SplitFlux: Learning to Decouple Content and Style from a Single Image

Supplementary Material

A. Additional Flux Architecture Analysis

For all experiments related to Flux analysis, we fixed the inference steps to 50 while keeping all other hyperparameters at their default settings. The input prompt in Flux is jointly encoded by CLIP [38] and T5 [39]: the CLIP embedding is utilized in modulation layers (e.g., fused with timestep embeddings), whereas the T5 embedding is employed for cross-modal interaction. Although the T5 embedding plays a dominant role in guiding the generative process, the semantic information carried by the CLIP embedding can occasionally influence the final outputs.

To systematically isolate and analyze the functional contribution of each block within Flux, we introduce a masking operation on the CLIP embeddings in our experiments. This operation effectively suppresses semantic cues originating from CLIP, enabling a more precise examination of the model's behavior under the exclusive influence of T5 embeddings. Two sets of ChatGPT-generated prompts were employed for evaluation (see Tab. A.1 and Tab. A.2). The corresponding quantitative results are reported in Tab. A.3, with visual comparisons provided in Fig. A.1 and Fig. A.2. As shown in Tab. A.3, when only a small number of content blocks are selected, the semantic injection capability of P_2 remains limited. As the number of selected blocks increases, the success rate of semantic injection improves markedly, exhibiting a consistent trend for the style blocks as well. Based on these observations, we define the content block range as Blocks 20–29 and the style block range as Blocks 30–57. As illustrated in columns 6 and 7 of Fig. A.1 and Fig. A.2, incomplete specification of block intervals results in images that contain blended or ambiguous semantic features. These findings further substantiate the validity of our chosen content and style block ranges.

Table A.1. Prompt set used for content evaluation.

P_1	P_2
a photo of a dog	a photo of a tiger
a photo of a cat	a photo of a monkey
a photo of a horse	a photo of a lion
a photo of a shark	a photo of a turtle
a photo of a panda	a photo of a fox
a photo of a penguin	a photo of an elephant
a photo of a giraffe	a photo of a wolf
a photo of a kangaroo	a photo of a zebra
a photo of a bear	a photo of a snake
a photo of a dolphin	a photo of a chicken

Table A.2. Prompt set used for style evaluation.

P_1	P_2
a photo of a white dog	a photo of a black dog
a photo of a white cat	a photo of a black cat
a photo of a white horse	a photo of a black horse
a photo of a white rabbit	a photo of a brown rabbit
a photo of a white monkey	a photo of a black monkey
a photo of a white chicken	a photo of a black chicken
a photo of a yellow fox	a photo of a white fox
a photo of a white wolf	a photo of a black wolf
a photo of a white bear	a photo of a black bear
a photo of a white pig	a photo of a black pig

Table A.3. Ablation study on different blocks. B1–19 denotes injecting P_2 into Blocks 1–19 (inclusive); other notations follow the same rule. The percentage represents the probability of successfully injecting the semantics of P_2 into the generated results. Content is evaluated using a detection model [27], and style is assessed with Qwen3-VL [3].

Block replacement	**Content set**	**Style set**
B1–19	0%	0%
B20–57	100%	100%
B20–24	0%	-
B20–27	35%	-
B20–29	**100%**	0%
B35–57	-	4%
B32–57	-	65%
B30–57	0%	**95%**

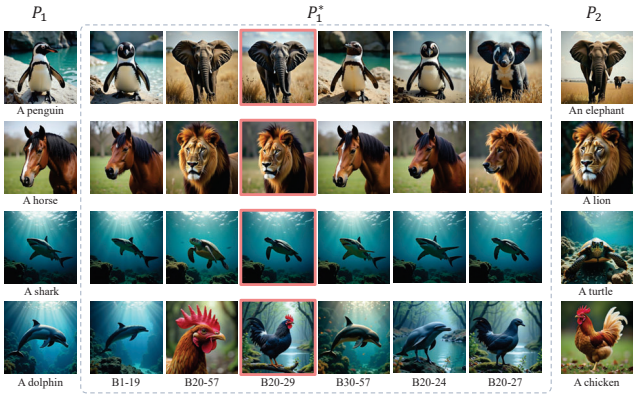

Figure A.1. Results of prompt replacement in different blocks for content evaluation. B1–19 denotes injecting P_2 into all blocks from Block 1 through Block 19, inclusive.

