nature portfolio

https://doi.org/10.1038/s41586-023-06782-2

Supplementary information

Multiscale stress deconcentration amplifies fatigue resistance of rubber

In the format provided by the authors and unedited

Supplementary Information for

Multiscale stress deconcentration amplifies fatigue resistance of rubber

Jason Steck[1†], Junsoo Kim[1†], Yakov Kutsovsky[2]*, Zhigang Suo[1]**

[1] John A. Paulson School of Engineering and Applied Science, Harvard University, Cambridge, MA 02138, USA

[2] Expert-in-Residence, Office of Technology Development, Harvard University

[†] These authors equally contributed to this work.

To whom correspondence should be addressed. Email: *yakov@materiax.ai, **suo@seas.harvard.edu

This PDF file includes:

Supplementary Figs. S1 to S13

Captions for Supplementary Videos 1 to 4

Other Supplementary Materials for this manuscript include the following:

Supplementary Videos 1 to 4

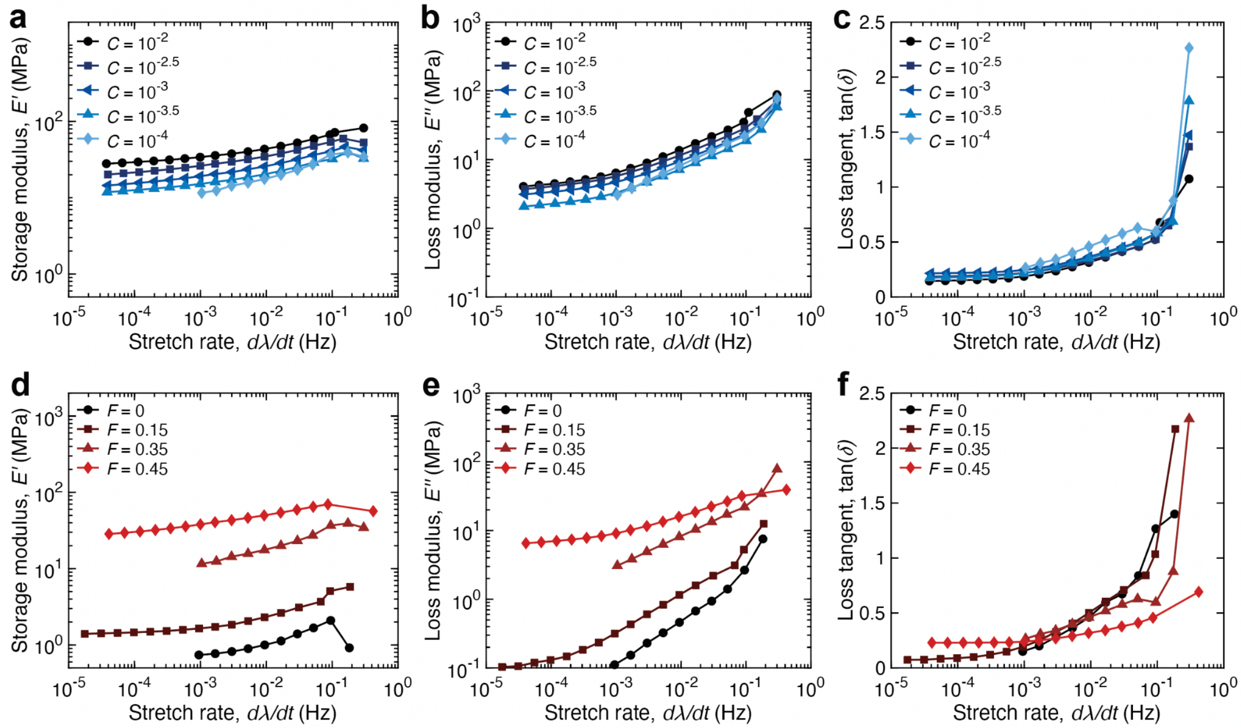

Figure S1. Dynamic mechanical analysis (DMA). The storage modulus *E'*, loss modulus *E''*, and loss tangent tan(δ) are measured as functions of stretch rate $d\lambda/dt$ for (**a, b, c**) $F = 0.35$ and various *C* and for (**d, e, f**) $C = 10^{-4}$ with various *F*. *E*' is insensitive to the stretch rate, but *E*'' increases with the stretch rate. Consequently, tan(δ) increases with the stretch rate. We also observe that *E*', *E*'', and tan(δ) are insensitive to *C*, but are sensitive to *F*.

