

## 324 A Proofs

### 325 A.1 Proof of Morse Flow Construction

326 **Proposition 3** (Morse Flow Construction). *Let  $B \subset \mathbb{R}^n$  be a bounded open set with smooth boundary,*  
 327 *and let  $j : B \rightarrow \mathbb{R}$  be a smooth Morse function that extends to  $C^\infty(\overline{B})$ . There exists  $\varepsilon > 0$ , a smooth*  
 328 *vector field  $X \in \mathfrak{X}(B)$ , and a unique smooth flow*

$$\Phi : B \times [0, \varepsilon] \rightarrow B$$

329 *satisfying:*

$$\begin{aligned} \frac{d}{dt}\Phi(x, t) &= X(\Phi(x, t)), \\ \Phi(x, 0) &= x, \end{aligned}$$

330 *such that:*

- 331 1.  $j(\Phi(x, \varepsilon)) \geq j(x)$  for all  $x \in B$
- 332 2. Each  $\Phi(\cdot, t) : B \rightarrow B$  is a diffeomorphism
- 333 3. Trajectories remain bounded away from  $\partial B$  for  $t \in [0, \varepsilon]$

334 *Constructive Proof.* We proceed through coordinated geometric and analytic constructions.

#### 335 Step 1: Geometric Preparations

336 1. *Smooth Defining Function:* By the smooth boundary assumption, there exists  $\mu \in$   
 337  $C^\infty(\overline{B}, [0, \infty))$  with:

- 338 •  $\mu^{-1}(0) = \partial B$
- 339 •  $\nabla \mu(x) \neq 0$  for  $x \in \partial B$
- 340 •  $\mu(x) \sim \text{dist}(x, \partial B)$  near  $\partial B$

341 For explicit construction, take  $\mu(x) = f(\text{dist}(x, \partial B))$  where  $f \in C^\infty([0, \infty))$  satisfies  
 342  $f(r) = r$  near 0.

343 2. *Critical Point Isolation:* Since  $j$  is Morse on compact  $\overline{B}$ , its critical points  $\mathcal{C}(j) =$   
 344  $\{p_1, \dots, p_N\}$  are finite and non-degenerate. Choose pairwise disjoint neighborhoods  
 345  $U_i \ni p_i$  with:

$$\overline{U_i} \subset B \setminus \partial B \quad \text{and} \quad \overline{U_i} \cap \mathcal{C}(j) = \{p_i\}$$

#### 346 Step 2: Vector Field Construction

347 1. *Partition of Unity:* Let  $\{\rho_i\}_{i=1}^N$  be smooth functions with:

$$\text{supp}(\rho_i) \subset U_i, \quad 0 \leq \rho_i \leq 1, \quad \sum_{i=1}^N \rho_i \leq 1$$

348 Define the cutoff function:

$$\eta(x) := 1 - \sum_{i=1}^N \rho_i(x)$$

349 Note  $\eta \equiv 0$  near critical points and  $\eta \equiv 1$  outside  $\bigcup U_i$ .

350 2. *Decay Modulation:* Fix  $m \geq n + 1$ . Define the boundary decay factor:

$$\mu_m(x) := \mu(x)^m$$

351 This ensures sufficient regularity at  $\partial B$ .

352 3. *Synthesized Vector Field*: Define

$$X(x) := \eta(x)\mu_m(x)\nabla j(x)$$

353 This field vanishes at critical points and near  $\partial B$ .

### 354 **Step 3: Flow Analysis**

355 *Boundary Avoidance*: For  $x \in B$ , let  $r(t) = \mu(\Phi(x, t))$ . Compute:

$$\frac{dr}{dt} = \nabla\mu(\Phi) \cdot X(\Phi) = \eta(\Phi)\mu_m(\Phi)\nabla\mu(\Phi) \cdot \nabla j(\Phi)$$

356 Using  $|\nabla\mu \cdot \nabla j| \leq C$  near  $\partial B$ :

$$\left| \frac{dr}{dt} \right| \leq C\eta(\Phi)\mu(\Phi)^{m+1} \leq Cr(t)^{m+1}$$

357 Solutions to  $\dot{r} \leq Cr^{m+1}$  satisfy it will never reach 0 in finite time, establishing boundary avoidance.

### 358 **Step 4: Monotonicity & Diffeomorphism**

359 1. *Energy Gain*: Along trajectories:

$$\frac{d}{dt}j(\Phi(x, t)) = \nabla j(\Phi) \cdot X(\Phi) = \eta(\Phi)\mu_m(\Phi)\|\nabla j(\Phi)\|^2 \geq 0$$

360 Thus  $j$  is non-decreasing, with strict increase except at critical points.

361 2. *Flow Diffeomorphisms*: The differential  $D\Phi(x, t)$  satisfies:

$$\frac{d}{dt}D\Phi(x, t) = DX(\Phi(x, t))D\Phi(x, t)$$

362 Since  $X$  is smooth with bounded derivatives on  $\overline{B}$ , Grönwall's inequality gives:

$$\|D\Phi(x, t)\| \leq \exp\left(\int_1^{1+\varepsilon} \|DX(\Phi(x, s))\| ds\right) < \infty$$