B. VLM-based Preference Study

Inspired by the Visual Question Answering (VQA) paradigm, we constructed single-choice questions and em-

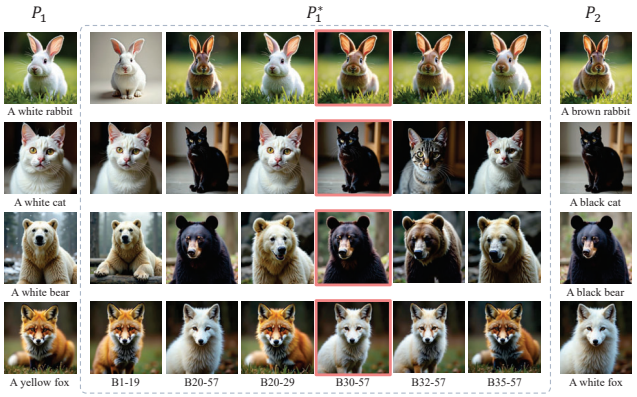

Figure A.2. Results of prompt replacement in different blocks for style evaluation. B1–19 denotes injecting P_2 into all blocks from Block 1 through Block 19, inclusive.

ployed Qwen3-VL [3] to independently evaluate the quality of content preservation and style transfer, as illustrated in Fig. B.3. Specifically, for content preservation assessment, we provided the model with the following message:"*Given the image above, choose the one from the following four images that is most similar to the original image in terms of content, identity, and structure.\n A: Image 1\n B: Image 2\n C: Image 3\n D: Image 4\n Please answer with only one option (A, B, C, or D).*" to determine which generated image best matches the original image in terms of content. For the style quality assessment, the input message was "*Given the image above, choose the one from the following four images that has the best style transfer quality.\n A: Image 1\n B: Image 2\n C: Image 3\n D: Image 4\n Please answer with only one option (A, B, C, or D).*" used to determine which generated image best matches the original image in terms of style.

C. UnZipLoRA Discussion

UnZipLoRA [26] employs three types of prompts during training: a composite prompt ("`A <c> subject in <s> style`"), a content prompt ("`A <c> subject`"), and a style prompt ("`An image in <s> style`"). In practice, the placeholder "`subject`" in these prompts must be replaced with the specific object depicted in the image (e.g., cat, dog), while in the style prompt, "`An image`" must also be replaced by the corresponding subject (e.g., A dog), and "`<s>`" must be substituted with a concrete style descriptor such as sketch or watercolor. During inference, all three types of prompts are required for content-style recontextualization. However, this prompt design inherently limits flexibility. As noted in the UnZipLoRA paper, it becomes particularly challenging to generate appropriate replacements for "`subject`" and "`<s>`" when dealing

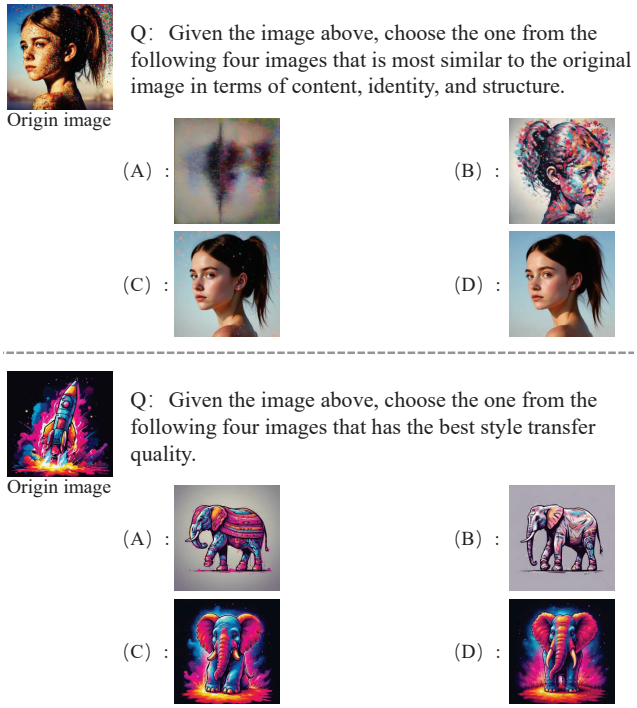

Figure B.3. Example of VLM-based preference study.