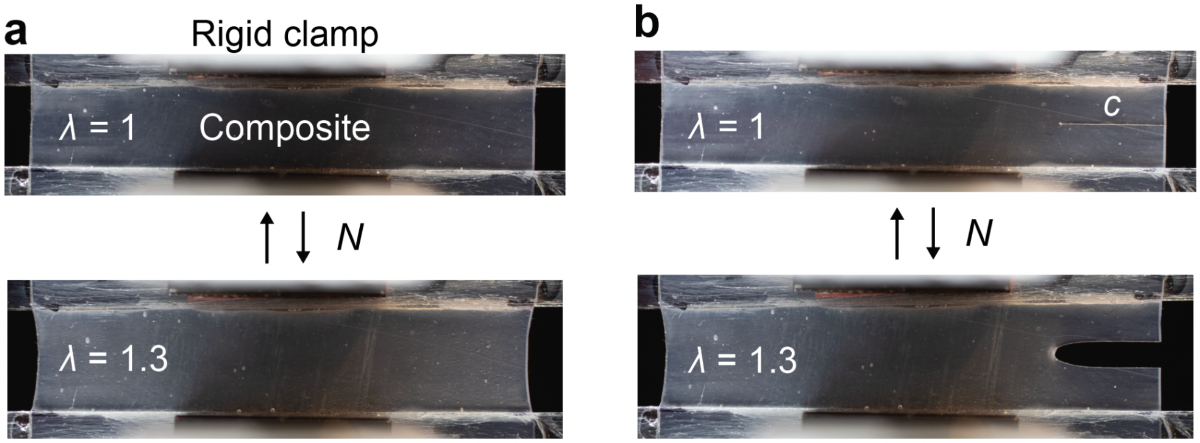

Figure S2. Pure shear test. (a) A rectangular membrane of composite is cyclically stretched using two rigid clamps. Stress-stretch curves are measured using an uncut sample, and used to calculate the energy release rate G. (**b**) A crack of length c is cut into the side of a membrane with scissors. The membrane is cyclically stretched, and crack growth is measured as the change in c. Representative photographs are of composites with $F = 0.45$ and $C = 10^{-4}$.

