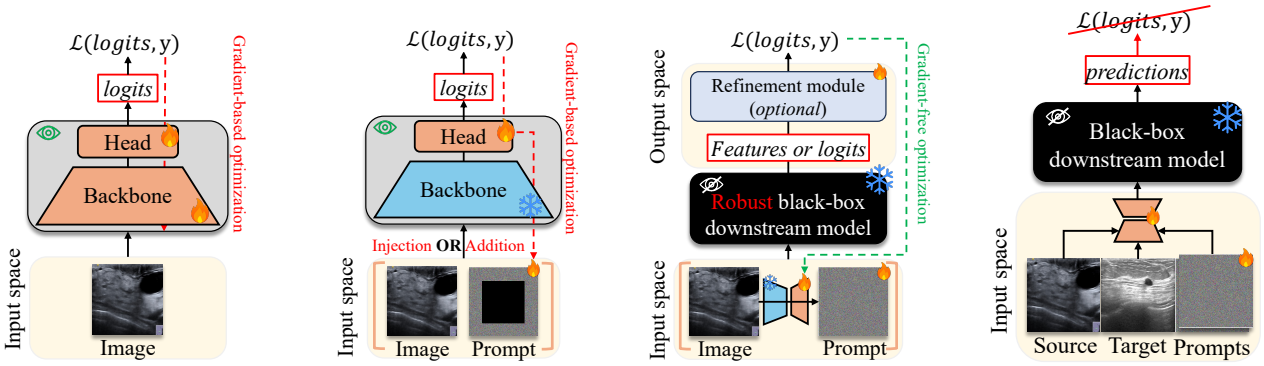
UI-Styler: Ultrasound Image Style Transfer with Class-Aware Prompts for Cross-Device Diagnosis Using a Frozen Black-Box Inference Network

Supplementary Material



(a) Full white-box setting. (b) Prompt white-box setting. (c) Black-box setting. (d) Inference-blackbox setting.

Figure 1. **Prompt Setting Comparison.** We illustrate four prompt-based training and deployment scenarios with increasing constraints: (a) *Full white-box setting* allows end-to-end fine-tuning via backpropagation over the entire model using ground-truth labels. (b) *Prompt white-box setting* injects learnable prompts into the input while freezing the backbone, but still requires gradients and supervision. (c) *Black-box setting* removes gradient access but assumes availability of intermediate features or logits for prompt tuning or refinement. (d) *Inference-blackbox setting* reflects the most realistic and constrained scenario, where only final predictions are available.

Overview

We organize the supplementary content into nine sections. Sec. A introduces key notations, and Sec. B provides the pseudo-code of UI-Styler. Sec. C compares full fine-tuning and prompt-tuning paradigms under different levels of model access, while Sec. D details the content and style losses. Sec. E reports black-box downstream performance on target domains. Sec. F presents additional experiments on loss contributions, weight configurations, and pattern-matching sensitivity. Sec. G further analyzes diagnostic semantic preservation and t-SNE failure cases. Sec. H discusses scalability, generalization, and robustness to noisy pseudo labels. Finally, Sec. I provides qualitative results across all 12 cross-device tasks.

Contents

A. Notation

We summarize the notations and their corresponding definitions frequently used in our method in Tab. 1.

B. Pseudo Code

We provide the pseudo code of **UI-Styler** in Algorithm 1, which outlines the core procedures for training and testing.

C. Problem Setting Comparison

In this section, we categorize and compare four increasingly constrained training and deployment scenarios, ranging from full fine-tuning in white-box settings to prompt tuning under inference-blackbox conditions. Each setting imposes distinct assumptions on parameter accessibility, label availability, and interaction scope, as summarized in Fig. 1. We highlight the practical limitations of existing methods in real-world deployment scenarios, thus motivating our inference-blackbox prompt tuning.

Symbol	Description
Abbreviations	
BDM	Black-box downstream model
PT	Prompt tuning
PDA	Prompt-based domain adaptation
UIT	Unpaired image translation
PM	Pattern-matching mechanism (domain-level adaptation)
CP	Class-aware prompting (class-level alignment)
ViT	Vision transformer
Data Setting	
$\mathcal{D}_s$	Unlabeled source domain
$\mathcal{D}_t$	Unlabeled target domain
x_s, x_t	Source and target images
$\hat{y}_t$	Pseudo target label
$\hat{\mathbf{y}}_t$	One-hot encoding of the pseudo target label
C	Number of classes
$H \times W$	Input image size (256×256)
UI-Styler Architecture	
P	Patch size (set to 8)
h, w	Patch grid size, $h = H/P$, $w = W/P$
L	Number of image tokens ($L = h \times w$)
d	Embedding dimension of each token
E_s, E_t	source and target encoders
W_q	Projection matrix for query from source features
W_k, W_v	Projection matrices for key and value from target features
$\mathcal{E}_f(\cdot)$, $\mathcal{E}_p(\cdot)$	Feature and prompt embedders
$H(\cdot)$	A classifier head
D	Decoder to reconstruct stylized images
$\widetilde{x}_s$	Stylized image
Features & Representations	
$\mathcal{F}_s$, $\mathcal{F}_t$	Extracted features from source and target images
Q	Query, projected from $\mathcal{F}_s$ using W_q
K, V	Key and Value, projected from $\mathcal{F}_t$ using W_k, W_v
$\widetilde{\mathcal{F}}_{s\to t}$	Stylized features (after domain-level alignment)
$\widetilde{\mathcal{F}}^{+}_{s\to t}$	Final stylized features (after class-aware prompting)
$\mathcal{P}$	Learnable template prompts
$\mathcal{P}_c$	Class-specific prompts
$\hat{\mathcal{P}}_c$	Supervised prompts derived from the pseudo target label
Loss Functions	
$\mathbf{a}$	Class–prompt correlation vector
$\mathbf{p}_t$	Probabilities from classifier head $H(\mathcal{F}_t + \hat{\mathcal{P}}_c)$
$\mathcal{L}_c$	Content loss (structure/content preservation)
$\mathcal{L}_s$	Style loss (appearance/style alignment)
$\mathcal{L}_{\text{dir}}$	Direction loss for prompt selection
$\mathcal{L}_{\text{sup}}$	Supervised loss for prompt supervision
$\mathcal{L}_{\text{total}}$	Overall training objective
Evaluation Metrics	
KID ↓	Kernel Inception Distance
Acc ↑	Classification accuracy
AUC ↑	Area under ROC curve
Dice ↑	Dice score
IoU ↑	Intersection over Union

Table 1. Summary of notations used throughout the paper.

Algorithm 1 The pseudo code of UI-Styler

Problem Setting (Sec. 3.1):
❒ Data Setting:
- The unlabeled source dataset $\mathcal{D}_s = \{x_s^i\}_{i=1}^{N_s}$.
- The pseudo-labeled target dataset $\mathcal{D}_t = \{(x_t^j, \hat{y}_t^j)\}_{j=1}^{N_t}$.

Note: Unpaired source and target data, $\mathcal{D}_s \cap \mathcal{D}_t = \emptyset$.

❒ Black-box Downstream Model: classification network: $C(\cdot)$ and segmentation network: $S(\cdot)$.

UI-Styler Architecture (Sec. 3.2):
❒ Feature Extractors: a source encoder $E_s(\cdot;\boldsymbol{\theta}_{E_s})$ and a target encoder $E_t(\cdot;\boldsymbol{\theta}_{E_t})$.
❒ Dual-level Stylization:
- Pattern-matching Mechanism:

$\text{PM}(c, s; \boldsymbol{\theta}_{PM}) = \{W_q(c;\boldsymbol{\theta}_{W_q}), W_k(s;\boldsymbol{\theta}_{W_k}), W_v(s;\boldsymbol{\theta}_{W_v})\}$.
- Class-aware Prompting:

$\text{CP}(\cdot,\cdot;\boldsymbol{\theta}_{CP}) = \{\mathcal{P}(\boldsymbol{\theta}_{\mathcal{P}}), \mathcal{E}_f(\cdot;\boldsymbol{\theta}_{\mathcal{E}_f}), \mathcal{E}_p(\cdot;\boldsymbol{\theta}_{\mathcal{E}_p}), H(\cdot;\boldsymbol{\theta}_H)\}$.

❒ Decoder: $D(\cdot;\boldsymbol{\theta}_D)$.
Note: The UI-Styler parameters: $\theta = \{\boldsymbol{\theta}_{E_s}, \boldsymbol{\theta}_{E_t}, \boldsymbol{\theta}_{PM}, \boldsymbol{\theta}_{CP}, \boldsymbol{\theta}_D\}$ is initialized using Xavier and optimized with Adam using learning rate η.