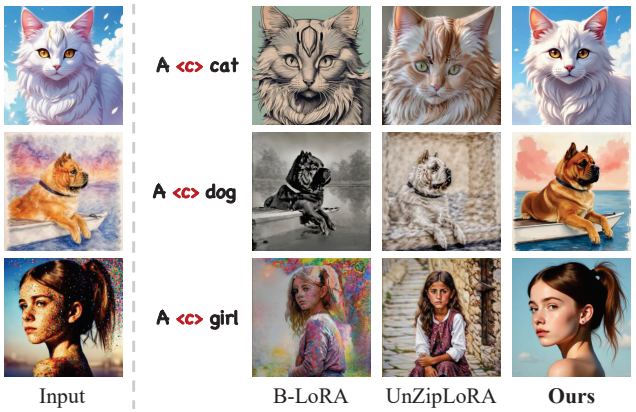

Figure C.4. Results from the loaded content LoRA.

with abstract concepts or styles, thereby restricting the method's applicability. In contrast, our method adopts a unified and fixed prompt formulation for both training and inference, eliminating the dependence on manually crafted descriptions.

To enable a comprehensive comparison with UnZipLoRA [26], we follow the same prompt configuration as in its original setup. In contrast, our method replaces only "`object`" with the corresponding subject depicted in the image while retaining the style token "`<s>`" unchanged. The qualitative comparison results are presented in Fig. C.4, Fig. C.5, Fig. C.6, and Fig. C.7. As illustrated in

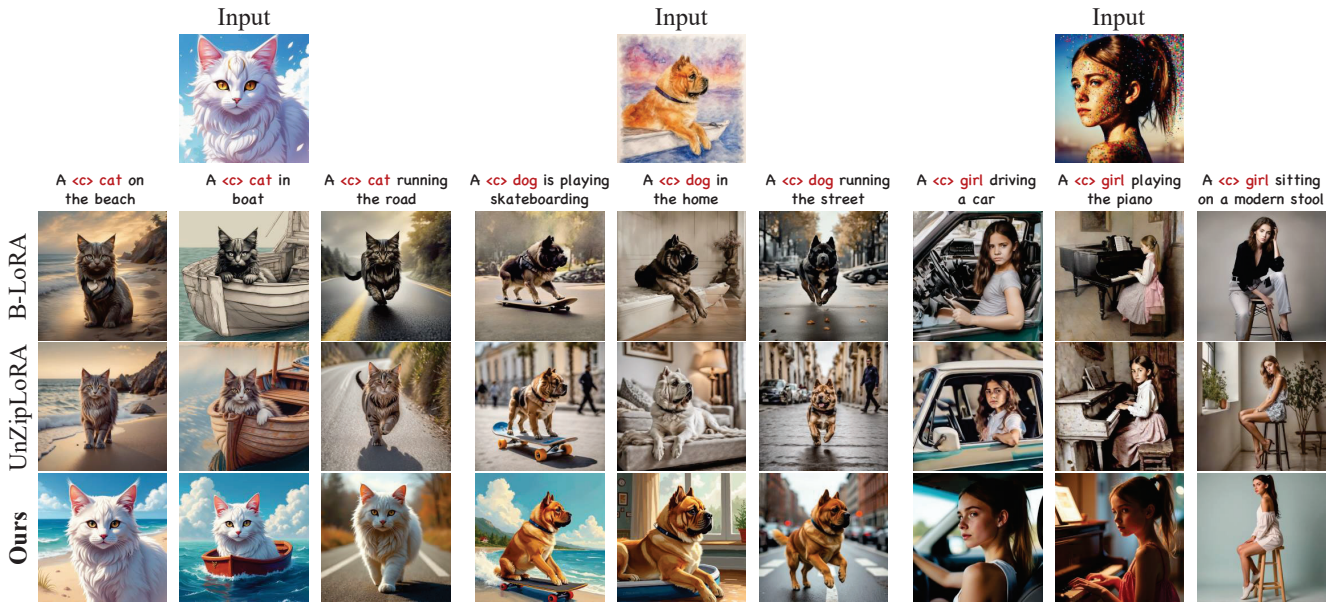

Figure C.5. Results of recontextualization with loaded content LoRA.

Figure C.6. Results of recontextualization with the loaded content and style LoRAs.