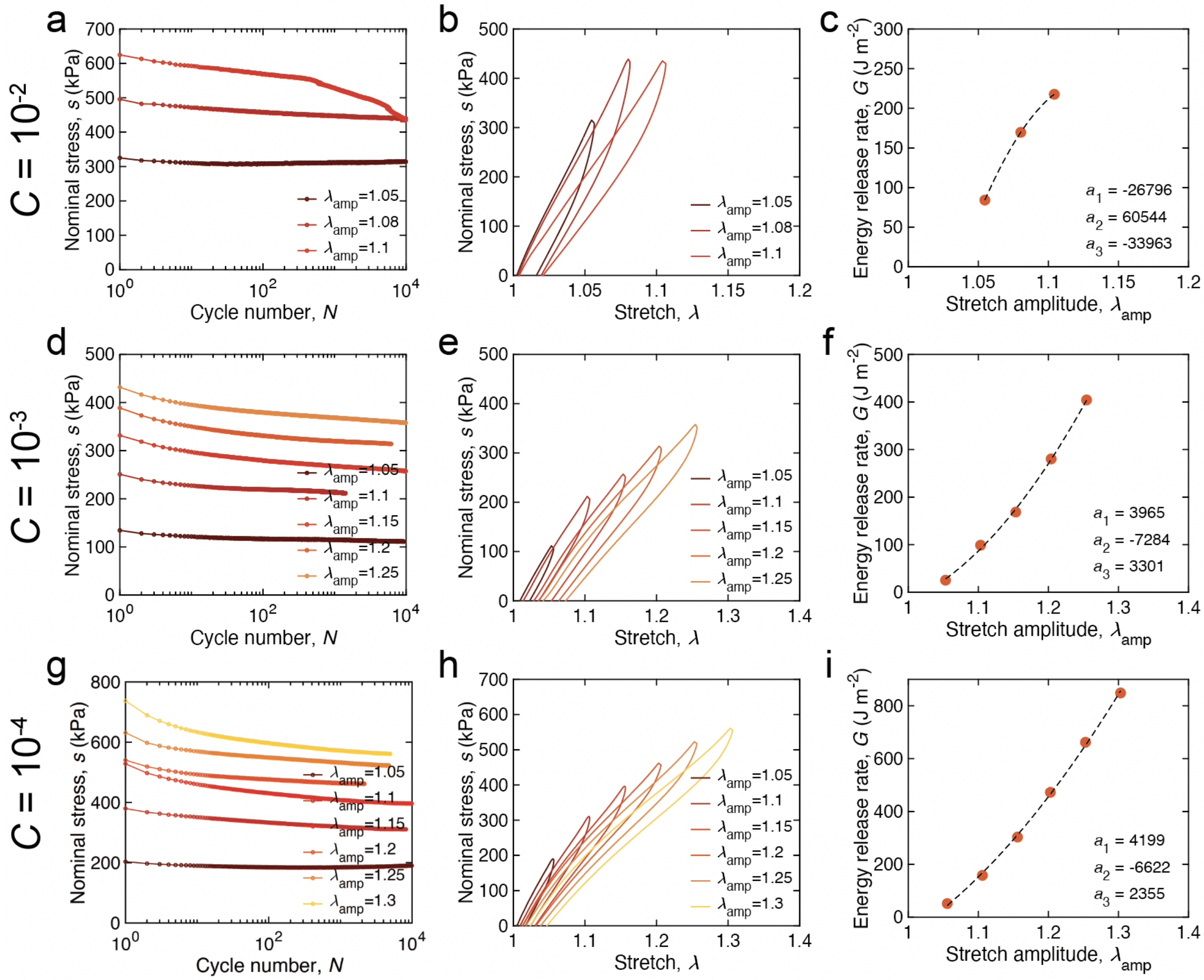

Figure S3. Cyclic loading of F = 0.15 with various C. Composites are cyclically stretched to a stretch of amplitude λ_{amp} in the pure shear geometry. For all samples, (**a**, **d**, **g**) the peak stress decays with cycles. (**b**, **e**, **h**) The stress-stretch curves are plotted in the steady state. (**c**, **f**, **i**) The steady-state energy release rate G as a function of λ_{amp} is fit with a quadratic function $G = a_1\lambda_{amp}^2 + a_2\lambda_{amp} + a_3$.