363 Thus  $\Phi(\cdot, t)$  remains locally diffeomorphic, and properness follows from boundary avoid-  
364 ance.

### 365 **Step 5: Isotopy Synthesis**

366 The time- $\varepsilon$  map  $\Phi(0, \varepsilon)$  provides the required isotopy through diffeomorphisms.  $\square$

## 367 **A.2 Proof of Cascading Improvement at Adjoint Timesteps**

368 **Proposition 4** (Cascading Improvement at Adjoint Timesteps). *Consider two consecutive timesteps*  
369  *$t, t - \delta$ . Following Algorithm 1, when comparing the cases with and without updating  $\theta$  at  $t$ , updating*  
370  *$\theta$  results in an equal or lower cross-entropy for  $\widehat{\mathbf{x}}_0$  at  $t - \delta$  when  $\delta$  is sufficiently small and all*  
371 *functions are smooth.*

372 *Proof.* We want to show

$$\mathbb{CE}(f(\widehat{\mathbf{x}}_0^2), c) \leq \mathbb{CE}(f(\widehat{\mathbf{x}}_0^1), c)$$

373 for sufficiently small  $\delta$  with the following statement:

$$\widehat{\mathbf{x}}_0^1 = \mathbf{x}_{t-\delta} - \mathbf{v}_\theta(\mathbf{x}_{t-\delta}, t - \delta)(t - \delta)$$

374 and

$$\widehat{\mathbf{x}}_0^2 = \mathbf{x}_{t-\delta} - \mathbf{v}_{\theta+\Delta\theta}(\mathbf{x}_{t-\delta}, t - \delta)(t - \delta),$$

375 where

$$\Delta\theta = -l_r \nabla_\theta \left( \mathbb{CE}(f(\widehat{\mathbf{x}}_0^0), c) \right), \quad \widehat{\mathbf{x}}_0^0 = \mathbf{x}_t - \mathbf{v}_\theta(\mathbf{x}_t, t)t.$$

376 **Step 1. Relating  $\widehat{\mathbf{x}}_0^1$  and  $\widehat{\mathbf{x}}_0^0$ .** Since  $\mathbf{x}_{t-\delta}$  is close to  $\mathbf{x}_t$  for small  $\delta$ , the smoothness of  $\mathbf{v}_\theta(\cdot, \cdot)$   
 377 implies

$$\|\widehat{\mathbf{x}}_0^1 - \widehat{\mathbf{x}}_0^0\| = \left\| [\mathbf{x}_{t-\delta} - \mathbf{v}_\theta(\mathbf{x}_{t-\delta}, t - \delta)(t - \delta)] - [\mathbf{x}_t - \mathbf{v}_\theta(\mathbf{x}_t, t)t] \right\|$$

378 can be made arbitrarily small by taking  $\delta$  sufficiently small (and using continuity/Lipschitz arguments).  
 379 Consequently,

$$\nabla_\theta \mathbb{CE}(f(\widehat{\mathbf{x}}_0^1), c) \quad \text{and} \quad \nabla_\theta \mathbb{CE}(f(\widehat{\mathbf{x}}_0^0), c)$$

380 are also close for small  $\delta$ .

381 **Step 2. First-order comparison at  $t - \delta$ .** By a first-order expansion of  $\mathbf{v}_{\theta+\Delta\theta}$  around  $\theta$  and the  
 382 smoothness of  $\mathbf{v}_\theta(\cdot, \cdot)$ , we have

$$\mathbf{v}_{\theta+\Delta\theta}(\mathbf{x}_{t-\delta}, t - \delta) = \mathbf{v}_\theta(\mathbf{x}_{t-\delta}, t - \delta) + \nabla_\theta \mathbf{v}_\theta(\mathbf{x}_{t-\delta}, t - \delta) \Delta\theta + \mathcal{O}(\|\Delta\theta\|^2).$$

383 Hence,

$$\widehat{\mathbf{x}}_0^2 - \widehat{\mathbf{x}}_0^1 = -\left[ \mathbf{v}_{\theta+\Delta\theta}(\mathbf{x}_{t-\delta}, t - \delta) - \mathbf{v}_\theta(\mathbf{x}_{t-\delta}, t - \delta) \right] (t - \delta) \approx -(t - \delta) \nabla_\theta \mathbf{v}_\theta(\mathbf{x}_{t-\delta}, t - \delta) \Delta\theta.$$