Fig. C.4, although specific subject text is introduced, both B-LoRA [13] and UnZipLoRA [26] still fail to disentangle content effectively, as the generated content often suffers from severe structural distortions and loss of identity, making it difficult to apply to downstream tasks. In Fig. C.5, despite the compared methods' ability to embed disentangled content into new contexts, they often lose fine-grained identity details; for instance, a white cat may incorrectly appear as a black cat, or a yellow dog as a black dog. Fig. C.6 shows the results of introducing new semantics to the original images, further demonstrating that our method produces superior results, as illustrated in columns 2, 4, and 7.

Additionally, Fig. C.7 presents the results of disentangled style transfer. We observe that UnZipLoRA requires detailed and explicit textual descriptions of the style to achieve satisfactory style transfer. This dependency becomes problematic when handling complex or abstract styles, where precise textual descriptions are difficult to craft and often necessitate additional assistance from large vision-language models (VLMs). In contrast, our approach relies solely on a fixed prompt without any manually designed style descriptions. Despite this, it still achieves superior style transfer quality. These experiments clearly demonstrate that our method outperforms alternative ap-

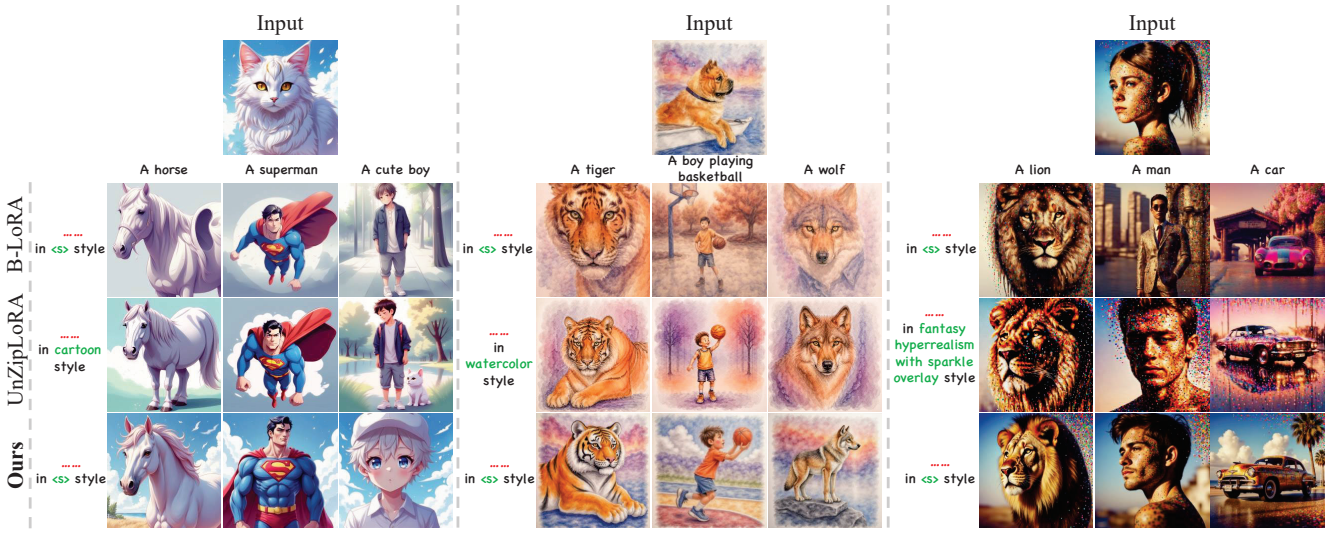

Figure C.7. Results from the loaded style LoRA.

proaches across a wide range of scenarios.

D. Additional Comparison Results

Implementation Details of Compared Methods. For B-LoRA [13], we adopt its official open-source implementation with default settings. The prompt "`A <c>`" is used for content training, while "`A <s>`" is used for style training. For UnZipLoRA [26], prompts for training and inference require explicit subject and style descriptions, which is impractical. Therefore, we adopt its main framework prompts as follows:"`A <c> subject in <s> style`"; "`A <c> subject`"; "`An image in <s> style`". For LoRA-Flux, we use the prompt "`A <c> in <s> style`", which enables simultaneous training of content and style.

