Neural-Assisted Homogenization of Yarn-Level Cloth – Supplementary Materials

Xudong Feng*
xudongfeng18@gmail.com
State Key Laboratory of CAD&CG, Zhejiang University
Hangzhou, China

Huamin Wang
wanghmin@gmail.com
Style3D Research
Hangzhou, China

Yin Yang
yin.yang@utah.edu
The University of Utah
USA
Style3D Research
Hangzhou, China

Weiwei Xu†
xww@cad.zju.edu.cn
State Key Laboratory of CAD&CG, Zhejiang University
Hangzhou, China

1 DETAILS IN DATA PREPARATION WITH WARM-START

As introduced in our main paper, we first split the node points into different sectors for parallelism. In each sector, we employ Algorithm 1 to determine the warm-start dependency for node points and solution propagation order that obeys the dependencies. The warm-start id array $\mathbf{w}$ produced by Algorithm 1 suggests which solution one node point should take to warm-start itself. In this section, we first discuss our method of solution migration to perform warm-start. Experiments in Table 1 show that our warm-start strategy speeds up the from-scratch yarn pattern simulation of all nodes by tenfold.

Solution migration in warm-start. Warm-start essentially involves migrating the solution from one node point to another. Assume the solution vector $\mathbf{u}_{\bar{\mathbf{s}}}$ optimized at node point $\bar{\mathbf{s}}$:

$$\mathbf{u}_{\bar{\mathbf{s}}} = [\mathbf{u}_{\bar{\mathbf{s}}}^0, \tau_{\bar{\mathbf{s}}}^0, \mathbf{u}_{\bar{\mathbf{s}}}^1, \tau_{\bar{\mathbf{s}}}^1, \ldots, \mathbf{u}_{\bar{\mathbf{s}}}^{n-1}] \tag{1}$$

According to the propagation order, we migrate solution $\mathbf{u}_{\bar{\mathbf{s}}}$ to be the initial value $\mathbf{u}_{\mathbf{s}}^{\text{start}}$ of the solution vector for optimization at its neighboring node point $\mathbf{s}$, by keeping the relative displacement w.r.t. the midsurface unchanged between the two node points. The initial solution $\mathbf{u}_{\mathbf{s}}^{\text{start}}$ is defined as follows:

$$\mathbf{u}_{\mathbf{s}}^{\text{start}} = [(\mathbf{R}_{\mathbf{s}}^{0\,T}\mathbf{R}_{\bar{\mathbf{s}}}^0)\mathbf{u}_{\bar{\mathbf{s}}}^0, \tau^0, (\mathbf{R}_{\bar{\mathbf{s}}}^1)\mathbf{u}_{\bar{\mathbf{s}}}^1, \tau^1, \ldots, (\mathbf{R}_{\mathbf{s}}^{n-1\,T}\mathbf{R}_{\bar{\mathbf{s}}}^{n-1})\mathbf{u}_{\bar{\mathbf{s}}}^{n-1}] \tag{2}$$

where $\mathbf{R_s}^i$ is the orthogonal matrix that transforms the normal of the i-th vertex from the rest pose of the mid-surface to the orientation specified by strain $\mathbf{s}$.

In addition to the vertices migration, we also migrate reference vectors $\mathbf{r}^i$ associated with each i-th yarn segment. The reference vector is initially introduced in [Bergou et al. 2008]. Specifically, we migrate the optimized reference vector $\mathbf{r}_{\bar{\mathbf{s}}}^i$ to $\mathbf{r}_{\mathbf{s}}^i$ using spatial parallel transport, detailed in Eq. 3:

$$\mathbf{r}_{\mathbf{s}}^i = \left(\mathbf{r}_{\bar{\mathbf{s}}}^i \cdot \mathbf{t}_{\bar{\mathbf{s}}}^i\right)\mathbf{t}_{\mathbf{s}}^i + \left(\mathbf{r}_{\bar{\mathbf{s}}}^i \cdot \mathbf{b}^i\right)\mathbf{b}^i + \left(\mathbf{r}_{\bar{\mathbf{s}}}^i \cdot \left(\mathbf{b}^i \times \mathbf{t}_{\bar{\mathbf{s}}}^i\right)\right)\left(\mathbf{b}^i \times \mathbf{t}_{\mathbf{s}}^i\right) \tag{3}$$

*The author was also partly affiliated with Style3D Research during this work.
†Corresponding author.

