450 | 220 | 3 | 14.1143 | 1.2686 | 0.7314 | 0 | 0 | 3.2076 | 81.9641 | 0.618 | 25.01 | 0.1046 | 71.1 | 7.4785 | 2.9767 | 59.89 |
| 0 | 1 | 0 | 500 | 200 | 3 | 22.1687 | 0.0893 | 0.1003 | 0.9689 | 2.8416 | 3.4121 | 81.4978 | 9.198 | 25.95 | 0.2784 | 0.201 | 0.3243 | 0.2015 | 37.47 |
| 0 | 1 | 0 | 450 | 200 | 3 | 23.3004 | 0.1187 | 0.0997 | 0.5827 | 3.1989 | 3.1187 | 77.8665 | 14.09 | 26.14 | 0.1016 | 0.419 | 1.0473 | 0.2826 | 72.84 |
| 0 | 1 | 0 | 500 | 180 | 3 | 24.177 | 0.2401 | 0.3358 | 1.5283 | 1.8959 | 2.9156 | 52.3292 | 41.2 | 28.53 | 0.1488 | 0.096 | 1.379 | 0.2317 | 83.12 |
| 0 | 1 | 0 | 450 | 180 | 3 | 14.5167 | 0.1906 | 0.0307 | 1.2368 | 2.5419 | 3.3084 | 84.5259 | 3.953 | 25.41 | 0.0863 | 0.205 | 0.6948 | 0.2009 | 70.93 |
| 0 | 1 | 0 | 500 | 140 | 3 | 22.9347 | 0.1701 | 0.691 | 0.8932 | 2.2457 | 3.2313 | 69.9945 | 17.68 | 27.06 | 0.1096 | 0.6 | 7.9543 | 1.7811 | 77.51 |
| 0 | 1 | 0 | 450 | 140 | 3 | 25.9757 | 0.3087 | 0.7112 | 1.4129 | 1.5672 | 3.2328 | 79.3309 | 15.32 | 26.48 | 0.0655 | 0.1 | 0.476 | 0.1154 | 75.65 |

*0 (1) represent precursor was not used (was used)

# C Codes

Listing S1: *Machine-Learning Codes for Data Splitting and Strategy Validation based on Python*

```
#Packages Import
import numpy as np
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.base import clone
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor,
   AdaBoostRegressor
from sklearn.svm import SVR
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
from xgboost import XGBRegressor
import warnings
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import GridSearchCV
from sklearn.cluster import KMeans, AgglomerativeClustering
from sklearn.exceptions import ConvergenceWarning
from sklearn.model_selection import LeaveOneOut

# define cross validation function
def cross_val_mse(model, X, y):
   n_splits = 5
   kfold = KFold(n_splits, shuffle=True, random_state=42)
   mse_scores = cross_val_score(model, X, y, cv=kfold,
      scoring='neg_mean_squared_error')
   return -1 * np.mean(mse_scores)

# Define features_cols, properties_cols and targets_cols

feature_cols = ["TTIP","TiEt","TiBu","Synthesis Temperature",
   "Bubbler Temperature", "Synthesis Time"]
properties_cols = ["Crystal Size", "101 Phase", "200 Phase", "211
   Phase", "220 Phase",
            "Bandgap", "VLT", "Rq", "SSA", "Charpop", "Charlife"]
targets_cols = ["NO","NOx"]
all_target_cols = properties_cols+targets_cols

# Initializing Regressors
regressors = [
   ("Linear Regression", LinearRegression()),
   ("Random Forest Regressor", RandomForestRegressor(n_estimators =
      120, random_state = 42)),
   ("SVR", SVR(kernel="linear",C = 0.2,epsilon = 0.05)),
```

```
    ("Adaboost Regressor", AdaBoostRegressor(n_estimators=100,
        random_state=42)),
    ("XGBoost Regressor", XGBRegressor())
]

best_models = {}

# Data Input
data = pd.read_excel('Master ML Data.xlsx')

# LOOCV Method
loo = LeaveOneOut()
all_data_copies = []
all_data_splits = []

for train_index, test_index in loo.split(data):
    # data splitting
    data_copy, data_split = data.iloc[train_index],
        data.iloc[test_index]
    all_data_copies.append(data_copy)
    all_data_splits.append(data_split)


# Define parameter grids for each regressor
param_grids = {
    "Linear Regression": {
        'fit_intercept': [True, False]
    },
    "Random Forest Regressor": {
        'n_estimators': [50, 100, 200],
        'max_depth': [None, 10, 20, 30],
        'min_samples_split': [2, 5, 10],
        'min_samples_leaf': [1, 2, 4],
    },
    "SVR": {
        'C': [0.1, 1, 10, 100],
        'gamma': [1, 0.1, 0.01, 0.001],
        'kernel': ['rbf', 'sigmoid'],
        "max_iter":[10000]
    },
    "Adaboost Regressor": {
        'n_estimators': [50, 100, 200],
