Supplementary Information for:

Active Learning-Guided Exploration of Thermally Conductive Polymers Under Strain

Renzheng Zhang[1], Jiaxin Xu[1], Hanfeng Zhang[1], Guoyue Xu[1], Tengfei Luo[1,2*]

1. Department of Aerospace and Mechanical Engineering, University of Notre Dame, Notre Dame, Indiana 46556, United States

2. Department of Chemical and Biomolecular Engineering, University of Notre Dame, Notre Dame, Indiana 46556, United States

3. Lucy Family Institute for Data & Society, University of Notre Dame, Notre Dame, Indiana 46556, United States

* Corresponding email: tluo@nd.edu

Supplementary Note 1. Optimization process in MD simulation

In the optimization process, the polymer systems were initially simulated with deactivated electrostatic interactions, while a 0.300 nm cutoff was implemented for Lennard-Jones interactions. The system was first subjected to a simulation under the NPT (constant number of atoms, pressure, and temperature) ensemble at 100 K for 2 ps, with a 0.1 fs timestep applied. The system was then gradually heated from 100 K to 1000 K over 1 ns under the NVT ensemble, followed by a simulation under the NPT ensemble for 50 ps at 0.1 atm and 1000 K. This stage was intended to further relax the structure and ensure the complete eradication of close contacts. Subsequently, the system underwent further simulation under NPT at 1000 K for 1 ns, during which the pressure was permitted to rise from 0.1 atm to 500 atm, with a 1 fs timestep and SHAKE constraints [1] implemented. The SHAKE constraint facilitates the usage of a larger 1 fs timestep in the presence of lightweight hydrogen atoms within the system. Throughout these stages, the deactivation of electrostatic interactions and the use of a range of ad-hoc simulation procedures were adopted to eliminate close atomic contacts, a vital measure to prevent system disruption in subsequent simulation steps.

These simulation procedures are collectively referred to as the initialization process. The obtained polymer system was then annealed with activated electrostatic interactions, using the PPPM (Particle–Particle–Particle–Mesh)-based Ewald sum method for computation. A 0.800 nm cutoff was used for the Lennard-Jones interactions. In this annealing phase, the system was first run in an NPT ensemble at 1 atm and 1000 K for 2 ps using a 0.1 fs timestep. Subsequently, it was cooled from 1000 K to 300 K at a rate of 140 K/ns, under an NPT

ensemble at 1 atm. This was followed by another NPT run at 300 K and 1 atm for 8 ns, utilizing a 1 fs timestep and SHAKE constraints to achieve the final amorphous state. Such a procedure has been shown to yield converged thermal conductivity (TC) results with NEMD [2].

Supplementary Note 2. Gaussian process regression model

A GPR model is a nonparametric Bayesian method that models the distribution over functions. In a GPR framework, we assume the observed data y consists of a latent function $f(\boldsymbol{x})$ corrupted by independent and identically normally distributed noise ϵ, where $\epsilon \sim N(0, \sigma^2)$. The function f is modeled as a Gaussian process, meaning that any finite set of function values follows a multivariate normal distribution with a specified mean function $\mathrm{m}(\boldsymbol{x})$ and a kernel (covariance) function $k(\boldsymbol{x}, \boldsymbol{x}')$. The kernel encodes the similarity between different input points. Given a dataset D, the joint distribution over the observed outputs y and the function values at a new point $\boldsymbol{x}_*$ is:

$$\begin{bmatrix} y \\ f(\mathbf{x}_*) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(\mathbf{X}) \\ m(\mathbf{x}_*) \end{bmatrix}, \begin{bmatrix} K(\mathbf{X},\mathbf{X}) + \sigma^2 I & k(\mathbf{X},\mathbf{x}_*) \\ k(\mathbf{x}_*,\mathbf{X}) & k(\mathbf{x}_*,\mathbf{x}_*) \end{bmatrix} \right)$$

Here, $K(\boldsymbol{X}, \boldsymbol{X})$ is the covariance matrix between the training points, and $k(\boldsymbol{X}, \boldsymbol{x}_*)$ is the covariance vector between the training points and the new point. Using Bayes' rule, we can derive the posterior distribution for the predicted mean $\boldsymbol{\mu}_*(\boldsymbol{x}_*)$ and the variance $\boldsymbol{\sigma}_*^2(\boldsymbol{x}_*)$, given by:

$$\mu_*(\mathbf{x}_*) = m(\mathbf{x}_*) + k(\mathbf{x}_*,\mathbf{X})^T [K(\mathbf{X},\mathbf{X}) + \sigma^2 I]^{-1} (y - m(\mathbf{X}))$$

$$\sigma_*^2(\mathbf{x}_*) = k(\mathbf{x}_*,\mathbf{x}_*) - k(\mathbf{x}_*,\mathbf{X})^T [K(\mathbf{X},\mathbf{X}) + \sigma^2 I]^{-1} k(\mathbf{X},\mathbf{x}_*)$$