Table 1: Yarn pattern simulation cost with or without our sector-based warm-start. An order of magnitude speedup can be achieved with our sector-based warm-start strategy.

Pattern Type	Basket	Stockinette	Honeycomb	Cartridge
No Warm-Start	1.98h	3.39h	5.81h	7.65h
With Warm-Start	**0.25h**	**0.29h**	**0.76h**	**1.03h**

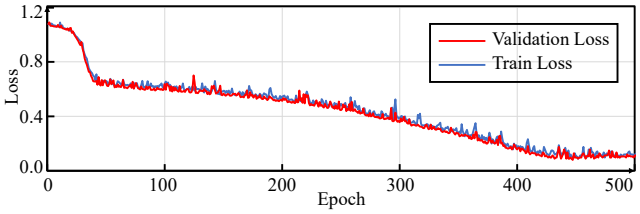
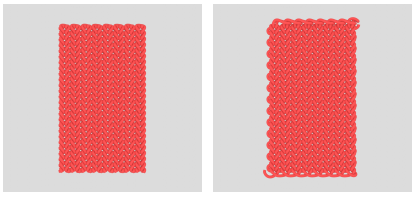


Figure 1: Network train loss plot for stockinette 1D at dof 1.

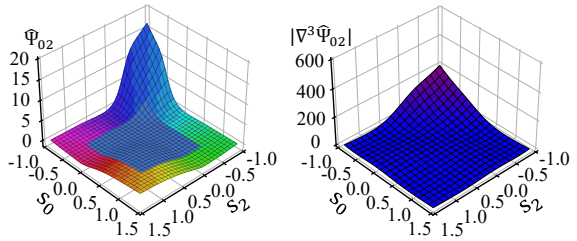
where $\mathbf{t}_{\mathbf{s}}^i, \mathbf{t}_{\bar{\mathbf{s}}}^i$ are unit tangent vectors for i-th yarn segments at node point $\bar{\mathbf{s}}$ and $\mathbf{s}$, $\mathbf{b}^i = \mathbf{t}_{\mathbf{s}}^i \times \mathbf{t}_{\bar{\mathbf{s}}}^i$ is defined as the minus binomial vector. This process ensures that the reference vectors remain consistent with the transformation of the solution vector between node points. We observe the absence of warm starting reference vectors does not affect the homogenization result, but may slow down the yarn pattern simulation.

The Choice of RVE Size. In yarn pattern simulation, the Representative Volume Element (RVE) is a patch of the yarn pattern we simulate. It's crucial to select the RVE size as it affects the accuracy and efficiency of homogenized models. A larger RVE can improve accuracy by better capturing yarn contacts, but it also reduces computational efficiency significantly. To find a balance between accuracy and efficiency, we tested different RVE sizes, as shown in Table 2. The smallest (Size-1) and largest (Size-4) RVE sizes were less accurate. We choose Size-2 for our later experiments, as it offers a good balance between computation cost and accuracy.



(a) Unconnected yarn model. (b) Connected yarn model with boundary.

Figure 2: Yarn-level cloth generation is demonstrated, where (a) shows the unconnected yarn model and (b) the connected yarn model with boundary yarns, following the method by [Yuksel et al. 2012]. The cloth in (b) is composed of a continuous single yarn, replicating real-world cloth characteristics. This model is utilized in subsequent accuracy tests.



(a) Strain energy density $\hat{\Psi}^{02}$. (b) Third-order derivatives of $\hat{\Psi}^{02}$.

Figure 3: Visualization of neural constitutive model $\hat{\Psi}^{02}$ with sector-based safeguard (left) and its cubic-order derivatives (right) of basket pattern. In (a), we visualize the $\hat{\Psi}_{02}$ in the trained region (center blue) and extrapolated values produced by our sector-based safeguard strategy (colored part out of the center blue region). Different sectors have different colors. Our sector-based safeguard policy can produce analytic extrapolation with low cubic-order derivatives, as shown in (b).