Training Strategy:
for $i \leftarrow 1$ to I **do**
  ✔ Feature Extraction (Sec. 3.2):
  $\mathcal{F}_s = E_s(x_s^i), \quad \mathcal{F}_t = E_t(x_t^i)$,
  ✔ Dual-level Stylization (Sec. 3.3):
  ✐ 1. Domain-level adaptation
  # Stylized Features
  $\widetilde{\mathcal{F}}_{s\to t} = \text{PM}(\mathcal{F}_s, \mathcal{F}_t)$, ▷ Eqs. 1, 2.
  ✐ 2. Class-level adaptation
  # Class-specific Prompts
  $\mathcal{P}_c = \text{one-hot-max}\left(\mathcal{E}_f(\widetilde{\mathcal{F}}_{s\to t})\mathcal{E}_p(\mathcal{P})^\top\right)\mathcal{P}$, ▷ Eq. 3.
  # Class-aligned Features
  $\widetilde{\mathcal{F}}^{+}_{s\to t} = \widetilde{\mathcal{F}}_{s\to t} + \mathcal{P}_c$, ▷ Eq. 4.
  ✔ Reconstruction (Sec. 3.2):
  $\widetilde{x}_s = D(\widetilde{\mathcal{F}}^{+}_{s\to t})$,
  ➠ **Final Objective Function** (Sec. 3.4):
  # Direction Loss
  $\mathbf{a} = \text{sigmoid}(\mathcal{E}_f(\mathcal{F}_t)\cdot\mathcal{E}_p(\mathcal{P})^\top) \in \mathbb{R}^C$,
  $\mathcal{L}_{\text{dir}} = -\frac{1}{C}\sum_{c=1}^{C}[\hat{y}_c \log a_c + (1-\hat{y}_c)\log(1-a_c)]$, ▷ Eq. 5.
  # Supervised Loss
  $\hat{\mathcal{P}}_c = \hat{\mathbf{y}}_t \cdot \mathcal{P} \in \mathbb{R}^{L\times d}$,
  $\mathcal{L}_{\text{sup}} = -\hat{\mathbf{y}}_t \cdot \log(\mathbf{p}_t)$,
  where $\mathbf{p}_t = \text{softmax}(H(\mathcal{F}_t + \hat{\mathcal{P}}_c))$ ▷ Eq. 6.
  # Backpropagation
  $\mathcal{L}_{\text{total}} = \lambda_{\text{dir}}\mathcal{L}_{\text{dir}} + \lambda_{\text{sup}}\mathcal{L}_{\text{sup}} + \lambda_c\mathcal{L}_c + \lambda_s\mathcal{L}_s$,
  $\theta \leftarrow \theta - \eta\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{total}}$.
end for

Testing:
❒ Style Transfer: $\widetilde{x}_s = \text{UI-Styler}(x_s, x_t)$,
❒ Reused Black-box Downstream Model:
- Predicted Class: $\hat{y}_{s\to t} = C(\widetilde{x}_s)$,
- Predicted Mask: $\hat{M}_{s\to t} = S(\widetilde{x}_s)$.

C.1. Full Fine-Tuning in White-box Setting

As shown in Fig. 1a, full fine-tuning (FT) enables end-to-end optimization of both the backbone and task-specific head using supervised loss $\mathcal{L}(logits, y)$, where y is the ground truth. Despite achieving strong task-specific performance [10, 21], FT demands full access to model parameters and gradients, making it infeasible in proprietary or privacy-sensitive deployments. Moreover, it incurs high computational overhead and risks of overfitting or catastrophic forgetting under distribution shifts.

C.2. Prompt Tuning in White-box Setting

Prompt tuning (PT) alleviates the limitations of FT by inserting learnable prompts into the input space while freezing the backbone [2, 9]. As shown in Fig. 1b, this strategy greatly reduces trainable parameters and improves efficiency [8]. It has been shown to enhance model interpretability and fine-grained recognition via class-specific prompts [5]. However, PT still assumes white-box access to model parameters and requires supervision, making it unsuitable in label-scarce or black-box environments.

C.3. Prompt Tuning in Black-box Setting

To overcome gradient restrictions, recent methods introduce *gradient-free prompt tuning* for black-box models. As illustrated in Fig. 1c, BlackVIP [13] and BAPs [14] optimize prompts directly in the input space to manipulate downstream outputs for classification and segmentation via zeroth-order optimization [13]. CraFT [20] extends this by combining input prompts (optimized via CMA-ES) and a refinement module (trained via gradients on logits).

To reduce reliance on labels, VDPG [3] and L2C [4] propose learning *domain prompt generators*, trained with gradients from a refinement module, to adapt black-box features without ground-truth supervision. However, these methods assume: (1) access to features or logits; (2) pre-trained robust black-box downstream models (e.g., CLIP [16]); and (3) in the case of VDPG and L2C, multiple source domains for domain-generalizable prompt generation. These assumptions are impractical in real-world, privacy-constrained environments such as healthcare.

C.4. Prompt Tuning in Inference-blackbox Setting

The inference-blackbox setting, illustrated in Fig. 1d, is the most restrictive scenario, where only the final predictions, **including image class IDs and segmentation masks (optional)**, are provided from the black-box downstream model. **NO** gradients, intermediate features, logits, and model parameters are accessible—conditions often encountered in real-world healthcare deployments.

To address this challenge, we propose **UI-Styler**, a prompt tuning framework designed explicitly for the inference-blackbox regime. Unlike previous approaches that still require supervision or logits [13, 20], UI-Styler leverages unpaired target samples and pseudo labels to drive adaptation via class-aware prompts. Our method operates entirely in the input space and applies a dual-level stylization strategy, aligning source images with the target domain in both appearance and semantics.

D. Detailed Content and Style Losses

Following style transfer works [6, 15, 23], we adopt perceptual losses computed from a pre-trained VGG-19 network to guide structural preservation and appearance alignment.

Content Loss. The content loss $\mathcal{L}_c$ measures the ℓ_2 distance between the feature representations of the stylized image $\widetilde{x}_s$ and the original source image x_s, extracted from two higher-level layers of VGG-19:

$$\mathcal{L}_c = \left\|\phi^{4,1}(\widetilde{x}_s) - \phi^{4,1}(x_s)\right\|_2^2 + \left\|\phi^{5,1}(\widetilde{x}_s) - \phi^{5,1}(x_s)\right\|_2^2, \tag{1}$$

where $\phi^{l,1}(\cdot)$ denotes the activation from the first convolutional layer after the l-th ReLU block.

Style Loss. To capture multi-scale stylistic characteristics, we define the style loss $\mathcal{L}_s$ using the mean and standard deviation statistics of VGG features from multiple layers:

$$\begin{aligned}\mathcal{L}_s = \sum_{l=2}^{5} \big(&\left\|\mu(\phi^{l,1}(\widetilde{x}_s)) - \mu(\phi^{l,1}(x_t))\right\|_2^2 \\ &+ \left\|\sigma(\phi^{l,1}(\widetilde{x}_s)) - \sigma(\phi^{l,1}(x_t))\right\|_2^2\big),\end{aligned} \tag{2}$$

where $\mu(\cdot)$ and $\sigma(\cdot)$ represent the mean and standard deviation of the extracted features, respectively.

E. Downstream Performance on Target Domains

To provide reference results, we report the performance of the black-box downstream model when directly evaluated on each target domain with the 30% **testing** set.

As listed in Tab. 2, the black-box model delivers strong performance on all target domains, with accuracy above

Target Domains	Acc↑	AUC↑	Dice↑	IoU↑
BUSBRA [7]	89.17	94.71	90.99	84.16
BUSI [1]	92.82	96.09	86.63	78.53
UCLM [18]	93.75	97.63	88.28	80.31
UDIAT [22]	91.84	97.65	90.51	83.29

Table 2. **Downstream Performance on Target Domains.** We report the performance of the black-box downstream models on each domain for reference. The results are evaluated on the 30% **testing** set. The high classification/segmentation performance indicates that these black-box downstream models are reliable enough to deploy clinical diagnosis applications.

$\mathcal{L}_{\text{dir}}$	$\mathcal{L}_{\text{sup}}$	Tasks	KID↓	Acc↑	AUC↑	Dice↑	IoU↑	Tasks	KID↓	Acc↑	AUC↑	Dice↑	IoU↑	Tasks	KID↓	Acc↑	AUC↑	Dice↑	IoU↑
–	✓	BUSBRA	11.73	73.89	75.06	83.80	73.89	BUSBRA	17.23	74.96	76.49	81.28	70.88	BUSBRA	12.11	65.90	68.83	85.82	76.94
✓	–	↓	12.66	75.13	75.77	84.47	**74.80**	↓	17.63	74.25	76.10	81.74	71.24	↓	12.71	69.09	70.84	85.83	76.87
✓	✓	BUSI	**11.20**	**75.84**	**76.33**	**84.52**	74.74	UCLM	**16.91**	**75.13**	**76.78**	**82.06**	**71.73**	UDIAT	**9.14**	**72.47**	**71.52**	**86.04**	**77.52**
–	✓	BUSI	**10.50**	84.10	87.12	83.01	73.97	BUSI	12.43	70.77	74.91	78.30	68.31	BUSI	4.39	**74.36**	75.31	80.30	71.21
✓	–	↓	12.74	84.62	87.22	83.04	73.97	↓	11.25	71.79	76.13	78.13	68.40	↓	3.78	73.85	77.74	80.19	71.27
✓	✓	BUSBRA	11.25	**85.13**	**88.14**	**83.15**	**74.05**	UCLM	**11.05**	**74.36**	**77.15**	**78.83**	**68.61**	UDIAT	**3.61**	**74.36**	**78.89**	**80.49**	**71.61**
–	✓	UCLM	10.22	87.50	92.49	81.71	71.60	UCLM	13.13	78.75	83.43	78.82	68.69	UCLM	15.76	62.50	65.58	82.97	72.91
✓	–	↓	12.91	83.75	91.01	82.07	71.71	↓	13.85	76.25	81.95	79.67	69.40	↓	14.91	65.00	70.18	83.02	73.19
✓	✓	BUSBRA	**9.60**	**88.75**	**94.93**	**82.79**	**72.65**	BUSI	**12.40**	**80.00**	**85.60**	**80.22**	**69.78**	UDIAT	**13.56**	**71.25**	**73.36**	**83.16**	**73.27**
–	✓	UDIAT	5.70	83.67	76.07	88.32	79.99	UDIAT	4.73	89.80	93.80	83.36	73.89	UDIAT	16.02	83.67	85.47	85.72	76.21
✓	–	↓	6.71	81.63	77.35	88.38	80.12	↓	4.59	89.80	92.95	83.92	74.52	↓	13.03	81.63	81.84	85.32	75.87
✓	✓	BUSBRA	**5.25**	**87.76**	**79.27**	**88.45**	**80.13**	BUSI	**4.47**	**91.84**	**96.15**	**85.39**	**76.09**	UCLM	**12.33**	**85.71**	**88.25**	**85.83**	**76.46**

Table 3. **Ablation Study on Loss Contributions.** We evaluate the impact of $\mathcal{L}_{\text{dir}}$ and $\mathcal{L}_{\text{sup}}$ in the final objective across 12 cross-device ultrasound tasks. Each result is reported under 5 metrics: KID, Acc, AUC, Dice, and IoU. **Bold** denotes the best result, and underline indicates the second-best.