384 **Step 3. Cross-entropy decrease.** Using the smoothness of the cross-entropy and another first-order  
 385 expansion,

$$\begin{aligned} \mathbb{CE}(f(\widehat{\mathbf{x}}_0^2), c) - \mathbb{CE}(f(\widehat{\mathbf{x}}_0^1), c) \\ \approx \langle \nabla_{\widehat{\mathbf{x}}_0} \mathbb{CE}(f(\widehat{\mathbf{x}}_0^1), c), \widehat{\mathbf{x}}_0^2 - \widehat{\mathbf{x}}_0^1 \rangle + \mathcal{O}(\|\widehat{\mathbf{x}}_0^2 - \widehat{\mathbf{x}}_0^1\|^2) \\ \approx \langle \nabla_\theta \mathbb{CE}(f(\widehat{\mathbf{x}}_0^1), c), \Delta\theta \rangle + \mathcal{O}(\|\Delta\theta\|^2, \|\delta\|). \end{aligned}$$

386 By definition of the gradient step  $\Delta\theta = -l_r \nabla_\theta \mathbb{CE}(f(\widehat{\mathbf{x}}_0^0), c)$  and the fact that  $\nabla_\theta \mathbb{CE}(f(\widehat{\mathbf{x}}_0^0), c)$   
 387 is close to  $\nabla_\theta \mathbb{CE}(f(\widehat{\mathbf{x}}_0^1), c)$  for small  $\delta$ , the above inner product is non-positive up to higher-order  
 388 (small) terms. Concretely,

$$\langle \nabla_\theta \mathbb{CE}(f(\widehat{\mathbf{x}}_0^1), c), -l_r \nabla_\theta \mathbb{CE}(f(\widehat{\mathbf{x}}_0^0), c) \rangle \leq 0$$

389 when  $\delta$  is sufficiently small so that these gradients align (up to small errors). Therefore,

$$\mathbb{CE}(f(\widehat{\mathbf{x}}_0^2), c) \leq \mathbb{CE}(f(\widehat{\mathbf{x}}_0^1), c),$$

390 which completes the proof.  $\square$

## 391 B Related Works

### 392 B.1 Targeted and Untargeted Attacks

393 **Targeted Attacks.** The objective of targeted attacks is to force the classifier to output a specified  
 394 label. In other words, the attacker seeks to cause the model to produce incorrect classification results  
 395 and aims for the result to be a specific target class. This type of attack is more hazardous due to its  
 396 ability to manipulate the model's output precisely but is typically more challenging to execute.

397 **Untargeted Attacks.** The goal of untargeted attacks is to make the classifier output any incorrect  
 398 label. The attacker merely needs to mislead the model so that its classification result does not  
 399 match the true label. Despite having lower requirements, untargeted attacks can still have severe  
 400 consequences in certain situations.

### 401 B.2 White-Box and Black-Box Attacks

402 **White-Box Attacks.** White-box attacks assume that the attacker has complete access to the target  
 403 model, including its architecture, parameters, and gradient information. Using this information, the  
 404 attacker can generate efficient adversarial examples through iterative optimization methods.

405 **Black-Box Attacks.** Black-box attacks assume that the attacker does not have access to the internal  
406 information of the target model. A common method to implement black-box attacks is to utilize  
407 transferability, where adversarial examples are first generated against a known surrogate model and  
408 then used to attack the unknown target model.

### 409 **B.3 Instance-Specific and Instance-Agnostic Attacks**

410 **Instance-Specific Attacks.** Instance-specific attacks [10, 18, 14, 58, 52, 34, 30] and *instance-*  
411 *agnostic* generate adversarial perturbations for specific input samples. The attacker uses gradient  
412 information from the target model and iterative optimization algorithms to create minimal pertur-  
413 bations that achieve the attack on a given sample. Such attacks usually have high success rates on  
414 individual samples but lack generalization and transferability.

415 **Instance-Agnostic Attacks.** Instance-agnostic attacks [56, 35, 29, 37, 38, 16] do not target specific  
416 input samples but instead learn universal adversarial perturbations or generative functions based on  
417 data distribution. These attack methods have better generalization across different samples, thus  
418 exhibiting stronger transferability.

### 419 **B.4 Subcategories of Instance-Agnostic Attacks**

420 Instance-agnostic attacks can be further subdivided into the following categories:

421 **Universal Adversarial Perturbations.** These methods learn a universal perturbation [36, 61]  
422 applicable to the entire dataset. The classifier can be misled by superimposing this perturbation on  
423 any input sample.

424 **Generative Models.** Generative attacks [41, 37] train a generator that, upon receiving an input  
425 sample, can produce specific adversarial perturbations. This approach often surpasses universal  
426 adversarial perturbations regarding flexibility and attack efficacy.

### 427 **B.5 Single-Target and Multi-Target Attacks**

428 **Single-Target Attacks.** Single-target attacks train an individual generative model for each target  
429 class [37, 38, 16, 54]. Although these models achieve high success rates for single-target classes,  
430 the training cost becomes substantial when the number of target classes is large, thereby limiting  
431 practical usability.

432 **Multi-Target Attacks.** Multi-target attacks simultaneously train the attack capabilities for multiple  
433 target classes within a single model [21, 60, 15]. Class labels or text embeddings are typically used  
434 as conditional inputs to generate corresponding adversarial perturbations. This method significantly  
435 reduces training costs and enhances feasibility in real-world applications.