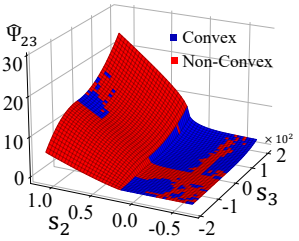


Figure 4: Visualization of convexity for $\hat{\Psi}_{23}$ of stockinette pattern. We observe that there are many non-convex energy density points (red) and convex energy density points (blue).

Table 2: The time cost and mean stress difference in the wale direction for the homogenized model with different RVE sizes. The mean stress difference is explained in Table 5. In the Size-1 experiment, the RVE size is set to 1.8 times the size of a single pattern. In the Size-2, Size-3, and Size-4 experiments, we set the RVE size to 2.2, 2.6, and 3.0 times the single pattern dimension, respectively. We choose Size-2 for our subsequent experiments, as it maintains a good balance between efficiency and accuracy.

Pattern Type	Size-1	**Size-2**	Size-3	Size-4
Basket	0.25h,1.30	**0.35h,1.30**	0.53h,1.30	0.74h,1.56
Stockinette	0.29h,1.17	**0.42h,1.94**	0.59h,1.94	0.88h,2.10
Honeycomb	1.37h,2.13	**2.15h,1.56**	2.96h,1.55	3.80h,1.73
Cartridge	3.39h,1.14	**5.09h,0.34**	7.01h,0.35	9.22h,0.67

Table 3: The validation loss and elapsed training time of neural constitutive models with different architectures trained on datasets of basket yarn pattern sampled for 1D stretch function $\hat{\Psi}_i$ with $i = 0$. The number of neurons in Net-1, Net-2, Net-3, Net-4 are [16,16], [32,32], [16,32,16], [32,64,32] respectively. We choose Net-3 in our method as it achieves the lowest validation loss with moderate training cost.

Network Type	Net-1	Net-2	Net-3	Net-4
Validation Loss	0.232	0.181	**0.148**	0.197
Training Time	83.5s	90.2s	**103.2s**	114.3s

2 DETAILS IN NETWORK TRAINING

We train all neural constitutive models on a PC with a NVIDIA GeForce RTX 3060 GPU and an Intel i9-10850K CPU. The training of all 13 constitutive networks costs about 10 minutes. In our training, We split 90% normalized dataset for training, and the others are left for validation. The initial learning rate in the Adam optimizer is set to 1e-3, and we reduce our learning rate on plateau with factor 0.5, patience 80 and threshold 0.01. We set the batch size to 4 and 128 in 1D and 2D network training. Networks are trained by 500 epochs to convergence. A typical train and validation loss curve is shown in Fig. 1. In our training, the coefficients of loss terms are set to $w^Z = 1$, $w^F = 0.05$, $w^C = 10^{-3}$, $w^S = 10^2$. In each batch, we randomly select 50 strain points outside the training dataset boundary. We use these points in the strain concentration loss function L_I^S, to ensure the network's minimization point remains inside the trained region.

Table 4: The influence of baking on the accuracy of the constitutive model. Therefore, we calculate the #Hausdorff distance/#stress difference with respect to ground truth yarn-level simulation of different baking resolutions. In this experiment, we stretch 5cm×10cm fabrics in wale direction. The resolutions applied during the baking process for 1D neural constitutive models in scenarios from Bake-1 to Bake-3 are 100, 150, and 200, respectively. Additionally, for 2D models, the corresponding baking resolutions are established at [25, 25], [50, 50], and [100, 100].

Pattern Type	Bake-1	Bake-2	Bake-3	Neural
Basket	0.93/0.72	0.87/0.69	**0.87/0.69**	0.87/0.69
Stockinette	4.26/0.82	3.08/0.78	**3.04/0.75**	3.04/0.75
Honeycomb	5.32/1.42	3.63/1.10	**3.54/1.10**	3.54/1.10
Cartridge	7.39/0.93	6.12/0.81	**6.02/0.74**	6.02/0.74

Choice of Network Architecture. To ascertain the most effective and compact architecture, we train various networks and compare their validation loss to assess their performance and efficiency. As

ALGORITHM 1: Sector-based Node Traversal Algorithm.