89% and AUC consistently exceeding 94%. Segmentation results are also reliable, as Dice scores remain above 86% and IoU above 78% across all cases. These results confirm that the black-box downstream model can serve to evaluate unpaired image translation methods in cross-domain tasks. Furthermore, its reliable performance suggests suitability for deploying clinical diagnosis applications.

F. Additional Experiments

F.1. Ablation Study on Loss Contributions

Since the content loss ($\mathcal{L}_c$) and style loss ($\mathcal{L}_s$) are standard components in style transfer frameworks, we focus on evaluating the additional contributions of the proposed direction loss ($\mathcal{L}_{\text{dir}}$) and supervised loss ($\mathcal{L}_{\text{sup}}$), as reported in Tab. 3. Specifically, we find that using only $\mathcal{L}_{\text{sup}}$—*without the explicit guidance from* $\mathcal{L}_{dir}$—often causes the stylized features ($\widetilde{\mathcal{F}}_{s\to t}$) to be matched with *incorrect* class-specific prompts ($\mathcal{P}_c$). From Tab. 3, we observe that the accuracy drops **drastically** from 71.25 (full setting) to 62.50 in the UCLM→UDIAT task.

Moreover, when using only $\mathcal{L}_{\text{dir}}$—*without the supervision from* $\mathcal{L}_{\text{sup}}$—the prompts lack supervision from the target domain and thus fail to learn class-specific characteristics. As a result, in the UDIAT→BUSI task, the Dice score declines from 85.39 to 83.92, and the AUC drops from 96.15 to 92.95.

Consequently, the superior performance achieved with the full setting of $\mathcal{L}_{\text{dir}}$ and $\mathcal{L}_{\text{sup}}$ provides strong evidence that the stylized features ($\widetilde{\mathcal{F}}_{s\to t}$) are effectively aligned with the correct class while preserving diagnostic traits.

F.2. Loss Weight Configurations

We investigate different combinations of loss functions across 12 cross-device tasks. Since the content loss ($\mathcal{L}_c$) and style loss ($\mathcal{L}_s$) are the baseline objectives in the style transfer process, we divide the study into **two main groups** (G) with distinct optimization goals: **(1)** *style transfer*, where $\mathcal{L}_c$ and $\mathcal{L}_s$ are computed to guide the transformation $\left(I^{s-style}_{s-content}, I^{t-style}_{t-content}\right) \to I^{t-style}_{s-content}$; and **(2)** *prompt learning*, where the direction loss $\mathcal{L}_{\text{dir}}$ and the supervised loss $\mathcal{L}_{\text{sup}}$ are used to optimize the template prompt set $\mathcal{P}$. For each group, we assess three pairwise settings—$(1,1)$, $(2,1)$, and $(1,2)$—with the averaged results in Tab. 4.

	λ_c	λ_s	λ_{dir}	λ_{sup}	KID↓	Acc↑	AUC↑	Dice↑	IoU↑
G(1)	2	1	1	1	12.38	77.71	80.53	82.90	73.36
G(1)	1	2	1	1	**8.75**	78.20	80.75	82.77	73.12
G(2)	1	1	2	1	10.62	79.71	80.20	82.96	73.44
G(2)	1	1	1	2	10.40	78.12	80.65	82.97	73.44
	1	1	1	1	10.06	**80.22**	**82.20**	**83.41**	**73.89**

Table 4. **Loss Weight Configurations.** We report the *averaged results* of different loss weight configurations over 12 cross-device tasks under 5 metrics: KID, Acc, AUC, Dice, and IoU. **Bold** denotes the best result, and underline indicates the second-best. *The per-task results are reported in Tab. 5.*

For the G(**1**), we find that increasing $\mathcal{L}_c$ tends to overshadow $\mathcal{L}_s$, resulting in insufficient transfer of the target style, especially when the domain gap is large. Conversely, increasing $\mathcal{L}_s$ may over-stylize the content information, causing content degradation. Therefore, balancing content and style information proves essential, yielding improvements across all metrics. In the G(**2**), we observe that balancing $\mathcal{L}_{\text{dir}}$ and $\mathcal{L}_{\text{sup}}$ yields consistently higher Acc, AUC, Dice, and IoU compared to overwhelming-weight settings. This trend can be further explained by examining the effect of unbalanced weights: when $\mathcal{L}_{\text{dir}}$ dominates, prompt learning leans toward directional alignment but lacks pseudo target label guidance, reducing discriminability. Conversely, increasing $\mathcal{L}_{\text{sup}}$, the supervision from pseudo target labels overshadows the correlation-alignment

λ_c	λ_s	λ_{dir}	λ_{sup}	Tasks	KID↓	Acc↑	AUC↑	Dice↑	IoU↑	Tasks	KID↓	Acc↑	AUC↑	Dice↑	IoU↑	Tasks	KID↓	Acc↑	AUC↑	Dice↑	IoU↑
2	1	1	1		15.45	75.31	74.67	84.41	74.69		16.30	**75.13**	75.76	82.15	71.74		13.90	68.21	70.73	85.97	77.03
1	2	1	1	BUSBRA	**8.46**	73.00	75.69	83.76	73.89	BUSBRA	**13.39**	74.60	76.61	82.16	71.84	BUSBRA	**8.87**	67.14	68.68	85.87	76.93
1	1	2	1	↓	13.06	75.49	75.55	84.43	74.71	↓	15.05	74.96	**76.98**	82.29	71.90	↓	12.48	69.09	71.13	85.83	76.85
1	1	1	2	BUSI	13.25	74.25	74.28	84.25	74.46	UCLM	16.71	74.42	76.30	**82.41**	**72.09**	UDIAT	12.52	67.50	70.67	85.80	76.85
1	1	1	1		11.20	**75.84**	**76.33**	**84.52**	**74.74**		16.91	**75.13**	76.78	82.06	71.73		9.14	**72.47**	**71.52**	**86.04**	**77.52**
2	1	1	1		10.89	84.62	88.09	**83.27**	**74.17**		12.52	70.77	75.67	77.93	68.20		4.29	73.85	76.96	80.40	71.41
1	2	1	1	BUSI	**5.52**	82.56	86.22	83.08	73.84	BUSI	**10.84**	**75.90**	**78.07**	78.34	**68.62**	BUSI	**3.43**	74.36	76.05	79.77	70.68
1	1	2	1	↓	7.61	**85.13**	87.16	82.86	73.89	↓	10.92	72.82	75.87	77.96	68.29	↓	4.45	**75.38**	76.49	80.35	71.40
1	1	1	2	BUSBRA	7.46	**85.13**	88.05	82.74	73.57	UCLM	11.98	74.36	75.91	78.12	68.55	UDIAT	3.69	73.85	78.46	80.19	71.18
1	1	1	1		11.25	**85.13**	**88.14**	83.15	74.05		11.05	74.36	77.15	**78.83**	68.61		3.61	74.36	**78.89**	**80.49**	**71.61**
2	1	1	1		15.02	86.25	93.37	81.67	71.70		14.85	75.00	83.77	78.63	68.31		16.17	66.25	72.62	82.80	72.82
1	2	1	1	UCLM	**8.98**	85.00	93.31	81.73	71.65	UCLM	**11.84**	**80.00**	84.18	78.40	67.95	UCLM	15.09	68.75	70.39	82.76	72.64
1	1	2	1	↓	11.82	**90.00**	93.31	82.73	72.32	↓	13.57	77.50	82.76	79.32	69.20	↓	14.20	68.75	71.26	82.62	72.75
1	1	1	2	BUSBRA	12.02	85.00	91.55	**83.01**	**72.75**	BUSI	12.27	76.25	82.35	78.82	68.59	UDIAT	**13.13**	67.50	**73.83**	82.76	72.77
1	1	1	1		9.60	88.75	**94.93**	82.79	72.65		12.40	**80.00**	**85.60**	**80.22**	**69.78**		13.56	**71.25**	73.36	**83.16**	**73.27**
2	1	1	1		7.07	83.67	**79.49**	88.19	79.84		4.27	89.80	92.52	83.95	74.37		17.78	83.67	82.69	85.42	75.98
1	2	1	1	UDIAT	**3.13**	83.67	77.78	88.04	79.56	UDIAT	**3.13**	89.80	94.02	84.14	74.26	UDIAT	12.32	83.67	88.03	85.14	75.63
1	1	2	1	↓	5.62	85.71	76.71	87.94	79.63	↓	4.16	**93.88**	90.81	83.57	74.15	↓	14.55	**87.76**	84.40	85.57	76.15
1	1	1	2	BUSBRA	5.93	83.67	78.21	88.37	**80.13**	BUSI	4.35	91.84	92.95	83.89	74.49	UCLM	**11.47**	83.67	85.26	85.31	75.86
1	1	1	1		5.25	**87.76**	79.27	**88.45**	**80.13**		4.47	91.84	**96.15**	**85.39**	**76.09**		12.33	85.71	**88.25**	**85.83**	**76.46**

Table 5. **Loss Weight Configurations.** We report the per-task performance of different loss weight configurations across 12 cross-device tasks, evaluated under 5 metrics: KID, Acc, AUC, Dice, and IoU. **Bold** denotes the best result, and underline indicates the second-best.

effect of $\mathcal{L}_{\text{dir}}$, thereby limiting the selection of suitable class-specific prompts, $\mathcal{P}_c$.

Based on these findings, the balanced loss weighting provides the most reliable performance, achieving $4/5$ best metrics, including Acc of 80.22, AUC of 82.20, Dice of 83.41, and IoU of 73.89. *For a comprehensive comparison, we provide the per-task results in Tab. 5.*

F.3. Sensitivity of Pattern-matching Parameters

We analyze the sensitivity of our pattern-matching module with respect to the number of ViT blocks as shown in Tab. 6, which reports the averaged results over 12 cross-device tasks. The floating-point operations (FLOPs) are measured with an input image size of 256×256. We observe that the configuration with 3 ViT blocks achieves the best overall trade-off, obtaining the lowest KID (10.06) and highest Acc (80.22). Specifically, compared to 5 blocks, the performance gap is marginal (only 0.37 in AUC and 0.16 in Dice), while the FLOPs are reduced from 64.30G to 55.70G. More importantly, compared to the 2-block setting, 3 blocks show a substantial improvement of 2.48% in Acc (from 77.74 to 80.22) and consistent gains across other metrics.