## 436 **C Comparison with Other Diffusion-Based Methods**

437 Currently, diffusion models have been explored in several adversarial attack methods [5, 59, 6, 4, 7].  
438 Among them, ACA [6] and DiffAttack [4] leverage DDIM inversion to obtain the latent space  
439 of diffusion models and optimize adversarial examples within this latent space. AdvDiffuser [5],  
440 DiffPGD [59], and AdvDiff [7] incorporate adversarial guidance during the reverse denoising process  
441 of diffusion models. Notably, although these methods are diffusion-based, they are all instance-  
442 specific attacks, requiring access to the target classifier’s gradient information during inference for  
443 each input sample to perform the attack.

444 In contrast, our method, once trained, does not require any further information from the classifier,  
445 enabling more efficient generation of adversarial examples.

446 Furthermore, since these prior methods either only support untargeted attacks[5, 59, 6, 4] or focus  
447 on unrestricted adversarial examples[5, 6, 4, 7], their settings are not directly compatible with ours,  
448 making direct comparisons infeasible.

## D Method Details

### D.1 Target Class Condition Representation

For each label  $c$  in the target label set  $\mathcal{C}$ , we first obtain its class description and format it into a text condition using the template "a photo of a {class}"[42]. Subsequently, we utilize CLIP’s text encoder to derive this textual input’s embedding  $\mathbf{e}$ . Finally, this embedding is fed into our model via cross-attention mechanisms:

$$Q = \mathbf{z}W_Q, K = \mathbf{e}W_K, V = \mathbf{e}W_V, \quad (6)$$

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right) \cdot V,$$

where  $\mathbf{z} \in \mathbb{R}^{d_z}$  denotes the flattened intermediate features of the unet model,  $W_Q \in \mathbb{R}^{d_z \times d}$ ,  $W_K \in \mathbb{R}^{d_e \times d}$ ,  $W_V \in \mathbb{R}^{d_e \times d}$  are learnable parameters.

By employing this approach, we can leverage the rich semantic priors associated with the target classes embedded in the pre-trained diffusion model, thereby facilitating a more effective training process.

---

#### Algorithm 3 Single-Target Fine-Tuning Mechanism

---

**Input:**  $\tau = N\delta$ , stepsize  $\delta$ , model param.  $\phi, \theta$ , victim model  $f$ , target label  $c$ , training dataset  $\{\mathbf{I}^i\}_{i \in \mathcal{I}}$ , learning rate  $l_r$

**repeat**

**for**  $i \in \mathcal{I}$  **do**

    get  $\mathbf{x}_0 = \mathbf{I}^i$

**for**  $t = 1$  **to**  $N$  **do**

$\mathbf{x}_{t\delta} = \mathbf{x}_{(t-1)\delta} + \mathbf{v}_\phi(\mathbf{x}_{(t-1)\delta}, (t-1)\delta, \emptyset)\delta$

**end for**

**for**  $t = N$  **to**  $1$  **do**

$\mathbf{x}_{(t-1)\delta} = \mathbf{x}_{t\delta} - \mathbf{v}_\theta(\mathbf{x}_{t\delta}, t\delta, c)\delta$

$\widehat{\mathbf{x}}_0 = \mathbf{x}_{t\delta} - \mathbf{v}_\theta(\mathbf{x}_{t\delta}, t\delta, c)t\delta$

      get random mask  $M$

$\widehat{\mathbf{x}}_0 = \mathbf{x}_0 + M \cdot (\widehat{\mathbf{x}}_0 - \mathbf{x}_0)$

$\widehat{\mathbf{x}}_0^{i,j,k} = \text{clip}\left(\widehat{\mathbf{x}}_0^{i,j,k}, \mathbf{x}^{i,j,k} - \epsilon, \mathbf{x}^{i,j,k} + \epsilon\right)$

$\theta = \theta - l_r \cdot \nabla_\theta(\mathbb{E}(f(\widehat{\mathbf{x}}_0), c))$

**end for**

**end for**

**until**  $\mathbf{v}_\theta$  convergence

**Return:** Dual-Flow  $\{\mathbf{v}_\phi, \mathbf{v}_\theta\}$

---

### D.2 Fine-Tuning on Single-Target Tasks

We fine-tune our model for single-target tasks to enhance its performance further. Specifically, we fix the target label during training, enabling the model to focus on targeted attacks for a specific label. To mitigate the perturbations being confined to some areas of the image, which can reduce the robustness and transferability of adversarial examples in single-target training, we apply the mechanism introduced in [15].

In detail, we generate a random mask  $M$  of the same size as the image, where several randomly positioned square pixel areas are set to 0, and the rest are set to 1. By multiplying this mask with the perturbation, we ensure the generated adversarial samples remain consistent with the original image in the masked square areas. This forces the model to create adversarial patterns distributed across the entire image rather than being localized to specific regions, as illustrated in Algorithm 3.

Like other single-target methods, we must fine-tune a separate model for each target. However, due to our model’s powerful capabilities in multi-target attacks, once the model is trained on the

multi-target task, it requires only a few additional steps to adapt to each single-target task. This results in significantly lower training overhead compared to other methods.