Input: Node list $\mathbf{s}[N]$ in a single sector. // N is the number of node points in a sector.
Output: Solution propagation order list $\mathbf{O}[N]$, Warm-start id array $\mathbf{w}[N]$.
for all index i in range(0, N) **do**
  $\mathbf{w}[i] \leftarrow -2$ // Uninitialized flag = -2
end for
Normalize node list $\mathbf{s}$ to unit hypersquare.
for all index i in range(0, N) **do**
  if $\|\mathbf{s}^i\| = 0$ **then**
    //the node point at the origin
    $\mathbf{w}[i] \leftarrow -1$
  else
    $\mathbf{w}[i] \leftarrow \underset{j\in[0,N-1], j\neq i}{\operatorname{argmin}} \{\|\mathbf{s}^i - \mathbf{s}^j\| \mid \|\mathbf{s}^j\| < \|\mathbf{s}^i\|\}$
  end if
end for
// Breadth-first Search from the node point at the origin
$\mathbf{O} \leftarrow \mathrm{BFS}(\mathbf{w}, \|\mathbf{s}^i\| = 0)$

in Table. 3, The Net-3 network with [16, 32, 16] fully-connected layers outperforms the others, proving to be our best choice balancing performance and efficiency.

Dicussions about Energy Density Convexity. We observe that the homogenized energy density function does not exhibit global convexity, as shown in Fig. 4. A convexity loss during training leads to bad accuracy in the constitutive model, as the homogenized energy density function is not convex.

3 DETAILS IN BAKING AND SAFEGUARDING

We bake neural constitutive models to analytic Hermite functions to avoid costly runtime inferencing. However, during baking, it should take performance and accuracy into consideration simultaneously. In Table 4, we present a detailed comparison of the accuracy of our proposed neural constitutive model, both pre- and after-baking. Based on these test results, we choose the Bake-3 type for our implementation. As a result, we bake 1D and 2D neural models at resolutions of 200 and [100, 100], respectively. This decision is grounded in Bake-3's demonstrated capability to preserve an accuracy level comparable to the original neural constitutive models, as shown in Table. 4.

Safeguarding of Constitutive Model. In Fig. 3 we demonstrate the strain energy density function with sector-based safeguarding. Our proposed sector-based safeguard can keep smooth and relatively low cubic-order derivatives for extreme strains, as shown in Fig. 3(b).

4 DETAILS IN EVALUATIONS

4.1 Details in Quadratic Expansion Error

For a given knit pattern, we define its quadratic expansion error E, as the mean of five 1D and nine 2D networks' Ψ_I quadratic expansion errors E_I, where I is the unified indices as defined in Sec. 5 of the main paper.

$$E = \frac{1}{5}\sum_{I\in\{0,1,2,3,5\}} E_I + \frac{1}{9}\sum_{\substack{I\in\{(0,1),(0,2),(0,3),(0,5),\\(1,2),(1,3),(1,5),(2,3),(2,5)\}}} E_I \tag{4}$$

For a neural network Ψ_I, we calculate its quadratic expansion error as $E_I = \frac{1}{N_I}\sum_{i=0}^{N_I} D_I(\mathbf{s}_i^d)$. Here, N_I represents the number of sampling points in the dataset, and $D_I(\mathbf{s}_i^d)$ calculates the expansion error around the local point $\mathbf{s}_i^d$, defined as:

$$D_I(\mathbf{s}_i^d) = \int_{\Omega_I(\mathbf{s}_i^d)} \Big| \Psi_I(\mathbf{s}) - \Big[\Psi_I(\mathbf{s}_i^d) + \nabla\Psi_I(\mathbf{s}_i^d)(\mathbf{s}-\mathbf{s}_i^d) + \frac{1}{2}(\mathbf{s}-\mathbf{s}_i^d)^T\nabla^2\Psi_I(\mathbf{s}_i^d)(\mathbf{s}-\mathbf{s}_i^d)\Big]\Big| d\mathbf{s}, \tag{5}$$

We set $\Omega_I(\mathbf{s}_i)$ to be a square with its width equal to one-hundredth of the dataset range accordingly. We uniformly distribute 1024 quadruple samples in $\Omega_I(\mathbf{s}_i)$ when calculating the integration in Eq. 5.