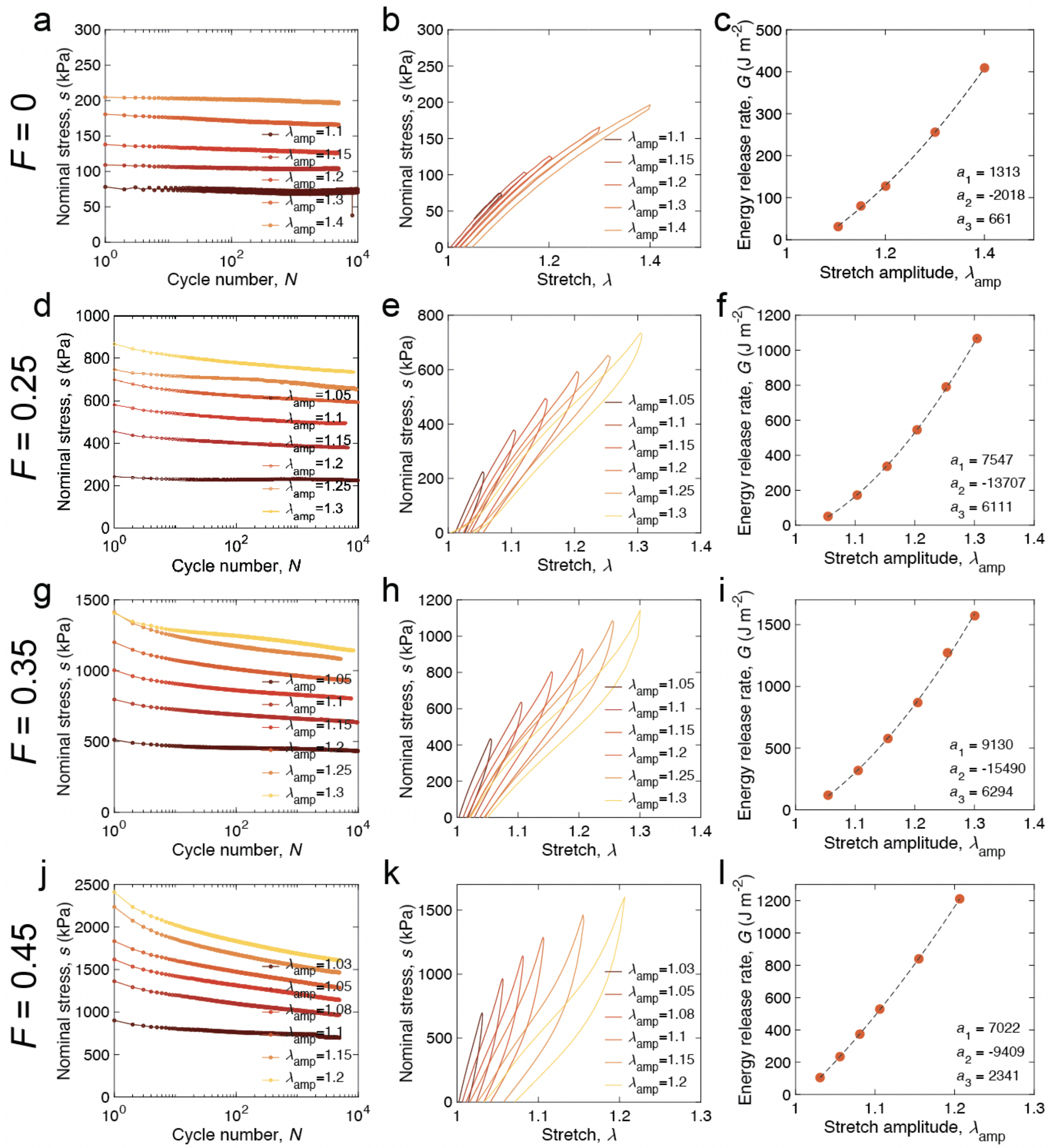

Figure S4. Cyclic loading of $C = 10^{-4}$ with various F. Composites are cyclically stretched to a stretch of amplitude λ_{amp} in the pure shear geometry. For all samples, (**a**, **d**, **g**, **j**) the peak stress decays with cycles. (**b**, **e**, **h**, **k**) The stress-stretch curves are plotted in the steady state. (**c**, **f**, **i**, **l**) The steady-state energy release rate G as a function of λ_{amp} is fit with a quadratic function $G = a_1\lambda_{amp}^2 + a_2\lambda_{amp} + a_3$.

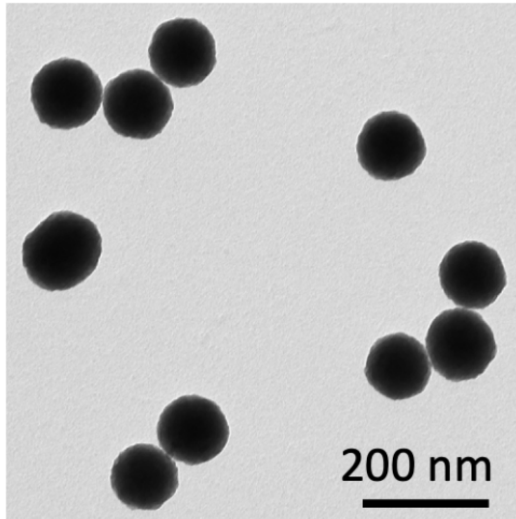

Figure S5. Transmission electron microscope (TEM) image of TPM-functionalized silica nanoparticles. This micrograph was provided by Cabot Corporation.

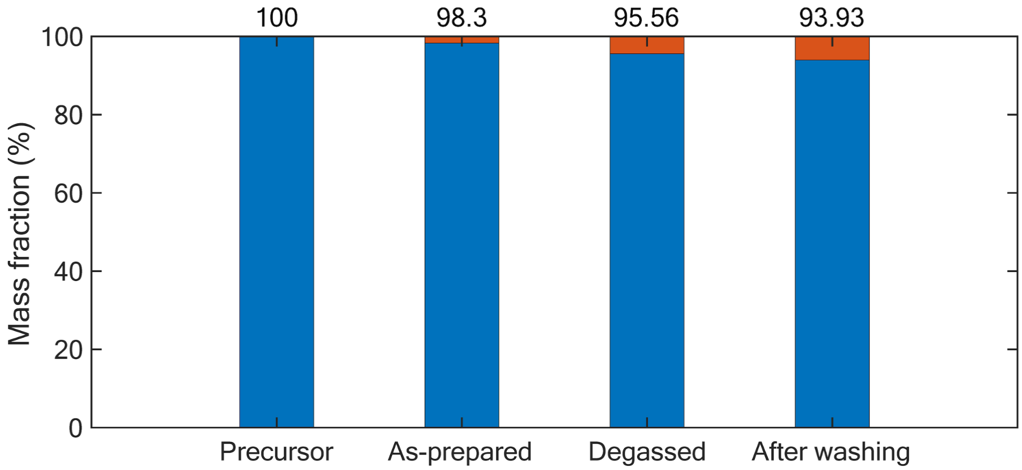

Figure S6. The fraction of PEA chains that crosslink into a network. The mass of precursor for a PEA elastomer of $C = 10^{-4}$ is measured before curing. The PEA elastomer is then weighed after polymerization, as well as after degassing in air for two days. The degassed sample is submerged in ethyl acetate for one week, after which the sample is removed from the solvent and the solvent in the sample is allowed to fully evaporate in air. The gel fraction is 95.54 wt%, calculated as the mass of the washed sample divided by the mass of the as-prepared sample. Most polymer chains are crosslinked into a polymer network.