## E More Experiments

**Evaluation on Transformer Models.** We adhered to the settings of prior works and tested our attack method’s success rate when transferred to transformer models. Specifically, we utilized Res152 as the substitute model. The results, included in Table 5, demonstrate that despite the fundamental architectural differences between transformer models and our source model based on convolutional networks (Res152), our method maintains a high attack success rate, significantly outperforming baseline methods. This further corroborates the advantage of our method in terms of transferability.

Table 5: Attack success rates (%) for multi-target attacks on transformer models. The surrogate model is Res-152.

Method	ViT-B/16	CaiT-S/24	Visformer-S	DeiT-B	LeViT-256	TNT-S
C-GSP	11.78	32.00	36.60	35.58	37.85	31.00
CGNC	19.46	54.56	58.70	59.90	57.53	48.40
Dual-Flow	<b>36.39</b>	<b>74.24</b>	<b>76.72</b>	<b>78.50</b>	<b>79.34</b>	<b>67.86</b>

**Evaluation on DiffPure.** We evaluate our attack method using Diffusion Models for Adversarial Purification (DiffPure)[40]. The experimental results show the attack success rates of our method under various DiffPure  $t^*$  settings and compare them with the baseline method. As illustrated in Table 6, the baseline method is easily nullified by the purification process, whereas our method maintains a significant success rate. This further demonstrates the robustness of our approach.

Table 6: Attack success rates (%) for multi-target attacks on normally trained models with DiffPure. The surrogate model is Res-152.

$t^*$	Method	Inc-v3	Inc-v4	Inc-Res-v2	Res-152	DN-121	GoogleNet	VGG-16
0.05	CGNC	16.26	19.91	8.53	67.76	49.81	21.79	29.81
	Ours	49.60	51.92	37.56	79.50	70.20	51.30	52.26
0.10	CGNC	2.41	2.96	1.25	14.65	10.10	3.46	4.59
	Ours	24.31	25.32	18.76	48.05	39.81	25.06	24.78
0.15	CGNC	0.47	0.46	0.34	1.84	1.46	0.62	0.92
	Ours	7.20	7.87	5.89	16.70	13.72	7.71	8.34

Table 7: Attack success rates (%) for multi-target attacks on normally trained models using the ImageNet validation set. The perturbation budget is constrained to  $l_\infty \leq 16/255$ . \* indicates white-box attacks. The results are averaged across 8 different target classes, and the overall average on the far right is computed solely for black-box attacks.

Source	Method	Inc-v3	Inc-v4	Inc-Res-v2	Res-152	DN-121	GoogleNet	VGG-16	Average
Inc-v3	CGNC	96.59*	57.82	46.84	44.13	65.90	53.40	56.27	54.06
	Ours	89.89*	<b>75.74</b>	<b>65.05</b>	<b>75.73</b>	<b>82.75</b>	<b>72.21</b>	<b>66.20</b>	<b>72.95</b>
Res-152	CGNC	56.00	50.37	32.26	96.44*	86.69	63.84	63.90	58.84
	Ours	<b>69.75</b>	<b>72.53</b>	<b>54.11</b>	92.70*	<b>86.71</b>	<b>74.08</b>	<b>68.22</b>	<b>70.90</b>

**Transferability Evaluation On ImageNet Validation Set.** In addition to the evaluation on the ImageNet-NeurIPS (1k) dataset[39], we conducted an assessment of our attack method on the ImageNet validation set (50k)[8] and compared it with the state-of-the-art multi-target attack method, CGNC[15]. The experimental results presented in Table 7 indicate that our method achieves a significantly higher average black-box attack success rate than CGNC, demonstrating its superior transferability. This outcome is consistent with the results observed on the ImageNet-NeurIPS (1k) dataset.

## F More Ablation Studies

**Ablation on LoRA Fine-tuning.** To demonstrate that the adversarial attack capability of our model stems from LoRA fine-tuning, we evaluated the model’s performance when performing attacks directly without applying LoRA. As shown in Table 8, the original model fails to achieve effective adversarial attacks, which validates the necessity of incorporating LoRA for fine-tuning.

Table 8: Attack success rates comparing the model with and without LoRA. The surrogate model is Res-152.

Method	Inc-v3	Inc-v4	Inc-Res-v2	Res-152	DN-121	GoogleNet	VGG-16
w/o LoRA	0.075	0.075	0.05	0.05	0.075	0.05	0.075
w/ LoRA	69.58	71.92	56.10	92.39	85.73	73.65	67.59

**Ablation on Dynamic Gradient Clipping.** To evaluate the effect of dynamic gradient clipping, we designed a variant that does not apply dynamic clipping during training and only clips the outputs to satisfy the norm constraints during inference. As shown in Table XXX, if the model is not trained with clipping, it fails to adapt to the norm constraints at inference time and thus cannot perform effective attacks.

Table 9: Attack success rates comparing the model with and without Dynamic Gradient Clipping. The surrogate model is Res-152.