These results indicate that using 3 ViT blocks provides the most efficient balance between computational cost and performance. Hence, we adopt 3 blocks as the default configuration of the pattern-matching module. *For comprehensive comparison, we also provide the per-task performance in Tab. 7.*

#Blocks	KID↓	Acc↑	AUC↑	Dice↑	IoU↑	FLOPs↓
2	10.07	77.74	79.89	82.85	73.30	**51.40G**
3	**10.06**	**80.22**	82.20	83.41	73.89	55.70G
5	10.61	80.21	**82.57**	**83.57**	**73.97**	64.30G

Table 6. **Sensitivity of Pattern-matching Parameters.** We present the *average performance* of different numbers of ViT blocks in the pattern-matching module across 12 cross-device tasks, evaluated on 5 metrics (KID, Acc, AUC, Dice, IoU) and computational cost (FLOPs). **Bold** denotes the best result, and underline indicates the second-best. *The per-task results are reported in Tab. 7.*

G. Additional Analyses

G.1. Comparison on Diagnostic Semantics

To demonstrate the capability of UI-Styler in preserving diagnostic semantics, we conduct a qualitative comparison of stylized results produced by unpaired image translation methods. Each comparison is performed on the same source image from BUSBRA with target-style counterparts from BUSI, UCLM, and UDIAT. According to the medical ultrasound literature [11, 12, 17], the tumor region is a critical feature for accurate diagnosis.

As shown in Fig. 2, previous methods often produce **inconsistencies** in tumor areas (highlighted by red boxes □), as they mainly operate at the domain level, which imposes the target style onto the source content. As a result, different target devices can yield varying outcomes even for the same source image. In contrast, UI-Styler consistently preserves tumor regions across all tasks, providing strong evidence of its ability to maintain diagnostic semantics when incorpo-

#Blocks	Tasks	KID↓	Acc↑	AUC↑	Dice↑	IoU↑	Tasks	KID↓	Acc↑	AUC↑	Dice↑	IoU↑	Tasks	KID↓	Acc↑	AUC↑	Dice↑	IoU↑
2	BUSBRA	12.17	74.78	74.33	83.92	74.05	BUSBRA	**14.40**	74.07	77.27	82.08	71.68	BUSBRA	11.67	66.61	68.56	85.78	76.83
3	↓	**11.20**	75.84	76.33	**84.52**	**74.74**	↓	16.91	75.13	76.78	82.06	71.73	↓	**9.14**	**72.47**	71.52	86.04	77.52
5	BUSI	11.87	**76.55**	**77.40**	84.24	74.46	UCLM	15.19	**77.62**	**78.30**	**82.29**	**71.98**	UDIAT	13.61	69.45	**72.52**	**86.83**	**77.80**
2	BUSI	7.04	83.59	86.33	83.14	74.03	BUSI	**10.67**	73.85	75.76	77.80	68.08	BUSI	4.12	74.36	77.63	80.21	71.09
3	↓	11.25	**85.13**	88.14	83.15	74.05	↓	11.05	**74.36**	77.15	**78.83**	**68.61**	↓	**3.61**	74.36	**78.89**	**80.49**	71.61
5	BUSBRA	**6.43**	84.62	**89.17**	**83.20**	**74.26**	UCLM	11.02	**74.36**	**79.20**	78.07	68.35	UDIAT	4.29	**76.41**	78.62	**80.49**	**72.48**
2	UCLM	12.24	86.25	93.44	82.54	72.64	UCLM	12.82	77.50	82.08	78.35	67.91	UCLM	**12.94**	68.75	71.33	82.61	72.65
3	↓	**9.60**	**88.75**	**94.93**	82.79	72.65	↓	**12.40**	**80.00**	85.60	**80.22**	**69.78**	↓	13.56	71.25	73.36	83.16	**73.27**
5	BUSBRA	13.45	**88.75**	94.46	**83.05**	**72.86**	BUSI	12.57	**80.00**	**85.73**	79.71	69.28	UDIAT	13.20	**71.50**	**74.92**	**83.96**	73.06
2	UDIAT	7.26	83.67	77.99	88.73	80.58	UDIAT	**3.78**	89.80	90.38	84.37	74.99	UDIAT	**11.69**	79.59	83.55	84.64	75.08
3	↓	**5.25**	**87.76**	**79.27**	88.45	80.13	↓	4.47	**91.84**	**96.15**	85.39	**76.09**	↓	12.33	**85.71**	88.25	85.83	76.46
5	BUSBRA	6.56	85.71	77.99	**88.96**	**80.78**	BUSI	4.21	**91.84**	94.02	**85.94**	75.51	UCLM	14.91	**85.71**	**88.48**	**86.11**	**76.81**

Table 7. **Sensitivity of Pattern-matching Parameters.** We report the per-task performance of different numbers of ViT blocks in the pattern-matching module across 12 cross-device ultrasound tasks, under 5 metrics: KID, Acc, AUC, Dice, and IoU. **Bold** denotes the best result, and underline indicates the second-best.

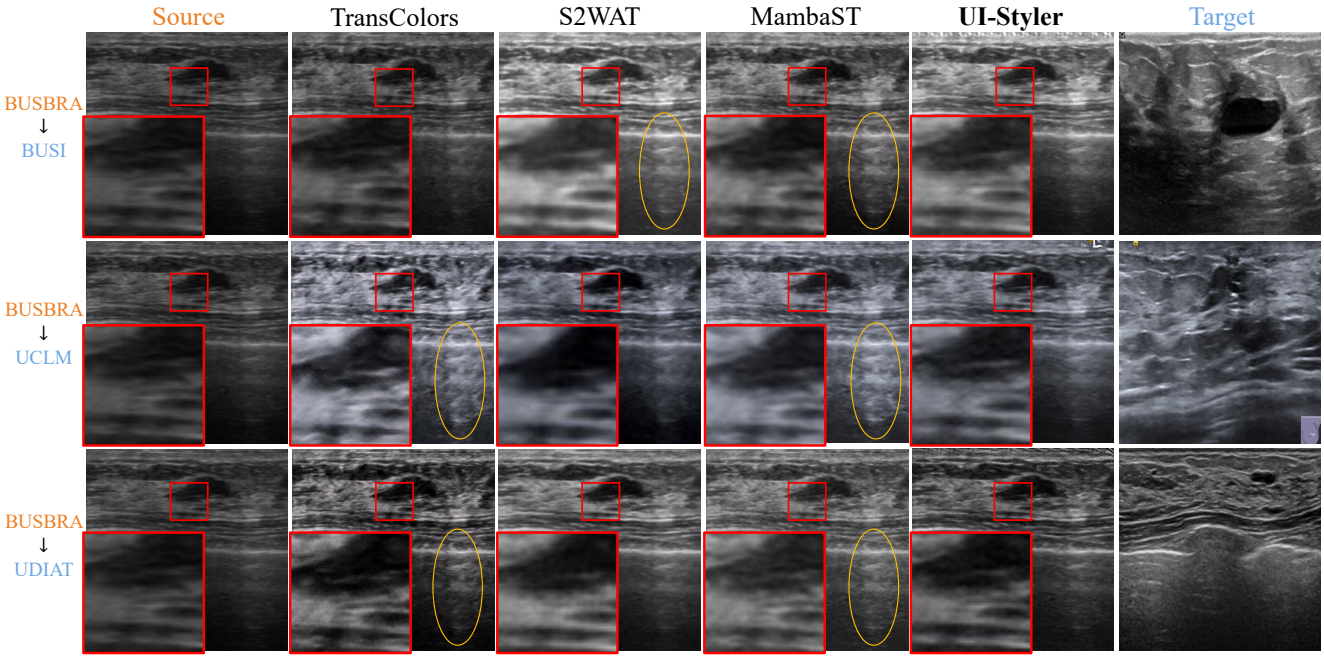


Figure 2. **Comparison on Diagnostic Semantics.** We show stylized outputs from unpaired image translation methods, where each row displays the results generated from the same source-content image alongside target-style counterparts. Red boxes □ indicate zoomed tumor regions, while yellow ellipses ⬭ highlight artifact areas where competing methods fail to preserve diagnostic semantics. *Please zoom in to view details more easily*.

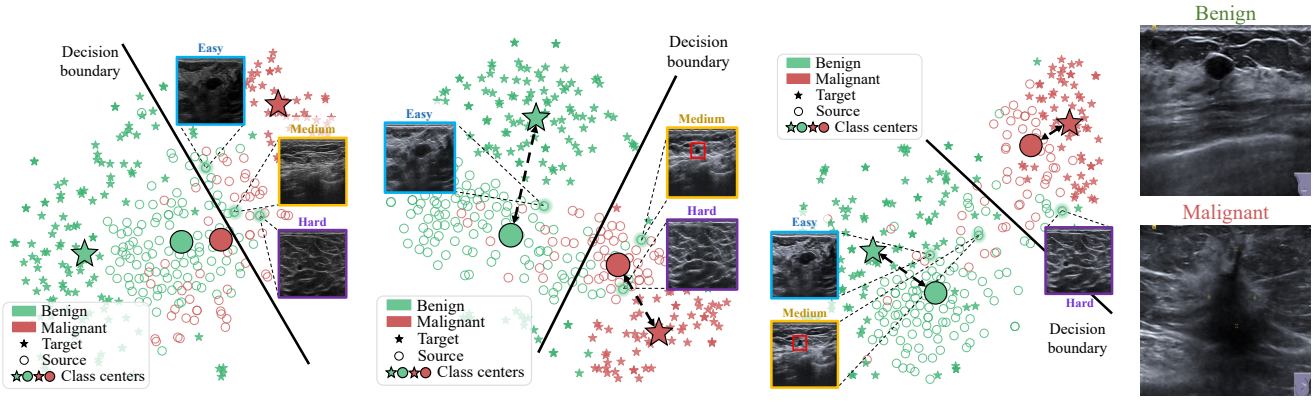
rating class-aware transfer.