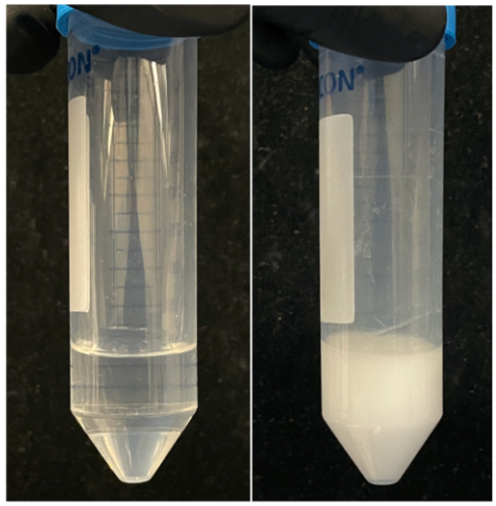

Figure S7. Precursor solution of composite. Ethyl acrylate (left) without silica nanoparticles and (right) with silica nanoparticles (F = 0.15). The nanoparticles readily dissolve in ethyl acrylate, and the resulting mixture is opaque.

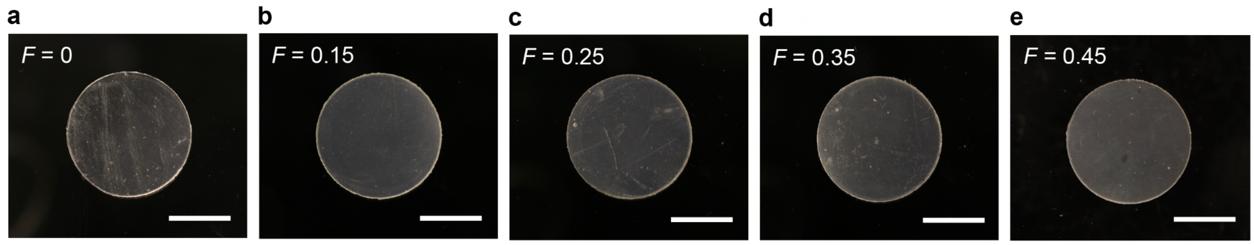

Figure S8. PEA elastomers reinforced with silica nanoparticles. (**a-e**) Photographs of composites of various F and fixed $C = 10^{-4}$. The scale bars are 1 cm.

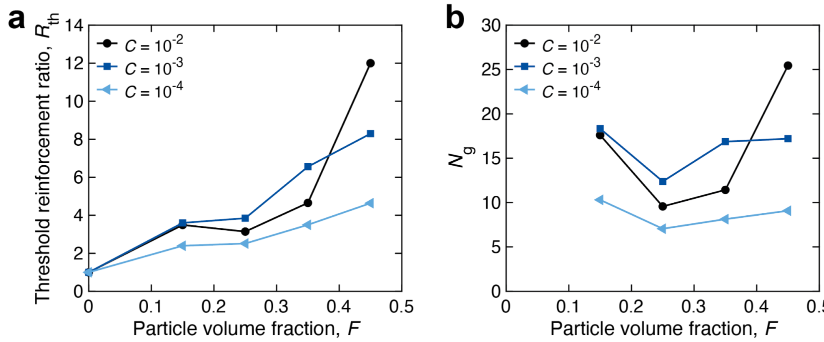

Figure S9. Reinforcement of fatigue threshold. (a) Threshold reinforcement ratio, $R_{th} = G_{th}/G_{th,m}$, and (b) N_g plotted as functions of F. The value of N_g is evaluated from R_{th} and Eqn. (1) for each value of F.