To further complement the content presented in the main paper, we provide additional qualitative visual comparisons, as shown in Fig. D.8, Fig. D.10, and Fig. D.9. Consistent with the findings reported in the main paper, these results clearly demonstrate the limitations of existing methods across various tasks.

B-LoRA. As shown in Fig. D.8, B-LoRA attempts to disentangle the content and style of an image. Although it can successfully extract specific content in some simple cases, it fails in most scenarios. The disentangled content, while semantically aligned, often suffers from poor visual quality, making it difficult to apply such representations to downstream tasks (as illustrated in Fig. D.10). In contrast, the disentangled style generally exhibits better performance. Furthermore, when combining multiple LoRA, B-LoRA tends to lose certain content information (see Fig. D.9).

UnZipLoRA. Built upon B-LoRA, UnZipLoRA extends its capability to achieve image disentanglement. Specifically, it expands the block separation strategy to all decoder blocks in SDXL, leading to slightly improved disentanglement performance compared to B-LoRA. However, as shown in Fig. D.8, this approach still suffers from noticeable structural distortion and degraded image quality. In the merging tasks, its performance is even worse than that of B-LoRA (see Fig. D.9). This degradation arises because UnZipLoRA requires a dedicated set of merging weights for LoRA combination, which either demands careful manual tuning or task-specific retraining for different combinations.

LoRA-Flux. LoRA-Flux is designed based on our observations. Due to the absence of the Rank-Constrained Adaptation module, the disentangled content sometimes fails to preserve the original content information. Moreover, the lack of the Visual-Gated LoRA module prevents the disentangled content from being re-embedded into new contexts.

Ours. In contrast, our method effectively disentangles the content and style of images while achieving superior visual quality. With the introduction of the Rank-Constrained Adaptation module, the disentangled content preserves the original identity and structural information without degrading the quality of style transfer. Furthermore, the proposed Visual-Gated LoRA mitigates content overfitting, enabling the disentangled content to be seamlessly integrated into new contexts.

E. Training Parameters and Computational Requirements

Parameters. Both B-LoRA [13] and UnZipLoRA [26] require two sets of LoRA weights — one for content and one for style. Therefore, when calculating the total number of parameters, we sum the parameters of these two weight sets.