Furthermore, competing approaches tend to generate undesired artifacts (marked by yellow ellipses ⬭), whereas UI-Styler remains unaffected.

G.2. Failure Case Analysis

We analyze failure cases within the feature space of the black-box downstream model using t-SNE [19], categorizing them into three cases—easy, medium, and hard—as shown in Fig. 3. For clarity, we further examine them under three settings:

1. Setting 1 (**S1**): We denote the *before style transfer* setting as no style transfer applied. As shown in Fig. 3a, the source and target domains remain misaligned.
2. Setting 2 (**S2**): We introduce our pattern-matching module to alleviate the domain gap. We refer to this configuration as *only domain level*, since the alignment focuses solely on transferring domain-specific appearance, as shown in Fig. 3b.



(a) Before style transfer. (b) Only domain level. (c) Domain and class levels. (d) Target samples.

Figure 3. **Failure Case Analysis.** We illustrate the t-SNE [19] feature space of the black-box downstream model on the UDIAT→UCLM task. The analysis is presented under three settings: (a) before style transfer, (b) with domain-level alignment only, and (c) with both domain- and class-level alignment. We illustrate three failure cases: easy, medium, and hard, using the same samples across settings. The easy case is misclassified only before style transfer, the medium case remains misclassified after domain-level alignment, and the hard case persists under all settings. Meanwhile, by comparing the same sample across different settings, we show the progressive influence of style transfer under different settings. *Please zoom in for better visibility.*

3. Setting 3 (**S3**): Finally, we simultaneously minimize both domain-level and class-level discrepancies through our proposed dual-level stylization module. This configuration is referred to as *domain and class levels*, as shown in Fig. 3c.

In the easy case, the source sample (blue-bordered image) is initially misclassified in **S1**. In **S2**, the same sample successfully matches the appearance of the target data (see more Fig. 3d for the comparison), leading to a correct classification. Furthermore, this alignment continues improvements with **S3**, the sample moves further from the decision boundary, providing more robust predictions.

However, when we consider the medium case (example by the orange-bordered image), **S2** is insufficient to preserve class-discriminative properties (e.g., the tumor region highlighted in red-square □ of Fig. 3b), leading to ambiguous class confusion. In contrast, with **S3**, the benign-specific characteristics are preserved (see the red-square □ in Fig. 3c), which effectively drives the misclassified sample toward the correct class.

More critically, we observe the hard case (shown by the purple-bordered image), where the sample exhibits inherent differences in structure and tissue characteristics compared with the target data. As a result, even with **S3**, we still encounter a misclassification for this specific sample.

H. Discussion

H.1. Can UI-Styler Achieve Scalability and Generalization?

Scalability. To demonstrate the scalability of UI-Styler in real-world deployments with multiple source domains, we explore two training strategies:

1. *Single*-source setting: the model is trained on one source domain (either BUSBRA or BUSI) and evaluated on the corresponding source→UDIAT task.
2. *Multi*-source setting: the model is trained jointly on (BUSBRA+BUSI)→UDIAT and then evaluated on both source→UDIAT tasks within a unified model, which alleviates the need for training $N \times (N-1)$ separate models as required by the *single*-source setting, where N denotes the number of devices.

As shown in the seen part of Tab. 8, *multi*-source training achieves performance comparable to *single*-source training, with only a small gap (e.g., BUSBRA→UDIAT AUC 71.52 vs. 71.31 and BUSI→UDIAT Dice 80.49 vs. 80.39), while consistently outperforming the baseline without style transfer (w/o ST).

Generalization. We further evaluate the generalization ability of UI-Styler by selecting BUSBRA and BUSI as the seen source domains, UCLM as the unseen source domain, and keeping UDIAT as the fixed target.

1. *Single*-source setting: the model is trained on BUSBRA→UDIAT and then evaluated on UCLM→UDIAT.
2. *Multi*-source setting: the model is trained jointly on (BUSBRA+BUSI)→UDIAT and evaluated on UCLM→UDIAT.

As shown in the unseen part of Tab. 8, the single-source model already achieves solid performance, while the multi-source setting further improves results across multiple metrics, with Acc increasing from 65.00 to 67.50 and AUC from 70.32 to 72.62. These findings provide strong evidence of UI-Styler's effectiveness in adapting to new, un-

	Tasks	Settings	KID↓	Acc↑	AUC↑	Dice↑	IoU↑
Seen	BUSBRA ↓ UDIAT	w/o ST	13.81	55.95	64.29	84.76	75.71
		Single	**9.14**	**72.47**	**71.52**	**86.04**	**77.52**
		Multi	12.24	68.74	71.31	85.83	76.93
	BUSI ↓ UDIAT	w/o ST	7.23	73.33	73.16	79.53	70.61
		Single	**3.61**	74.36	**78.89**	**80.49**	**71.61**
		Multi	4.00	**75.38**	78.43	80.39	71.34
Unseen	UCLM ↓ UDIAT	w/o ST	20.90	63.75	68.15	82.22	72.06
		Single	10.84	65.00	70.32	**82.71**	**72.64**
		Multi	**9.67**	**67.50**	**72.62**	82.66	72.57

Table 8. **Can UI-Styler Achieve Scalability and Generalization?** We assess scalability and generalization with BUSBRA and BUSI as the seen source domains, UCLM as the unseen source domain, and UDIAT as the fixed target. In the seen setting, models are trained *and evaluated* on the corresponding source→UDIAT tasks (single: one source; multi: BUSBRA+BUSI). In the unseen setting, models are trained on BUSBRA→UDIAT (single) or (BUSBRA+BUSI)→UDIAT (multi) and evaluated on UCLM→UDIAT. w/o ST denotes training without style transfer.

seen devices in practical scenarios.

H.2. How Noisy Pseudo Target Labels Affect Performance?

Since pseudo target labels are generated by a black-box downstream model, *label noise is an inevitable factor in realistic deployments*. To investigate the robustness of UI-Styler against noisy labels, we conduct experiments on the BUSI→BUSBRA task by progressively injecting noise from 0% to 40% into the target domain. Specifically, we randomly replaced the ground truths with incorrect classes.

As shown in Tab. 9, we observe that introducing a mild noise level of 10% keeps the results almost unchanged compared to the clean setting (0%). Even higher noise levels (20–30%) lead to only **marginal** degradation across most metrics (e.g., AUC drops only slightly to 87.87 and 87.61), while all metrics continue to surpass the baseline without style transfer (w/o ST). These findings indicate that UI-Styler can tolerate moderate noise levels without noticeable performance loss. Only at 40% noise, we observe a more visible decline, with AUC reduced to 86.77 and Dice to 82.39, yet UI-Styler still surpasses the w/o ST baseline on 3/5 metrics (KID, Acc, and IoU).

These findings suggest that although UI-Styler does not incorporate any explicit noise-mitigation module, its design exhibits a certain degree of robustness to label noise. We acknowledge that heavy noise can accumulate errors through the proposed losses ($\mathcal{L}_{\text{dir}}$ and $\mathcal{L}_{\text{sup}}$), which may limit reliability in extreme cases. Nonetheless, the **stability under low-to-moderate noise** demonstrates that UI-Styler can operate effectively in realistic settings where the black-box downstream model achieves at least 70% accuracy.

Task	Noisy Levels	KID↓	Acc↑	AUC↑	Dice↑	IoU↑
BUSI ↓ BUSBRA	w/o ST	19.73	82.56	87.30	82.41	73.37
	0%	11.25	**85.13**	**88.14**	**83.15**	**74.05**
	10%	11.20	**85.13**	87.93	82.92	73.97
	20%	**11.14**	84.10	87.87	82.70	73.70
	30%	11.19	83.59	87.61	82.68	73.67
	40%	11.26	83.08	86.77	82.39	73.45

Table 9. **How Noisy Pseudo Target Labels Affect Performance?** We report results on the BUSI→BUSBRA task under different noise levels (0%, 10%, 20%, 30%, and 40%), where noise is introduced by randomly replacing ground truths with incorrect class assignments. Even with 40% noisy labels, UI-Styler still surpasses the baseline without style transfer (w/o ST) on 3/5 metrics (KID, Acc, and IoU).

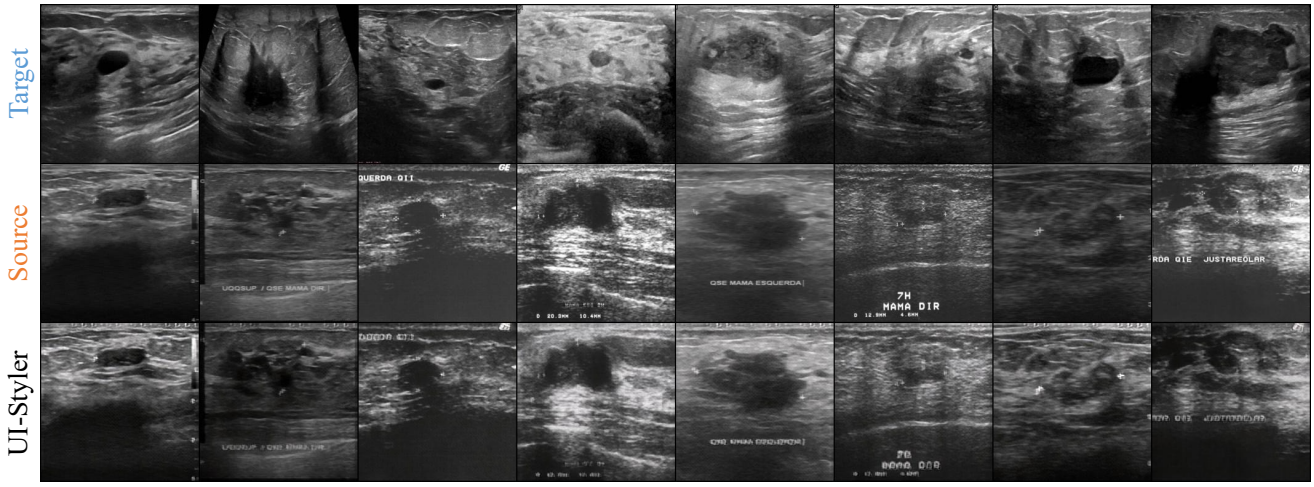
Obviously, black-box downstream models *must* achieve accuracy well above 70% to be meaningful in medical applications. Models falling below this accuracy level are essentially random in outcome and often biased toward a single class. Consequently, their predictions are unsafe for diagnosis and provide clinicians with no reliable basis for decision-making.