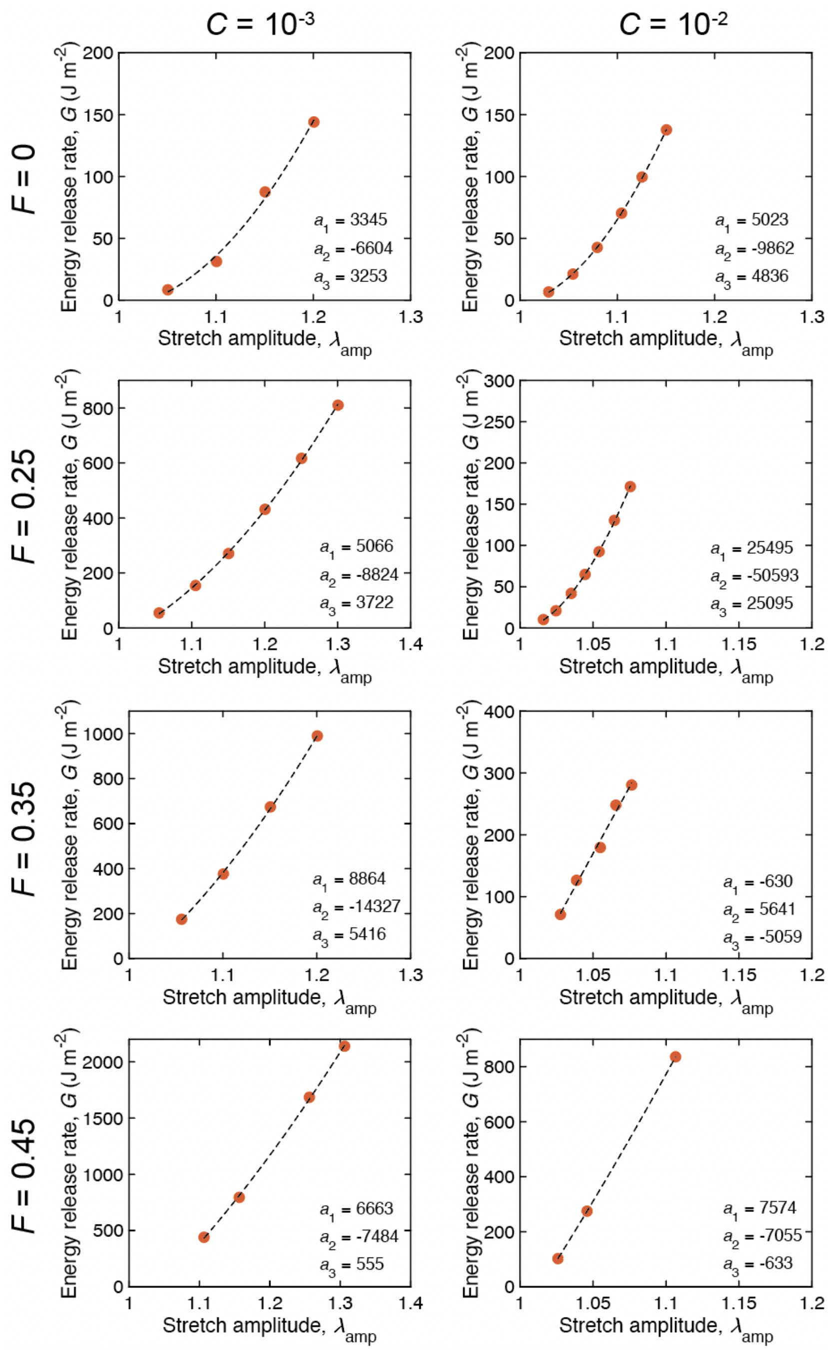

Figure S10. Energy release rate for composites in cyclic stretch. The steady-state energy release rate G is measured as a function of stretch amplitude λ_{amp}. For each C and F, G as a function of λ_{amp} is fit with a quadratic function $G = a_1\lambda_{amp}^2 + a_2\lambda_{amp} + a_3$.

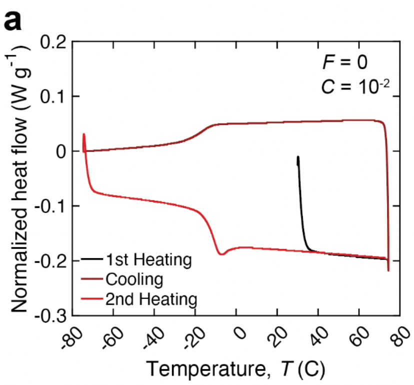

Figure S11. Differential scanning calorimetry of composites. The mass normalized heat flow is plotted as a function of temperature for (**a**) $C = 10^{-2}$ and $F = 0$, (**b**) $C = 10^{-4}$ and $F = 0$, and (**c**) $C = 10^{-4}$ and $F = 0.35$, where the glass transition temperature upon cooling is measured, respectively, to be T_g = -17.49 °C, T_g = -20.96 °C, and T_g = -21.93 °C. For pure PEA of $C = 10^{-4}$, T_g upon cooling is -20.96 °C, which agrees with the expected values for PEA.

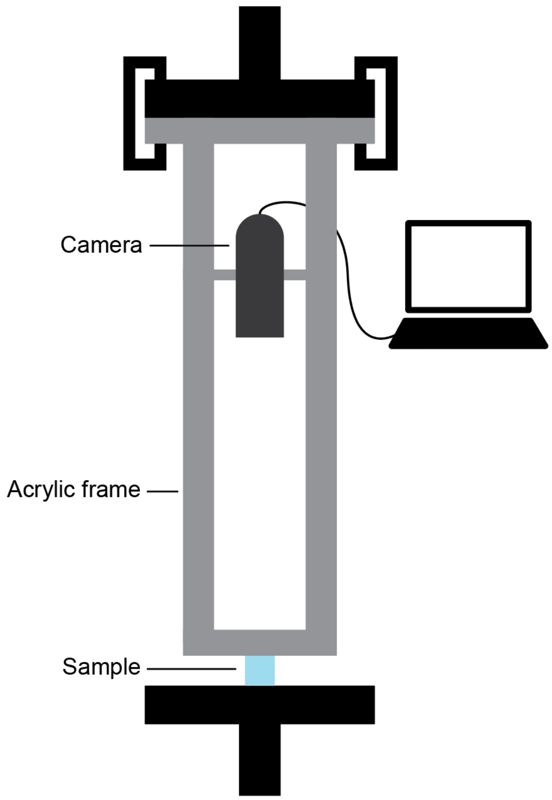

Figure S12. Compression test. The sample is placed between two rigid plates. The top plate is made of a transparent acrylic frame. A camera monitors the sample from above during the compression test. The acrylic frame is clamped to the top plate of the Instron. The force applied by the Instron transmits through the frame to the sample.

