Appendix

For each condition $\mathscr{D}_c = \{(x_i, y_i)\}_{i=1}^{N_c}$, where x_i denotes the spectra and y_i the corresponding target property, we trained a separate one-dimensional convolutional neural network (1D CNN). The input to the network consisted of spectra of length 4996. The architecture began with a batch-normalization layer applied to the input, followed by two convolutional blocks: Conv1D with 2 channels, kernel size 3, and stride 1, followed by average pooling with a factor of 2; and Conv1D with 4 channels, kernel size 3, and stride 2, followed again by average pooling with a factor of 2. The convolutional features were passed to four fully connected layers of sizes 2048, 1024, 512, and 128, each with Leaky ReLU activations and dropout ($p = 0.1$). The final output was produced by separate linear layers (128 $\rightarrow$ 1) for each elemental concentration.

Training was performed using the Adam optimizer with a learning rate of 0.001, combined with a cosine annealing scheduler over 40 epochs. We used a batch size of 128 and mean squared error (MSE) as the loss function. The dataset was split into 80% training, 10% validation, and 10% test sets. Spectral preprocessing included clipping values below zero, normalizing total emissivity, and standardizing the input, while target properties were also standardized.

Predictions on transferred spectra were obtained as

$$y_{\text{transf}} = f_{\text{CNN}_\varepsilon}\big(F(x, c, c')\big),$$

where $F(x, c, c')$ denotes the transfer model. For clarity, condition-specific notation has been omitted.