For the kernel function $k(\boldsymbol{x}, \boldsymbol{x}')$, we use a Rational Quadratic kernel combined with a White kernel to account for noise in the observations. The Rational Quadratic Kernel is chosen because of it can model both small- and large-scale variations in the data, and the White Kernel is chosen to account for observational noise, ensuring that the model is robust to noisy data and does not overfit small fluctuations due to measurement errors. The combined kernel function is defined as:

$$k(\boldsymbol{x}, \boldsymbol{x}') = (1 + \frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\alpha \cdot l^2})^{-\alpha} + \sigma^2 \cdot \delta(\boldsymbol{x} - \boldsymbol{x}')$$

where, $\|\boldsymbol{x} - \boldsymbol{x}'\|$ is the squared Euclidean distance between input points $\boldsymbol{x}$ and $\boldsymbol{x}'$, α controls the relative weighting of large and small-scale variations in the data, l is the characteristic length scale, determining how quickly the function varies with the input. σ^2 represents the noise variance, and $\delta(\boldsymbol{x} - \boldsymbol{x}')$ is the Kronecker delta function, which ensures that noise only affects the diagonal of the covariance matrix.

Supplementary Note 3. Bayesian optimization process

Fig. S1 shows the parity plots of each iteration and GPR prediction & uncertainty on the PoLyInfo database after iteration 2. For each iteration, the newly incorporated data are marked by red circles. The predictive accuracy, estimated by the R^2, is above 0.8 in all iterations. The upper-bound GPR-predicted TC increases overall and reaches 1.1358 W/m·K in iteration 8. The amount of polymers that have GPR-predicted TC over 1.0 W/m·K increases from 0 in iteration 2 to 485 in iteration 8. **Table S1** provides detailed information, including PID, MD-labeled TC and SMILES, of the 35 BO-recommended polymers.