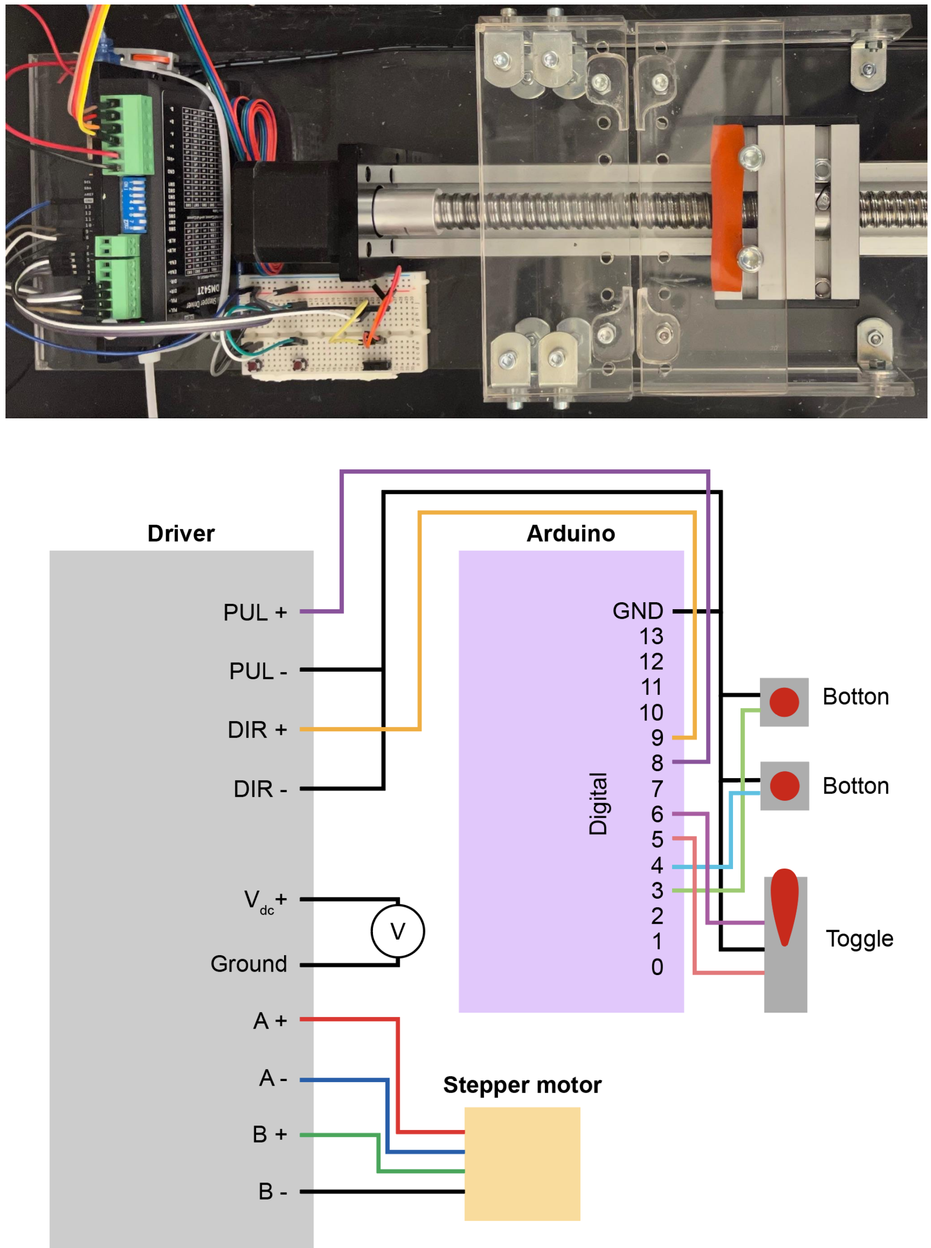

Figure S13. Fatigue tester. The photo of the fatigue tester (top). The connection of linear motor, driver, and controller (bottom).