I. Cross-device Visual Results

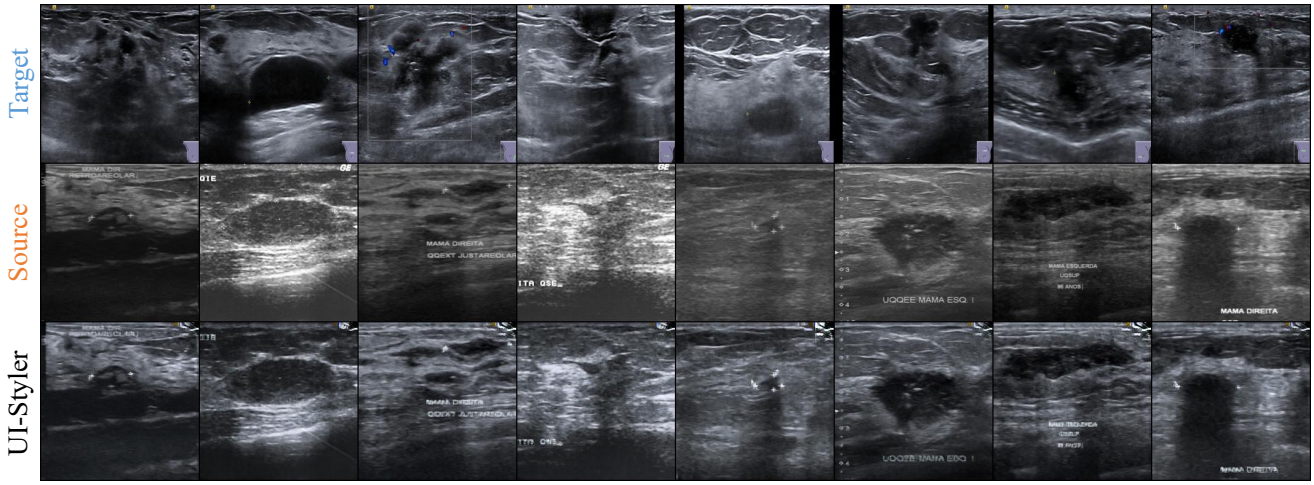
To further assess the effectiveness of the proposed UI-Styler, we present visual results for all 12 source-to-target transfer tasks, alongside representative examples that highlight the unique appearance characteristics of each ultrasound dataset, as shown in Fig. 4. Each subfigure corresponds to a specific domain adaptation scenario, where the top row shows target domain samples, the middle row displays source domain inputs, and the bottom row presents the stylized outputs produced by UI-Styler.

Visually, UI-Styler consistently adapts the source image style to match the target domain while preserving tumor structure and lesion boundaries. The translated images demonstrate improved textural consistency and contrast characteristics aligned with the target domain, including probe artifacts, intensity ranges, and noise profiles. Notably, the stylized outputs retain key diagnostic features essential for downstream classification and segmentation tasks.

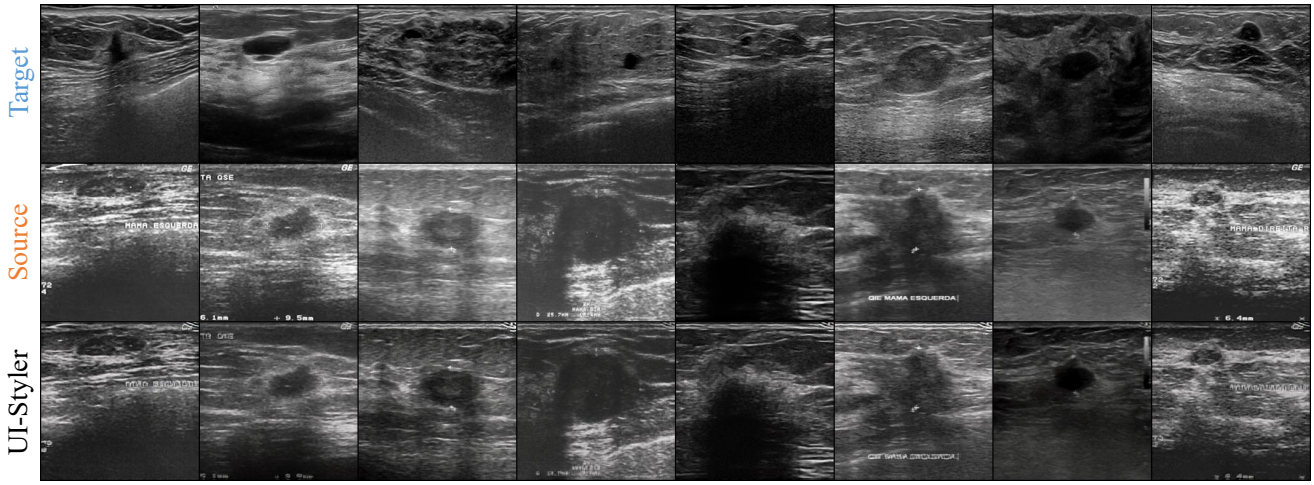
Beyond enhancing model performance, this visual consistency also supports clinical interpretation. By translating unfamiliar input styles into the target domain's appearance, UI-Styler facilitates diagnostic reasoning for physicians, especially when deploying models trained on known devices to new acquisition environments. This alignment reduces adaptation burden and promotes safe model deployment in device-diverse clinical settings.



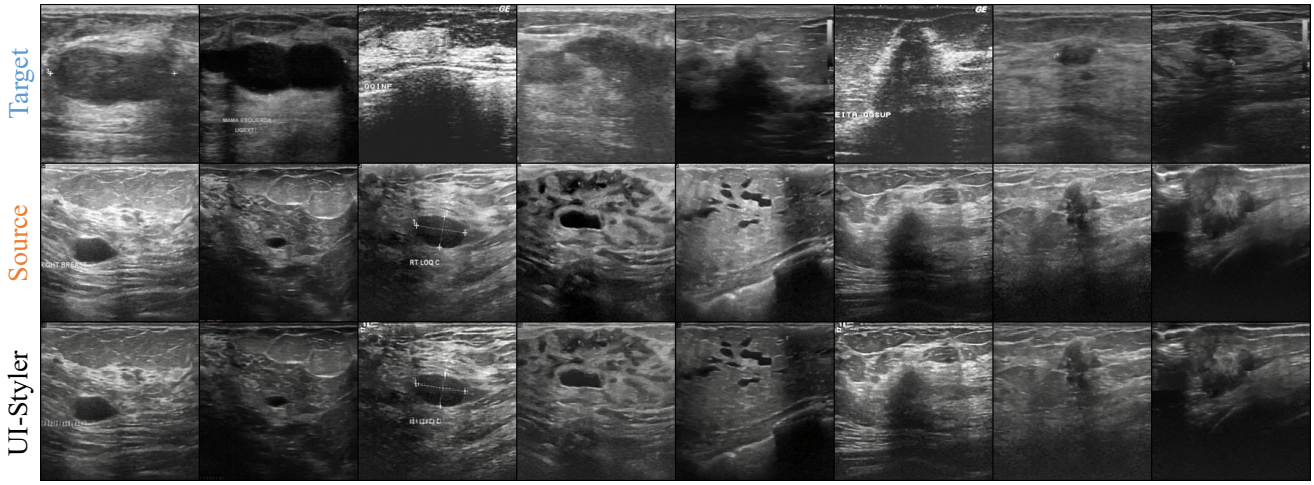
(a) BUSBRA→BUSI.



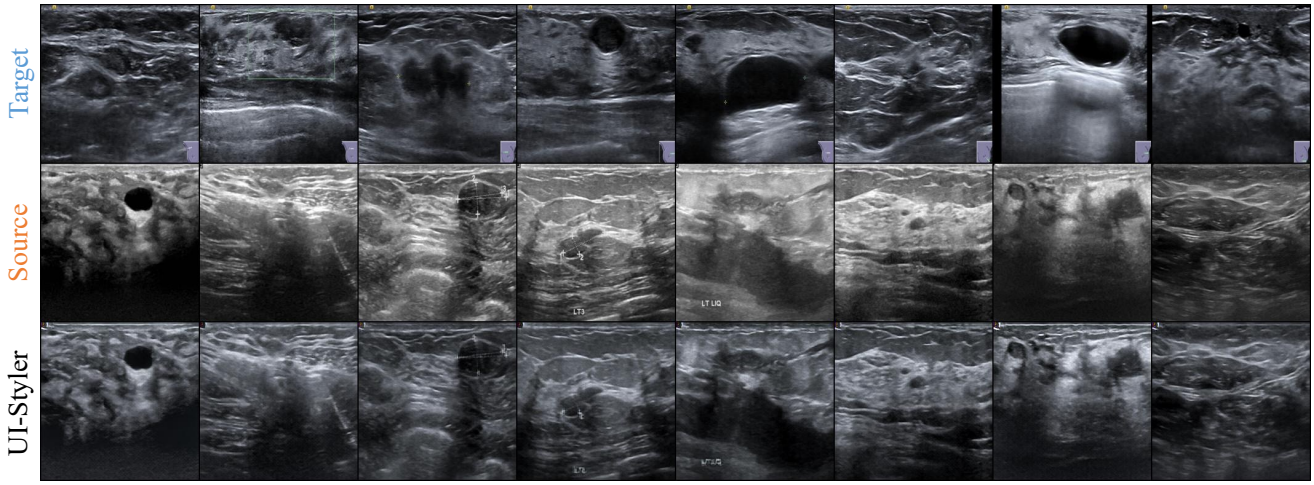
(b) BUSBRA→UCLM.