4.2 Details in Accuracy Test

In order to justify the accuracy of our method, we conduct uniaxial tests to compare the mechanical response for yarn-level simulation and continuum-based simulation. More specifically, we stretch 5cm×10cm fabric strips in the course and wale directions, and compare the Hausdorff distance and stress difference as metrics to evaluate the homogenization accuracy. In this experiment, we employ the boundary connection method from [Yuksel et al. 2012] to connect tiled periodic yarn patterns (Fig. 2(a)) and create yarn-level fabric strips (Fig. 2(b)) in the following accuracy tests. Otherwise, the yarn will get loose and produce invalid simulation results in uni-axial stretch. As our homogenization method doesn't take boundary yarns into account, in order to produce a fair accuracy test, we eliminate the effects of boundary yarn segments by adjusting their Young's modulus, bending and twisting moduli to be 1/200 of the raw physical parameters of other yarns.

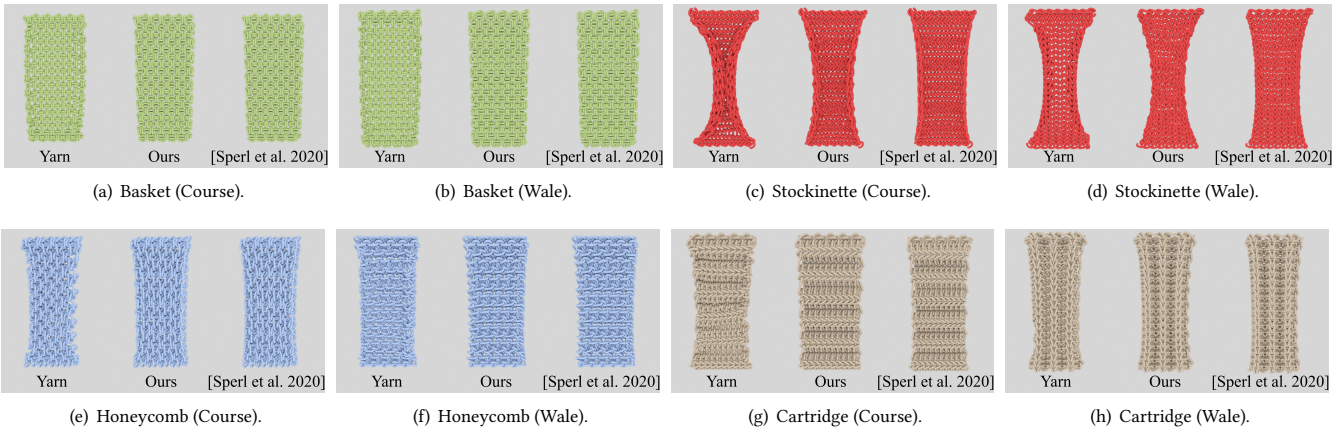
Fig. 5 shows the fabric strips with 130% stretch ratio, simulated in the yarn-level cloth simulator and continuum-based simulator. The accuracy comparison between our method and [Sperl et al. 2020] is presented in Table. 5.

REFERENCES

Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. 2008. Discrete Elastic Rods. In *ACM SIGGRAPH 2008 Papers* (Los Angeles, California) *(SIGGRAPH '08)*. Article 63, 12 pages.

Georg Sperl, Rahul Narain, and Chris Wojtan. 2020. Homogenized Yarn-Level Cloth. *ACM Trans. Graph. (SIGGRAPH)* 39, 4, Article 48 (July 2020), 16 pages.

Cem Yuksel, Jonathan M Kaldor, Doug L James, and Steve Marschner. 2012. Stitch meshes for modeling knitted clothing with yarn-level detail. *ACM Trans. Graph.* 31, 4 (2012), 1–12.



(a) Basket (Course). (b) Basket (Wale). (c) Stockinette (Course). (d) Stockinette (Wale).

(e) Honeycomb (Course). (f) Honeycomb (Wale). (g) Cartridge (Course). (h) Cartridge (Wale).

Figure 5: 5x10 cm fabric stretch comparison.