        'learning_rate': [0.001, 0.01, 0.1, 1],
    },
    "XGBoost Regressor": {
        'n_estimators': [50, 100, 200],
        'learning_rate': [0.001, 0.01, 0.1, 1],
        'max_depth': [3, 5, 7, 9],
    },
}
```

```
# Run 80 cycles for strategy I
best_models = {col: None for col in properties_cols + targets_cols}
best_model_params = {col: None for col in properties_cols +
   targets_cols}
best_model_names = {col: "" for col in properties_cols +
   targets_cols}

all_mse_division = {col: [] for col in properties_cols +
   targets_cols}

for i, (data_train, data_test) in enumerate(zip(all_data_copies,
   all_data_splits)):
   print(f"Data copy: {i+1}")
   X_train = data_train[feature_cols]
   X_test = data_test[feature_cols]

   for col in best_models.keys():
      y_train = data_train[col]
      y_test = data_test[col]

      min_mse = np.inf
      for name, model in regressors:
         if name in param_grids:
            model = clone(model)
            grid_search = GridSearchCV(model, param_grids[name],
               scoring='neg_mean_squared_error', cv=5,n_jobs=-1)
            grid_search.fit(X_train, y_train)
            model = grid_search.best_estimator_
            mse_train = -grid_search.best_score_

            if mse_train < min_mse:
               min_mse = mse_train
               best_models[col] = model
               best_model_params[col] = grid_search.best_params_
               best_model_names[col] = name

      best_models[col].fit(X_train, y_train)
      pred_test = best_models[col].predict(X_test)
      mseerror_test = mean_squared_error(y_test, pred_test)
      mseerror_division = (mseerror_test)

      print(f"real {col} value is: ", data_test[col].values, f"pred
         {col} value is: ", pred_test)
      all_mse_division[col].append(mseerror_division)

      print(f"Best model for {col} is {best_model_names[col]} with
         parameters: {best_model_params[col]} with mse error:
         {np.nanmean(mseerror_division)}")
   print("\n")

for col, mseerror_division in all_mse_division.items():
```

```
    print(f"Average mseerror for {col}:
        {np.nanmean(mseerror_division)}")

# Run 80 cycles for strategy II

best_models = {col: None for col in targets_cols}
best_model_params = {col: None for col in targets_cols}
best_model_names = {col: "" for col in targets_cols}

all_mse_division = {col: [] for col in targets_cols}

for i, (data_train, data_test) in enumerate(zip(all_data_copies,
    all_data_splits)):
    print(f"Data copy: {i+1}")
    X_train = data_train[properties_cols]
    X_test = data_test[properties_cols]

    for col in best_models.keys():
        y_train = data_train[col]
        y_test = data_test[col]

        min_mse = np.inf
        for name, model in regressors:

            if name in param_grids:
                model = clone(model)
                grid_search = GridSearchCV(model, param_grids[name],
                    scoring='neg_mean_squared_error', cv=5,n_jobs=-1)
                grid_search.fit(X_train, y_train)
                model = grid_search.best_estimator_
                mse_train = -grid_search.best_score_

                if mse_train < min_mse:
                    min_mse = mse_train
                    best_models[col] = model
                    best_model_params[col] = grid_search.best_params_
                    best_model_names[col] = name

        best_models[col].fit(X_train, y_train)
        pred_test = best_models[col].predict(X_test)
        mseerror_test = mean_squared_error(y_test, pred_test)
        mseerror_division = (mseerror_test)

        print(f"real {col} value is: ", data_test[col].values, f"pred
            {col} value is: ", pred_test)
        all_mse_division[col].append(mseerror_division)