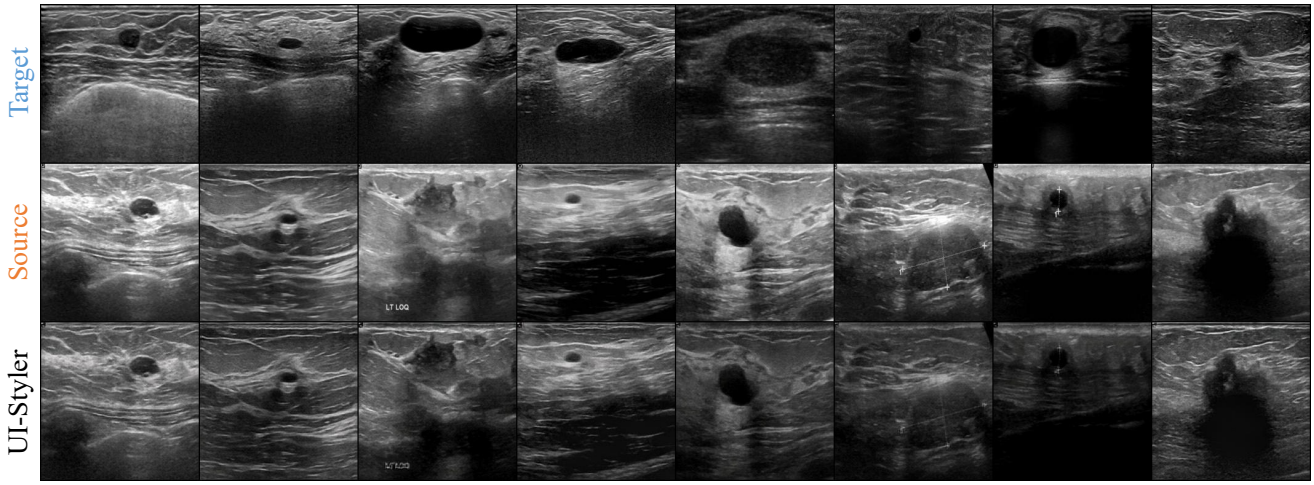
(c) BUSBRA→UDIAT.



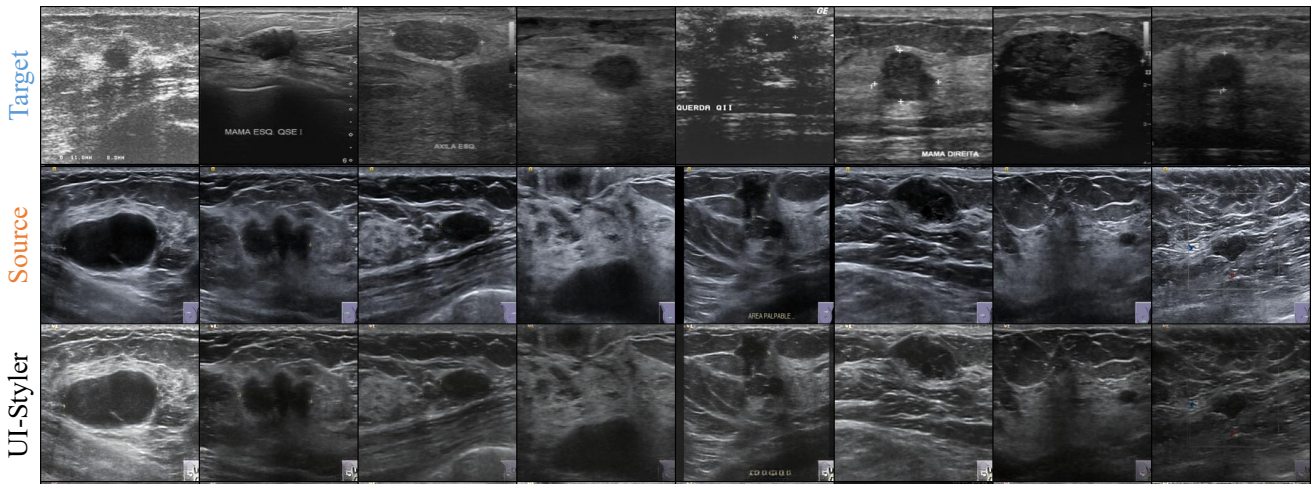
(d) BUSI→BUSBRA.



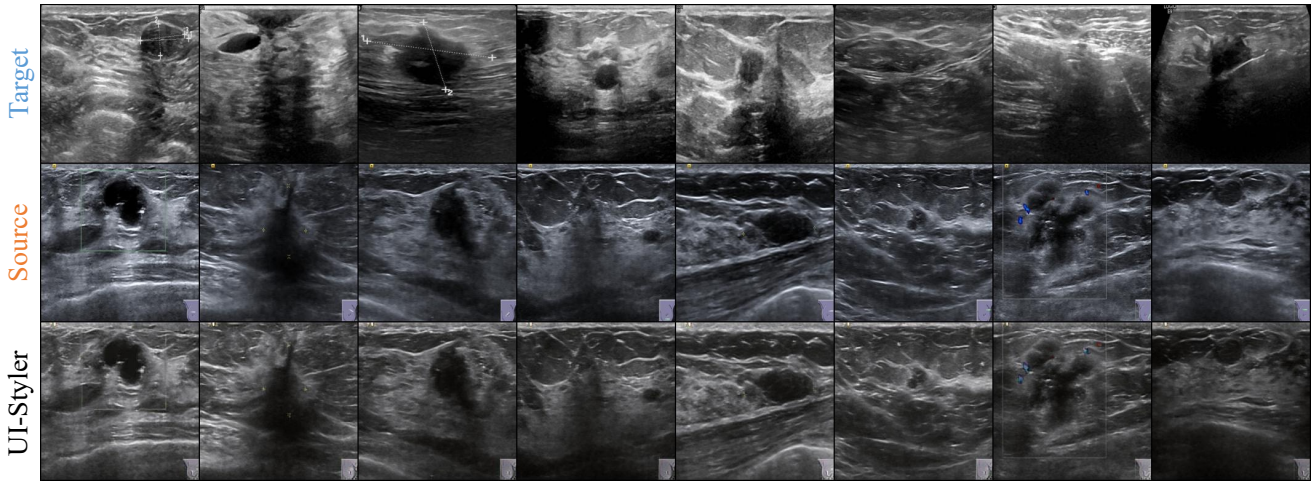
(e) BUSI→UCLM.



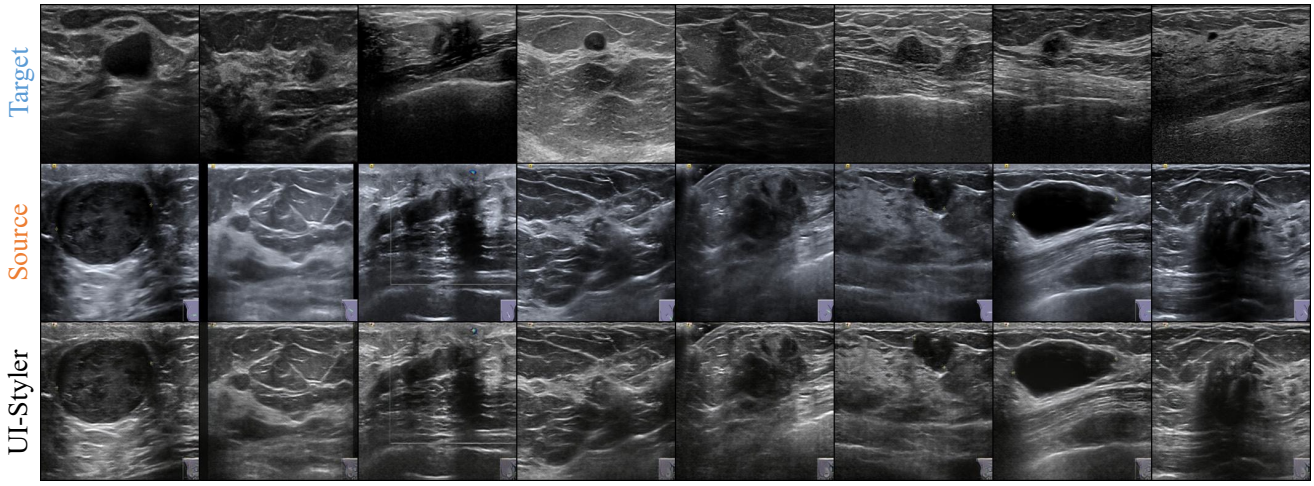
(f) BUSI→UDIAT.



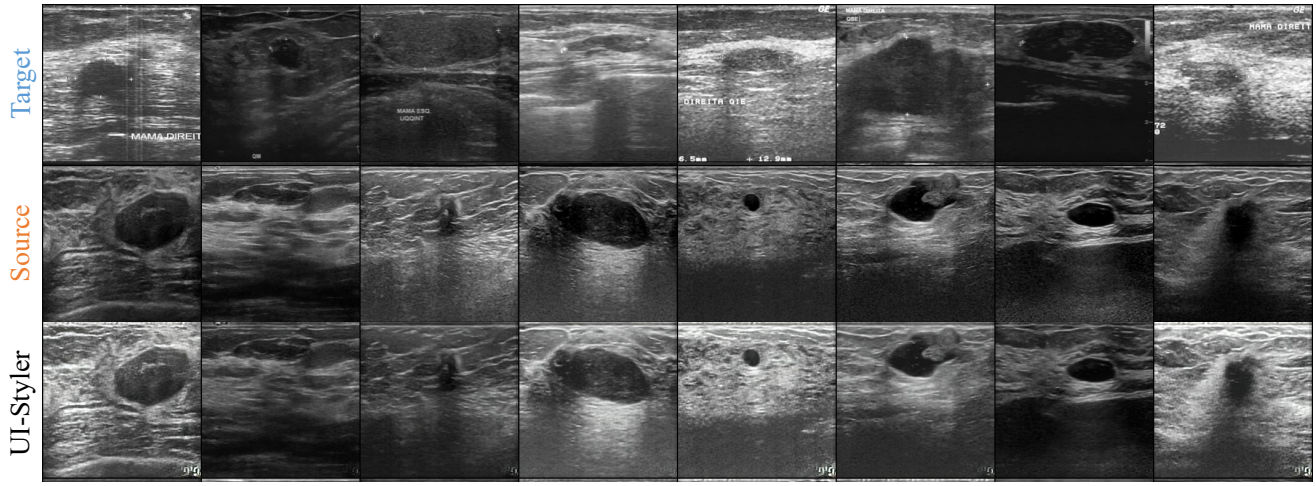
(g) UCLM→BUSBRA.