Method	Inc-v3	Inc-v4	Inc-Res-v2	Res-152	DN-121	GoogleNet	VGG-16
w/o clip	1.85	2.32	1.96	2.84	3.51	1.82	1.19
w/ clip	69.58	71.92	56.10	92.39	85.73	73.65	67.59

**Ablation on Loss.** We designed a variant: during training, we use a new  $L_2$  loss function to make the model’s output as close as possible to the original ODE trajectory, ensuring it does not deviate too far from the original ODE flow. We call this variant Dual-Flow- $L_2$ . The experimental results in Table 10 indicate that the attack capability of this variant is not ideal, as described in Section 3.2.

Table 10: Comparison of Dual-Flow and Dual-Flow- $L_2$ . The surrogate model is Res-152.

Method	Inc-v3	Inc-v4	Inc-Res-v2	Res-152	DN-121	GoogleNet	VGG-16	Average
Dual-Flow	69.58	71.92	56.10	92.39	85.73	73.65	67.59	70.76
Dual-Flow- $L_2$	54.90	56.26	42.94	86.80	78.41	57.71	46.36	56.10

## G More Analysis

**Semantic Adversarial Attack.** We compared the adversarial perturbations generated by our method and those produced by the state-of-the-art multi-target attack method, CGNC[15]. As illustrated in Figure 5, the visual results indicate that while CGNC’s perturbations contain some semantic features of the target class, they are primarily confined to small, repetitive patterns. In contrast, our method generates perturbations that are semantically more representative of the complete target class.

To further validate this observation, we directly input the adversarial perturbations generated by CGNC and our method into the target classifier. As shown in Table 11, our adversarial perturbations alone can induce the classifier to predict the target class with a relatively high probability. Conversely, the perturbations produced by CGNC exhibit a lower likelihood, particularly when transferred to black-box models. This demonstrates that our perturbations incorporate more semantic features of the target class.

Moreover, as depicted in Figure 2, the unclipped samples generated by our method are semantically very close to the target class. We also input these unclipped samples directly into the target classifier. Table 11 shows that these samples are highly likely to be classified as the target class. This confirms that semantic proximity to the target class effectively increases attack success rates.

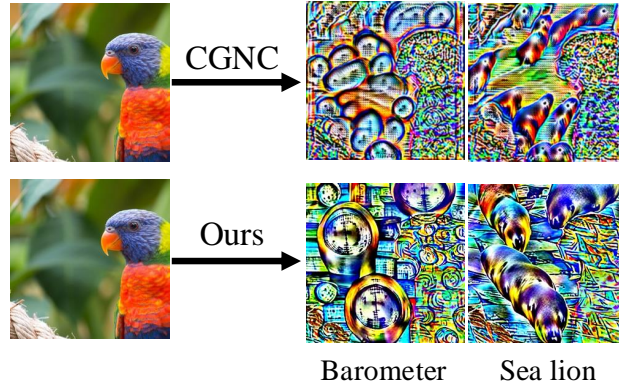


Figure 5: Visualization results comparing the adversarial perturbations generated by our method with those produced by CGNC.

Table 11: Comparison of different types of adversarial inputs. CGNC-P and Dual-Flow-P represent the adversarial perturbations generated by CGNC and our method, respectively, while Dual-Flow-A denotes the unclipped adversarial samples produced by our method. The adversarial perturbations are scaled to a range between 0 and 1 before input into the classifier.

Source	Method	Inc-v3	Inc-v4	Inc-Res-v2	Res-152	DN-121	GoogleNet	VGG-16
Inc-v3	CGNC-P	56.80*	19.15	22.06	10.56	13.14	14.56	3.41
	Dual-Flow-P	86.55*	81.20	74.55	74.55	70.66	76.15	55.10
	Dual-Flow-A	99.12*	95.31	92.79	97.39	96.80	95.62	87.95
Res-152	CGNC-P	23.61	25.72	39.07	58.29*	39.09	37.47	17.21
	Dual-Flow-P	68.44	81.48	78.26	86.29*	80.44	74.59	55.35
	Dual-Flow-A	94.38	96.60	94.62	99.19*	97.92	93.54	90.85

These findings collectively suggest that our attack method’s robustness and transferability result from embedding substantial target class semantics into the images, thereby reducing dependence on specific target model decision boundaries.

## H More Visualization

## I Limitations

While our proposed Dual-Flow framework demonstrates strong adversarial attack performance and transferability, several limitations remain. First, the training process involves additional computational overhead compared to some simpler attack methods, which may affect scalability to extremely large datasets or models. Second, although we observe improved semantic consistency in generated perturbations, there is still room to enhance the interpretability and controllability of adversarial patterns in more complex scenarios. Lastly, our evaluations focus primarily on standard image classification benchmarks; extending the approach to other tasks or modalities warrants further investigation.