        print(f"Best model for {col} is {best_model_names[col]} with
            mse error: {np.nanmean(mseerror_division)}")
    print("\n")
```

```
for col, mseerror_division in all_mse_division.items():
   print(f"Average mseerror for {col}:
      {np.nanmean(mseerror_division)}")

# Run 80 cycles for strategy III

combined_cols = feature_cols + properties_cols
best_models = {col: None for col in targets_cols}
best_model_params = {col: None for col in targets_cols}
best_model_names = {col: "" for col in targets_cols}

all_mse_division = {col: [] for col in targets_cols}

for i, (data_train, data_test) in enumerate(zip(all_data_copies,
   all_data_splits)):
   print(f"Data copy: {i+1}")
   X_train = data_train[combined_cols]
   X_test = data_test[combined_cols]

   for col in best_models.keys():
      y_train = data_train[col]
      y_test = data_test[col]

      min_mse = np.inf
      for name, model in regressors:

         if name in param_grids:
            model = clone(model)
            grid_search = GridSearchCV(model, param_grids[name],
               scoring='neg_mean_squared_error', cv=5,n_jobs=-1)
            grid_search.fit(X_train, y_train)
            model = grid_search.best_estimator_
            mse_train = -grid_search.best_score_

            if mse_train < min_mse:
               min_mse = mse_train
               best_models[col] = model
               best_model_params[col] = grid_search.best_params_
               best_model_names[col] = name

      best_models[col].fit(X_train, y_train)
      pred_test = best_models[col].predict(X_test)
      mseerror_test = mean_squared_error(y_test, pred_test)
      mseerror_division = (mseerror_test)

      print(f"real {col} value is: ", data_test[col].values, f"pred
         {col} value is: ", pred_test)

      all_mse_division[col].append(mseerror_division)
      print(f"Best model for {col} is {best_model_names[col]} with
         mse error: {np.nanmean(mseerror_division)}")
```

```
    print("\n")

for col, mseerror_division in all_mse_division.items():
    print(f"Average mseerror for {col}:
        {np.nanmean(mseerror_division)}")

# Run 80 cycles for strategy IV

best_grids_prop = {col: None for col in properties_cols}
best_grids_target = {col: None for col in targets_cols}

all_mseerror_divisions_target = {col: [] for col in targets_cols}

for i, (data_train, data_test) in enumerate(zip(all_data_copies,
    all_data_splits)):
    print(f"Data copy: {i+1}")
    X_train = data_train[feature_cols]
    X_test = data_test[feature_cols]

    for col in properties_cols:
        y_train = data_train[col]
        best_score = np.inf
        for name, model in regressors:
            if name in param_grids:
                grid = GridSearchCV(model, param_grids[name],
                    scoring='neg_mean_squared_error', cv=5,n_jobs=-1)
                grid.fit(X_train, y_train)
                if -grid.best_score_ < best_score:
                    best_score = -grid.best_score_
                    best_grids_prop[col] = grid
        print(f"Best model for {col} is
            {best_grids_prop[col].best_estimator_}")

    X_train_properties = data_train[properties_cols]

    for col in targets_cols:
        y_train = data_train[col]
        best_score = np.inf
        for name, model in regressors:
            if name in param_grids:
                grid = GridSearchCV(model, param_grids[name],
                    scoring='neg_mean_squared_error', cv=5,n_jobs=-1)
                grid.fit(X_train_properties, y_train)
                if -grid.best_score_ < best_score:
                    best_score = -grid.best_score_
                    best_grids_target[col] = grid
        print(f"Best model for {col} is