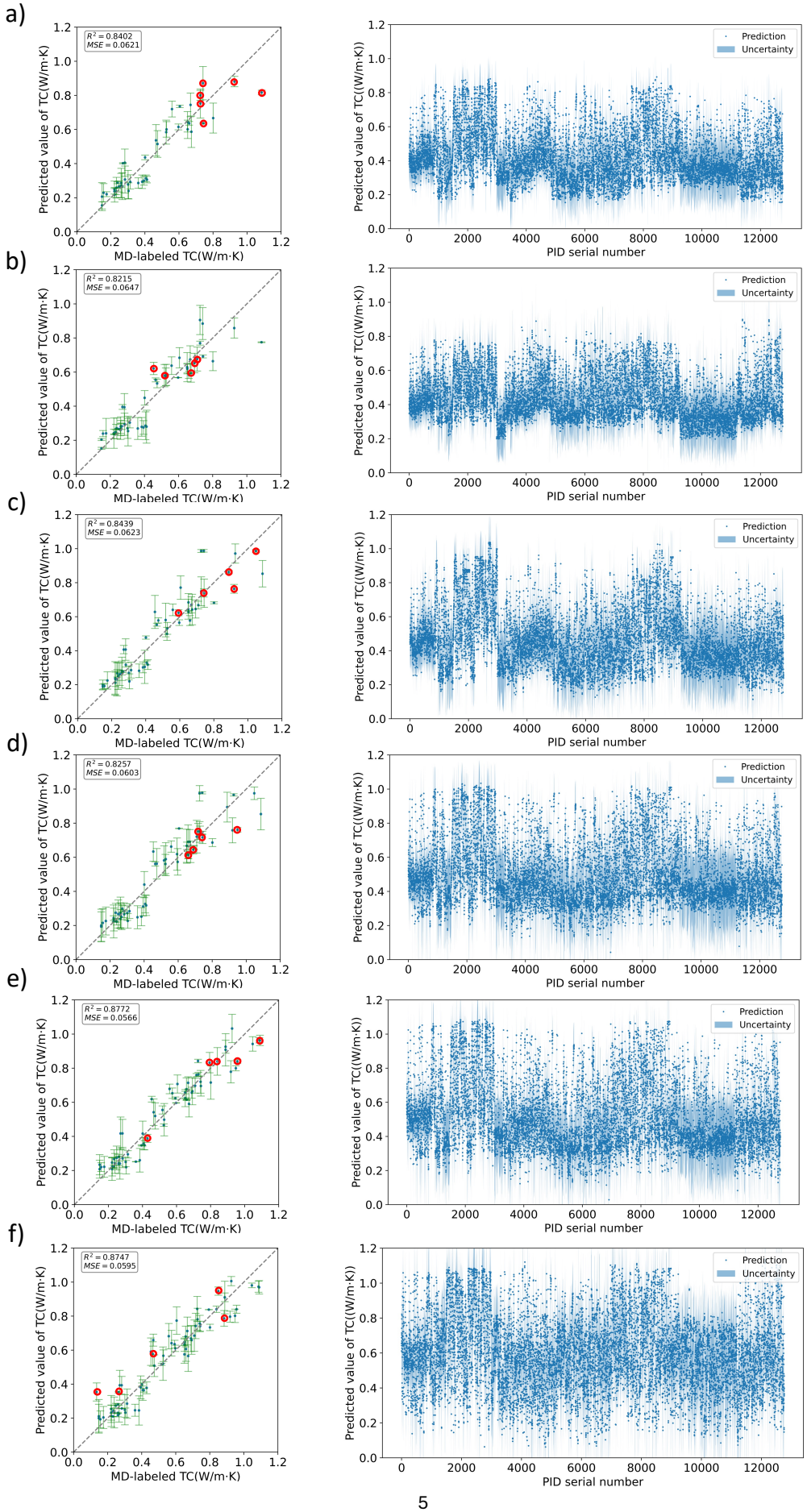

a)
$R^2 = 0.8402$
$MSE = 0.0621$
Predicted value of TC(W/m·K)
MD-labeled TC(W/m·K)
Prediction
Uncertainty
Predicted value of TC((W/m·K))
PID serial number
b)
$R^2 = 0.8215$
$MSE = 0.0647$
Predicted value of TC(W/m·K)
MD-labeled TC(W/m·K)
Prediction
Uncertainty
Predicted value of TC((W/m·K))
PID serial number
c)
$R^2 = 0.8439$
$MSE = 0.0623$
Predicted value of TC(W/m·K)
MD-labeled TC(W/m·K)
Prediction
Uncertainty
Predicted value of TC((W/m·K))
PID serial number
d)
$R^2 = 0.8257$
$MSE = 0.0603$
Predicted value of TC(W/m·K)
MD-labeled TC(W/m·K)
Prediction
Uncertainty
Predicted value of TC((W/m·K))
PID serial number
e)
$R^2 = 0.8772$
$MSE = 0.0566$
Predicted value of TC(W/m·K)
MD-labeled TC(W/m·K)
Prediction
Uncertainty
Predicted value of TC((W/m·K))
PID serial number
f)
$R^2 = 0.8747$
$MSE = 0.0595$
Predicted value of TC(W/m·K)
MD-labeled TC(W/m·K)
Prediction
Uncertainty
Predicted value of TC((W/m·K))
PID serial number

Fig. S1 | GPR model performances from iteration 3 to iteration 8. Parity plot between GPR-predicted TC versus MD-labeled TC and GPR-predicted TC mean and uncertainty of each polymer in the PoLyInfo database in **a** iteration 3, **b** iteration 4, **c** iteration 5, **d** iteration 6, **e** iteration 7, **f** iteration 8, with the five new BO-guided data marked by red circles.