## J Statistical Significance

To rigorously evaluate the reliability of our attack success rates, we partitioned the test dataset into 5 disjoint subsets, each containing 200 images. We computed the attack success rate for each subset independently and derived the overall 95% confidence intervals (CIs) for our method. These confidence intervals provide a quantitative measure of variability and statistical uncertainty. We further compared our results against the baseline method by examining the overlap of their respective confidence intervals. As shown in Table 12, the non-overlapping intervals observed in our experiments





Figure 6: Visualization results of different input images targeting various classes. For each text prompt of the target class, the left column displays the adversarial examples generated before clipping, the middle column shows the adversarial examples after clipping, and the right column presents the corresponding adversarial perturbations, which represent the differences between the clipped adversarial examples and the original images. Note that the perturbations are scaled to a range between 0 and 1. The surrogate model used is Inc-v3.

544 indicate that the improvement in attack success rate achieved by our method is statistically significant  
545 with high confidence.

Table 12: Attack success rates with 95% confidence intervals over 5 data splits, compared to the baseline method. The surrogate model is Res-152.

Method	Inc-v3	Inc-v4	Inc-Res-v2	DN-121	GoogleNet	VGG-16
CGNC	53.39 $\pm$ 2.77	51.53 $\pm$ 2.52	34.24 $\pm$ 1.72	85.66 $\pm$ 1.53	62.23 $\pm$ 2.20	63.36 $\pm$ 2.95
Dual-Flow	69.58 $\pm$ 3.70	71.92 $\pm$ 3.27	56.10 $\pm$ 3.31	85.73 $\pm$ 1.57	73.65 $\pm$ 2.83	67.59 $\pm$ 2.28

## 546 K Societal Impacts

547 Our work contributes to a deeper understanding of adversarial vulnerabilities in deep learning models,  
548 which is essential for improving model robustness and security. By developing more effective attack  
549 methods, we provide valuable tools for evaluating and strengthening defenses against malicious  
550 exploitation. However, as with any adversarial technique, there is a potential risk of misuse in  
551 compromising AI systems. We advocate for responsible use of such methods strictly within research  
552 and security auditing contexts, and encourage the community to develop corresponding mitigation  
553 strategies to safeguard AI applications.

## 554 References

- 555 [1] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square  
556 attack: a query-efficient black-box adversarial attack via random search. In *European conference*  
557 *on computer vision*, pp. 484–501. Springer, 2020.
- 558 [2] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust ad-  
559 versarial examples. In *International conference on machine learning*, pp. 284–293. PMLR,  
560 2018.
- 561 [3] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In  
562 *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57. Ieee, 2017.

- [4] Jianqi Chen, Hao Chen, Keyan Chen, Yilan Zhang, Zhengxia Zou, and Zhenwei Shi. Diffusion models for imperceptible and transferable adversarial attack. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- [5] Xinquan Chen, Xitong Gao, Juanjuan Zhao, Kejiang Ye, and Cheng-Zhong Xu. Advdiffuser: Natural adversarial example synthesis with diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4562–4572, 2023.
- [6] Zhaoyu Chen, Bo Li, Shuang Wu, Kaixun Jiang, Shouhong Ding, and Wenqiang Zhang. Content-based unrestricted adversarial attack. *Advances in Neural Information Processing Systems*, 36: 51719–51733, 2023.
- [7] Xuelong Dai, Kaisheng Liang, and Bin Xiao. Advdiff: Generating unrestricted adversarial examples using diffusion models. In *European Conference on Computer Vision*, pp. 93–109. Springer, 2025.
- [8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.
- [9] Gavin Weiguang Ding, Luyu Wang, and Xiaomeng Jin. Advertorch v0. 1: An adversarial robustness toolbox based on pytorch. *arXiv preprint arXiv:1902.07623*, 2019.
- [10] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9185–9193, 2018.
- [11] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4312–4321, 2019.
- [12] Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M Roy. A study of the effect of jpg compression on adversarial images. *arXiv preprint arXiv:1608.00853*, 2016.
- [13] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*.
- [14] Kevin Eykholt, Ivan Evtimov, Earlece Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1625–1634, 2018.
- [15] Hao Fang, Jiawei Kong, Bin Chen, Tao Dai, Hao Wu, and Shu-Tao Xia. Clip-guided generative networks for transferable targeted adversarial attacks. In *European Conference on Computer Vision*, pp. 1–19. Springer, 2025.
- [16] Weiwei Feng, Nanqing Xu, Tianzhu Zhang, and Yongdong Zhang. Dynamic generative targeted attacks with pattern injection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16404–16414, 2023.
- [17] Kuofeng Gao, Yang Bai, Jiawang Bai, Yong Yang, and Shu-Tao Xia. Adversarial robustness for visual grounding of multimodal large language models. In *ICLR Workshop*, 2024.
- [18] Lianli Gao, Yaya Cheng, Qilong Zhang, Xing Xu, and Jingkuan Song. Feature space targeted attacks by statistic alignment. *arXiv preprint arXiv:2105.11645*, 2021.
- [19] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.
- [20] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