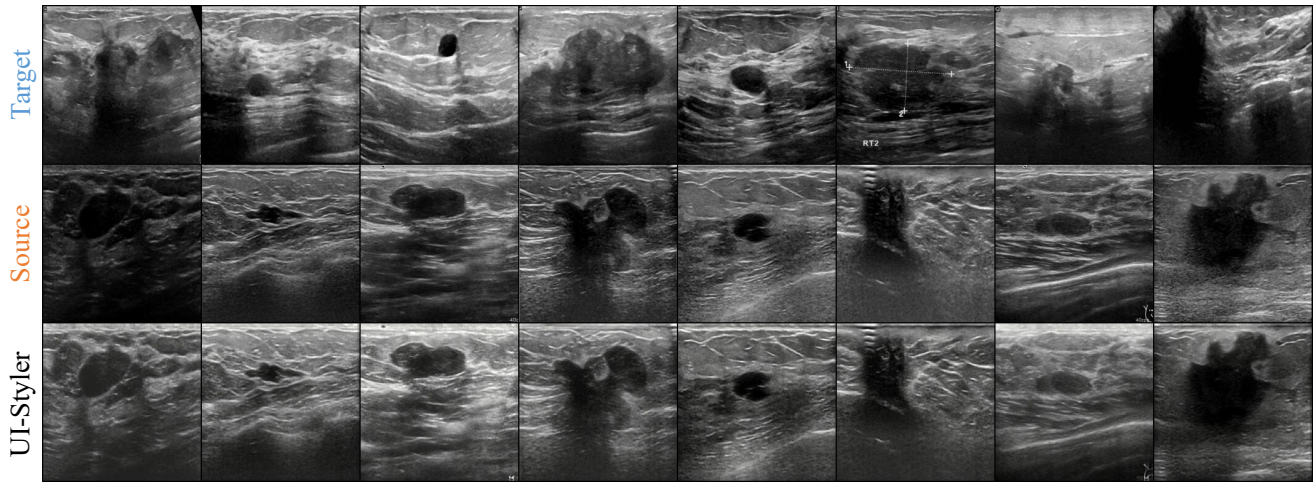
(h) UCLM→BUSI.



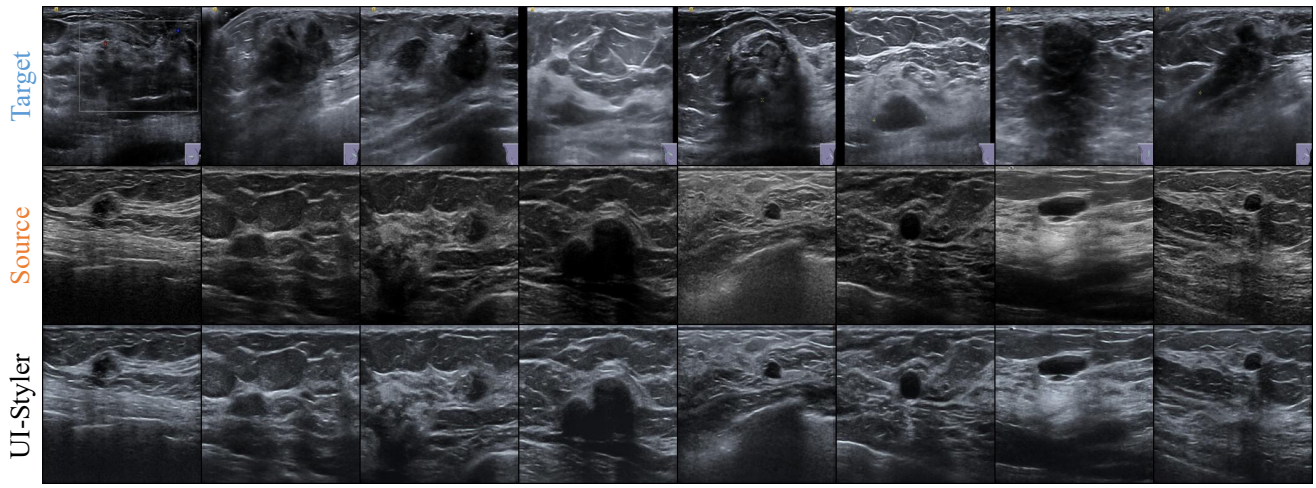
(i) UCLM→UDIAT.



(j) UDIAT→BUSBRA.



(k) UDIAT→BUSI.



(l) UDIAT→UCLM.

Table 4. **Cross-device Visual Results.** We present qualitative results of UI-Styler across all 12 cross-device ultrasound translation tasks. Each group shows representative examples from the target domain (top), source domain (middle), and the stylized results by UI-Styler (bottom).

References

[1] Walid Al-Dhabyani, Mohammed Gomaa, Hussien Khaled, and Aly Fahmy. Dataset of breast ultrasound images. *Data in brief*, 28:104863, 2020. 3

[2] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring visual prompts for adapting large-scale models. *arXiv preprint arXiv:2203.17274*, 2022. 3

[3] Zhixiang Chi, Li Gu, Tao Zhong, Huan Liu, Yuanhao Yu, Konstantinos N Plataniotis, and Yang Wang. Adapting to distribution shift by visual domain prompt generation. In *International Conference on Learning Representations (ICLR)*, 2024. 3

[4] Zhixiang Chi, Li Gu, Huan Liu, Ziqiang Wang, Yanan Wu, Yang Wang, and Konstantinos N Plataniotis. Learning to adapt frozen clip for few-shot test-time domain adaptation. In *International Conference on Learning Representations (ICLR)*, 2025. 3

[5] Arpita Chowdhury, Dipanjyoti Paul, Zheda Mai, Jianyang Gu, Ziheng Zhang, Kazi Sajeed Mehrab, Elizabeth G Campolongo, Daniel Rubenstein, Charles V Stewart, Anuj Karpatne, et al. Prompt-cam: Making vision transformers interpretable for fine-grained analysis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 4375–4385, 2025. 3

[6] Yingying Deng, Fan Tang, Weiming Dong, Chongyang Ma, Xingjia Pan, Lei Wang, and Changsheng Xu. Stytr2: Image style transfer with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11326–11336, 2022. 3

[7] Wilfrido Gómez-Flores, Maria Julia Gregorio-Calas, and Wagner Coelho de Albuquerque Pereira. Bus-bra: a breast ultrasound dataset for assessing computer-aided diagnosis systems. *Medical Physics*, 51(4):3110–3123, 2024. 3

[8] Cheng Han, Qifan Wang, Yiming Cui, Zhiwen Cao, Wenguan Wang, Siyuan Qi, and Dongfang Liu. E^2vpt: An effective and efficient approach for visual prompt tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 17445–17456, 2023. 3

[9] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision (ECCV)*, pages 709–727. Springer, 2022. 3

[10] Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations (ICLR)*, 2022. 3

[11] Ellen B Mendelson, Marcela Böhm-Vélez, Wendie A Berg, GJ Whitman, MI Feldman, H Madjar, et al. Acr bi-rads® ultrasound. *ACR BI-RADS® atlas, breast imaging reporting and data system*, 2013, 2013. 5

[12] Woo Kyung Moon, Chung-Ming Lo, Jung Min Chang, Chiun-Sheng Huang, Jeon-Hor Chen, and Ruey-Feng Chang. Quantitative ultrasound analysis for classification of bi-rads category 3 breast masses. *Journal of digital imaging*, 26(6):1091–1098, 2013. 5

[13] Changdae Oh, Hyeji Hwang, Hee-young Lee, YongTaek Lim, Geunyoung Jung, Jiyoung Jung, Hosik Choi, and Kyungwoo Song. Blackvip: Black-box visual prompting for robust transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24224–24235, 2023. 3

[14] Jay N Paranjape, Shameema Sikder, S Swaroop Vedula, and Vishal M Patel. Black-box adaptation for medical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 454–464. Springer, 2024. 3

[15] Dae Young Park and Kwang Hee Lee. Arbitrary style transfer with style-attentional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5880–5888, 2019. 3

[16] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763. PMLR, 2021. 3

[17] A Thomas Stavros, David Thickman, Cynthia L Rapp, Mark A Dennis, Steve H Parker, and Gale A Sisney. Solid breast nodules: use of sonography to distinguish between benign and malignant lesions. *Radiology*, 196(1):123–134, 1995. 5

[18] Noelia Vallez, Gloria Bueno, Oscar Deniz, Miguel Angel Rienda, and Carlos Pastor. Bus-uclm: Breast ultrasound lesion segmentation dataset. *Scientific Data*, 12(1):242, 2025. 3

[19] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9 (11), 2008. 6, 7

[20] Zhengbo Wang, Jian Liang, Ran He, Zilei Wang, and Tieniu Tan. Connecting the dots: Collaborative fine-tuning for black-box vision-language models. In *International Conference on Machine Learning (ICML)*, 2024. 3

[21] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7959–7971, 2022. 3

[22] Moi Hoon Yap, Gerard Pons, Joan Marti, Sergi Ganau, Melcior Sentis, Reyer Zwiggelaar, Adrian K Davison, and Robert Marti. Automated breast ultrasound lesions detection using convolutional neural networks. *IEEE Journal of Biomedical and Health Informatics*, 22(4):1218–1226, 2017. 3

[23] Chiyu Zhang, Xiaogang Xu, Lei Wang, Zaiyan Dai, and Jun Yang. S2wat: Image style transfer via hierarchical vision transformer using strips window attention. In *AAAI Conference on Artificial Intelligence*, pages 7024–7032, 2024. 3