Table S1. MD-labeled TC of 35 BO-guided polymers and their SMILES

Iteration	PID	TC (W/m·K)	Smiles
Random	P432586	0.348	*c1ccc(Oc2ccc(-c3nc4ccc(Oc5ccc(N6C(=O)c7ccc(C(c8ccc9c(c8)C(=O)N(c8ccc(Oc%10ccc%11nc(*)c(-c%12ccccc%12)nc%11c%10)cc8)C9=O)(C(F)(F)F)C(F)(F)F)cc7C6=O)cc5)cc4nc3-c3ccccc3)cc2)cc1
	P460108	0.2	*/C(=N\c1ccccc1)N(*)c1ccccc1
	P010100	0.223	*C1CCC1*
	P040310	0.238	*CC(*)(CC(=O)OCCCCCCC)C(=O)OCCCCCCC
	P100773	0.521	*CCCCNC(=O)CCCCCCCC(=O)NCCCCNC(=O)C(=O)N*
Iter_2	P120025	0.801	*CCCCCCCCCCCCCNC(=O)NCCCCCCCCCCCNC(=O)N*
	P400028	0.726	*CCCCCCCCCCCCCNC(=O)CCCCCCCCC(=O)NCCCCCCCCCCCCCNC(=O)C(=O)N*
	P340177	0.525	*CC(*)(C)C(=O)NCCCCCCCCCCC(=O)NCCCCCCCCCCC(=O)OC
	P340179	0.527	*CC(*)(C)C(=O)NCCCCCCCCCCC(=O)NCCCCCCCCCCC(=O)O
	P100719	0.605	*CCCCCCNC(=O)C(CCCCCCCCCCCCCCCCC)C(=O)N*
Iter_3	P100029	0.926	*CCCCCCCCCCCCCCNC(=O)CCCCCCCCCCCCCCCCC(=O)N*
	P100235	0.741	*CCCCCCCCCCCCNC(=O)CCCCCCCCCCCCCCCCCCC(=O)N*
	P100238	0.726	*CCCCCCCCCCCCCCNC(=O)CCCCCCCCCCCCCCCCCCC(=O)N*
	P332306	0.743	*CCCCCCCCCCCCCCCCCCCCC(*)COCCOCCOCCOCCOCCCCCCCCCCCCC
	P402128	1.086	*CCCCCCCCCCCCNC(=O)CCCCCCCCCCCCCCCCCC(=O)N*
Iter_4	P090293	0.517	*CCCCCC(=O)NCCCCCCNC(=O)CCCCCOC(=O)CCCCC(=O)O*
	P090368	0.454	*CCCCCCCCOC(=O)CCCCCNC(=O)CCCCC(=O)NCCCCCC(=O)O*
	P100747	0.709	*CCCCCCCCCNC(=O)CCCCCCCCC(=O)NCCCCCCCCCNC(=O)C(=O)N*
	P100809	0.651	*CCCCCCCNC(=O)CCCCCCCCC(=O)NCCCCCCCNC(=O)C(=O)N*
	P400027	0.693	*CCCCCCCCNC(=O)CCCCCCCCC(=O)NCCCCCCCCNC(=O)C(=O)N*
Iter_5	P090253	0.888	*CCCCCCCCCCCCCCCCCCCCCOC(=O)CCCCCCCC(=O)O*
	P100030	1.136	*CCCCCCCCCCCCCCCCCCCNC(=O)CCCCCCCCCCCCCCCCC(=O)N*
	P100696	0.919	*CCCCCCCCCCCCCCCCCCCCC(=O)N*
	P332076	0.741	*/C=C(/*)CCCCCCCCCCCCCCCCCCC(=O)O
	P372661	0.593	*CCOCCOCCOCCOCc1cc(CO*)cc(OCCCCCCCCCCCCCCCCCC)c1
Iter_6	P332081	0.728	*/C=C(/*)C#CCCCCCCCCCCCCCCCCCCCC(=O)O
	P080116	0.689	*CCCCCCCCC(=O)NCCc1ccc(CCNC(=O)CCCCCCCCS*)cc1
	P100124	0.717	*CCCCCCCCCCCCCCCCCCCNC(=O)CCc1ccc(CCC(=O)N*)cc1
	P100063	0.658	*CCc1ccc(CCNC(=O)CCCCCCCCCCCCCCCCC(=O)N*)cc1
	P390132	0.948	*CCCCCCCCCCCCCCCCCOC(=O)CCCCCCCCC(=O)O*
Iter_7	P372375	0.429	*CCCNC(=O)CNC(=O)C(NC(=O)C(Cc1ccccc1)NC(=O)CCCCCCCCC(=O)NC(Cc1ccccc1)C(=O)NC(C(=O)NCC(=O)NCCCOCCOCCO*)C(C)C)C(C)C
	P402222	1.036	*CCCCCCCCCCCCCNC(=O)CCCCCCCCCCCCC(=O)N*
	P402250	1.09	*CCCCCCCCCCCNC(=O)CCCCCCCCCCCCCCCCCC(=O)N*
	P110210	0.839	*CCCCCCCCCCCCCCNC(=O)CCCCCCCCCCCC(=O)N*
	P392523	0.824	*CCCCCCCCCCCCCCCCCCCCCCCOC(=O)CCC(=O)O*
Iter_8	P450068	0.263	*CC(OC(=O)OC1CCC2(C)C(=CCC3C2CCC2(C)C(C(C)CCCC(C)C)CCC32)C1)C(COC(=O)O*)OC(=O)OC1CCC2(C)C(=CCC3C2CCC2(C)C(C(C)CCCC(C)C)CCC32)C1
	P402218	0.884	*CCCCCCCCCNC(=O)CCCCCCCCCCCCCCCC(=O)N*
	P100754	0.878	*CCCCCNC(=O)CCCCCCCCCCCCCCCCCC(=O)N*
	P522013	0.135	*CC(*)(F)C(=O)OCC(F)(F)C(F)(F)C(F)(F)C(F)(F)C(F)(F)C(F)(F)C(F)(F)C(F)(F)C(F)(F)C(F)(F)C(F)(F)F
	P382405	0.468	*c1nc(CCCCCCCC)c(-c2sc(-c3sc(-c4cc(CCCCCCCC)c(*)s4)cc3CCCCCCCC)nc2CCCCCCCC)s1