<u>Arduino code</u>

```
// Fatigue tester v3. 12/17/2021 Made by Junsoo Kim
byte vccPin = 2;
byte upswitchPin = 3;
byte downswitchPin = 4;
byte updownmodePin = 5;
byte fatiguemodePin = 6;
byte directionPin = 8;
byte stepPin = 9;
```

```
double stepsperrevolution = 6400;
int stepdelay = 23;

// Put your parameters
double maxcount = 50000;
double distance = 3.0; // mm to move
double velocity = 2;// mm/s
double t = 0;

// Calculate the actuation parameters
double NumberOfSteps = double(stepsperrevolution*distance) / 5;
double frequency = 1/(distance*2/velocity); // Hz
int microbetweenSteps = 1000000 / double(stepsperrevolution) * 5 / velocity;
int millisbetweenloading = 0;

void setup() {
 Serial.begin(9600);
 Serial.println();
 Serial.println("Starting Test");
 Serial.print("Steps per revolution: ");
 Serial.println(stepsperrevolution);
 Serial.print("Distance: ");
 Serial.print(distance);
 Serial.println(" mm");
 Serial.print("Velocity: ");
 Serial.print(distance * frequency * 2);
 Serial.println(" mm/s");
 Serial.print("Frequency: ");
 Serial.print(frequency);
 Serial.println(" Hz");
 Serial.print("The number of cycles: ");
 Serial.println(maxcount);
 Serial.print("The duration of the test: ");
 Serial.print(distance *2/velocity*maxcount/3600);
 Serial.println(" hours");
 Serial.print("Delay between steps: ");
 Serial.print(microbetweenSteps-stepdelay);
 Serial.println(" us");
 Serial.print("time for loading: ");
 Serial.print(double(microbetweenSteps)*double(NumberOfSteps) / 1000000);
 Serial.println(" s");
 pinMode(upswitchPin, INPUT_PULLUP);
 pinMode(downswitchPin, INPUT_PULLUP);
 pinMode(updownmodePin, INPUT_PULLUP);
 pinMode(fatiguemodePin, INPUT_PULLUP);

 pinMode(directionPin, OUTPUT);
 pinMode(stepPin, OUTPUT);
}
void loop()   {
 if (digitalRead(fatiguemodePin) == 0) {
   Serial.println("Error: Go to the updown mode and reset");
   delay(1000);
   exit(0);
 }
 if (microbetweenSteps < stepdelay) {
   Serial.println("Error: The frequency is too fast");
   delay(1000);
   exit(0);
 }
 if (digitalRead(updownmodePin) == 0) {
   Serial.println("The updown mode. Please adjust the holder");
```

```
    while (digitalRead(updownmodePin) == 0) {
      if (digitalRead(upswitchPin) == 0) {
        digitalWrite(directionPin, HIGH);
        digitalWrite(stepPin, HIGH);
        digitalWrite(stepPin, LOW);
        delayMicroseconds(500);
      }
      if (digitalRead(downswitchPin) == 0) {
        digitalWrite(directionPin, LOW);
        digitalWrite(stepPin, HIGH);
        digitalWrite(stepPin, LOW);
        delayMicroseconds(500);
      }
    }
  }
  Serial.println("Fatigue test start");
  delay(1000);
  double m = 0;
  while (m < maxcount) {
    if (digitalRead(updownmodePin) == 0) {
      continue;
    }
    t = millis();
    digitalWrite(directionPin, LOW);
    for (double n = 0; n < NumberOfSteps; n = n + 1) {
      digitalWrite(stepPin, LOW);
      digitalWrite(stepPin, HIGH);
      delayMicroseconds(microbetweenSteps - stepdelay);
    }

    digitalWrite(directionPin, HIGH);
    for (double n = 0; n < NumberOfSteps; n = n + 1) {
      digitalWrite(stepPin, LOW);
      digitalWrite(stepPin, HIGH);
      delayMicroseconds(microbetweenSteps - stepdelay);
    }
    m = m + 1;
    Serial.print("Cycle number: ");
    Serial.print(m);
    Serial.print("  Actual frequency: ");
    Serial.print(1000 / (millis() - t));
    Serial.print("  Remaing time: ");
    Serial.print((maxcount-m)*distance*2/velocity/3600);
    Serial.println("  hours");

  }
  Serial.println("Fatigue test is completed");
  delay(1000);
  exit(0);
}
```

Supplementary Videos

Video 1. Cyclic compression of mounts with various *C* and *F*.

Video 2. Fatigue test of a mount with $C = 10^{-4}$ and $F = 0.45$ for 33,000 cycles under compression.

Video 3. Cyclic stretch of compliant grippers with $C = 10^{-2}$ and $C = 10^{-4}$ at fixed $F = 0.45$.

Video 4. Lift force measurement of compliant grippers with $F = 0$ and $F = 0.45$ at fixed $C = 10^{-4}$.