- [21] Jiangfan Han, Xiaoyi Dong, Ruimao Zhang, Dongdong Chen, Weiming Zhang, Nenghai Yu, Ping Luo, and Xiaogang Wang. Once a man: Towards multi-target attack via learning multi-target adversarial network once. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5158–5167, 2019.
- [22] Jie Hang, Keji Han, Hui Chen, and Yun Li. Ensemble adversarial black-box attacks against deep learning systems. *Pattern Recognition*, 101:107184, 2020.
- [23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pp. 630–645. Springer, 2016.
- [24] Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*, 2019.
- [25] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [26] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.
- [27] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.
- [28] Rongjie Huang, Jiawei Huang, Dongchao Yang, Yi Ren, Luping Liu, Mingze Li, Zhenhui Ye, Jinglin Liu, Xiang Yin, and Zhou Zhao. Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models. *arXiv preprint arXiv:2301.12661*, 2023.
- [29] Zelun Kong, Junfeng Guo, Ang Li, and Cong Liu. Physgan: Generating physical-world-resilient adversarial examples for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14254–14263, 2020.
- [30] Qizhang Li, Yiwen Guo, and Hao Chen. Yet another intermediate-level attack. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pp. 241–257. Springer, 2020.
- [31] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. *arXiv preprint arXiv:1908.06281*, 2019.
- [32] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*.
- [33] Xingchao Liu, Chengyue Gong, et al. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*.
- [34] Yantao Lu, Yunhan Jia, Jianyu Wang, Bai Li, Weiheng Chai, Lawrence Carin, and Senem Velipasalar. Enhancing cross-task black-box transferability of adversarial examples with dispersion reduction. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pp. 940–949, 2020.
- [35] Jinqi Luo, Tao Bai, and Jun Zhao. Generating adversarial yet inconspicuous patches with a single image (student abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.
- [36] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1765–1773, 2017.
- [37] Muhammad Muzammal Naseer, Salman H Khan, Muhammad Haris Khan, Fahad Shahbaz Khan, and Fatih Porikli. Cross-domain transferability of adversarial perturbations. *Advances in Neural Information Processing Systems*, 32, 2019.

- [38] Muzammal Naseer, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Porikli. On generating transferable targeted perturbations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7708–7717, 2021.
- [39] NeurIPS. <https://www.kaggle.com/c/nips-2017-defense-against-adversarial-attack/data>. Kaggle, 2017.
- [40] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification. In *International Conference on Machine Learning (ICML)*, 2022.
- [41] Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. Generative adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4422–4431, 2018.
- [42] 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*, pp. 8748–8763. PMLR, 2021.
- [43] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.
- [44] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [45] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.
- [46] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.
- [47] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.
- [48] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.
- [49] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.
- [50] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.
- [51] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.
- [52] Xiaosen Wang and Kun He. Enhancing the transferability of adversarial attacks through variance tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1924–1933, 2021.
- [53] Xiaosen Wang, Kun He, and John E Hopcroft. At-gan: A generative attack model for adversarial transferring on generative adversarial nets. *arXiv preprint arXiv:1904.07793*, 3(4):3, 2019.
- [54] Zhibo Wang, Hongshan Yang, Yunhe Feng, Peng Sun, Hengchang Guo, Zhifei Zhang, and Kui Ren. Towards transferable targeted adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20534–20543, 2023.

- 709 [55] Zhipeng Wei, Jingjing Chen, Zuxuan Wu, and Yu-Gang Jiang. Enhancing the self-universality  
710 for transferable targeted attacks. In *Proceedings of the IEEE/CVF Conference on Computer*  
711 *Vision and Pattern Recognition*, pp. 12281–12290, 2023.
- 712 [56] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating  
713 adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610*, 2018.
- 714 [57] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L  
715 Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of*  
716 *the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2730–2739, 2019.
- 717 [58] Yifeng Xiong, Jiadong Lin, Min Zhang, John E Hopcroft, and Kun He. Stochastic variance  
718 reduced ensemble adversarial attack for boosting the adversarial transferability. In *Proceedings*  
719 *of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14983–14992,  
720 2022.
- 721 [59] Haotian Xue, Alexandre Araujo, Bin Hu, and Yongxin Chen. Diffusion-based adversarial  
722 sample generation for improved stealthiness and controllability. *Advances in Neural Information*  
723 *Processing Systems*, 36:2894–2921, 2023.
- 724 [60] Xiao Yang, Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Boosting transferability of  
725 targeted adversarial examples via hierarchical generative networks. In *European Conference on*  
726 *Computer Vision*, pp. 725–742. Springer, 2022.
- 727 [61] Chaoning Zhang, Philipp Benz, Tooba Imtiaz, and In So Kweon. Understanding adversarial  
728 examples from the mutual influence of images and perturbations. In *Proceedings of the*  
729 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14521–14530, 2020.
- 730 [62] Zhengyu Zhao, Zhuoran Liu, and Martha Larson. On success and simplicity: A second look  
731 at transferable targeted attacks. *Advances in Neural Information Processing Systems*, 34:  
732 6115–6128, 2021.