Supplementary Note 4. Comparative analysis of other ML models

To compare the performance of the GPR model, we test two other popular ML models: random forest (RF) and gradient boosting regressor (GBR), both of which are insensitive to dimensionality, so no dimensionality reduction process is performed on both models. **Fig. S2a&d** show the parity plot between RF-predicted TC and MD-labeled TC in iteration 1 and iteration 8. **Fig. S2b&e** show the parity plot between GPR-predicted TC and RF-predicted TC in iteration 1 and iteration 8, where R^2 are 0.72 and 0.53, respectively. Since neither the RF model nor the GBR model provides the uncertainty, we only have the predicted value of TC, as shown in **Fig. S2c&f**. Similarly, **Fig. S2g&j** show the parity plot between GBR-predicted TC and MD-labeled TC in iteration 1 and iteration 8. **Fig. S2h&k** show the parity plot between GPR-predicted TC and GBR-predicted TC in iteration 1 and iteration 8, where R^2 are 0.64 and 0.52, respectively. The predicted value of TC in iteration 1 and iteration 8 are shown in **Fig. S2i&l.** It is noteworthy that with the initial 36 training data, the distributions of the RF and GBR models in iteration 1 are more similar to the GPR model, where the agreement is more reasonable than that in iteration 8. With more data incorporated, the R^2 of RF-GPR and GBR-GPR decreased instead. It has been validated that the GPR model has accurate predicted TC values that match the MD simulation, especially for high TC polymers. However, the boundaries of RF and GBR models are both around 1, which means both RF and GBR models underestimate the strained polymers TC to some extent (more datapoints are above the gray dashed line).