            {best_grids_target[col].best_estimator_}")

    predicted_properties = {col:
        best_grids_prop[col].predict(X_test) for col in
```

```
        properties_cols}
    X_test_pred = np.column_stack(([predicted_properties[col] for
        col in properties_cols]))

    for col in ['NO', 'NOx']:
        pred_test_target = best_grids_target[col].predict(X_test_pred)
        Logerror_test_target = log_error(data_test[col],
            pred_test_target)
        mseerror_test_target = mean_squared_error(data_test[col],
            pred_test_target)
        Logerror_division = (Logerror_test_target)
        mseerror_division = (mseerror_test_target)
        print(f"real {col} value is: ", data_test[col].values, f"pred
            {col} value is: ", pred_test_target)

        all_mseerror_divisions_target[col].append(mseerror_division)
        print(f"MSE error for {col} is:
            {np.nanmean(mseerror_division)}")
    print("\n")

for col, mseerror_divisions in
    all_mseerror_divisions_target.items():
    print(f"Average mseerror for {col}:
        {np.nanmean(mseerror_divisions)}")


# Run 80 cycles for strategy V

best_grids_prop = {col: None for col in properties_cols}
best_grids_target = {col: None for col in targets_cols}
all_mseerror_divisions_target = {col: [] for col in targets_cols}

for i, (data_train, data_test) in enumerate(zip(all_data_copies,
    all_data_splits)):
    print(f"Data copy: {i+1}")
    X_train = data_train[feature_cols]
    X_train_prop = data_train[properties_cols]
    X_test = data_test[feature_cols]

    for col in properties_cols:
        y_train = data_train[col]
        best_score = np.inf
        for name, model in regressors:
            if name in param_grids:
                grid = GridSearchCV(model, param_grids[name],
                    scoring='neg_mean_squared_error', cv=5,n_jobs=-1)
                grid.fit(X_train, y_train)
                if -grid.best_score_ < best_score:
                    best_score = -grid.best_score_
                    best_grids_prop[col] = grid
```

```
            print(f"Best model for {col} is
                {best_grids_prop[col].best_estimator_}")

        predicted_properties = {col:
            best_grids_prop[col].predict(X_train).reshape(-1, 1) for col
            in properties_cols}
        X_taget_fit_all = np.concatenate([X_train, X_train_prop], axis=1)
        X_train_all = np.concatenate([X_train,
            *predicted_properties.values()], axis=1)

        for col in targets_cols:
            y_train = data_train[col]
            best_score = np.inf
            for name, model in regressors:
                if name in param_grids:
                    grid = GridSearchCV(model, param_grids[name],
                        scoring='neg_mean_squared_error', cv=5,n_jobs=-1)
                    grid.fit(X_taget_fit_all, y_train)
                    if -grid.best_score_ < best_score:
                        best_score = -grid.best_score_
                        best_grids_target[col] = grid
            print(f"Best model for {col} is
                {best_grids_target[col].best_estimator_}")

        predicted_properties_test = {col:
            best_grids_prop[col].predict(X_test).reshape(-1, 1) for col
            in properties_cols}
        X_test_pred = np.concatenate([X_test,
            *predicted_properties_test.values()], axis=1)

        for col in ['NO', 'NOx']:
            pred_test_target = best_grids_target[col].predict(X_test_pred)
            Logerror_test_target = log_error(data_test[col],
                pred_test_target)
            mseerror_test_target = mean_squared_error(data_test[col],
                pred_test_target)
            Logerror_division = (Logerror_test_target)