Table 5: Accuracy test results for four yarn patterns stretched with different ratios in course and wale direction. In each cell, the first number represents the difference of stress between yarn-level simulation results and continuum-based simulation results in $N \cdot m^{-1}$. The second number represents the Hausdorff distance which measures the geometric similarity between yarn-level simulation results and continuum-based simulation results with unit cm. The last column shows mean absolute stress error, mean Hausdorff distance and mean relative stress error. These mean values are calculated through 15 uni-axial tests with a stretch ratio ranging from 100% to 170% (with a 5% gap between test samples), as shown in the last column.

Stretch Ratio		110%	120%	130%	140%	150%	160%	170%	Mean
Basket(Course)	Ours	0.04, 0.62	0.20, 0.69	0.43, 0.52	2.63, 0.54	2.59, 0.61	2.14, 0.69	1.07, 0.77	**1.30**, 0.63, **5.41%**
	[Sperl et al. 2020]	0.23, 0.62	0.80, 0.68	1.42, 0.53	1.67, 0.54	1.96, 0.61	2.10, 0.69	2.49, 0.77	1.52, **0.63**, 6.73%
Basket(Wale)	Ours	0.69, 0.69	0.81, 0.72	1.62, 0.56	1.38, 0.58	0.10, 0.68	0.09, 0.78	1.42, 0.87	0.87, **0.69**, **3.8%**
	[Sperl et al. 2020]	0.23, 0.72	0.53, 0.74	0.65, 0.56	0.81, 0.58	0.97, 0.68	1.11, 0.78	1.43, 0.87	**0.82**, 0.70, 4.6%
Stockinette(Course)	Ours	0.56, 0.65	2.69, 0.90	4.76, 1.16	3.22, 1.01	1.50, 1.00	0.51, 1.06	0.37, 1.15	**1.94**, **0.99**, **7.3%**
	[Sperl et al. 2020]	0.20, 0.71	1.83, 1.12	3.31, 1.32	4.07, 1.45	2.94, 1.53	0.74, 1.35	2.72, 1.18	2.26, 1.24, 14.9%
Stockinette(Wale)	Ours	2.11, 0.81	3.32, 0.68	1.60, 0.69	2.10, 0.66	1.23, 0.72	1.80, 0.81	4.80, 0.86	**3.04**, **0.75**, **13.7%**
	[Sperl et al. 2020]	1.19, 0.78	4.02, 0.67	4.89, 1.01	4.62, 1.14	3.84, 1.13	4.53, 1.20	3.75, 1.21	3.83, 1.02, 20.1%
Honeycomb(Course)	Ours	0.57, 0.82	0.90, 0.81	0.94, 0.75	0.69, 0.85	1.35, 0.86	2.71, 0.85	3.75, 0.81	1.56, 0.82, 14.3%
	[Sperl et al. 2020]	0.64, 0.88	1.17, 0.81	0.77, 0.77	0.79, 0.72	1.02, 0.67	1.73, 0.82	4.25, 0.84	**1.48**, **0.79**, **13.8%**
Honeycomb(Wale)	Ours	00.05, 1.14	0.40, 1.18	0.45, 1.17	0.91, 1.08	4.36, 1.04	8.20, 1.10	10.39, 1.00	**3.54**, **1.10**, **25.7%**
	[Sperl et al. 2020]	1.23, 1.13	0.25, 1.19	0.09, 1.17	2.61, 1.24	5.72, 1.30	8.51, 1.37	11.83, 1.40	4.32, 1.26, 27.3%
Cartridge(Course)	Ours	0.02, 0.80	0.09, 0.81	0.42, 0.81	0.23, 0.83	0.20, 0.82	0.57, 0.82	0.82, 0.85	**0.34**, **0.82**, **12.2%**
	[Sperl et al. 2020]	0.08, 0.79	0.16, 0.80	0.16, 0.81	0.32, 0.83	0.69, 0.82	0.04, 0.85	1.42, 0.92	0.41, 0.83, 15.6%
Cartridge(Wale)	Ours	1.16, 0.75	0.13, 0.75	0.25, 0.73	4.76, 0.73	11.34, 0.75	13.92, 0.73	10.61, 0.73	**6.03**, **0.74**, **11.3%**
	[Sperl et al. 2020]	0.84, 0.74	0.86, 0.75	1.08, 0.73	6.68, 0.74	13.43, 0.81	15.85, 0.88	12.86, 0.97	7.37, 0.80, 15.8%