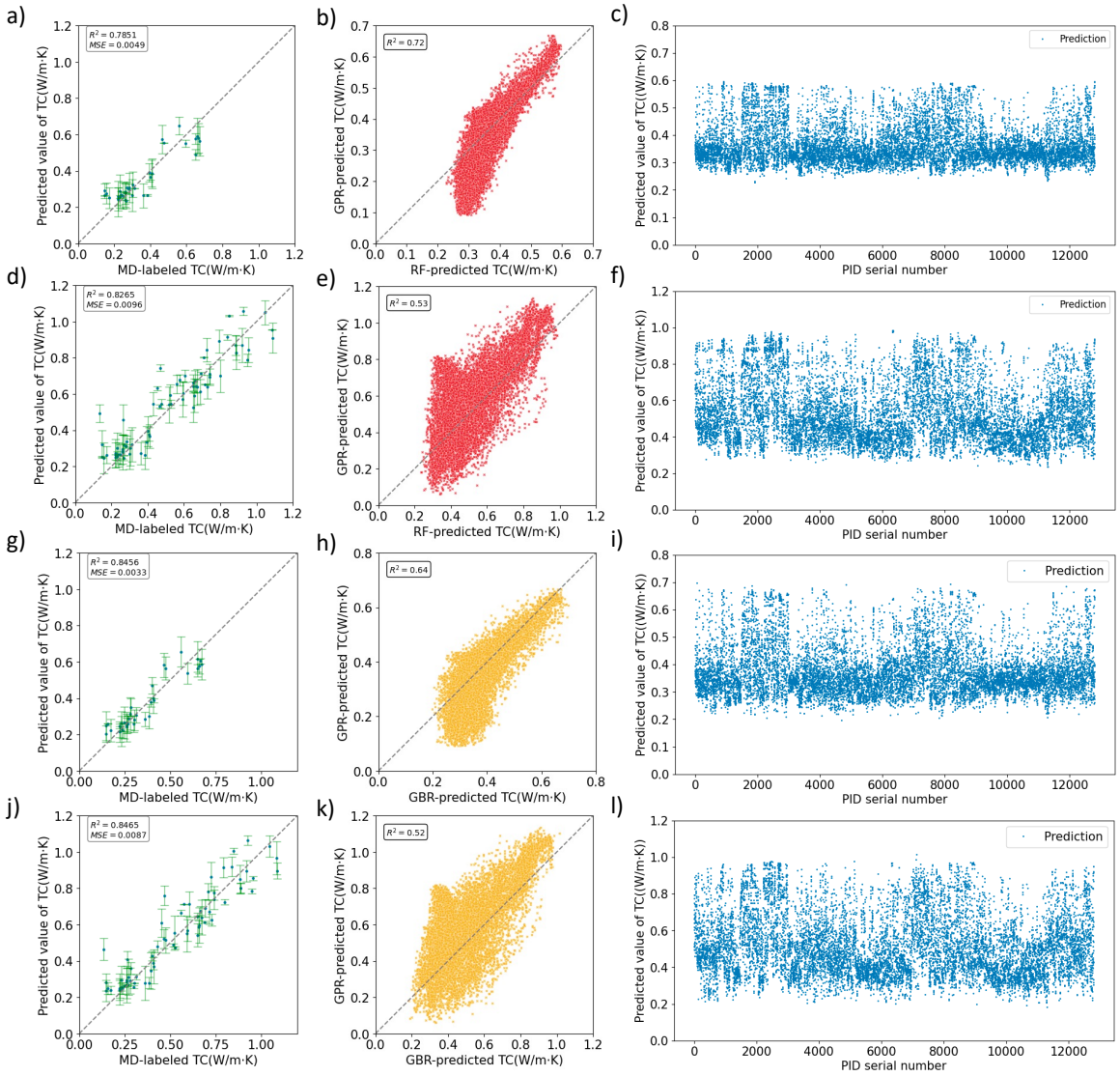


Fig. S2 | Comparative analysis of RF and GBR models. a Parity plot between RF-predicted TC and MD-labeled TC in iteration 1. **b** Parity plot between GPR-predicted TC and RF-predicted TC in iteration 1. **c** RF-predicted TC of each polymer in the PoLyInfo database in iteration 1. **d** Parity plot between RF-predicted TC and MD-labeled TC in iteration 8. **e** Parity plot between GPR-predicted TC and RF-predicted TC in iteration 8. **f** RF-predicted TC of each polymer in the PoLyInfo database in iteration 8. **g** Parity plot between GBR-predicted TC and MD-labeled TC in iteration 1. **h** Parity plot between GPR-predicted TC and GBR-predicted TC in iteration 1. **i** GBR-predicted TC of each polymer in the PoLyInfo database in iteration 1.

j Parity plot between GBR-predicted TC and MD-labeled TC in iteration 8. **k** Parity plot between GPR-predicted TC and GBR-predicted TC in iteration 8. **l** GBR-predicted TC of each polymer in the PoLyInfo database in iteration 8.

Supplementary Note 5. Summary of the 30 high-TC (>0.8 W/m·K) polymers' categories

Fig. S3 shows the classes of the 30 MD-labeled high-TC (>0.8 W/m·K) polymers. Polyamides is the majority (~ 50%) since it contains amide bonds (-CONH-) linking monomers, which can be formed by the condensation reaction between an amine group ($–NH_2$) and a carboxylic acid group (–COOH). The formation of amide linkages during polymerization facilitates the establishment of strong hydrogen bonds between the polymer chains, enhancing the material's thermal transport properties.