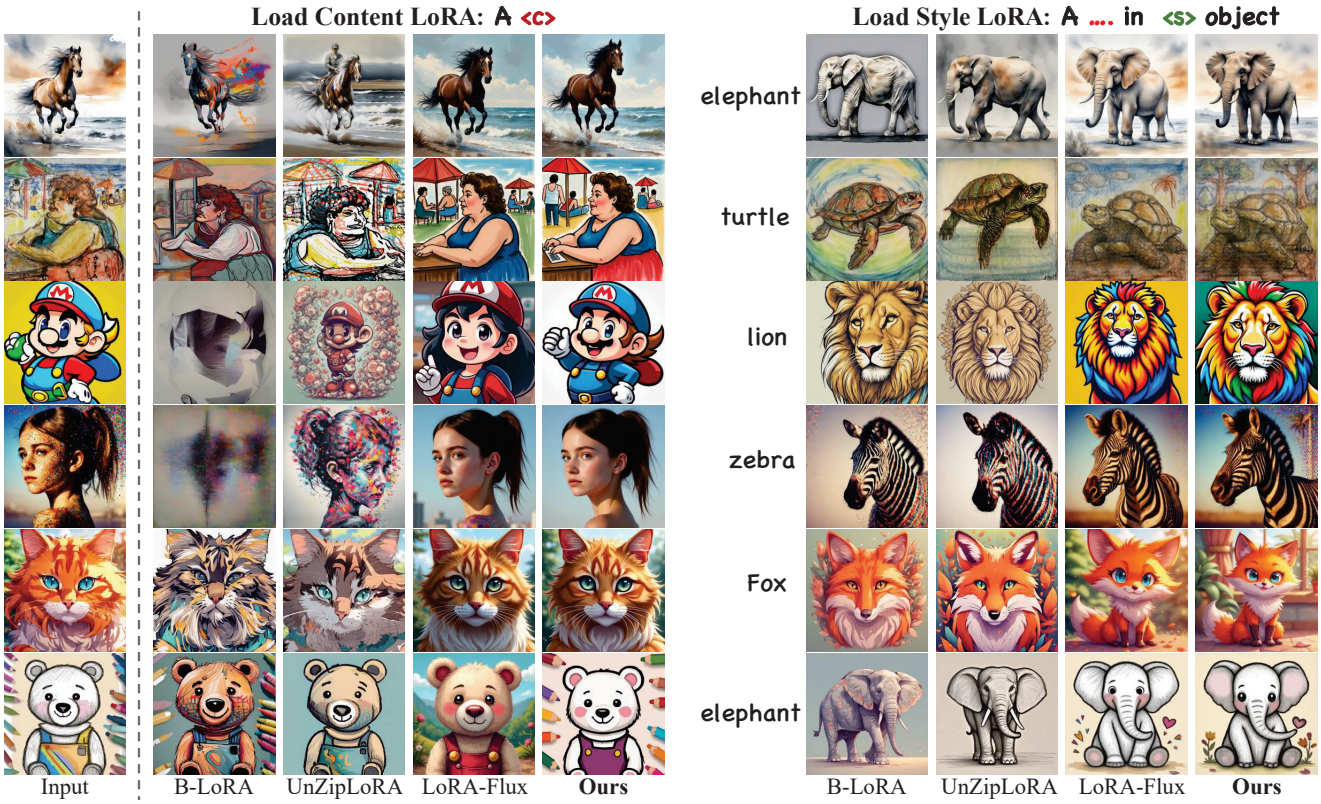

Figure D.8. Qualitative comparison for disentanglement. Our results show that our method achieves superior content and style disentanglement, with the disentangled content preserving the original identity and structure better than other methods.

Figure D.9. Comparison of different methods for Recontextualization. Our approach flexibly adapts to varying contexts while accurately preserving both the subject and style.

In contrast, LoRA-Flux and our method require only a single set of LoRA weights.

Time. For B-LoRA [13], the number of training steps is 1,000. Since it requires two separate trainings to obtain the content and style LoRA weights, the total training time includes both runs, amounting to approximately 23 minutes.

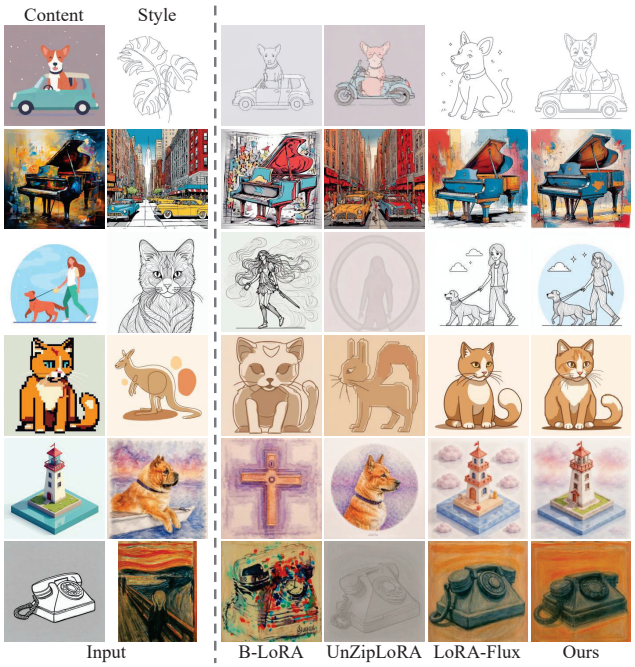

Figure D.10. Qualitative comparison for Merger. Compared to other methods, our approach achieves the best combination, maintaining high fidelity in both content and style.

Figure D.11. More results of Merger.

UnZipLoRA [26] uses 600 training steps and can obtain both content and style LoRA weights in a single training process, so only one run is required, totaling approximately 16 minutes. However, UnZipLoRA [26] notes that for some complex images, the number of training steps may need to be increased to 800, which results in a total training time that exceeds that of our method. Both LoRA-Flux and our method use 1,000 training steps, taking approximately 14 minutes. To prevent specific concepts from being inadvertently bound to other tokens during training, we input different prompts for different blocks. Unlike SDXL, however, the text embeddings in the Flux model are iteratively updated across blocks, which introduces additional computational overhead to obtain block-specific embeddings. As a result, our method incurs slightly higher time costs than LoRA-Flux, taking approximately 18 minutes. Moreover, the inference time for all these approaches is determined by the model itself and is independent of the specific methods employed.