            mseerror_division = (mseerror_test_target)
            print(f"real {col} value is: ", data_test[col].values, f"pred
                {col} value is: ", pred_test_target)

            all_mseerror_divisions_target[col].append(mseerror_division)
            print(f" MSE error for {col} is:
                {np.nanmean(mseerror_division)}")
        print("\n")

    for col, mseerror_divisions in
        all_mseerror_divisions_target.items():
        print(f"Average mseerror for {col}:
            {np.nanmean(mseerror_divisions)}")
```

```
# Run 80 cycles for strategy VI

best_grids_prop = {col: None for col in properties_cols}
best_grids_target = {col: None for col in targets_cols}
all_mseerror_divisions_target = {col: [] for col in targets_cols}

for i, (data_train, data_test) in enumerate(zip(all_data_copies,
   all_data_splits)):
   print(f"Data copy: {i+1}")
   X_train = data_train[feature_cols]
   X_train_prop = data_train[properties_cols]
   X_test = data_test[feature_cols]

   for col in properties_cols:
      y_train = data_train[col]
      best_score = np.inf
      for name, model in regressors:
         if name in param_grids:
            grid = GridSearchCV(model, param_grids[name],
               scoring='neg_mean_squared_error', cv=5,n_jobs=-1)
            grid.fit(X_train, y_train)
            if -grid.best_score_ < best_score:
               best_score = -grid.best_score_
               best_grids_prop[col] = grid
      print(f"Best model for {col} is
         {best_grids_prop[col].best_estimator_}")

   predicted_properties = {col:
      best_grids_prop[col].predict(X_train).reshape(-1, 1) for col
      in properties_cols}
   X_taget_fit_all = np.concatenate([X_train, X_train_prop], axis=1)
   X_train_all = np.concatenate([X_train,
      data_train[properties_cols], *predicted_properties.values()],
      axis=1)

   for col in targets_cols:
      y_train = data_train[col]
      best_score = np.inf
      for name, model in regressors:
         if name in param_grids:
            grid = GridSearchCV(model, param_grids[name],
               scoring='neg_mean_squared_error', cv=5,n_jobs=-1)
            grid.fit(X_train_all, y_train)
            if -grid.best_score_ < best_score:
               best_score = -grid.best_score_
               best_grids_target[col] = grid
      print(f"Best model for {col} is
         {best_grids_target[col].best_estimator_}")

   predicted_properties_test = {col:
      best_grids_prop[col].predict(X_test).reshape(-1, 1) for col
```

```
        in properties_cols}
    X_test_pred = np.concatenate([X_test,
        data_test[properties_cols],
        *predicted_properties_test.values()], axis=1)

    for col in ['NO', 'NOx']:
        pred_test_target = best_grids_target[col].predict(X_test_pred)
        Logerror_test_target = log_error(data_test[col],
            pred_test_target)
        mseerror_test_target = mean_squared_error(data_test[col],
            pred_test_target)
        Logerror_division = (Logerror_test_target)
        mseerror_division = (mseerror_test_target)
        print(f"real {col} value is: ", data_test[col].values, f"pred
            {col} value is: ", pred_test_target)

        # Store MSE_division
        all_Logerror_divisions_target[col].append(Logerror_division)
        all_mseerror_divisions_target[col].append(mseerror_division)
        print(f" MSE error for {col} is:
            {np.nanmean(mseerror_division)}")
    print("\n")

for col, mseerror_divisions in
    all_mseerror_divisions_target.items():
    print(f"Average mseerror for {col}:
        {np.nanmean(mseerror_divisions)}")
```

# References

[1] J. Z. Bloh, A. Folli and D. E. Macphee, *RSC Adv.*, 2014, **4**, 45726–45734.