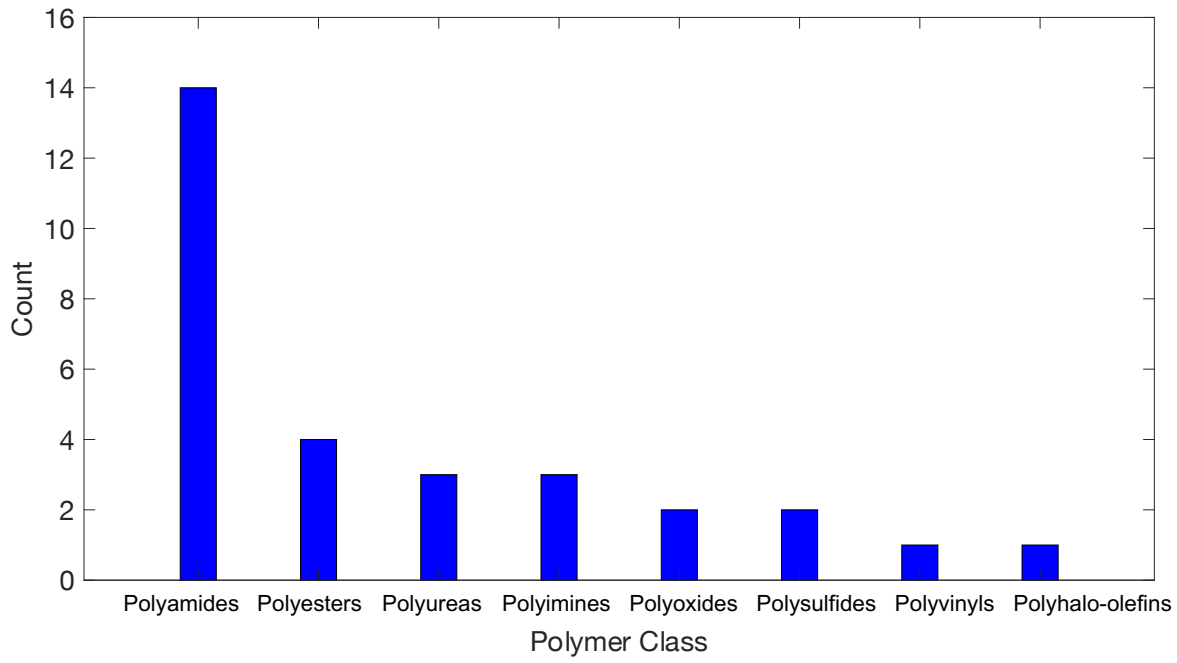
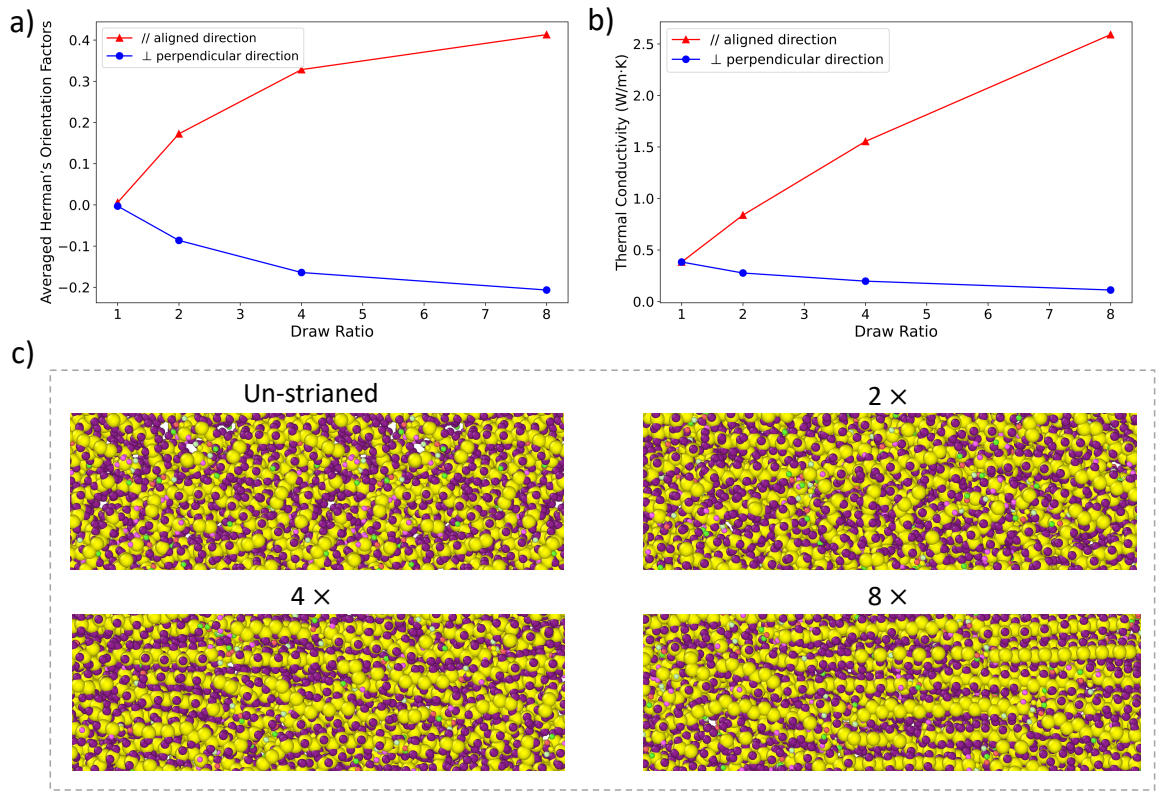


Fig. S3 | Summary of the 30 MD-labeled high TC (>0.8 W/m·K) polymers' categories.

Supplementary Note 6. Herman's orientation factor

Herman's orientation factor (f) [3,4] is used to characterize the segment alignment along different directions: $f = 1.5 < cos^2\theta > -0.5$, where θ is the angle between a carbon-carbon bond and the reference direction. A value of f = -0.5 means that the orientation is perpendicular to the selected direction, and f = 1 means that the orientation is parallel to the selected direction. A value of 0 indicates a completely random orientation. We select polymer P110210 to simulate its f and TC in the oriented and perpendicular directions at different draw ratio. **Fig. S4a** shows f increases with larger draw ratio along the draw direction, while decreasing in the perpendicular direction. **Fig. S4b**, respectively, indicates the changes of TC in the oriented and perpendicular direction at different draw ratio, which is consistent with **Fig. S4a**. **Fig. S4c** visualizes the structures of this polymer under different strain ratios. The main chain carbon atoms, indicated in yellow, are more aligned with the draw ratio increasing.



136

137 **Fig. S4 | Herman's orientation factor simulation for polymer P110210 at different draw**

138 **ratio. a** Averaged Herman's orientation factor and **b** TC in oriented and perpendicular

139 directions at different draw ratios of polymer P110210. **c** Structures visualization of polymer

140 P110210 at different draw ratios.

Supplementary Note 7. Structural comparison of high TC and low TC polymers

Fig. S5 shows the polymers that have linear chain structures (right) and non-linear chain structures (left). For linear chain polymers, they have much higher TC and *f* than non-linear chain polymers due to the higher orientation factor and chain alignments.

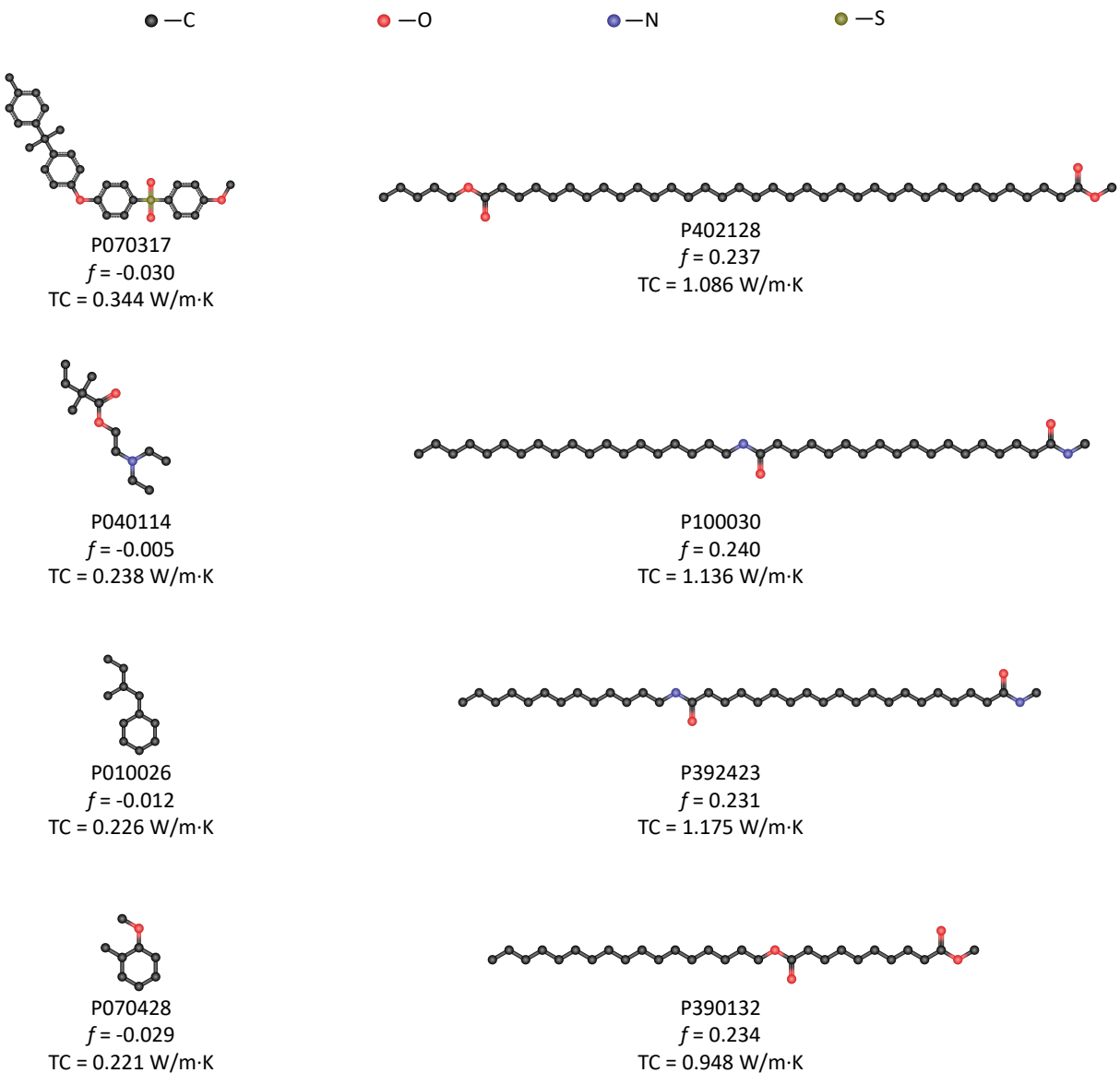


Fig. S5 | Structural comparison of high TC polymers (right) and low TC polymers (left).

Reference

(1) Ryckaert JP, Ciccotti G, Berendsen HJC. Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes. J Comput Phys. 1977; 23(3):327–341.

(2) Ma R, Zhang H, Xu J, Sun L, Hayashi Y, Yoshida R, Shiomi J, Wang J, Luo T. Machine learning-assisted exploration of thermally conductive polymers based on high-throughput molecular dynamics simulations. Mater Today Phys. 2022;28:100850.

(3) Zhang T, Luo T. Role of chain morphology and stiffness in thermal conductivity of amorphous polymers. J Phys Chem B. 2016;120(4):803–812.

(4) Pal S, Balasubramanian G, Puri IK. Modifying thermal transport in electrically conducting polymers: effects of stretching and combining polymer chains. J Chem Phys. 2012;136(4):044901–044907.